Thèse

présentée pour obtenir le grade de docteur

de l'Ecole nationale supérieure
des télécommunications

Spécialité : Electronique et Communications

Pietro Michiardi

Mécanismes de sécurité et de coopération entre noeuds d'un réseaux mobile ad hoc

soutenue le 14 Decembre 2004 devant le jury composé de

Jon Crowcroft	Rapporteurs
Ludivoc Mé	
Eitan Altman	Examinateurs
Christian Bonnet	
Gene Tsudik	
Refik Molva	Directeur de thèse

Ecole nationale supérieure des télécommunications

PhD thesis

Ecole nationale supérieure des télécommunications

Communications and Electronics department

Digital communications group

Pietro Michiardi

Cooperation enforcement and network security mechanisms for mobile ad hoc networks

Defense date: December, 14th 2004

Committee in charge:

Jon Crowcroft	Reporters
Ludovic Mé	
Eitan Altman	Examiners
Christian Bonnet	
Gene Tsudik	
Refik Molva	Advisor

Ecole nationale supérieure des télécommunications

To Alison, my Family and my Friends

Acknowledgements

Acknowledgements pages are always the last to be written. This is totally unfair! Well, I have been working for the past four years on some absolutely abstract concepts that in the end have the simple goal to stimulate people to get along together, behave correctly and, very importantly, to be fair!
So I think it is my turn to be fair and express my gratitude to those people who contributed to my personal and scientific growth and, most of all, that helped and encouraged me through all these years.

First of all I have to thank my parents. Without their great help in dealing with all that administrative paperwork I always tried to avoid when finishing my graduate studies I wouldn't be here to write these words. I'd like to thank them for their support too: if I had a wonderful life in the Cote d'Azur it's because they have always been there whenever I needed.

Many thanks go to my thesis advisor Refik Molva: if I had to count all the things that I learned and that I have done thanks to Refik, I would need another thesis manuscript. These years have been full of amazing experiences that helped me to understand and chose what I will do when I grow up!

I have to loudly thank my girlfriend Alison. She has always been there for my night-time rehearsals before any presentation I did: I'm sure she could present my thesis work better then I could. She has also contributed to my happiness through all these years, and she has inspired many of my ideas from her valuable comments and because she is an excellent listener!

I want to thank my friends as well: living with eight roommates helped me in building strong relationships that had and still have a great positive impact on my way of living.

Last but not least, I have a special thanks to give to "mother Eurecom": she's been part of my life since that first encounter 1998 and I often consider her premises as my second home!!

Resumé

Introduction

Un réseau ad hoc mobile (MANET) est un réseau maillé temporaire constitué par une collection de noeuds sans fil et mobiles sans aide d'une infrastructure préétablie utilisée pour exécuter les fonctions de base de gestion de réseau comme l'acheminement et l'expédition de paquets. Dans un tel environnement, il peut être nécessaire qu'un noeud mobile demande l'aide d'autres noeuds pour expédier un paquet à sa destination, a cause de la couverture limitée du champ radio disponible à chaque noeud.

En origine, des applications exploitant les réseaux ad hoc ont été envisagés principalement pour des situations de crise (par exemple, dans les champs de bataille ou pour des opérations de secours). Dans ces applications, tous les noeuds du réseau appartiennent à une même autorité (par exemple, une unité militaire ou une équipe de secours) et ont un but commun. Cependant, les technologies sans fil se sont sensiblement améliorées ces dernières années et des dispositifs peu coûteux basés sur la norme 802.11 ont envahi le marché. Par conséquence, le déploiement des réseaux ad hoc pour des applications commerciales est devenu réaliste. Des exemples incluent le déploiement des réseaux ad hoc pour l'automobile ou pour la fourniture d'équipements de communication pour les régions éloignées ou les périmètres physiquement délimités (par exemple, centres commerciaux, aéroports, etc.....). Dans ces réseaux, les noeuds typiquement n'appartiennent pas à la même organisation ni à une même autorité et ils ne poursuivent pas un but commun. En outre, les réseaux commerciaux pourraient être plutôt grands et avoir une vieplus longue; de plus ils peuvent être complètement autonomes, signifiant que le réseau fonctionnerait seulement par l'opération des utilisateurs.

Selon le type d'application, il est possible de définir deux catégories principales des réseaux ad hoc : les réseaux contrôlé et les réseaux ouvert. Nous nous référons aux réseaux ad hoc contrôlés quand la phase d'initialisation du réseau peut être appuyée par une infrastructure provisoire et quand une confiance a priori entre les noeuds du réseau est disponible. Des relations de confiance a priori peuvent être établies, par exemple, par une autorité contrôlant les noeuds du réseau ou par une organisation commune entre les utilisateurs. D'autre part, dans un réseau ad hoc ouvert, les noeuds sont entièrement autonomes et dans la majorité des cas ne peuvent pas compter sur une infrastructure préétablie pour l'initialisation et l'opération de réseau. De plus, puisque les noeuds sont actionnés par des utilisateurs qui n'appartiennent pas nécessairement à la même organi-

sation ni partagent une même autorité, une relation de confiance a priori entre les noeuds n'est pas disponible. La confiance entre les noeuds doit être établie par des mécanismes spécifiques conçus en fonction du scénario offert par un environnement ouvert.

Dans la thèse, deux protocoles d'acheminement spécifiquement conçus pour l'environnement ad hoc sont brièvement présentés. Même si une attention particulière a été consacrée à la conception des protocoles d'acheminement optimaux qui réduisent au minimum la consommation d'énergie ou qui sont bien adaptés à une topologie dynamique, pour cette thèse il est suffisant de décrire les propositions fondamentales qui ont contribué au développement successif des mécanismes d'acheminement plus avancés. En effet, le dénominateur commun entre les propositions d'acheminement ad hoc existantes est qu'aucune d'elles n'a défini des conditions de sécurité. En outre, les protocoles de routage ad hoc courants font confiance à tous les participants. Le point principal exposé dans cette thèse est que l'opération du réseau peut être facilement compromise si des contre-mesures ne sont pas mis en place dés leur conception, afin de protéger les fonctions de base de la gestion du réseau. La sécurité pour les MANET est un élément essentiel parce que, au contraire des réseaux qui utilisent des noeuds dédié pour exécuter les fonctions de base de gestion du réseau (comme l'Internet), dans les réseaux ad hoc ces fonctions sont effectuées par tous les noeuds disponibles. Cette principale différence est au coeur des nouveaux problèmes de sécurité qui sont spécifiques aux réseaux ad hoc. Au contraire des noeuds d'un réseau classique, les noeuds d'un réseau ad hoc ne peuvent pas être retenues fiable pour la correcte exécution des fonctions critiques du réseau. Ainsi, les réseaux ad hoc offrent des défis pas present dans les réseaux traditionnels. Les objectifs de recherches de cette thèse sont discuté dans la suite tandis que le plan de la thèse est présenté a la fin de cette section.

Sécurité des réseaux ad hoc

Traditionnellement, la sécurité a été considérée comme une question importante pour les réseaux avec infrastructure (i.e. réseaux dans lesquels des composants dédié, comme des router, fournissez le fonctions de base du réseau), particulièrement pour ces dédies a l'exécution d'applications sensibles. Pour sécuriser ce type de réseau et les applications conçu ainsi, les services de sécurité de base suivants ont été largement adressés par la communauté scientifique: confidentialité, intégrité, authentification, et non répudiation. La confidentialité assure que une certaine information n'est jamais révélée aux entités non autorisées. La transmission sur le réseau d'information sensible, comme des informations stratégique ou aient une valeur monétaire, exige de la confidentialité. La fuite d'information vers des oreilles indiscrètes pourrait avoir des conséquences graves. L'intégrité garantit qu'un message étant transféré n'a pas été corrompu. Un message pourrait être corrompu en raison des faute bienveillants ou en raison des attaques malveillantes sur le réseau. L'authentification permet à un noeud d'assurer l'identité du noeud avec qu'il communique. Sans authentification, un adversaire pourrait se faire passer pour un autre noeud, en sorte de gagner accès à une ressource non autorisé ou a une l'information sensible et interférant l'opération correcte d'autres noeuds. Enfin, la non répudiation as-

sure que la source d'un message ne peut pas nier en suite l'envoi du message. Ils existent d'autres objectifs de sécurité (par exemple, autorisation, détection d'intrusion, etc.....) qui sont une préoccupation pour certaines applications, mais nous ne poursuivrons pas ces problèmes dans cette thèse.

Dans le suivant, nous présentons une vue d'ensemble des conditions de sécurité pour les réseaux ad hoc qui ont leur analogue dans les réseaux reliant sur une infrastructure; nous devisons aussi les mécanismes de base qui permettent leur mise en place. Nous présentons en suite une nouvelle problématique de sécurité qui est spécifique au paradigme ad hoc et présentons brièvement les mécanismes qui répondent aux exigences de sécurité accrues par une telle menace additionnelle. Nous discutons les difficultés additionnelles pour fournir des mécanismes de sécurité traditionnels imposés par un environnement sans infrastructure et arguons que les nouvelles solutions de sécurité conçues initialement pour des réseaux ad hoc peuvent être étendu également aux réseaux reliant sur une infrastructure.

Sécurité réseaux traditionnelle

Les services de sécurité traditionnels tels que la confidentialité, intégrité, authentification, et la non répudiation ont été le sujet d'une intense recherche ces dernières années. Il est hors de la portée de cette section fournir un aperçu sur le divers type d'algorithmes cryptographiques qui ont été proposés dans la littérature. Au lieu de cela, nous discutons sur les aspects fondamentaux des deux familles principales d'algorithmes cryptographiques qui représentent la clef pour déployer des services de sécurité traditionnels dans les réseaux d'infrastructure. Le choix du matériel de base qui est employé pour fournir des services de sécurité de réseau est d'importance fondamentale.
Les algorithmes cryptographiques employé par les mécanismes de sécurité se basent sur un service de gestion et de distribution de clefs et peuvent être basés sur des clef symétriques ou asymétriques. Selon le type de clef, des algorithmes basés sur des clefs symétriques s'appellent **algorithmes symétriques** tandis que des algorithmes basés sur des clefs asymétriques s'appellent algorithmes asymétriques ou **algorithmes a clef publique**. Le choix du type d'algorithme a un impact immédiat sur la nécessite en termes de puissance de calcul disponible à chaque noeud du réseau. On le connaît de la littérature (voyez par exemple [88]) que la nécessite de puissance de calcul est sensiblement plus haute pour des algorithmes asymétriques. D'ailleurs, les effets du type d'algorithme peuvent être étendus pour tenir compte du coût énergique lié à l'exécution du protocole de sécurité. En outre, selon le type d'algorithme, un protocole de sécurité peut présenter des coûts non négligeables liées au trafic introduit dans le réseau : la distribution et l'utilisation des clefs ont un impact sur la longueur des messages a échangé par les noeuds menant de ce fait à un gaspillage significatif de ressources de largeur de bande.

D'autre part, les mécanismes de sécurité basés sur du matériel asymétrique ont quelques applications intéressantes. Le système cryptographique RSA constitue un travail séminal qui a permis la naissance des algorithmes asymétriques modernes. RSA [98], est généralement employé pour fournir la confidentialité et assurer l'authenticité des données numériques. De nos jours, RSA est déployé dans beaucoup de systèmes commerciaux. Il

est employé par des serveurs et des navigateurs Web pour securiser le trafic Internet, il est employé pour assurer la confidentialité et l'authenticité des e-mail, il est employé pour sécuriser des sessions de login a distance, et il est au coeur des systèmes électroniques de paiement . En bref, RSA est fréquemment employé dans les applications où la sécurité des données numériques est un souci. Les systèmes cryptographique a clef publique traditionnels sont basés sur RSA et exigent que la clef publique de n'importe quel utilisateur soit certifiée avec un certificat délivré et signé par une autorité de certification (CA) : ceci est fait afin de prouver l'authenticité de la clef publique appartenant à un utilisateur. N'importe quel participant qui veut employer une clef publique doit d'abord vérifier le certificat correspondant pour vérifier la validité de la clef publique. Quand beaucoup de CA sont impliqués entre deux utilisateurs, des rapports de confiance entre ces CA doivent également être vérifiés. En fait, les systèmes a clef publique se fondent sur une infrastructure a clef publique (PKI) qui est employée afin de contrôler le rapport de confiance entre les entités (d'une façon hiérarchique). Cependant, les solutions basées sur les certificats traditionnels sont affectées par l'inconvénient principal qui est la révocation des clefs publiques et du certificat correspondant, ce qui représente un grand problème et exige une grande quantité de stockage et de calcul. D'ailleurs, les services de révocation exigent souvent que l'autorité de confiance soit en ligne.

Nouvelles vulnérabilités spécifiques aux réseaux ad hoc

Les réseaux ad hoc sont des systèmes répartis composés d'entités autonomes qui nécessitent de coopération afin de fonctionner correctement. Grâce a la technologie ad hoc, les noeuds du réseau se connectent dynamiquement et d'une façon autonome entre eux et forment des topologies temporaires. Ceci permet à des personnes et à des dispositifs de communiquer sans besoin d'une infrastructure réseau préexistante. Le concept fondamental est l'exploitation de la synergie, résultat de la collaboration entre les composants du réseau, afin de fournir des services a chacun. La condition de base pour rendre opérationnel le paradigme coopératif est la contribution supposée de toutes les entités qui composent et, en même temps, se servent du système. Cependant, le déploiement récent des réseaux ad hoc pour des applications civiles, détend l'hypothèse initiale sur une coopération implicite entre les noeuds. D'un coté, les applications des réseaux ad hoc pour des situations de crise (militaire ou urgence) prévoient que tous les noeuds du réseau appartiennent à une même autorité et que tous aient un objectif commun. Par conséquence, la coopération des noeuds peut être assumée. Dans le contexte des applications civiles, les noeuds typiquement n'appartiennent pas à une même autorité et, de conséquence, la coopération de noeuds ne peut pas directement être assumée. Le manque de toute autorité centrale, pour garantir la collaboration, motive une tendance possible des entités a se conduire mal en n'adhérant pas au paradigme coopératif.

Le manque de coopération peut être provoqué par les entités qui sont malveillantes ou qui sont égoïstes. Les entités malveillantes visent intentionnellement à endommager l'opération normale du réseau. Ce genre de malveillance provoque plusieurs problèmes de sécurité qui peuvent être résolus par des services de sécurité traditionnels. Comme exemple, une entité malveillante peut effectuer une attaque de rupture d'acheminement par lequel l'attaquant envoie des paquets forgé pour créer une boucle d'acheminement ou pour

diviser le réseau. Ce genre d'attaque peut être empêché par la vérification d'intégrité et d'authenticité des messages de contrôle d'acheminement. D'autre part, une entité égoïste ne prévoit pas de endommager directement le fonctionnement global du système, mais est peu disposé à dépenser ses ressources au nom des autres. Ce genre de comportement entame une nouvelle classe des problèmes que nous groupons en problèmes de non coopération. Le manque de coopération a gagné beaucoup d'attention dans le contexte de la gestion de réseau ad hoc et plusieurs techniques ont été proposées pour mitiger ce nouveau type de menace. Nous présentons ces techniques en chapitre 3

Discussion

Les mécanismes traditionnels de sécurité réseau sont d'importance fondamentale pour les réseaux ad hoc. Les services de sécurité de base peuvent être efficacement employés pour protéger le réseau contre le comportement malveillant des entités visant à compromettre l'opération normale de réseau. Cependant, une analyse étendue des nécessites de sécurité de base exigées par les réseaux ad hoc indique que la majorité de conditions requises ne sont pas nouvelle ni spécifiques au paradigme ad hoc, mis a part du problème fondamental de la gestion et de la distribution de clefs. La difficulté accrue pour fournir le matériel cryptographique approprié aux entités du système est due à la conjonction de plusieurs facteurs qui s'étendent du manque de sécurité physique des noeuds au manque de rapports de confiance a priori entre les utilisateurs du système et au manque d'infrastructure. Les solutions classiques de sécurité développés pour les réseaux a infrastructure se fondent sur des composants dédiées aient des rôles prédéfinis : cependant, ces exigences fondamentales ne sont pas compatibles avec la définition de base des réseaux ad hoc par lequel aucun composant n'ait un rôle pré assigné. Il est important de rappeler que le choix des primitives cryptographiques employés par n'importe quel mécanisme de sécurité proposé pour l'environnement ad hoc doit être pense en considérant les conditions spécifiques dues à la nature des noeuds mobiles formant le réseau : souvent, la puissance de calcul et la capacité de mémoire peuvent être considérées en tant que ressources rares. Les frais additionnels dus au trafic introduit par des mécanismes de sécurité doivent également être considérés avec un oeil attentif. La largeur de bande et la disponibilité énergétique sont une autre ressource rare qui ne devrait pas être gaspillée par des mécanismes de sécurité sub-optimaux. D'autre part, le manque de coopération ce traduit dans des exigences de sécurités spécifiques et nouvelles aux réseaux ad hoc qui n'ont apparemment pas une analogie avec les réseaux infrastructure. Les problèmes de sécurité adressés par la communauté scientifique se concentrent sur des réseaux overlay et des systèmes de pair à pair, cependant, nous prouve que quelques similitudes qui lient les réseaux sans et avec infrastructure peuvent être trouvées. En effet, la gestion de réseau ad hoc partage beaucoup de concepts, comme la distribution et la coopération, avec le modèle de calcul du pair à pair (P2P) [101], ce qui constitue un paradigme normal pour le modèle de calcul ad hoc. Une caractéristique importante des systèmes de P2P est leur capacité de fournir un acheminement efficace, fiable, et résilient entre les noeuds qui le constituent en formant des topologies "ad hoc" structurées sur une vraie infrastructure de réseau. La différence avec les systèmes traditionnels de l'informatique répartie est le manque d'une autorité centrale commandant les divers composants ; au lieu de cela, les noeuds forment dynamiquement

et d'un façon autonome le système. Les applications plus adaptées pour l'exécution de P2P sont ceux où la centralisation n'est pas possible, les relations sont passagères, et les ressources sont fortement distribuées [91]. En particulier, l'extension des applications couverts par le modèle de P2P inclut le partage de fichier, recherche et indexation distribuées, stockage de ressource, travail de collaboration, etc... L'atout principal des systèmes P2P est leur capacité de fournir une plateforme peu coûteuse mais en même temps scalable, tolérant aux fautes, et robuste. Par exemple, le système P2P, comme Gnutella [4], est un système réparti où la contribution de beaucoup de participants avec un peu d'espace disque a comme conséquence une base de données très grande distribuée parmi tous les noeuds participant. La nature volontaire de la contribution des noeuds a également des inconvénients sérieux car les ressources du système peuvent être fortement variables et imprévisibles. En outre, le manque d'une autorité centrale coordonnant les ressources aux quelles chaque pair devrait contribuer, mène des utilisateurs à employer le système sans contribuer beaucoup. Le problème de non coopération, " freeriding " dans la terminologie de P2P, affecte fortement l'exploitation du système et le mène éventuellement a tourner son esprit de pair à pair vers un model client serveur plus traditionnel [100].

Objectifs de recherche de la thèse

La provision de services de sécurité pour la couche réseau ad hoc représente une terre fertile dans laquelle les chercheurs ont récemment exprimé leur créativité. Le scénario stimulant offert par les réseaux ad hoc où il n'y a aucune confiance a priori entre les noeuds (i.e., les réseaux ad hoc ouvert) motive l'étude initiale présentée dans cette thèse. En effet, dans un environnement ouvert, les mécanismes classiques de sécurité disponible pour les réseaux traditionnels ne peuvent pas être employé. Dans les réseaux qui se fondent sur des noeuds dédié (opéré par des autorités de confiance ou globalement connues) pour effectuer la gestion du réseau (par exemple, acheminement ou expédition de paquets), l'authentification d'entité peut être suffisante pour garantir l'exécution correcte de ces fonctions. Au contraire, si les noeuds du réseau sont actionnés par des différentes autorités pour les quelles une base de confiance n'existe pas et qui ne poursuivent pas un but commun, l'authentification d'entité n'est pas suffisante et un nouveau ensemble de exigence de sécurité doit être défini ainsi que des mécanismes appropriés de sécurité qui sont adaptés à un environnement sans infrastructure. Le travail présenté dans cette thèse se concentre initialement sur la définition des nécessitées de sécurité spécifiques pour les réseaux ad hoc ouvert, basé soit sur des attaques traditionnelles effectuées par des noeuds illégitimes ou compromis et sur un nouveau type de comportement des noeuds spécifique a un environnement fortement coopératif. Les solutions de sécurité qui font face aux attaques traditionnelles, i.e. les noeuds malveillants qui empêchent activement l'exécution normale de la gestion de réseau, exigent une évaluation bien pensée des caractéristiques spécifiques inhérentes à un réseau sans infrastructure et sans organisation. D'autre part, un nouveau type de comportement malveillant des noeuds qui est accentué dans cette thèse exige la définition de nouveaux paradigmes de sécurité. Le principe de fonctionnement des réseaux ad hoc rend la coopération des noeuds une condition essentielle. La coopération est vue comme la bonne volonté d'un noeud d'exécuter des fonctions de gestion du réseau au profit d'autres noeuds. Cependant, la coopération a un coût énergique

non négligeable ce qui peut mener à un comportement égoïste, particulièrement dans un environnement reliant sur des batteries tel que les réseaux ad hoc mobiles. En effet, il n'y a aucune raison de supposer que les noeuds participeront à l'opération du réseau si, par exemple, une simple modification de la fonction de expédition de paquets prolongerait sensiblement leur dure de vie. Les conditions de sécurité définies dans cette thèse mènent à la conclusion que les comportements malveillant et égoïste doivent être prise en compte pour fournir un ensemble complet de services de sécurité de réseau. En conséquence, les objectifs de recherches développe dans cette thèse visent la conception des mécanismes de sécurité faisant face au comportement égoïste et malveillant des noeuds.
Le premier mécanisme de sécurité proposé dans ce travail fournit des incitations pour les noeuds à coopérer. Notre mécanisme de coopération n'empêche pas un noeud de nier la coopération ou de dévier d'un comportement légitime mais s'assure que les noeuds se conduisant mal soient punis en niant graduellement les services de communication. L'efficacité de notre mécanisme de coopération a été analysée par une évaluation basée sur des simulations. Cependant, une simple évaluation numérique a montré de n'être pas suffisante pour fournir l'évidence des capacités de notre solution due à une limitation du model utilise pour simuler un comportement égoïste réaliste des noeuds formant le réseau. Par conséquence, notre recherche a été consacrée au développement d'un cadre formel dans lequel l'interaction entre les noeuds égoïste peut être fidèlement modelée tenant compte des caractéristiques du réseau et du mécanisme employé pour stimuler la coopération.

Le deuxième mécanisme de sécurité a présenté dans cette thèse ait face au comportement malveillant des noeuds qui visent à perturber l'opération normale du réseau. Notre but consiste a proposer une solution conçue en fonction d'un environnement ouvert que compte ni sur une infrastructure préétablie ni exige une organisation entre les utilisateurs.

Exigences de sécurité

Un réseau ad hoc se compose d'un ensemble de noeuds sans fil qui fonctionnent soit comme terminaux que comme routeurs afin de former spontanément un réseau temporaire sans compter sur la présence d'une infrastructure dédié. La sécurité des réseaux ad hoc a reçu récemment beaucoup d'attention de la communauté scientifique et un grand nombre de solutions pour protéger les réseaux ad hoc contre de divers types d'attaques ont été publiés. La sécurité dans les réseaux ad hoc est un problème grave dû à la conjonction de plusieurs facteurs :

- **absence de sécurité physique** : les vulnérabilités dues aux communications radio et la facilité d'écouter anonymement et de " spoofing " nécessite des mécanismes de sécurité forte afin d'obtenir la sécurité qui est équivalente aux communications standard câblées ;

- **absence de confiance a priori** : la plupart des réseaux ad hoc se composent d'un ensemble de noeuds qui ne font pas une partie d'aucune organisation partagée donc les paradigmes classiques de sécurité basés sur une relation de confiance préétablie parmi les utilisateurs ne sont pas applicables ;

- **absence d'infrastructure** : les solutions de sécurité basées sur les composants consacrés avec des rôles prédéfinis tels que les tiers et les serveurs de confiance de clef ne sont pas compatibles avec la définition de base des réseaux ad hoc par lequel aucun composant n'ait un rôle prédéfini;
- **exigence de coopération** : en raison du manque de composants consacrés, les fonctions de base de gestion du réseau doivent être effectuées d'une façon distribuée par la collaboration d'un ensemble de noeuds ordinaires, ainsi l'exécution des opérations de base de réseau comme l 'expédition de paquets et l'acheminement peut être fortement affectée par manque malveillant ou accidentel de coopération.

La recherche sur la sécurité des réseaux ad hoc a été initialement concentrée sur des protocoles d'acheminement puisque ceux-ci ont été considérés la partie la plus critique du réseau. Cependant, une analyse fine des travaux de sécurité du routage indique que la majorité des exigences adressées par les solutions et les mécanismes suggérés ne sont pas nouveaux ou spécifiques aux réseaux ad hoc, mis a part d'un problème fondamental qui a été souvent laissé de côté qui est la gestion de clefs sans de confiance a priori et en absence d'infrastructure. Dans ce chapitre nous présentons une classification des attaques visées à la couche réseau ad hoc. D'abord, nous nous concentrons sur des attaques aux mécanismes d'acheminement: nous considérons soit les attaquants externes ces joignent avec malveillance au réseau pour perturber son fonctionnement de base, que les attaquants interne qui visent a compromettre des noeuds légitimes pour perturber leur correct fonctionnement. Dans le reste de ce chapitre, les attaques qui exigent un effort en terme de puissance de calcul et qui coûtent de l'énergie sont définies comme attaques actives. Les conséquences des attaques actives sont analysées afin de dériver un ensemble de conditions pour la sécurité du routage ad hoc. Nous arguons du fait que la sécurité doit être prise en considération dés la conception des protocoles de routage ad hoc. Cependant, si la conception de protocoles de routage sécurise pour MANET n'a pas des exigences en particulier, la fourniture de services de sécurité de base qui constitue les briques d'une architecture de sécurité représente toujours un grand défi pour les chercheurs. Afin d'atténuer la complexité présentée par le paradigme ad hoc pur, les solutions proposées dans le chapitre 3 se fondent sur des hypothèses spécifiques (et limitatives) qui minimisent la complexité introduite par la mise en place d'associations de sécurité requises avant l'exécution des protocoles d'acheminement sécurise. Par conséquence, notre attention est en particulier concentrée sur les exigences de sécurité demandées par des services de gestion de clefs complètement autonomes qui peuvent être employés par les protocoles de routage sécurise. De plus, nous présentons un nouveau type d'attaque qui n'exige aucun effort ni a un coût énergétique pour l'attaquant, ce que nous appelons attaque passive. Les attaques passives sont spécifiques à l'environnement ad hoc et sont provoquées par noeuds égoïstes qui sont inclinés à sauver leur énergie et leur puissance de calcul pour effectuer des opérations visant a leur propre intérêt. Un comportement égoïste implique un manque de coopération qui peut prendre la forme d'échecs systématiques à la participation aux opérations de expédition ou d'aheminement. Les noeuds égoïstes ne visent pas a compromettre directement l'opération normale du réseau, mais les conséquences du manque de coopération ont un impact grave sur les performances. Nous fournissons une analyse basée sur des simulations de l'influence des attaques dues à

l'égoïsme des noeuds sur l'expédition et acheminement des paquets et analysons de plus l'impact de la mobilité des noeuds en présence des noeuds égoïstes. En conclusion, nous fournissons les directives et les conditions de sécurité requises pour concevoir un mécanisme de incitation à la coopération efficace pour atténuer les effets d'un comportement égoïste.

Composantes de bases pour la sécurité des réseaux ad hoc

La sécurité des réseaux ad hoc traite les menaces qui s'étendent des attaques actives aux attaques passives. Les exigences fondamentales de sécurité présenté dans le chapitre2 indiquent le besoin de sécuriser l'exécution des protocoles d'acheminement ad hoc aussi bien que la nécessité de mécanismes qui encouragent la coopération de noeud a l'opération du réseau. Les protocoles de routage sécurise se fondent sur la disponibilité d'associations de sécurité entre les noeuds : l'authentification des entités impliquées dans le routage et la vérification d'intégrité des messages d'acheminement réclament un schéma de gestion de clef pour fournir aux parties concernées dans l'exécution du protocole le matériel cryptographique approprié. Les mécanismes de coopération exigent également l'authentification des noeuds afin d'empêcher les attaques de " spoofing "; encore, les schémas de distribution de clef offrent la solution pour empêcher des attaques exploitant la mascarade d'identité. En outre, l'information sur le comportement des noeuds qui participent à l'opération de réseau rassemblée par des mécanismes coopération peut être employée par des protocoles de routage sécurise pour modifier le critère de choix des routes afin d'éviter des noeuds potentiellement nocifs, une technique qui va sous le nom de " path rater " [61].

Dans ce chapitre, nous présentons les efforts courants disponibles dans la littérature visant la conception des blocs fonctionnels de base pour sécuriser la couche réseau ad hoc, i.e. securiser l'acheminement, mécanismes de gestion de clef et de coopération, comme présenté dans [77], [68], [71] et [73]. Nous arguons du fait qu'une panoplie complète de services de sécurité ad hoc réclame une intégration de ces blocs fonctionnels de base qui adressent seulement un sous-ensemble spécifique de toutes les menaces possibles. En effet, le recouvrement d'exigence présenté par différents mécanismes de sécurité demande la détermination exacte de l'interface entre les divers modules de sécurité afin d'adresser le problème difficile de l'initialisation du réseau en l'absence d'un appui externe tel qu'une infrastructure ou une organisation partagée entre les utilisateurs du système.

La définition d'une architecture complète de sécurité est hors de la portée de cette thèse, mais nous espérons fournir un ensemble fondamental de directives pour permettre une conception efficace d'une architecture de sécurité pour le scénario stimulant offert par les réseaux ad hoc ouverts. En outre, nous concluons le chapitre avec une discussion sur les défis de recherches imposés par les réseaux ad hoc ouverts : nous réclamons que l'effort principal doit être investi dans la conception des mécanismes de coopération et de gestion de clef. Cette discussion justifie le reste de la thèse, ou nous proposons nos solutions pour fournir une incitation à la coopération et un mécanisme de gestion de clef pour les réseaux ad hoc ouverts.

CORE

L'étude a base de simulation effectué dans notre laboratoire [65] et présenté en chapitre 2 a prouvé que les performances d'un réseau MANET dégrade sévèrement en présence d'une simple comportement illégitime des noeuds. Indépendamment des cas spéciaux comme pour les réseaux militaires pour lesquels une confiance a priori existe entre tous les noeuds, les noeuds d'un réseau ad hoc ne peuvent pas être considéré fiables pour l'exécution correcte des fonctions critiques du réseau. Des opérations essentielles assurant la connectivité de base peuvent être fortement compromises par les noeuds qui n'exécutent pas correctement le leur part des opérations du réseau comme le routage, l'expédition de paquets, mappage nom adresse, etc... Un mauvais comportement des noeuds qui affecte ces opérations peut s'étendre de l'égoïsme ou du manque simple de collaboration due au besoin d'économie de batterie aux attaques actives comme le déni de service et la subversion du trafic. En raison de leur vulnérabilité accrue, les réseaux ad hoc devraient tenir compte des problèmes de sécurité comme condition de base indépendamment des scénarios d'application et des contre-mesures doivent être intégrés aux mécanismes de base de gestion de réseau dés leurs conception.

Une autre conclusion importante de notre étude de simulation est que la dégradation de performance due aux noeuds égoïstes s'abstenant à expédier de paquets est plus significative que l'impact du comportement égoïste simulé par des attaques sur le protocole d'acheminement comme le protocole " dynamic source routing " (DSR). Nous croyons que des résultats semblables qui accentuent la sensibilité inhérente de MANET à l'égoïsme des noeuds peuvent être obtenus avec des fonctions de réseau autres que l'acheminement et l'expédition de paquets. Les mécanismes de sécurité qui imposent seulement l'exactitude ou l'intégrité des opérations de réseau ne seraient ainsi pas suffisants dans MANET. Une condition de base pour maintenir le réseau opérationnel consiste a imposer la contribution ad hoc des noeuds aux opérations du réseau en dépit de la tendance contradictoire de chaque noeud vers l'égoïsme comme motivée par la pénurie des ressources énergétiques.

Dans ce chapitre nous proposons un mécanisme appelé CORE pour imposer la coopération entre les noeuds basée sur une technique de surveillance distribuée, ce qui a été présenté dans [66]. Notre mécanisme de coopération n'empêche pas un noeud de nier la coopération ou de dévier d'un comportement légitime mais s'assure que les entités se conduisant mal soient **punie** en leur refusant graduellement les services de communication. CORE est suggéré comme mécanisme générique qui peut être intégré avec n'importe quelle fonction de réseau comme l'expédition de paquets, découverte de route, gestion de réseau, et gestion de localisation. Dans CORE, chaque entité réseau encourages la collaboration d'autres entités en utilisant un métrique de coopération appelé **réputation**. La métrique de réputation est calculée sur la basé des données recueillies localement par chaque noeuds et peut se baser optionnellement sur l'information fournie par d'autres noeuds du réseau impliqués dans des échanges de message avec le noeud surveillé. Basé sur la réputation, un mécanisme de punition est adopté comme système de dissuasion pour empêcher un comportement égoïste en refusant graduellement les services de communication aux entités qui se conduisent mal. Comme précisé dans la thèse, la conséquence immédiate d'un réseau MANET qui adopte CORE est que les noeuds légitimes (i.e. noeuds qui coopèrent à l'opération de réseau) arrivent a économiser de l'énergie car il ne servent pas ceux qui ont

été détectés comme égoïstes. En outre, selon le modèle d'égoïsme adopté pour représenter des noeuds se conduisant mal, il est possible de prouver que CORE fournit une incitation efficace à coopérer. Dans le reste du chapitre on fournis une définition détaillée des objectifs de sécurité de CORE et un modèle d'égoïsme de noeud. Les hypothèses de base utilisées dans la conception de notre mécanisme de coopération sont aussi détaillées. La section 4.4 fournit un croquis du fonctionnement de CORE tandis que la section 4.4 se concentre sur les différents modules qui constituent notre mécanisme. Nous fournissons alors une discussion sur les propriétés de notre système en mettant en évidence l'évaluation de sécurité de CORE ; nous concluons avec un exemple d'application où CORE est utilisé pour stimuler la participation des noeuds à la fonction d'expédition de paquets.

Validation de CORE a base de simulation réseau

Dans ce chapitre nous effectuons une étude a base de simulation du mécanisme CORE, utilisé comme composant additionnel pour la suite logiciel de simulation réseau Glomosim. Le mécanisme CORE est analysés en termes de certaines métriques de simulation que nous considérons appropriés pour évaluer les propriétés de base d'un mécanisme de coopération: les coûts en terme d'énergie qui affectes les noeuds de d'un réseau qui utilise CORE et l'efficacité des mécanismes de détection et de punition adopté par CORE. D'un façon similaire aux approches a base de simulation disponibles dans la littérature (référez-vous au chapitre 3) nous effectuons nos expériences pour divers type de scénarios en tenant compte des réseaux statiques et dynamiques aussi bien que différents modèles de trafic. Dans notre étude de simulation, nous employons un modèle simple d'égoïsme de noeud (référez-vous au chapitre 2) par lequel des entités se conduisant mal sont définies au début de la simulation et leur comportement soit indépendant du temps de simulation. Des résultats de simulation sont employés pour comprendre si et quand un mécanisme pour distribuer l'information de réputation pourrait être nécessaire afin d'améliorer l'efficacité du mécanisme de punition: la distribution de réputation est une option du mécanisme CORE et constitue le principal discriminant entre CORE et autres mécanismes de coopération basé sur la réputation.

Même si des résultats intéressant peuvent être obtenus par une évaluation basée sur des simulations de des schémas de coopération, nous pensons que le modèle d'égoïsme utilisé dans la plupart des travails disponible dans la littérature n'est pas suffisant pour saisir les caractéristiques fondamentaux des mécanismes prévus pour stimuler la coopération entre entités qui sont intéressées que par leur propre survie. En assumant un comportement égoïste statique selon lequel les noeuds soient définis comme égoïstes pour la durée de vie totale de réseau, il est peu justifiable que les propriétés de stimuler la coopération puissent être correctement démontrées. Un modèle d'égoïsme qui ne tient pas compte des variations occasionnelles du comportement des noeuds n'est pas approprié pour la validation d'un mécanisme qui est prévu pour guider des noeuds égoïstes (ou des utilisateurs égoïste qui actionnent les noeuds) vers un comportement plus coopératif.

Puisqu'une grande fraction des mécanismes de coopération sont basées sur des principes apparentés aux systèmes de prise de décision et à la modélisation économique, un outil naturel qui a émergé pour valider tels mécanismes est la théorie des jeux. Dans le prochain

chapitre nous définissons deux modèles analytiques qui décrivent l'environnement MANET et les noeuds participant à l'opération du réseau en terme de théorie des jeux. Notre recherche a été présentée dans un travail préliminaire [67], et a été étendu dans [5–7, 70]. L'utilisation des modèles théoriques pour valider un mécanisme de coopération permet la définition d'un comportement dynamique des noeuds qui suivent une stratégie "rationnelle" imposée par un utilisateur qui actionnerait le noeud. Une stratégie rationnelle représente le comportement d'un utilisateur égoïste qui essais de maximiser ses bénéfices (en termes de consommation d'énergie) tout en sachant que d'autres utilisateurs dans le système pourraient faire la même chose ou pourraient adopter une stratégie de coopération pour forcer la coopération. Les modèles d'un réseau MANET basés sur la théorie des jeux sont toutefois limités car ils ne représentent pas d'une façon fine les mécanismes de base (et leurs limitations inhérentes) qui sont mis en place pour opérer le réseau ad hoc, comme les protocoles MAC de contrôle d'accès au moyen radio et de mechanismes de routage. Dans les approches présentées dans ce chapitre nous surmontons certaines des limitations dictés par une représentation de niveau élevé des interactions entre noeuds dans le réseau en incluant les aspects qui caractérisent les mécanismes qui sont analysés : par exemple, dans le premier modèle présenté dans la suite de ce chapitre nous tenons compte des issues liées à la technique de "watchdog" qui ont été exposées dans le chapitre 4. En combinant les résultats obtenus par l'analyse a base de simulation et par l'analyse en terme de théorie des jeux de CORE nous concluons que notre mécanisme répond non seulement à toutes les exigences qui ont été présentées en chapitre 2 mais que CORE a également des meilleures performance par rapport a d'autres stratégies de coopération évaluées dans la littérature quand un modèle réaliste de réseau est assumé.

Modélisation et validation analytique de CORE par la théorie des jeux

Plusieurs mécanismes de coopération ont été proposés par la communauté scientifique dans la tentative de faire face au comportement égoïste des noeuds présent dans les réseaux ad hoc mobiles (MANET). Comme défini en chapitre 2, un noeud est considéré égoïste quand il ne participe pas à la gestion ordinaire du réseau afin d'économiser de l'énergie. En opposition à la malveillance, l'égoïsme est une menace passive qui ne comporte aucune dégradation intentionnelle a l'opération du réseau au contraire des attaques actives, par exemple, comme la subversion de route, modification des données, etc... En ce chapitre nous présentons deux approches pour évaluer les mécanisme CORE, décrit en chapitre 4. Puisqu'une grande fraction des schémas existants sont basées sur des principes apparentent a la modélisation économique, un outil naturel qui a émergé pour la validation de tels mécanismes est la théorie des jeux. Dans ce chapitre, une méthode basée sur la théorie des jeux non coopérative est présenté, ce modèle étant utile pour démontrer les propriétés de base de CORE. En utilisant cette méthode nous adoptions un modèle qui décrit la stratégie d'un noeud égoïste qui doive prendre la décision de coopérer ou de ne pas coopérer avec un noeud voisin aléatoirement choisi. Nous traduisons alors le mécanisme CORE en guise de stratégie qui peut être ainsi comparé à d'autres stratégies bien connues dans la littérature. Sous l'hypothèse généralement utilisée de la " surveillance

parfaite ", nous démontrons l'équivalence entre CORE et un éventail de stratégies basées sur l'histoire des interactions, comme la stratégie " Tit-for-Tat ". De plus, en adoptant une hypothèse plus réaliste qui tient compte de l' **imperfection** des observations du comportement des noeuds voisin dues aux erreurs de communication, le modèle non coopératif met en évidence la supériorité (en termes de stabilité et robustesse) de CORE par rapport a d'autres mecanismes basé sur une histoire de interaction. Successivement, nous avons également défini une méthode que nous inclurons comme annexe de la thèse, là où nous développons notre modèle afin de tenir compte de l'information de topologie du réseau dans la phase de choix de stratégie, c.a.d. quand le noeud prend une décision au sujet de l'expédition ou moins du trafic. Nous dérivons également un mécanisme alternatif de punition qui fournit une punition plus douce pour les noeuds qui se conduisent mal.

Dans ce chapitre, nous explorons en suite un modèle différent qui tient compte d'une perception centré sur les noeuds autant qu'une perception centré sue le réseau des interactions entre les noeuds qui participent à un MANET, en employant la théorie des jeux *coopérative*. Nous démontrons d'abord la nécessité fondamentale d'un mécanisme de coopération pour stimuler la coopération entre les noeuds égoïstes en montrant que en absence d'un tel mécanisme, la meilleure stratégie pour un noeud serait de ne pas coopérer. D'ailleurs, nous analysons quelle serait la taille d'une coalition des noeuds coopératifs en se basant sur l'importance donnée par un noeud à la perspective centré sur lui-même ou centré sur le resau. Nous suggérons finalement comment le mécanisme CORE pourrait être utilisé pour stimuler les noeuds à joindre la coalition des coopérateurs. L'avantage d'employer la théorie des jeux coopératif dérive de la capacité de cette méthode de saisir la dynamique d'un groupe de joueurs : la stratégie choisie par un joueur dépend non seulement d'une perception égoïste du jeu mais tient compte également d'une politique de groupe au quel le joueur appartiendra. Bien que l'approche "de jeux coopératifs" semble être appropriée pour modeler la dynamique de grandes coalitions des noeuds formant un MANET, la limitation principale de cette méthode est qu'elle est basée sur une représentation à niveau élevé du mécanisme de réputation qui ne tient pas compte des dispositifs réels de CORE. Bien que les méthodes décrites dans ce chapitre se concentrent sur CORE comme mécanisme spécifique, quelques conclusions générales peuvent être tirées de cette analyse vers la conception des mécanismes de coopération en général.

Gestion de clef et authentification pour un réseau MANET

Dans ce chapitre nous nous concentrons sur les services de gestion de clef et d'authentification *pour les réseaux ad hoc ouverts*. Fournir des solutions de sécurité pour les réseaux ad hoc ouverts est une tâche particulièrement difficile en raison de la conjonction de plusieurs facteurs. **L'absence de confiance a priori** entre les noeuds du réseau et le faite qu'on ne puisse pas assumer que les utilisateurs partagent la même organisation, rend les paradigmes de sécurité classique basés sur un rapport de confiance préétablie parmi les parties pas applicables. En outre, les solutions de sécurité basées sur des composants dédiée aient des rôles prédéfinis tels que les tiers et les serveurs de confiance ne sont pas compatibles avec la définition de base des réseaux ad hoc ouverts dans lesquels aucun

composant n'ait un rôle prédéfini et pour lesquels une infrastructure n'est pas disponible.

La recherche sur la sécurité des réseaux ad hoc s'est d'abord concentrée sur des protocoles de routage puis que ceux-ci ont été considérés la partie la plus critique de du réseau. Cependant, une analyse fine des protocole de routage sécurise (référez-vous au chapitre 3) indique que la majorité des exigences adressées par les solutions et les mécanismes courants ne sont pas nouveaux ni spécifiques aux réseaux ad hoc, mis a part d'un problème fondamental qui a été souvent ignoré et qui consiste a initialiser les associations de sécurité entre les noeuds du réseau sans une relation de confiance préétablie et sans une infrastructure. Dans le chapitre 3 nous proposons un aperçu détaillé sur les techniques de gestion de clef et de routage sécurise disponibles dans la littérature qui adressent une variété de configurations ad hoc, y compris les réseaux ad hoc ouverts.
Les approches principales de gestion que nous avons évaluées 3 essaient de répondre à la difficile question de comment établir des associations de sécurité sans confiance a priori et manque d'infrastructure. Plusieurs mécanismes originaux de gestion de clef basé sur des constructions cryptographiques avancées comme la cryptographie a seuil et la cryptographie basée sur l'identité ont été suggérés dans la littérature mais ils sont tous loin du but final d'établir une infrastructure de sécurité "à partir de zéro" puisqu'ils impliquent tous une première phase d'initialisation souvent très lourde.

Dans ce chapitre nous proposons un mécanisme d'authentification appelé IDHC et basé sur un concept original qui combine une forme simple de cryptographie basée sur l'identité avec la méthode de chaîne de hachage de Lamport [55]. Dans notre solution [72], les utilisateurs peuvent produire localement une chaîne de matériel d'authentification que nous appelons ticket d'*authentication* en employant comme graine une information secrète (ce que nous appelons a ticket d'authentification *principale*) livré par un serveur centrale de distribution de clefs (KDC). En mitigeant la nécessite de confiance dans une infrastructure a clef publique, notre mécanisme est particulièrement approprié pour les réseaux avec des sources dynamiques multiples tandis que d'autres schémas d'authentification disponibles dans la littérature souffrent des limitations imposées par la gestion des certificats de clef publique. Dans ce chapitre, nous décrivons également une application intéressante de notre mécanisme : IDHC peut être employé comme base pour fournir un mécanisme de distribution de clef léger qui offre des services d'authentification particulièrement appropriés à un réseau ad hoc sans infrastructure réseau. En effet, dans la solution proposée, il n'y a aucun besoin d'infrastructure de réseau et la phase d'initialisation de sécurité est légère : le serveur de distribution n'est impliqué ni dans des opérations de gestion de réseau ni dans toute autre opération de sécurité au delà de la phase initiale.

Le reste du chapitre est organisé comme suit : d'abord nous fournissons des concepts de base requis pour comprendre les systèmes cryptographiques traditionnel basés sur identité ; en suite nous présentons le mécanisme d'authentification IDHC en se concentrant sur la technique cryptographique a base d'identité adopté pour générer le matériel d'authentification. Une évaluation détaillée des propriétés de sécurité de notre schéma est fournie : nous analysons la robustesse d'IDHC en ce qui concerne les menaces effectuées par les utilisateurs malveillants présent dans le système. Nous nous concentrons alors sur l'analyse de l'exécution du mécanisme IDHC en termes de exigences de puissance de calcul et de stockage. En conclusion, nous décrivons une application possible d'IDHC pour fournir des services de distribution de clef et d'authentification de message dans le

scénario stimulant offert par les réseaux ad hoc. En suite nous expliquons comment IDHC peut être utilisé comme module additionnel pour la distribution de clef dans un protocole de routage sécurise existant.

Conclusions

Les systèmes de communication basés sur entités autonomes qui forment un réseau sans besoin ou d'une infrastructure préétablie représentent un scénario intéressant qui jouera un rôle important dans la société et l'économie en présentant des moyens de création de réseaux et services *ad hoc*. Cependant, afin que ces services soient exploitables, ils doivent reposer sur un réseau qui soit sécurise. La grande sensibilité des réseaux ad hoc mobiles (MANETs) au problèmes de sécurité qu'on ne trouve pas dans les réseaux dédiés comme l'Internet, dérive de l'absence de noeuds aient un rôle prédéfini comme la responsabilité de la correcte opération du réseau. En outre, dans le cas extrême des réseaux ad hoc dans lesquels les utilisateurs qui actionnent le réseau n'appartiennent pas à la même organisation ni sont régis par la même autorité, le manque d'un rapport de confiance a priori entre les membres du réseau rend la sécurité un composant essentiel pour permettre un déploiement et une utilisation réalistes de ce genre de réseaux.

Dans cette thèse nous abordons les questions de sécurité offertes par les réseaux ad hoc ouverts. Nous étudions d'abord l'impact de plusieurs menaces qui ont été souvent négligées par la communauté scientifique en concevant les protocoles de routage ad hoc qui font souvent confiance au noeuds participant à l'acheminement. Si ces menaces peuvent être considérées en conformité avec l'expérience recueillie par l'étude d'une variété d'attaques sur les protocoles d'acheminement pour les réseaux classiques, dans cette thèse nous précisons et analysons un nouveau type de attaque, que nous avons appelé égoïsme, spécifique a l'environnement fortement coopératif offert par le paradigme ad hoc de gestion du réseau. Une analyse basée sur des simulations conduite dans nos laboratoires a indiqué que la coopération entre noeuds est essentielle. A différence des réseaux qui utilisent des noeuds dédiée pour exécuter les fonctions réseau de base, dans les réseaux ad hoc ces fonctions sont exécutée par tous les noeuds disponibles. Cependant, il n'y a aucune raison de supposer que les noeuds participeront à l'opération de réseau, particulièrement dans un environnement reliant sur une source énergétique limité comme les MANET. Nous proposons un état de l'art des services de sécurité de base pour les réseaux ad hoc, ainsi que de la gamme de protocoles de routage sécurise, les services de gestion de clef et les mécanismes de coopération disponibles dans la littérature. Notre recherche a précisé deux directions de recherches sur les quelles nous avons investi dans la thèse : la nouveauté en terme d'exigences de coopération et le problème difficile de fournir des associations de sécurité sans appui d'une infrastructure externe.

Directions futures de recherche

La recherche qui couvre les mécanismes de sécurité pour les réseaux ad hoc ouverts est toujours dans son enfance et beaucoup d'espace pour la créativité est disponible. Pour ce qui concerne les mécanismes de coopération, une direction qui est susceptible d'intérêt

est offerte par l'analyse des stratégies de coopération et des algorithmes de formation de coalition en utilisant la théorie des jeux mais en utilisant une représentation plus réaliste des mécanismes sous jacent de gestion du réseau. En effet, plusieurs limitations de notre schéma sont dues à nos hypothèses initiales : comme exemple, nous considérons seulement les réseaux homogènes tandis qu'il serait plus réaliste de considérer des réseaux hétérogènes. En outre, nous considérons au initialement des interactions parmi paires aléatoires de noeuds dans le réseau : cependant, nous croyons que l'espace de stratégie d'un noeud pourrait être réduit si nous considérons des interactions a joueur multiple en tenant compte de la stratégie de tous les noeuds qui appartiennent à un chemin d'une source a la destination correspondante.

Pour ce qui concerne le but final de concevoir des services de sécurité distribués fournissant des associations de sécurité a partir de zéro, la plupart des solutions proposées dans la littérature ne sont pas alignées avec les exigences contraignantes des resaux ad hoc ouverts. Le manque de composants consacrés qui fournissent de tels services exige la définition des services de sécurité distribués ne reliant pas sur la présence d'un tiers de confiance pour leur initialisation : une direction intéressante qui devrait être étudié est offerte par les mécanismes qui utilisent l'information provenant d'autres niveaux de la couche ISO/OSI pour optimiser l'architecture de sécurité. Dans toute cette thèse nous soulignons le besoin d'une architecture de sécurité qui définit les interactions et l'intégration entre les modules de sécurité qui fait face aux attaques de sécurité de niveau réseau. Nous considérons qu'un sujet intéressant qui a besoin davantage de recherche est lié à la définition d'une telle architecture. Comme exemple, nous avons mentionné dans cette thèse qu'il serait intéressant d'étudier les implications des interactions entre une application comme le systèmes de partage de fichier pair a pair et un mécanisme de incitation a la coopération au niveau réseau de la couche ISO/OSI.

Abstract

Communication systems based on self-organizing entities that build up the network without the need or reliance for a pre-established infrastructure represent a challenging scenario that will play an important role in society and economy by providing opportunities for the creation of *ad hoc* networks and services. However, in order for these services to be successful, they must rely on a network that is secure. The increased sensibility of mobile ad hoc networks (MANETs) with respect to dedicated networks like the Internet, derives from the lack of nodes with a predefined role that are responsible for the network operation.
Initially, applications of ad hoc networks have been envisioned mainly for crisis situations (e.g., in battlefields or in rescue operations). In these applications, all the nodes of the network belong to a single authority (e.g., a single military unit or a rescue team) and have a common goal.
However, wireless technologies have significantly improved in recent years and low-cost devices based on the 802.11 standard have invaded the market. As a consequence, the deployment of ad hoc networks for civilian applications has become realistic. Examples include the deployment of ad hoc networks for vehicular technologies or for the provision of communication facilities for remote areas or physically delimited perimeters (e.g. shopping malls, airports, etc...). In these networks, nodes typically do not belong to the same organizational structure nor to a single authority and they do not pursue a common goal. In addition, civilian networks could be rather large and have a longer lifetime; further they could be completely *self-organizing*, meaning that the network would run solely by the operation of end-users.
In the extreme case of ad hoc networks in which users that operate the network do not belong to the same organization nor are governed by the same authority, the *lack of an a-priori trust relationship* between the members of the network renders security an essential component to enable a realistic deployment and utilization of such *open* networks.
In this thesis we address the security issues raised by open ad hoc networks. We first investigate on the impact of several threats that have been often neglected by the research community when designing ad hoc routing protocols in which all participants were *inherently trusted*. If these threats can be considered in line with the experience gathered through the study of a variety of attacks on routing protocols for classical networks, in this thesis we point out and analyze a new type of misbehavior, that we called node selfishness, specific to the highly cooperative environment offered by the ad hoc networking paradigm. A simulation-based analysis conducted in our laboratories revealed that node cooperation is essential because unlike networks using dedicated nodes to support basic functions like packet forwarding, routing, and network management, in ad hoc networks

those functions are carried out by all available nodes. However, there is no reason to assume that nodes will participate in the network operation, especially in battery powered environment such as a MANET.
We propose a state of the art of basic security services for ad hoc networks, that range from secure routing protocols to key-management services and we focus on various type of cooperation enforcement mechanisms available in the literature. Our research pointed out two challenging research directions that we further investigated in the reminder of the thesis: the novelty of cooperation requirements and the difficult task of providing security associations without the support of an external infrastructure.

We thus propose a cooperation enforcement mechanism based on an original reputation system that we called CORE. CORE meets the design requirements that we presented in our preliminary study on cooperation enforcement schemes. Furthermore, we propose a detailed validation of the CORE mechanism using two different methodologies: in the first method, we use a classical network simulation tool to infer the basic properties of CORE. We then extend our work in order to cope with a sophisticated model of node selfishness that takes into account end-users' "rationality" when operating the nodes. Our validation study shows that CORE is an effective and robust mechanism that stimulates cooperation of selfish nodes; furthermore, through the evaluation of nodes' energy consumption, we provide evidence that CORE also offers incentives for legitimate nodes to use it as a cooperation mechanism in that they can save a non-negligible amount of energy.

We conclude the thesis by proposing a novel authentication scheme for mobile ad hoc networks that does not rely on a pre-established network infrastructure and that does not require any shared organization between the users of the network. In our scheme, a lightweight bootstrap phase is necessary for a node entering the network: by contacting an authentication server a node is able to locally generate authentication credentials that are globally verifiable in the network without the need for the distribution of public key certificates. We also propose a practical utilization of our scheme as a complementary key distribution scheme that enables authentication services demanded by secure routing protocols available in the literature.

Contents

List of Figures

List of Tables

Glossary

-A-

ACK Acknowledgment packet
AODV Ad hoc On demand Distance Vector

-C-

CA Certification Authority
CBR Constant Bit Rate
CCS Credit Clearance Service

-D-

DoS Denial of Service
DSR Dynamic Source Routing

-E-

ERC Equity, Reciprocity, Competition

-F-

FIR Finite Impulse Response

-G-

GT Game Theory
GTE Guaranteed Time of Encounter
GTFT Generous Tit For Tat strategy profile

-I-

INRT Intermediate Node Reply Token
ID Identity, node identifier
ID-based Identity based
IDHC IDentity based Hash Chains
IP Internet Protocol
IPD Iterated Prisoner's Dilemma
IT Information Technology

-K-

KDC Key Distribution Center

-M-

MAC Medium Access Control (specified in text)
MAC Message Authentication Code (specified in text)
MAD Mutual Authentication with Distance bounding
MANET Mobile Ad hoc Network
MIPv6 Mobile IP version 6

-O-

OLSR Optimized Link State Routing

-N-

NE Nash Equilibrium

-P-

PD Prisoner's Dilemma
PGP Pretty Good Privacy
PK Public Key

-R-

RREP, REP Route Reply packet
RREQ, RDP Route Request packet
RT Reputation Table

-S-

SA Security Association
SK Secret (private) Key
SPC Shortest Path Confirmation
SRP Secure routing protocol
SSL Secure Socket Layer
SUCV Statistically unique cryptographically verifiable

-T-

TFT Tit For Tat strategy profile
TTP Trusted Third Party

-U-

UID Unique identifier

-W-

WD Watchdog mechanism

Notations

Chapter 2

E_{in}	Initial energy available to a node
e_{T1} , e_{T2}	Thresholds for the energy-driven selfish model
PDR	Packet Delivery Ratio
d	Communication delay
p	Percentage of selfish nodes in the network
$G = (V, E)$	Arbitrary graph, where V is the set of vertex and E is the set of edges
$N_r(s)$	Neighborhood of the arbitrary node x

Chapter 3

$K_{x,y}$	Symmetric shared keys between entities x and y
KG	Group shared key
$H[\,]$	Generic hash function
$MAC_{K_{x,y}}$	Keyed message authentication code, using symmetric key $K_{x,y}$
K_{A_i}	TESLA key for entity A
$cert_A$	Public Key Certificate for entity A

Chapter 4

f	Generic function, e.g. packet forwarding
W_{TO}	Watchdog timeout
W_{freq}	Watchdog sampling frequency
P_{th}	Punishment threshold
σ_k	k-th observation
v_+, v_-	Outcomes associated to an observation
$r^k_{n_i}(n_j \vert f)$	Subjective reputation at time k by node n_i on node n_j on function f
$r^k_{n_i}(n_l, n_j \vert f)$	Indirect reputation provided by node n_j on node n_l
$r^k_{n_i}(n_j)$	Global reputation value associated to node n_j by node n_i

Chapter 5

B	Observation buffer size
$p^t_{D_i}$	Path diversity at time t
p_E	Punishment efficiency
G	Generic game
$I = 1, ..., n$	Set of players that participate to game G
$g_i : A \rightarrow \Re$	Von Neumann-Morgenstern utility function
$a^t = (a^t_1, ..., a^t_n)$	Players' stage-game actions
$h^t = (a^0, a^1, ..., a^{t-1})$	History at time t for game G
$A^t = \underset{j=0}{\overset{t-1}{X}} A$	Set of all possible histories h^t at time t
$s^t = (s^t_1, ..., s^t_n)$	Period-t stage game strategy profile
$s_i = \left(s^0_i, s^1_i, ..., s^T_i\right)$	Player i's strategy for the repeated game
$s = (s_1, ..., s_n)$	n-tuple profile of players' repeated-game strategies
δ	Discount factor
$\sum_{t=0}^{\infty} \delta^t u^t_i$	Payoff function of the stream $u^0_i, u^1_i, ...$
$W_n(A)$	Number of players adopting strategy profile A
N	Number of players
$\alpha_i u(y_i) + \beta_i r(\sigma_i)$	ERC global utility function for player i
$\alpha_i, \beta_i \geq 0$	ERC-types for player i
y_i	Absolute payoff for player i
$u_i(y_i)$	Absolute utility function for player i
σ_i	Relative payoff for player i
$r(\sigma_i)$	Relative utility function for player i

Chapter 6

p , $q \in \mathrm{Z}_n$	Large primes
$n = p \cdot q$	RSA modulus
$\phi(n)$	Euler's totient function
$e \in \mathrm{Z}^*_{\phi(n)}$, $k \in \mathrm{N}$	Public values
$d = e^{-k} \bmod \phi(n)$	Master secret key
M	Master authentication ticket
T_k	k-th authentication ticket
τ_i	Time interval
P_i	Generic packet
m_i	Data message
$MAC_{T_{k-i}}(m_i)$	MAC on m_i using authentication ticket T_{k-i} as the key

Chapter 1

Introduction

A mobile ad hoc network (MANET) is a temporary meshed network formed by a collection of mobile wireless nodes without the aid of a dedicated infrastructure used to carry out basic networking functions like routing and packet forwarding. In such an environment, it may be necessary for one mobile host to enlist the aid of other hosts in forwarding a packet to its destination, due to the limited range of each mobile host's wireless transmissions.
Initially, applications of ad hoc networks have been envisioned mainly for crisis situations (e.g., in battlefields or in rescue operations). In these applications, all the nodes of the network belong to a single authority (e.g., a single military unit or a rescue team) and have a common goal.
However, wireless technologies have significantly improved in recent years and low-cost devices based on the 802.11 standard have invaded the market. As a consequence, the deployment of ad hoc networks for civilian applications has become realistic. Examples include the deployment of ad hoc networks for vehicular technologies or for the provision of communication facilities for remote areas or physically delimited perimeters (e.g. shopping malls, airports, etc...). In these networks, nodes typically do not belong to the same organizational structure nor to a single authority and they do not pursue a common goal. In addition, civilian networks could be rather large and have a longer lifetime; further they could be completely *self-organizing*, meaning that the network would run solely by the operation of end-users.
Based on the application type, it is possible to define two main categories of ad hoc networks: *managed* and *open* ad hoc networks. We refer to *managed* ad hoc networks when the network initialization phase can be supported by a temporary infrastructure and when an *a-priori trust* between the nodes of the network is available. A-priori trust relations can be established, for example, by a single authority managing the nodes of the network or by a common organizational structure of the end-users. On the other hand, in an *open* ad hoc network, nodes are fully self-organized and do not (or cannot) rely on any established infrastructure for the network initialization and operation. Furthermore, since nodes are operated by end-users that do not necessarily belong to the same organization nor share a single authority, an *a-priori trust* relation between the nodes is not

available. Trust between nodes have to be built through specific mechanisms tailored to the challenging scenario offered by an open environment.
In section 2.1, two routing protocols specifically designed for the ad hoc environment are briefly introduced. Even if major attention has been devoted to the design of optimal routing protocols that minimize power consumption or that are well adapted to a dynamic topology, for the scope of this thesis it is sufficient to describe the seminal proposals that contributed to the later development of current routing mechanisms. Indeed, the common denominator of existing ad hoc routing proposals is that none of them have defined security requirements. Furthermore, current ad hoc routing protocols *inherently trust all participants*. The claim exposed in this thesis is that network operation can be easily jeopardized if countermeasures are not embedded into basic networking functions at the early stages of their design. Security in MANET is an essential component because unlike networks using dedicated nodes to support basic functions like packet forwarding, routing, and network management, in ad hoc networks those functions are carried out by all available nodes. This very difference is at the core of the security problems that are specific to ad hoc networks: as opposed to nodes of a classical network, the nodes of an ad hoc network *cannot be trusted* for the correct execution of critical network functions. Thus, securing ad hoc networks offers challenges not present in traditional networks.
The research objectives of this work are detailed in section 1.2 while the outline of the thesis is presented in section 1.3.

1.1 Ad hoc network security

Traditionally, security has been considered as an important issue for infrastructural networks (i.e. networks in which dedicated components, such as routers, provide the basic network operation), especially for those running security-sensitive applications. To secure the network infrastructure and the applications designed for such type of dedicated networks, the following basic security services have been widely addressed by the research community: confidentiality, integrity, authentication, and non-repudiation. *Confidentiality* ensures that certain information is never disclosed to unauthorized entities. Network transmission of sensitive information, such as strategic or economically valuable information, requires confidentiality. Leakage of such information to eavesdropper could have severe consequences. *Integrity* guarantees that a message being transferred is never corrupted. A message could be corrupted because of benign failures or because of malicious attacks on the network. *Authentication* enables a node to ensure the identity of the peer node it is communicating with. Without authentication, an adversary could masquerade a node, thus gaining unauthorized access to resource and sensitive information and interfering with the correct operation of other nodes. Finally, *non-repudiation* ensures that the origin of a message cannot deny having sent the message. There are other security goals (e.g., authorization, intrusion detection, etc...) that are of concern to certain applications, but we will not pursue these issues in this thesis.
In the following, we present an overview of security requirements for ad hoc networks that

have their analogue in infrastructural networks and discuss on the basic mechanisms that enable their provision. We then present a new security exposure that is specific to the ad hoc networking paradigm and briefly introduce the mechanisms that meet the security requirements raised by such additional threat. Finally, we discuss on the increased difficulties in providing traditional security mechanisms imposed by an infrastructure-less environment and argue that the new security exposures initially thought of as specific to ad hoc networks can be extended also to infrastructural networks.

1.1.1 Traditional network security

Basic security services such as confidentiality, integrity, authentication, and non-repudiation have been the subject of intense research in the last thirty years. It is out of the scope of this section to provide a survey on the various type of cryptographic algorithms that have been proposed in the literature. Instead, we discuss on the salient aspects of the two main families of cryptographic algorithms that represent the key-enabler to deploy traditional security services in infrastructural networks.
The choice of keying material that is used to provide network security services is of fundamental importance. The cryptographic algorithms used by basic security mechanisms all rely on a *key management and distribution* service and can be based on **symmetric** or **asymmetric** keys. Depending on the key type, algorithms based on symmetric keys are called *symmetric algorithms* while algorithms based on asymmetric keys are called *asymmetric algorithms* or public key algorithms.
The choice of the algorithm type has an immediate impact on the requirements in terms of *computational power* available at each node of the network. It is known from the literature (see for example [88]) that computational power requirements are significantly higher for asymmetric algorithms. Moreover, the effects of the algorithm type can be extended to take into account the *energetic cost* related to the execution of the security protocol. Furthermore, depending on the algorithm type, a security protocol can introduce a non negligible *traffic overhead* in the network: the distribution and the utilization of keying material have an impact the length of the messages exchanged by the nodes thus leading to a significant waste of bandwidth resources.
On the other hand, security mechanisms based on asymmetric keying material have some interesting applications. The RSA cryptosystem is a seminal work that allowed the growth of modern asymmetric algorithms. RSA [98], is most commonly used for providing confidentiality and ensuring authenticity of digital data. These days, RSA is deployed in many commercial systems. It is used by Web servers and browsers to secure Web traffic, it is used to ensure confidentiality and authenticity of e-mail, it is used to secure remote login sessions, and it is at the heart of electronic credit card payment systems. In short, RSA is frequently used in applications where security of digital data is a concern. Traditional public key cryptosystems are based on the RSA cryptosystem and require any user's public key to be certified with a certificate issued and digitally signed by a certification authority (CA): this is done in order to provide authenticity of the public key belonging to a user. Any participant who wants to use a public key must first verify the corre-

sponding certificate to check the validity of the public key. When many CAs are involved between two users, trust relationships between those CAs also need to be verified. In fact, public key cryptosystems rely on a public key infrastructure (PKI) that is used in order to manage the trust relationship between entities (eventually in a hierarchical manner). However, traditional certificate-based schemes are affected by the main drawback that is key revocation: revocation of public keys and the corresponding certificate is a big issue and requires a large amount of storage and computing. Moreover, key revocation services often require some trusted authority to be online.

1.1.2 New security exposures in MANET

Ad hoc networks are distributed systems composed of autonomous entities, which need *cooperation* in order to work properly. In the ad hoc networking technology, mobile nodes equipped with wireless network interfaces freely and dynamically self-organize into temporary network topologies, allowing people and devices to communicate without any pre-existing network infrastructure. The underlying concept is the exploitation of synergy, resulting from collaboration among the network components, to provide services to each other.
The basic requirement for making operational the cooperative paradigm is the supposed contribution of all entities that compose and, at the same time, make use of the system. However, the recent deployment of ad hoc networks for civilian applications, relaxes the assumption on nodes cooperation. As long as applications of mobile ad hoc networks envision mainly emergency and military situations, all nodes in the network belong to a single authority and have a common objective. Therefore, cooperation among nodes can be assumed. In the context of civilian applications, the nodes typically do not belong to a single authority, and consequently nodes cooperation cannot directly be assumed. The lack of any centralized authority, guaranteeing the overall collaboration, motivates a possible tendency of entities to misbehave by not adhering to the cooperative paradigm. Generically, an entity that does not cooperate is called misbehaving. Cooperation misbehavior can be caused by entities that are *malicious* or *self-interested*. Malicious entities aim at breaking the proper network operation to intentionally damage others. This kind of misbehavior gives rise to several security problems that can be countered through traditional security services. As an example, a malicious entity can perform a routing disruption attack whereby the attacker sends forged routing packets to create a routing loop or to partition the network. This kind of attack can be prevented through the integrity and authenticity verification of routing control messages.
On the other hand, a self-interested entity does not intend to directly damage the overall system functioning, but is unwilling to spend its resources on behalf of others. From this kind of misbehavior arises a *new class of problems* that we group in non-cooperation problems. Lack of cooperation has gained much relevance in the context of ad hoc networking and several techniques have been proposed to counter this new type of threat. We present these techniques in Chapter 3

1.1.3 Discussion

Traditional network security mechanisms are of fundamental importance for ad hoc networks. Basic security services can be effectively used to counter the malicious behavior of entities aiming at jeopardizing the proper network operation. However, a close analysis of basic security requirements demanded by ad hoc networks reveals that the majority of requisites are not new nor specific to the ad hoc paradigm, apart from the fundamental problem of *key management and distribution*. The increased difficulty in providing the proper keying material to the entities of the system is due to the conjunction of several factors that range from the lack of physical security of the nodes to the lack of a-priori trust relationships among the users of the system and the lack of infrastructure. Classical security schemes developed for infrastructural networks rely on dedicated components with pre-defined roles: however, these fundamental requirements are not compatible with the basic definition of ad hoc networks whereby no component has a preassigned role.
It is important to recall that the choice of the cryptographic primitives used by any security scheme proposed for the ad hoc environment has to be pondered by considering specific requirements due to the nature of the mobile nodes forming the network: in some cases, computational power and storage capabilities can be considered as scarce resources. The eventual traffic overhead introduced by security schemes designed for a MANET also needs to be considered with a watchful eye. Bandwidth and energetic availability are another scarce resource that should not be wasted by suboptimal security schemes.
On the other hand, lack of cooperation translates in specific and new security requirements raised by ad hoc networks that apparently have not an analogy with infrastructural networks. Security issues addressed by the research community that focus on overlay networks and peer-to-peer systems, however, shows us that some similarities that link infrastructure-less and infrastructural networks can be found. Indeed, ad hoc networking shares many concepts, such as *distribution and cooperation*, with the peer-to-peer (P2P) computing model [101], which constitutes a natural paradigm for the ad hoc computing model. A defining characteristic of P2P systems is their ability to provide efficient, reliable, and resilient routing between their constituent nodes by forming structured "ad hoc" topologies on top of a real network infrastructure. The difference with traditional distributed computing systems is the lack of a central authority controlling the various components; instead, nodes form a dynamically and self-organizing system. The applications best suited for P2P implementation are those where centralization is not possible, relations are transient, and resources are highly distributed [91]. In particular, the range of applications covered by the P2P model includes file sharing, distributed search and indexing, resource storage, collaborative work, etc. The key aspect of P2P systems is the ability to provide inexpensive but at the same time scalable, fault tolerant, and robust platforms. For example, P2P sharing systems, like Gnutella [4], are distributed system where the contribution of many participants with small amounts of disk space results in a very large database distributed among all participant nodes. The voluntary nature of nodes' contribution has also serious drawbacks as the system's resources can be highly variable and unpredictable. Furthermore, the lack of a central authority coordinating the resources that each peer should contribute, leads users to use the system without

contributing much to it. The non-cooperation problem, *free-riding* in P2P terminology, highly affects the system performance, leading it to turn its peer-to-peer spirit in a more traditional client-server one [100].
The discussion alleged in this section suggests that the research presented in the reminder of this thesis have a broader range of application than initially expected: we believe that the problems tackled in our work can be extended to infrastructural networks and we hope to provide a basis for new research that focus on higher layers of the protocol stack.

1.2 Research objectives

The provision of security services for the ad hoc network layer represents a fertile ground in which researchers have recently expressed their creativity. The challenging scenario offered by ad hoc networks where there is *no a-priori trust* between the nodes (i.e., *open ad hoc networks*) motivates the initial research presented in this thesis. Indeed, in an open setting, classical security mechanism available for traditional networks cannot be used. In networks that rely on dedicated nodes (operated by trusted or well-known authorities) to carry out basic networking functions (e.g., routing or packet forwarding), *entity authentication can be sufficient* to guarantee the correct execution of those functions. On the contrary, if the nodes of the network are operated by different authorities that cannot be trusted for the execution of basic networking functions and that do not pursue a common goal, *entity authentication is not sufficient* and a new set of security requirements have to be defined together with appropriate security mechanisms that are adapted to an infrastructure-less environment.
The work presented in this thesis initially focuses on the definition of security requirements specific to *open ad hoc networks*, based both on traditional attacks carried out by illegitimate or compromised nodes and on a new type of *node misbehavior* that rise in a highly cooperative environment. Security solutions that cope with traditional node misbehavior, i.e. *malicious* nodes that *actively* thwart the normal execution of basic networking functions, demand a thoughtful assessment of the particular requirements inherent to an infrastructure-less and organization-less network.
On the other hand, a new type of node misbehavior that is highlighted throughout this thesis requires the definition of new security paradigms. The operating principle of ad hoc networks renders *cooperation among nodes* an essential requirement. Cooperation is intended as the willingness of a node to perform networking functions *for the benefit of other nodes*. However, cooperation has an non-negligible energetic cost that can lead to a *selfish behavior*, especially in a battery powered environment such as mobile ad hoc networks. Indeed, there is no reason to assume that nodes will participate in the network operation if, for example, turning the forwarding function off would noticeably extend their battery life, that could be used for *self-interested* purposes.
The security requirements defined in this thesis lead to the conclusion that both malicious and selfish misbehavior must be taken into account to provide a comprehensive set of network security services. Consequently, the research objectives developed in this thesis

focus on the design of security mechanisms that cope with selfish and malicious behavior of the nodes.
The first security mechanism proposed in this work provides **incentives** for nodes to cooperate. Our *cooperation enforcement mechanism* does not prevent a node from denying cooperation or deviating from a legitimate behavior but ensures that misbehaving nodes are **punished** by gradually withholding communication services. The performance of our cooperation mechanism has been assessed through a simulation-based evaluation. However, a simulation-based analysis has shown not to provide sufficient evidence of the correctness of our solution due to a limitation in modeling a realistic selfish behavior of the nodes forming the network. Therefore our research is devoted to the development of a formal framework in which the interaction between self-interested nodes can be faithfully modeled taking into account both network characteristics and the mechanism used to stimulate cooperation.
The second security mechanism presented in this thesis copes with malicious behavior of nodes that aim at disrupting the normal network operation. Our research goal consists in proposing a solution tailored to an open environment that neither relies on an established network infrastructure nor requires an organizational structure among the end-users operating the nodes of the network.

1.3 Thesis organization

In Chapter 2, we present the security requirements that are specific to the ad hoc networking environment. We show the attacks that can be performed by *malicious* nodes against ad hoc routing protocols with an emphasis on their effect on the reactive protocols outlined in section 2.1. Further, we introduce a new type of attack that consists in lack of cooperation. We illustrate the impact of such misbehavior on network performance in terms of aggregate throughput and aggregate delay and study the consequences of node mobility when self-interested nodes that *selfishly* minimize power consumption are present in the network.

In Chapter 3, we illustrate the building blocks for ad hoc network security available in the literature and discuss their properties. We argue that countermeasures that cope with selfish and malicious behavior need to be tightly coupled in order to provide a complete set of security services for the ad hoc network layer. Furthermore, we present our vision about research challenges we believe interesting to target and remark that security solutions presented in the literature are sometimes inappropriate or not realistic.

In Chapter 4, we follow the research recommendations discussed in the previous Chapter and present our cooperation enforcement mechanism (CORE) that copes with selfish behavior. The design of the mechanism as well as significative examples and a thoughtful discussion of the mechanism's features are detailed.

Chapter 5 provides both a numerical and an analytical validation of CORE. First, we introduce a set of suitable performance metrics that we use to validate our cooperation enforcement mechanism. A simulation-based evaluation is carried out to provide significative insight on the basic properties of our scheme. Furthermore, we develop a formal framework based on game theory that is used to model the (ad hoc) network and the interaction between the users in terms of forwarding behavior. We also define an *evolutionary simulation environment* in which nodes that perform better (in game theoretical terms) survive whereas nodes that are not following a suitable forwarding policy disappear. Based on this framework, we assess the properties of CORE by analyzing the evolution of nodes running CORE that compete with selfish nodes in an evolutionary simulation environment.
Further, we extended the game theoretical modeling to take into account topology information and derive a complementary cooperation enforcement policy that is inspired by CORE but provides a milder punishment, as exposed in [Annexe 1].
Finally, we introduce the concept of nodes coalitions and use cooperative game theory to model the interaction between coalitions of cooperating nodes and coalitions of selfish nodes. We then provide a thoughtful discussion to understand how our cooperation enforcement mechanism can be translated into a coalition formation algorithm.

In Chapter 7, we follow the research recommendations discussed in Chapter 3 and present a viable solution (that we called IDHC) for node authentication in ad hoc networks. Our scheme is based on an original concept that combines a simple form of identity-based cryptography with the Lamport's keyed hash chain method. The security of the proposed scheme as well as a performance evaluation is detailed. Furthermore we propose an application of IDHC that enables the secure discovery and maintenance of multi-hop paths when using the dynamic source routing protocol (DSR) in an open ad hoc network environment.

Chapter 2

Security requirements for mobile ad hoc networks

An ad hoc network consists of a set of wireless nodes that act both as data terminals and data transfer equipments in order to spontaneously form a temporary network without relying on any dedicated infrastructure.
Security of ad hoc networks recently gathered much attention from the research community and a large number of solutions to protect ad hoc networks against various types of attacks have been published. Security in ad hoc networks is a severe problem due to the conjunction of several factors:

- **lack of physical security**: vulnerabilities due to radio communications and the ease of eavesdropping and spoofing call for strong security mechanisms in order to get security that is equivalent to standard wireline communications;

- **lack of a-priori trust**: most ad hoc networks consist of a set of nodes that are not part of any shared organization thus classical security paradigms based on pre-established trust among the parties are not applicable;

- **lack of infrastructure**: security solutions based on dedicated components with pre-defined roles such as trusted third parties and key servers are not compatible with the basic definition of ad hoc networks whereby no component has a pre-assigned role;

- **requirement for cooperation**: due to the lack of dedicated components like routers and servers implementing network services, basic networking functions have to be carried out in a distributed fashion by the collaboration of a set of ordinary nodes, thus the performance of basic network operations like packet forwarding and routing can be strongly affected by malicious or accidental lack of cooperation.

Research on ad hoc network security initially focused on routing protocols since these were deemed the most critical part of network control. However, a close analysis of routing security works reveals that the majority of the requirements addressed by the solutions and the suggested mechanisms are not new or specific to ad hoc networks, apart from a fundamental problem that was often left aside by routing security solutions, that is, key management with no a-priori trust and lack of infrastructure.
In this Chapter we present a classification of attacks targeted to the ad hoc network layer. First, we focus on attacks to the routing mechanisms: we consider both *external attackers* that maliciously join the network to disrupt its basic functioning and *internal attackers* that compromise legitimate nodes to blast their proper operation. In the rest of this Chapter, attacks that require a computational effort and that cost energy are defined as *active attacks*. The consequences of active attacks are analyzed in order to derive a set of prominent **routing security** requirements that are demanded by ad hoc routing protocols. We argue that security has to be taken into account at the early stages of the design of ad hoc routing protocols. However, if the design of secure routing protocols for MANET has not been a particularly demanding task, the provision of basic security services needed as **building blocks** for the secure routing protocol itself is still the greatest challenge that researchers are facing. In order to mitigate the complexity introduced by the pure ad hoc paradigm, the solutions proposed in the Chapter 3 rely on particular (and frequently limiting) assumptions that ease the task of bootstrapping security associations needed prior to the secure routing protocol execution. As a consequence, our attention is particularly focused on the security requirements demanded by self-organized **key management services** that can be used by secure routing protocols.
Moreover, we introduce a new type of attack that does not require any computational effort nor cost energy to the attacker, which we call *passive attack*. Passive attacks are specific to the ad hoc environment and are caused by *selfish nodes* that are inclined to save energy and computational power to carry out self-interested operations. A selfish behavior implies *lack of cooperation* that can take the form of systematic failures in participating to forwarding or routing operations. Selfish nodes do not intend to directly jeopardize the normal network operation, but the consequences of lack of cooperation have a severe impact on network performance. We provide a simulation-based analysis of exposures due to node selfishness and targeted to the routing and packet forwarding functions and further analyze the impact of node mobility in the presence of selfish nodes. Finally, we provide the guidelines and security requirements needed to design efficient **cooperation enforcement schemes** that mitigate the effects of a selfish behavior.

2.1 Background

An ad hoc network is formed when a set of wireless and eventually mobile nodes join together and create a network by agreeing to route messages for each other. There is no infrastructure in an ad hoc network, e.g., no centralized routers or administrative policy. Ad hoc routing protocols use the mobile nodes to route packets from a given source to

the corresponding destination by eventually using multi-hop paths. Due to the node mobility, the network topology can be highly dynamic; links are continuously established and broken due to movement of mobile nodes. As a consequence, conventional routing protocols designed for dedicated networks such as the Internet do not work in an ad hoc setting. Several ad hoc routing protocols have been proposed in the literature and performance-based comparative studies have been carried out by the research community in order to assess the properties of the different solutions.
In this thesis we limit our attention on two protocols that are under consideration by the IETF for standardization: DSR and AODV. However, a test-bed implementation of CORE built on top of the OLSR [33] routing protocol is under development.
Our attention is focused on *reactive routing* protocols for sake of simplicity and because of the inherent participatory design that characterize "on-demand" protocols. However, the cooperation enforcement scheme presented in the reminder of the thesis can be extended to cope also with *proactive routing* protocols.

Below is a brief overview of route acquisition and maintenance basics in the targeted protocols. Detailed and authoritative descriptions of the AODV and DSR algorithms can be found elsewhere [84], [48].

2.1.1 The Dynamic Source Routing Protocol (DSR)

The DSR lets the sender of data traffic determine the path for a packets travel toward a destination. The discovered path is listed in the data packet header and is called a *source route*. Each node in the network maintains a dynamic route cache in which it stores routes to other nodes in the network that it has learned either from initiating a route request (RREQ) to a destination, or from forwarding packets along active paths for other nodes. In addition to the route discovery procedure, a node may also learn routes by overhearing transmissions of packets along paths for which it is not an intermediate node. This optimization is called *promiscuous listening*, and is based on the promiscuous mode operation offered by an increasing number of wireless cards available on the market. When a node sends a packet to another node in the network, the sending node first checks its route cache for a source route to the destination node. If the sender has an entry for the destination node in its route cache, the sender inserts this source route into the packet's header, listing the addresses of nodes through which the packet will be forwarded on its path to the destination. The sending node transmits the packet to the first node on the route list. Upon receiving the packet, each intermediate node along the path forwards the packet to the next hop on the indicated path until the packet reaches the destination node.
If the sending node does not have a source route to the destination, it initiates the DSR route discovery process. This begins with a route request packet broadcast from the initiator node. Each route request packet is uniquely identified by the concatenation of the initiator address, the target address, and the request *ID*. Upon receiving a route request packet, if the node is either the target node or has a route to the target node

stored in its route cache, it responds with a route reply (RREP) packet to the initiator. The route reply is a route record of the nodes which constitute the path from initiator to target. If the receiving node does not have a route to the destination, it checks that it has not already processed this route request by checking the <*initiator address, request id*> pair of the route request. If it has processed this route request, the node discards the packet. Otherwise, if it is seeing this route request packet for the first time, the receiving node appends its address to the route record in the route request packet and forward broadcasts the packet.

2.1.2 The Ad hoc Distance Vector Routing Protocol (AODV)

In AODV, each mobile node discovers or maintains routing information to another node if it is actively communicating with that node, or if it is an intermediary between two end points. If a node does not lie on an active path between two nodes, it does not maintain routing information for that path. AODV dynamically maintains loop-free routes, even when links change on active routes.
Route discovery in AODV is achieved with a source-initiated broadcast message called a route request (RREQ). When the RREQ reaches either the destination or an intermediate node that has a valid route to the destination, a route reply message (RREP) is unicast back to the source. As the RREP propagates back to the source, intermediate nodes receiving the RREP update their routing tables with a route to the destination.
AODV maintains fresh routes by implementing a counter for each node called a *sequence number*. A node's sequence number is incremented each time the local connectivity for that node changes. A RREQ contains fields for both the source and destination nodes' sequence numbers, called the *source_ sequence_ num* and *destination_ sequence_ num*, respectively. The RREP also contains a field for the *destination_ sequence_ num*. When an intermediate node receives a RREQ, it determines the freshness of its routing table entry for the destination (if such an entry exists) by comparing the destination sequence number of the RREQ with the one of its routing table entry for the destination. The node then either responds to the RREQ with its own route to the destination (if that route is recent enough), or rebroadcasts the RREQ to its neighbors.
RREQ packets also contain a *broadcast_ id* field. This value is a counter maintained at each node that is incremented every time the node initiates a new RREQ. The *broadcast_ id*, together with the source node's IP address, serves as a unique identifier for a RREQ. Nodes can temporarily buffer this information for each received RREQ so that they can determine whether they have already processed a given RREQ.

2.2 Threats directed to ad hoc network layer

In this section we present a classification of the attacks directed to the network layer of a MANET based on the energetic cost that the attacker has to bear. We thus address *active*

attacks, that requires a non negligible energetic cost, and *passive attacks*, that in turn are perpetrated by selfish nodes that do not cooperate to the network operation in order to save precious battery life. We also present a more classical categorization of attacker types when considering active attacks but we believe fundamental to separate the threats that thwart the normal network operation based on energetic considerations since extending node's lifetime can be assumed as a major concern in a battery powered environment such as MANET. It is arguable that most of the users do have the required level of knowledge and skills to modify their nodes: nevertheless, our assumption is reasonable in the sense that, as a practical example, turning off the forwarding function is a very simple task and the experience of cellular networks shows that as soon as the nodes are under the control of end-users, there is a strong temptation to alter their behavior in one way or another.

2.2.1 Active attacks

In this section we consider attacks that are carried out in order to withhold the normal network operation by compromising the routing protocol.
Traditional routing security mechanisms consist of node and message authentication referring to an a-priori trust model in which legitimate routers are believed to correctly operate the network. In contrast, authentication of a node and its messages *does not guarantee* the correct execution of routing functions in *open environments* with lack of a-priori trust like a MANET.
In the existing ad hoc routing proposals, nodes are trusted in that they do not maliciously tamper with the content of protocol messages transferred among nodes. However, malicious nodes (i.e. active attackers) can easily perpetrate integrity attacks by altering protocol fields in order to subvert traffic, deny communication to legitimate nodes (denial of service) and compromise the integrity of routing computations in general. As a result, the attacker can cause network traffic to be dropped, redirected to a different destination or to take a longer route to the destination increasing communication delays. A special case of integrity attacks is spoofing, whereby a malicious node impersonates a legitimate node due to the lack of authentication in the current ad hoc routing protocols. The main result of spoofing attacks is the misrepresentation of the network topology that possibly causes network loops or partitioning. Lack of integrity and authentication in routing protocols can further be exploited through "fabrication" referring to the generation of bogus routing messages. Fabrication attacks cannot be detected without strong authentication means and can cause severe problems ranging from denial of service to route subversion.
A more subtle type of active attack [87] is the creation of a tunnel (or wormhole) in the network between two colluding malicious nodes linked through a private connection bypassing the wireless network. This exploit allows a node to short-circuit the normal flow of routing messages creating a virtual vertex cut in the network that is controlled by the two colluding attackers. The wormhole attack is a severe threat against ad hoc routing protocols that is particularly challenging to detect and prevent. In a wormhole attack, a malicious node can record packets (or bits) at one location in the network and tunnel them to another location through a private network shared with a colluding malicious

node.
Most existing ad hoc routing protocols, without some mechanism to defend them against the wormhole attack, would be unable to find consistent routes to any destination, severely disrupting communication. A dangerous threat can be perpetrated if a wormhole attacker tunnels all packets through the wormhole honestly and reliably since no harm seems to be done: the attacker actually seems to provide a useful service in connecting the network more efficiently. However, when an attacker forwards only routing control messages and not data packets, communication may be severely damaged.
As an example, when used against an on demand routing protocol such as DSR, a powerful application of the wormhole attack can be mounted by tunneling each RREQ message directly to the destination target node of the request. This attack prevents routes more than two hops long from being discovered because RREP messages would arrive to the source faster than any other replies or, worse, RREQ messages arriving from other nodes next to the destination than the attacker would be discarded since already seen.
Hu, Perrig and Johnson propose an approach to detect a wormhole based on *packet leashes* [87]. The key intuition is that by authenticating either an extremely precise timestamp or location information combined with a loose timestamp, a receiver can determine if the packet has traversed a distance that is unrealistic for the specific network technology used.
Temporal leashes rely on extremely precise time synchronization and extremely precise timestamps in each packet. The travel time of a packet can be approximated as the difference between the receive time and the timestamp. Given the precise time synchronization required by temporal leashes, the authors propose efficient broadcast authenticators based on symmetric primitives. In particular, they extend the TESLA broadcast authentication protocol [85] to allow the disclosure of the authentication key within the packet that is authenticated.
Geographical leashes are based on location information and loosely synchronized clocks. If the clocks of the sender and the receiver are synchronized within a certain threshold and the velocity of any node is bounded, the receiver can compute an upper bound on the distance between the sender and itself and use it to detect anomalies in the traffic flow. In certain circumstances however, bounding the distance between the sender and the receiver cannot prevent wormhole attacks: when obstacles prevent communication between two nodes that would otherwise be in transmission range, a distance-based scheme would still allow wormholes between the sender and the receiver. To overcome this problem, in a variation of the geographical leashes the receiver verifies that every possible location of the sender can reach every possible location of the receiver based on a radio propagation model implemented in every node.
In some special cases, wormholes can also be detected through techniques that don't require precise time synchronization nor location information. As an example, it would be sufficient to modify the routing protocol used to discover the path to a destination so that it could handle multiple routes: a verification mechanism would then detect anomalies when comparing the metric (e.g. number of hops) associated to each route. Any node advertising a path to a destination with a metric considerably lower than all the others could raise the suspect of a wormhole.
Furthermore, if the wormhole attack is performed only on routing information while drop-

ping data packets, other mechanisms can be used to detect this misbehavior. When a node does not correctly participate to the network operation by not executing a particular function (e.g. packet forwarding) a collaborative monitoring technique can detect and gradually isolate misbehaving nodes. Lack of cooperation and security mechanism used to enforce node cooperation to the network operation is the subject of section 3.2.3 in Chapter 3 and Chapter 4.
Misbehaving nodes can be part of the network and perform attacks by exploiting compromised nodes or by disrupting the normal routing operation (insider attacks) or can be unauthorized nodes aiming at causing congestion, propagate incorrect routing information, prevent services from working properly or shut them down completely (external attacks). These threats exist because of the inherently limited physical security of mobile ad hoc networks. Indeed, the wireless communication medium makes it easier to intercept communications and inject messages than in an equivalent wired network.
We now detail the attacks in terms of the AODV and DSR protocols that are introduced in section 2.1. These protocols are used as representatives of ad hoc on-demand (or reactive) routing protocols.

Threats using modification

Current routing protocols assume that nodes do not alter the protocol fields of messages passed among nodes. Malicious nodes can easily cause redirection of network traffic and denial of service (DoS) attacks by simply altering these fields or by injecting routing messages into the network with falsified values in these fields. For example, in the network illustrated in Figure 2.1, a malicious node M could keep traffic from reaching X by consistently advertising to B a shorter route to X than the route to X which C is advertising. Below we detail several of the attacks possible if particular fields of routing messages in specific routing protocols are altered or falsified. Such attacks compromise the integrity of routing computations. The first two attacks we present allow remote redirection, which allows an attacker to drop traffic, causing a denial of service attack, or forward the traffic on to a destination after eavesdropping or altering the data payload. The third attack is a simple denial of service attack.

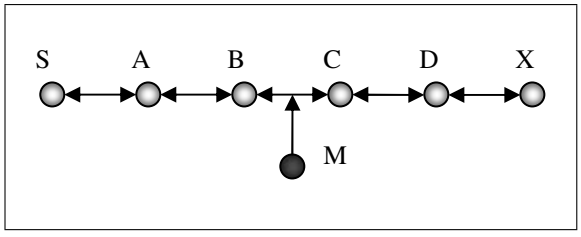


Figure 2.1: A simple ad hoc network with malicious node M

- Redirection with modified sequence number

Protocols such as AODV [84] and DSDV [42] instantiate and maintain routes by assigning monotonically increasing sequence numbers to routes toward specific destinations. We describe an attack on AODV that allows redirection of routes with the falsification of destination sequence numbers.
In AODV, any node may divert traffic through itself by advertising a route to a node with a *destination_sequence_num* greater than the authentic value. Figure 2.1 illustrates an example ad hoc network. Suppose a malicious node, M , receives the RREQ that originated from S for destination X after it is re-broadcast by B during route discovery. M redirects traffic towards itself by unicasting to B a RREP containing a significantly higher *destination_sequence_num* for X than the authentic value last advertised by X. Eventually, the RREQ broadcasted by B will reach a node with a valid route to X and a valid RREP will be propagated back toward S. However, at that point B will have already received the false RREP from M . If the *destination_sequence_num* for X that M used in the false RREP is higher than the *destination_sequence_num* for X in the valid RREP, B will drop the valid RREP, thinking that the valid route is stale. All subsequent traffic destined for X that travels through B will be directed toward M . The situation will not be corrected until either a legitimate RREQ or a legitimate RREP with a *destination_sequence_num* for X higher than that of M 's false RREP enters the network routing traffic.

- Redirection with modified hop counts

A redirection attack is also possible in certain protocols, such as AODV, by modification of the hop count field in route discovery messages. When routing decisions cannot be made by other metrics, AODV uses the hop count field to determine a shortest path. In AODV, malicious nodes can attract routes towards themselves by resetting the hop count field of the RREP to zero. Similarly, by setting the hop count field of the RREP to infinity, routes will tend to be created that do not include the malicious node. Such an attack is most threatening when combined with spoofing, as detailed later in this section.

- Denial of service with modified source routes

DSR is a routing protocol which explicitly states routes in data packets. These routes **lack any integrity checks** and a simple denial-of-service attack can be launched in DSR by altering the source routes in packet headers. Assume a shortest path exists from node S to node X as in Figure 2.2. Also assume that nodes C and X cannot hear each other, that nodes B and C cannot hear each other, and that node M is a malicious node attempting a denial of service attack. Suppose node S wishes to communicate with node X and that S has an unexpired route to X in its route cache. S transmits a data packet toward node X , with the source route $S \rightarrow A \rightarrow B \rightarrow M \rightarrow C \rightarrow D \rightarrow X$ contained in the packet's header. When the malicious node M receives the packet, it can alter the source route in the packet's header, such as deleting node D from the source route. Consequently, when

C receives the altered packet, it attempts to forward the packet to X . Since X cannot hear C, the transmission is unsuccessful. DSR provides a route maintenance mechanism such that a node forwarding a packet is responsible for confirming that the packet has been received by the next hop along the path. If no confirmation of receipt is received after retransmitting the packet a specified maximum number of attempts, this node should return a route error message to the source node. In this case, C would send a route error message to S. Since M would be the first hop the route error takes on its way back to S, M can continue the denial-of-service attack by dropping this route error message.

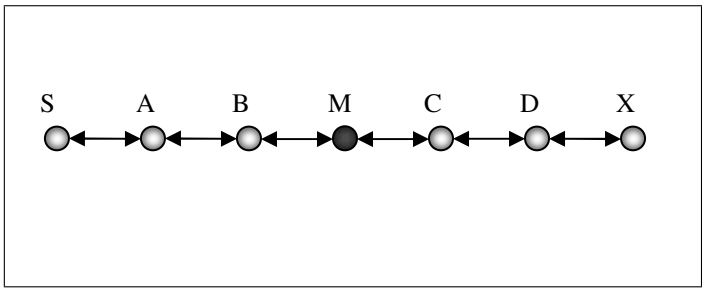


Figure 2.2: Another simple ad hoc network with malicious node M

DSR implements another route maintenance mechanism called *route salvaging* to recover from broken links along a path. When a break occurs, the node immediately upstream of the break can check its route cache, and if it has a different route to that destination, it can use that route instead. In Figure 2.2 C would check its route cache for an alternate route. If C only knows of the erroneous route to X , the DoS attack can be completed.
Modifications to source routes in DSR may also include the introduction of loops in the specified path. Although DSR prevents looping during the route discovery process, there are insufficient safeguards to prevent the insertion of loops into a source route after a route has been salvaged. Indeed, an intermediate node salvaging the path replaces the source route in the packet with a new route from its route cache. DSR prevents infinite looping in this case by allowing a packet to only be salvaged a finite number of times.

Threats using impersonation (spoofing attacks)

Current ad hoc routing protocols **do not authenticate** source IP addresses. By masquerading as another node, a malicious node can launch several types of attack in a network. This is commonly known as spoofing. Spoofing occurs when a node misrepresents its identity in the network, such as by altering its MAC (Medium Access Control) or IP address in outgoing packets. Spoofing is readily combined with modification attacks. The following example illustrates how this attack works in AODV. Similar attacks are possible in DSR.

- Forming loops by spoofing

Assume a path exists between the four nodes illustrated in Figure 2.3(a) towards destination X as would follow after an AODV RREQ/RREP exchange.
In this example, A can hear B and D; B can hear A and C; D can hear A and C; and C can hear B, D, and X . M can hear all nodes except X . A malicious attacker, M , can learn the topology by listening (for example by using the promiscuous mode operation of the 802.11 MAC layer) to the RREQ/RREP exchanges during route discovery. To start the attack, M changes its MAC address to match A's, moves closer to B and out of the range of A. It then sends a RREP to B that contains a hop count to X that is less than the one sent by C. B therefore changes its route to the destination to go through A, as illustrated in Figure 2.3(b).
M then changes its MAC to match B's, moves closer to C and out of range of B, and then sends to D an RREP with a hop-count lower than what X had sent. C then routes through B.
At this point (see Figure 2.3(c))a loop is formed and the network is partitioned while X is isolated. Our example shows how the attack is possible with a single malicious attacker, however, multiple attackers may collaborate for the same result.

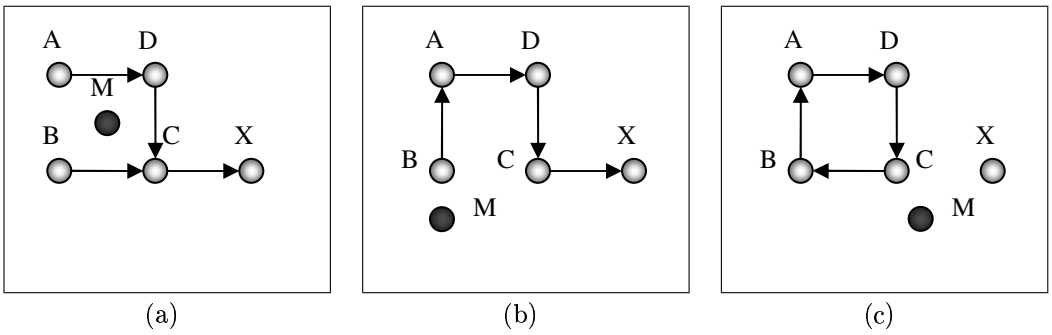


Figure 2.3: Topology loop forming by a spoofing attack by malicious node M

Threats using fabrication

We term the generation of *false routing messages* as fabrication attacks. Such attacks can be difficult to detect as invalid constructs, especially in the case of fabricated error messages claiming that a neighbor cannot be contacted, as we discuss below.

- Falsifying route error messages in AODV and DSR

AODV and DSR implement path maintenance mechanisms to recover broken paths when nodes move. If the source node moves, route discovery is re-initiated with a new route

request message if the route is still needed. If the destination node or an intermediate node along an active path moves, the node upstream of the link break broadcasts a route error message to all active upstream neighbors. The node also invalidates the route for this destination in its routing table (in DSR the source route is removed from the node's route cache). Upon receiving a route error message, if a node has an active route to this destination, and if the source of the route error message is the node's next hop along the path to the destination, the node deletes its routing table entry for the destination and forwards the route error message (the exception in AODV occurs when the RERR message has the"No delete flag" set. In this case the node forwards the RERR without deleting its routing table entry).
The vulnerability is that routing attacks can be launched by sending false route error messages. Suppose node S has a route to node X via nodes A, B, and C, as in Figure 2.1. A malicious node M can launch a denial of service attack against X by continually sending route error messages to B spoofing node C, indicating a broken link between nodes C and X . B receives the spoofed route error message thinking that it came from C. B deletes its routing table entry for X and forwards the route error message on to A, who then also deletes its routing table entry. If M listens and broadcasts spoofed route error messages whenever a route is established from S to X , M can successfully prevent communications between S and X.

- Route Cache Poisoning in DSR

Corrupting routing state is also an attack against routing integrity, but is passive in nature. This occurs when information stored in routing tables at routers is either deleted, altered or injected with false information.
Since the routers in an ad hoc network are the nodes themselves, centralized defense is not possible. Poisoning route caches is a common example of this attack. When a router forwards traffic to a destination listed in its routing tables, the router may unknowingly be forwarding the traffic to a false address.
The following details such an attack in DSR. In addition to learning routes from headers of packets which a node is processing along a path, routes in DSR may also be learned from promiscuously received packets. A node overhearing any packet may add the routing information contained in that packet's header to its own route cache, even if that node is not on the path from source to destination. For example, in Figure 2.1 a path exists from node S to node X via nodes A, B, and C. If a packet travelling along the source route from S to X is overheard by another node, that node may then add the route $< S, A, B, C, X >$ to its route cache. The vulnerability is that an attacker could easily exploit this method of learning routes and poison route caches. Suppose a malicious node M wanted to poison routes to node X . If M were to broadcast spoofed packets with source routes to X via itself, neighboring nodes that overhear the packet transmission may add the route to their route cache. Since this route discovery feature of caching overheard routing information is optional in DSR, this exploit can be easily patched by disabling this feature in the network. The downside of this is that without this feature DSR operates with a loss in its performance.

2.2.2 Passive attacks

In an *open ad hoc network*, nodes cannot rely on any a-priori trust relationship nor strong authentication infrastructure. As a consequence, the reliability of basic networking functions can be endangered by any node of the network. However, no classical security mechanism can help counter a misbehaving node in this context. The proper operation of the network requires not only the correct execution of critical network functions by each participating node (i.e., active attacks must be prevented) but it also requires each node to perform a *fair share* of the functions: all nodes must cooperate.
The latter requirement seems to be a strong limitation for wireless mobile nodes whereby power saving is a major concern. With lack of a-priori trust, *cooperative security schemes* seem to offer a reasonable solution to node selfishness. In a cooperative security scheme, malicious behavior can be detected through the collaboration between a number of nodes assuming that a majority of nodes do not behave maliciously.
In order to come up with the appropriate security requirements to build a viable countermeasure against *selfish nodes*, we analyzed the impact of various security threats on essential network functions through a simulation-based study.

Simulation-based analysis of node selfishness in MANET

Our simulation study has been carried out in order to analyze the effects of node selfishness on essential network functions such as routing and packet forwarding. We focused our attention on the evaluation of network performance in terms of aggregate *packet delivery ratio* and *communication delay* of a mobile ad hoc network where a predefined percentage of nodes were misbehaving.
The software we have used to simulate the MANET is the GloMoSim network simulator, which has been modified in order to simulate different types of selfish nodes. We assume that selfish nodes *operate independently* while attacks by several *colluding nodes are not taken into account*. Further, the DSR protocol is selected as the underlaying routing algorithm.

- Selfish node models

We propose three different models that have been evaluated for the DSR protocol. We believe that the selfishness problem is of great interest because nodes of a mobile ad hoc network are often battery-powered, thus, energy is a precious resource that they may not want to waste for the benefit of other nodes.
The node behavior has been added as a node definition type in the GloMoSim node model: the syntax that is used to define the node configuration has been enhanced with a new optional feature that allows selecting the selfishness model among three possible choices. It also has been necessary to modify the GloMoSim implementation of the network layer

(routing and packet forwarding) in order to override the routing protocol selected in the node configuration. Thus, the selfish behavior of a node can be independent from the routing protocol selected for the simulation.

Model 1: ***selfish forwarding.*** In the first model, a selfish node does not perform the packet forwarding function. When this behavior is selected, packet forwarding is disabled for all packets that have a (IP) source address or a destination address different from the current node. However, a selfish node that operates following this model participates in the Route Discovery and Route Maintenance phases of the DSR protocol. The consequence of the proposed model in terms of consumed energy is that the misbehaving node will save a significant portion of its battery life neglecting large data packets, while still contributing to the network maintenance.

Model 2: ***selfish routing.*** The second model focuses on selfish nodes that do not participate in the Route Discovery phase of the DSR protocol. The impact of this model on the network maintenance and operation can be more significant than the first one. Indeed, if the node does not participate in the Route Discovery phase, then the node will not be selected in any path: the consequence is that the packet forwarding function will never be executed. A selfish node of this type uses energy only for its own communications.

Model 3: ***energy-driven selfish behavior.*** The third model of selfishness is more complex: the node behavior follows the energy model implemented in GloMoSim. When the simulator creates an instance of a mobile node, it is possible to specify the initial energy (E_{in}) attributed to that node. During a normal operation, the node consumes energy while executing networking functions such as packet forwarding and routing.
We propose a selfishness model that uses two energy thresholds (e_{T1}, e_{T2}) to determine the node behavior. When the node's energy falls within the interval $[E_{in}, e_{T1})$ the node behaves properly, executing both the packet forwarding and the routing function. When the energy level falls in the interval $[e_{T1}, e_{T2})$ the node will behave as if it was a selfish node of type 1, thus disabling the packet forwarding function. If the energy level is within the interval $[e_{T2}, 0)$ then the same behavior as the one described for a selfish node of type 2 is selected. Whenever a node has no more energy it is possible to set a stochastic recharge phase: within a limited time interval the node's energy is set back to the initial value. We believe that this selfishness model is more realistic than the others; in our study, we evaluate the influence of simulation parameters such as node mobility over the global network performance when nodes behave following this selfishness model.

- Movement and communication patterns

In our simulations, the node chooses a destination and moves in a straight line towards it at a speed uniformly distributed between 1 meters/seconds (m/s) and some maximum

speed. This is called the *random waypoint* model. We limit the maximum speed of a node to 20 m/s and we set the run-time of the simulations to 50 seconds. Once the node reaches its destination it waits for a pause time before choosing another random destination and repeating the process. Additionally we developed a script that launches simulations with different random mobility scenarios for every simulation cycle.
The nodes communicate using constant bit rate (CBR) sources that are randomly bound to a subset of all the nodes forming the MANET. The packet size is set to 512 bits while the source throughput (expressed as packet per seconds) is different for each simulation. Additionally we developed a script that randomly chose sources and sinks among the nodes of a network and launches simulations with different random communication patterns for every simulation cycle.

NOTE: the random waypoint model is widely used as a mobility model to compare the performance of various mobile ad hoc network routing protocols. It has been shown in the literature [110] that the random waypoint model in its general form fails to reach a steady state in terms of instantaneous average node speed, but rather the speed continuously decreases as simulation progresses. Consequently, the model shouldn't be used to conduct performance evaluation measured as time averages. Such averages are based on metrics that change over time, sometimes substantially. Considering only these averages can result in misleading or incorrect conclusions.
To overcome these limitations, we set the minimum speed for the nodes to be different from 0 m/s (as it has been suggested in [110]). Furthermore we use a performance metric which is not averaged on time and the simulation time has been set to be short enough to mitigate the effects of nodes' speed drift.

- Performance metrics

The impact of selfish behavior was measured in terms of aggregate packet delivery ratio and communication delay. The measurements of the network performance were made using a script that parses and analyzes the trace file output provided by the GloMoSim software. The trace file provides information about a set of defined events that occurred in the simulation such as MAC layer events, routing events and application events. By analyzing the application event trace, it is possible to evaluate the total number of packets sent by every node of the MANET as well as the total number of packets that have been dropped. We used the following definition for the aggregate packet delivery ratio (PDR):

$$PDR = \frac{\text{Total \# of Received Packets}}{\text{Total \# of Sent Packets}} = \frac{\text{Total \# of Sent Packets - Total \# of Lost Packets}}{\text{Total \# of Sent Packets}}$$

The term "lost packets" covers all packet losses due to malicious drops, route failures, congestion and wireless channel losses.
The other performance characteristic that was measured is the average delay for all packets

that are correctly received (d). We believe that selfish nodes also have an influence on the network delay and we use it as a further comparative criterion when the network packet delivery ratio is not sufficient.

- Simulation results based on Model 1 and Model 2
  - Simulation configuration

    The effects of the selfishness models type 1 and type 2 are studied on different scenarios where the parameter that define each scenario is *node density*.[1] We define node density as the number of nodes that form the MANET deployed over an 800 by 800 meter flat space. On the other hand, node mobility is defined as the average speed each node moves at in the simulation space.
    Simulation results are classified in two categories: low node density (20 nodes) and low mobility (2 m/s), high node density (60 nodes) and low mobility. If it is not specified differently, the simulation run-time for all the families of graphs presented in this section is set to 50 seconds. Also, the CBR source throughput is set to 1 packet per second.
    The percentage of selfish nodes (p) is increased for each simulation run and takes values from 0% to 50%: in each simulation run, only p nodes are set to be selfish while the other nodes of the network behaves correctly.

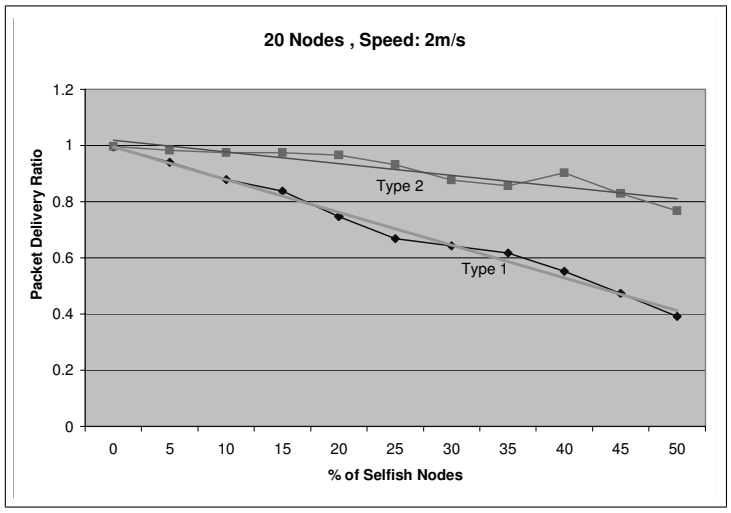
  - Results

    Figure 2.4 and Figure 2.5 respectively show the variations of aggregate packet deliver ratio (PDR) and communication delay (d) as a function of the percentage of selfish nodes.

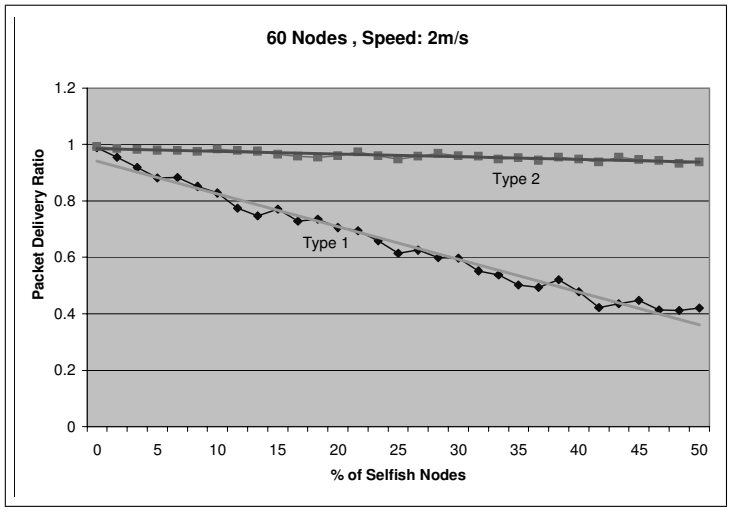
  - Observations

    Model 1: *selfish forwarding.* Figure 2.4(a) and 2.4(b) point out that PDR degrades by 60% when 50% of the nodes of the network misbehave. The linear regression of the experimental data shows that PDR degrades by 10%-15% every time the percentage of selfish nodes increases by 10%. On the other hand, Figure 2.5(a) and 2.5(b) show that d increases linearly with the percentage of selfish nodes. These observations are valid both for low and high node density, while node mobility have a negligible influence on the measurements.
    The results show that when the packet forwarding function is disabled by a

[1] *Node mobility* has been initially used as an alternative parameter to study the impact of selfish nodes. Simulation results [65] show however that the same conclusion exposed in this section apply when node mobility is used as a simulation parameter. For the sake of simplicity we report in this section results obtained when node mobility has an average of 2 m/s.



(a)



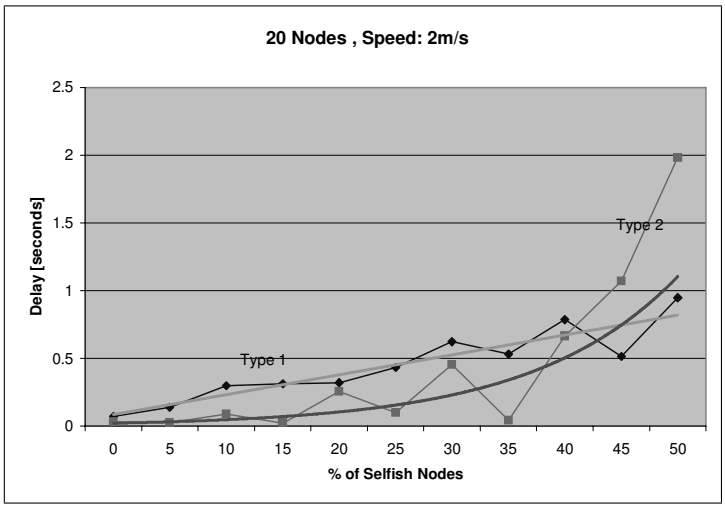
(b)

Figure 2.4: PDR for low and high node density, low mobility

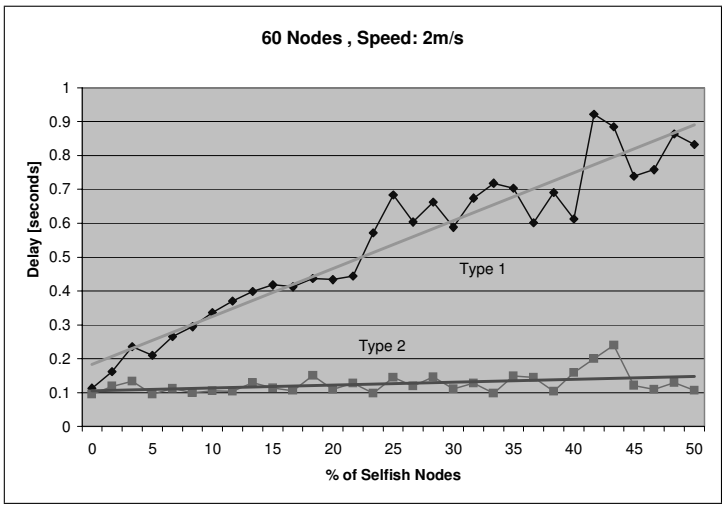
large percentage of nodes of the MANET the global network performance **severely degrade**.

Countermeasures for this type of selfishness have to be taken into account for the design of cooperative routing mechanisms. We claim that a cooperative security scheme offers a reasonable solution to the problem: an important requirement for the security mechanism consists in verifying the correct execution of the packet forwarding function and to enforce node cooperation.

Model 2: ***selfish routing.*** Figure 2.4(a) shows that PDR degrades by 20% when 50% of the nodes follow the second model of selfishness. Figure 2.5(a) indicates that d increases exponentially when the percentage of misbehaving nodes increase. On the other hand, when node density is high, it is possible to notice that network performance improve: PDR degrades only



(a)



(b)

Figure 2.5: d for low and high node density, low mobility

by 7%-10% and d has a linear growth.
When the Route Discovery phase of the DSR protocol is disabled, selfish nodes do not participate in the route construction and will never appear in a source route. The reason why the average packet delivery ratio degrades is that it is more difficult to find a route to the destination leading to a higher packet loss probability. Furthermore, delays are high, especially when node density is low. Depending on the MANET topology, it is possible that selfish nodes partially or totally isolate a legitimate node by not providing any routing information; the result is that the legitimate node loses both time and packets to find another route (if it exists) towards the destination. When node density is high, the effect of the selfish behavior is mitigated and the probability to find a route to the destination increases: packet loss and communication delay decrease.

Model 1 vs. Model 2 The analysis of the results obtained with the first two families of simulations indicate that the effects of a node selfishness of type 1 are more important than the one caused by a selfishness of type 2. The apparent conclusion is that a mechanism to enforce cooperation in MANET has to focus on the first type of selfishness, obliging misbehaving nodes to correctly perform the packet forwarding function.
However, if a selfish node does not participate in the Route Discovery phase of the DSR then it will never appear in any source route. It is implicit then that also the packet forwarding function will not be executed, thus a mechanism that simply force a node to perform the packet forwarding function can be easily tricked by disabling the DSR function. On the other hand, a mechanism that only force a selfish node to correctly perform the DSR function does not assure that also the packet forwarding function will be properly executed.
Concluding, it is necessary that the security scheme adopted to face the selfish behavior of a node have to enforce the execution of **both** the packet forwarding and the routing functions. Moreover, we believe that a selfish behavior that selectively disables the packet forwarding or the routing function is not realistic: it is more likely that the node behavior dynamically changes depending on the node's energy level. This is why we decided to design a third type of selfishness that is based on the energy model implemented in GloMoSim.

- Simulation results based on Model 3
  - Simulation configuration

    The last set of simulations focuses on the analysis of the network performance of a MANET when the third model of selfishness is used. Movement and communication patterns have been modified with respect to the one used in the first two families of simulations.
    Node movements follow the random waypoint model but the speed is selected among a predefined set: we defined the set of possible speeds as the values that go from 1m/s to 20m/s with a step of 5m/s: 1, 5, 10, 15, 20.
    Nodes communicate using constant bit rate (CBR) sources that are randomly bound to a sub-set of all the nodes forming the MANET. The packet size is set to 512 bits while the source throughput (calculate as packet per seconds) is set to 20 packets/second. Additionally we developed a script that randomly chose sources and sinks among the nodes of a network and launches simulations with different random communication patterns for every simulation cycle. The global communication delay d has not been evaluated.

  - The energetic model

The third model of selfishness is an energy-based model: the initial value for the energy (E_{in}) associated to each node is defined through the node configuration file. We decided to set different values for E_{in} using a uniform distribution in the interval $[E_{in} - 0.25J, E_{in} + 0.25J]$ where the energy is expressed in Joules (J) and $E_{in} = 2.75J$. The consequence of this choice is that every node will run out of energy at different times, adding a degree of randomness to the simulation.

– Results

As opposed to the analysis made for the first and the second model of selfishness, we decided to focus on the *effects of node mobility* over the aggregate packet delivery ration (PDR). As it is possible to see in Figure 2.6, the first series represent PDR when all nodes of the MANET behaves correctly whereas the second series represents PDR when **every node** of the MANET behaves following the third model of selfishness.

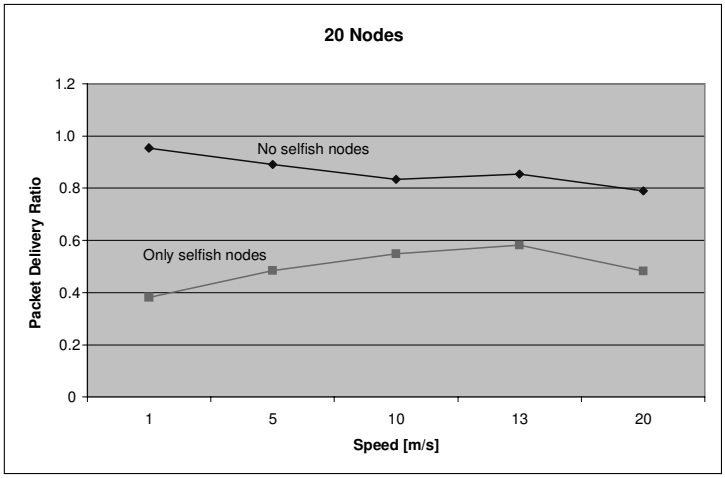


Figure 2.6: Impact of node mobility on aggregate packet delivery ratio under selfishness Model 3

– Observations

The last family of simulations pointed out an interesting characteristic of PDR. It has already been showed [47] that the global network throughput decreases when the node mobility increases: the reason is that link outage becomes more frequent causing a higher packet loss probability.
On the other side, when every node of the network is selfish (Model 3), simulation results indicate that PDR increases when node mobility increases until it reaches its maximum; then it decreases when node mobility increases.
We believe that this particular behavior depends on the mobile node topological position in the network. When the third model of selfishness is applied

to a mobile node, the node's behavior dynamically change depending on its energy level. Now, a MANET topology can be represented by an arbitrary graph $G = (V, E)$ where V is the set of (mobile) nodes and E is the set of edges. An edge exists if and only if the distance between two mobile nodes is less or equal than the node's radio range r. Accordingly, the neighborhood of a node x is defined by the set of nodes that are inside a circle with center at x and radius r, and it is denoted by

$$N_r(x) = N_x = \{n_j \,|d(x, n_j) < r, x \neq n_j, \forall j \in N, j \leq |V|\}$$

where x is an arbitrary node in the graph. The degree of a node x in G is the number of edges that are connected to x, and it is equal to $\deg(x) = |N_r(x)|$. Given that the communication pattern used in the simulation produce a dense traffic, a central node (i.e. a node that has a central position in the MANET) consume more energy than a peripheral node because it acts as relays for other nodes, wasting its energy for routing and packet forwarding. A central node has also another characteristic: its degree is high, which implies that nodes with a higher degree consume more energy than nodes with a lower degree.
When mobility is low, all nodes located in a central position stay in the central area of the network and consume more energy than peripheral nodes. Energy consumption leads to a selfish behavior: the packet forwarding and the routing functions will not be correctly executed and the network can be partitioned. As it is possible to see in Figure 2.6 for a 1m/s speed, the aggregate packet delivery ratio (PDR) is drastically reduced.
When node mobility increases, the location of a node changes from a central to a peripheral position and vice-versa with a high rate, implying that the energy consumption will be equally distributed among the nodes. The selfish behavior is mitigated and, as it is possible to see in Figure 2.6, PDR increases considerably.
However, when the node mobility reaches higher values the influence of the link outage over PDR is more important than the impact of a selfish behavior: speed affects negatively the network performance, as it is possible to see in Figure 2.6 for speed higher than 13m/s.

One interesting observation that is possible to draw from the literature on performance studies of ad hoc routing protocols [40], is that for the most protocols the packet delivery fraction uniformly decreases from low mobility rate to medium mobility rate (i.e. from pause time 900 seconds to 150) and then *fluctuates* as the mobility rate increases. This is because the occurrence of *multiuser diversity via relaying* increases with mobility as nodes are likely to be present ubiquitously. This multiuser diversity is best motivated by the information theoretic results of Knopp and Humblet [53]. They showed that the optimal strategy is to schedule at any one time only the user with the best channel to transmit to the base station. Grossglauer and Tse exploit multiuser diversity through relaying for mobile ad hoc networks [40]. The basic idea is for a source to distribute packets to as many different nodes as possible;

these nodes relay the packet to the final destination whenever they get close to the destination. Indeed, by using all the other node excluding the source and destination node as relays, the communication is then performed through *two* multiuser links: a downlink from the source to all the relays, and an uplink from the relays to the destination. Note that, the direct point-to-point link is a statistically poor channel, because it is only strong for a small fraction of time when the source-destination pair are close by. Therefore, due to a multiuser diversity effect, the throughput of the downlink is high because at any time, there is likely to be a relay node close to the source to whom the source can transmit the information. The same rule holds for the uplink in relation with the destination. Therefore, the expected path length remains constant. It has to be mentioned that the multiuser diversity via relaying require the information about the neighboring nodes and its results depends strictly on the movement model.

2.3 Security requirements and recommendations

This section concludes the Chapter on security requirements for mobile ad hoc networks. Based on the threats described in section 2.2 and their consequences on network functioning and performance, we derive the basic requirements needed to provide security services for the network layer of an *open* ad hoc network. Following, security requirements are organized in terms of the corresponding mechanisms that can be used to cope with *active* and *passive* attacks.
Active attacks performed on ad hoc routing protocols call for the design of **secure routing mechanisms**. However, the design of secure routing mechanisms for a pure ad hoc environment requires a thoughtful evaluation of the assumptions made in terms of **key management schemes**. Further, **bootstrapping security associations** needed prior to the secure routing protocol execution is a delicate task that have an impact on the viability of the security schemes that rely on a pre-distribution of keying material.
Similarly, the design of **cooperation enforcement mechanisms** that cope with node selfishness requires the definition of adequate guidelines that takes into account the distributed nature of the network and the limitations in terms of computational power and available energy to the nodes forming the network.

2.3.1 Secure routing requirements

Based on the attacks presented in section 2.2.1 a secure routing protocol must satisfy the following requirements to ensure that a path discovery from source to destination is correctly executed in the presence of malicious adversaries:

1. Route signalling cannot be spoofed;
2. Fabricated routing messages cannot be injected into the network;
3. Routing messages cannot be altered in transit, except for those changes that must be made according to the normal execution of the routing protocol;
4. Routing loops cannot be formed through malicious action;
5. Routes cannot be redirected from the shortest path through malicious action.

In addition to the these basic requirements, a secure routing protocol relies on the deployment of a key management scheme for the distribution and management of the keying material needed during the protocol execution. In the following section, we define the basic requirements for the deployment of key management schemes targeted to open ad hoc networks and we discuss about the choice of the keying material type (symmetric or asymmetric) based on computational power requirements and bandwidth utilization.

2.3.2 Key management requirements

A close analysis of secure routing requirements reveals that the majority of the requisites defined in section 2.3.1 are not new or specific to ad hoc networks, apart from the fundamental problem of key management with no a-priori trust and lack of infrastructure. We now present the requirements needed for the deployment of a key management scheme in a truly open environment, that is, where users that operate the nodes are not part of any common organizational structure and where dedicated components with a predefined role such as centralized servers or trusted third parties cannot exist.
A key management scheme for open ad hoc networks must:

1. Follow distributed approach;
2. Provide high availability;
3. Do not rely on any fixed or pre-established network infrastructure;
4. Do not rely on an organizational structure among the users that operate the nodes of the network;
5. Do not rely on an organizational structure between users and the security infrastructure;

The choice of keying material that is used to provide network security services is of fundamental importance. The cryptographic algorithms used by the security mechanisms

that rely on the key management and distribution service can be based on **symmetric** or **asymmetric** keys. Depending on the key type, algorithms based on symmetric keys are called *symmetric algorithms* while algorithms based on asymmetric keys are called *asymmetric algorithms* or public key algorithms.
The choice of the algorithm type has an immediate impact on the requirements in terms of *computational power* available at each node of the network. It is known from the literature (see [88]) that computational power requirements are significantly higher for asymmetric algorithms. As a consequence, security protocols for ad hoc networks should be based on symmetric algorithms: indeed, nodes can be assumed to have a limited capacity in terms of computational power.
However, the effects of the algorithm type can be extended to take into account the *energetic cost* related to the execution of the security protocol: again, security services based on symmetric algorithms seems to be better tailored to an environment where energy is a scarce resource. Furthermore, depending on the algorithm type, a security protocol can introduce a non negligible *traffic overhead* in the network: the distribution and the utilization of keying material have an impact the length of the messages exchanged by the nodes thus leading to a significant waste of bandwidth resources.
As a final remark, key management services need to be designed taking into account *availability requirements*: in a MANET, the network topology can be highly dynamic leading to temporary network partitioning or partial isolation of peripheral nodes. Security services must tackle with network dynamics and the underlying security protocols have to minimize the negative effects of **unreliable communications** typical of a wireless environment.

2.3.3 Cooperation enforcement requirements

In this section we summarize the basic requirements needed for the design of a viable cooperation enforcement mechanism, that is a mechanism that copes with selfish behavior of nodes.
A cooperation enforcement mechanism designed for an open ad hoc network must:

1. Follow a distributed approach;
2. Provide incentives to cooperate **both** in packet forwarding and routing;
3. Punish misbehaving nodes;
4. Be lightweight;
5. Be efficient in terms of power consumption;
6. Be robust: a malicious user must not be allowed to use the mechanism itself to attack the network;
7. Be secure: spoofing must be prevented.

Energetic requirements for the execution of the cooperation enforcement mechanism deserve particular attention. Detection of selfish behavior and incentives to cooperate should be provided by a lightweight mechanism that *do not waste limited energetic resources*. Indeed, it is realistic to assume that selfish behavior appears when the battery lifetime is limited. The condition in which the nodes of the network execute a heavy mechanism to cope with a non-cooperative behavior would fall into a paradox: wasting a considerable amount of energy to enforce node cooperation would have the undesirable effect to mutate a legitimate node into a selfish node. Moreover, it is also realistic to assume a MANET to be formed by **heterogeneous devices** with different energetic capabilities. In an heterogeneous setting, the proper operation of the network requires each node to perform a *fair share* of networking functions, proportional to the energetic availability of every node.
Furthermore, eventual *traffic overhead* originated by the cooperation enforcement mechanism should be *limited*, otherwise the gain in network performance as a consequence of a more cooperative behavior of the nodes would be reduced.
As a concluding remark, the cooperation enforcement mechanism should be **robust** in the sense that it must be impossible for a malicious node to launch an attack by exploiting the cooperation mechanism itself. Moreover, **identity spoofing** must be prevented: nodes that have been detected as selfish by the cooperation enforcement mechanism must not be able to modify their identity in order to trick legitimate nodes and persist with a non-cooperative behavior.

2.4 Summary

In this Chapter we presented a complete analysis of the threats directed to the ad hoc network layer and their consequences on the network operation and performance. Countermeasures that cope with malicious and selfish behavior of the nodes have been defined in terms of security requirements for an open environment.
Section 2.3 shows a high degree of overlapping in the prerequisites demanded by the network security mechanisms: for example secure routing protocols rely on key management services for the bootstrapping of security associations needed during the protocol execution; in the same way, a cooperation enforcement mechanism rely on key management services in order to prevent spoofing attacks. In the next Chapter we discuss on the need for a tight integration of network security mechanisms that constitute the basic **building blocks** for the definition of a network security architecture. We then present the related research that has been carried out in the literature in order to provide security services for the ad hoc network layer.

Relevant publication

P. Michiardi and R. Molva. Simulation-based analysis of security exposures in mobile ad hoc networks. In *Proceedings of the European Wireless Conference*, Florence, Italy, February 2001.

Chapter 3

Building blocks for ad hoc network security

3.1 Introduction

Ad hoc network security deals with threats that range from *active* to *passive* attacks. The fundamental security requirements presented in Chapter 2 reveal the need for securing the execution of ad hoc routing protocols as well as the necessity for incentive schemes that promote node cooperation in the network operation. Secure routing protocols rely on the availability of security associations among the nodes: authentication of peer entities involved in routing and the integrity verification of routing messages call for a key management scheme to provide the parties involved in the protocol execution with proper keying material. Cooperation enforcement schemes also require peer authentication in order to prevent spoofing attacks: again, key management schemes offer the solution to prevent attacks using impersonation. Furthermore, information on the behavior of the nodes that participate to the network operation collected through cooperation enforcement mechanisms can be optionally used by secure routing protocols to modify the route selection criterion in order to avoid potentially harmful nodes, a technique that goes under the name of "path rating" [61].
In this Chapter, we present current efforts available in the literature towards the design of basic building blocks for securing the ad hoc network layer, i.e. secure routing, key management and cooperation enforcement mechanisms, as presented in [77], [68], [71] and [73]. We argue that an exhaustive bundle of ad hoc security services call for a tight integration of these basic building blocks that only address a specific subset of all the possible threats. Indeed, overlapping requirements demanded by different security mechanisms and their evident inter-dependence call for an exact determination of the interface between the various building blocks mainly to address the difficult problem of the network bootstrap in the absence of an external support such as a network infrastructure or a shared organization between the users of the system.

The definition of a complete security architecture is out of the scope of this thesis, but we hope to provide a fundamental set of guidelines to enable an efficient and secure architecture design for the challenging scenario offered by open ad hoc networks. Furthermore, we conclude the chapter with a discussion on the research challenges imposed by open ad hoc networks: we claim that major effort has to be invested in the design of cooperation enforcement mechanisms and key management schemes. The discussion justifies the reminder of the thesis, in which we propose our solutions to provide an incentive and a key management scheme for open ad hoc networks.

3.2 Related research

Research in ad hoc network security has been particularly active in recent years and a multitude of security mechanisms that address secure routing, key management and distribution and cooperation enforcement issues have been proposed. In the following sections, the most significant approaches available in the literature are outlined. For each approach, we provide a description of the fundamental assumptions made by the authors as well as a discussion on the advantages and drawbacks inherent to the design of each proposals. Our remarks are made based on the security requirements provided in Chapter 2.

3.2.1 Secure routing proposals

In this section we present a limited set of current approaches that target the security of ad hoc routing protocols. The selection was made in order to cover a wide range of cryptographic techniques allowing a clear discernment of the advantages and drawbacks that the underlying cryptographic primitives imply.
Table 3.1 summarizes the different approaches presented hereafter.

- Secure routing protocol (SRP)

The Secure Routing Protocol (SRP) [82] is designed as an extension compatible with a variety of existing *reactive* routing protocols. SRP combats attacks that disrupt the route discovery process and guarantees the acquisition of correct topological information: SRP allows the initiator of a route discovery to detect and discard bogus replies. SRP relies on the availability of a security association (SA) between the source node (S) and the destination node (T).
The SA could be established using a **hybrid key distribution based on the public keys** of the communicating parties. S and T can exchange a secret symmetric key

Name	Routing type	Cryptographic Primitives	Bootstrap phase	Security Services
SRP	Reactive	Symmetric keys	Pair wise key distribution between *all* pairs of communicating parties	Authentic network topology discovery
ARIADNE	Reactive	Symmetric keys Hash chains TESLA keys	Pair wise key distribution between all possible <source,destination> Public key infrastructure needed for TESLA keys authentication	<source,destination> authentication Route discovery and maintenance authentication
ARAN	Reactive	Asymmetric keys Public key certificates	Public key infrastructure, distribution of public key certificates	Authentication, integrity and non-repudiation of routing messages (Optional: secure shortest path)
SEAD	Proactive	Symmetric keys Hash chains	Authentic distribution of hash chain seed	Authentication of routing control traffic

Table 3.1: Summary of secure routing proposals available in the literature

($K_{S,T}$) using the *public keys* of one another to establish a secure channel. S and T can then further proceed to mutual authentication of one another and the authentication of routing messages. SRP copes with non-colluding malicious nodes that are able to modify (corrupt), replay and fabricate routing packets.
In case of the Dynamic Source Routing (DSR) protocol [48], SRP requires including a 6-word header containing unique identifiers that tag the discovery process and a message authentication code (MAC) computed using a keyed hash algorithm.
In order to initiate a route request (RREQ) the source node has to generate the MAC of the entire IP header, the basic protocol RREQ packet and the shared key $K_{S,T}$. The intermediate nodes that relay the RREQ towards the destination measure the frequencies of queries received from their neighbors in order to regulate the query propagation process: each node maintains a priority ranking that is inversely proportional to the query rate. A node that maliciously pollutes network traffic with unsolicited RREQ will be served last (or ignored) because of its low priority ranking.
Upon reception of a RREQ, the destination node verifies the integrity and authenticity of the RREQ by calculating the keyed hash of the request fields and comparing them with the MAC contained in the SRP header. If the RREQ is valid, the destination initiates a route reply (RREP) using the SRP header the same way the source did when initiating the request. The source node discards replays that do not match with pending query identifiers and checks the integrity using the MAC generated by the destination.
The basic version of SRP suffers from the route cache poisoning attack: routing information gathered by nodes that operate in promiscuous mode in order to improve the efficiency of the DSR protocol could be invalid, because of potential fabrication by malicious nodes.

The authors propose two alternative designs of SRP that use an Intermediate Node Reply Token (INRT). INRT allows intermediate nodes that belong to the same group that share a common key (KG) to validate RREQ and provide valid RREP messages.
SRP suffers also from the lack of a validation mechanism for route maintenance messages: route error packets are not verified. However, in order to minimize the effects of fabricated error messages, SRP source-routes error packets along the prefix of the route reported as broken: the source node can thus verify that each route error feedback refers to the actual route and that it was originated at a node that is part of the route. A malicious node can harm only routes it actually belongs to.
Assuming that the neighbor discovery mechanism maintains information on the binding of the medium access control and the IP addresses of nodes, SRP is proven to be essentially immune to IP spoofing [82].
SRP is, however, not immune to the wormhole attack: two colluding malicious nodes can misroute the routing packets on a private network connection and alter the perception of the network topology by legitimate nodes.

- ARIADNE

In [88] Hu, Perrig and Johnson present an on-demand secure ad hoc routing protocol based on DSR that withstands node compromise and relies only on *highly efficient symmetric cryptography*. ARIADNE guarantees that the target node of a route discovery process can authenticate the initiator, that the initiator can authenticate each intermediate node on the path to the destination present in the RREP message and that no intermediate node can remove a previous node in the node list in the RREQ or RREP messages.
As for the SRP protocol, ARIADNE **needs some mechanism to bootstrap authentic keys required by the protocol**. In particular, each node needs a shared secret key ($K_{S,D}$, is the shared key between a source S and a destination D) with each node it communicates with at a higher layer, an authentic TESLA [85] [86] key for each node in the network and an authentic "Route Discovery chain" element for each node for which this node will forward RREQ messages.
ARIADNE provides point-to-point authentication of a routing message using a message authentication code (MAC) and a shared key between the two parties. However, for authentication of a broadcast packet such as RREQ, ARIADNE uses the *TESLA broadcast authentication protocol*. ARIADNE copes with attacks performed by malicious nodes that modify and fabricate routing information, with attacks using impersonation and, in an advanced version, with the wormhole attack. Selfish nodes are not taken into account.
In ARIADNE, the basic RREQ mechanism is enhanced by eight additional fields used to provide authentication and integrity to the routing protocol as follows:

$$< \text{ROUTE REQUEST}, initiator, target, id,$$
$$\text{time interval}, \text{hash chain}, \text{node list}, \text{MAC list} >$$

The *initiator* and *target* are set to the address of the initiator and target nodes, respectively. As in DSR, the initiator sets the *id* to an identifier that it has not recently

used in initiating a Route Discovery. The time interval is the TESLA time interval at the pessimistic expected arrival time of the request at the target, accounting for clock skew. The initiator of the request then initializes the hash chain to $MAC_{K_{S,D}}$ $(initiator, target, id,$ time interval) and the node list and MAC list to empty lists.
When any node A receives a RREQ for which it is not the target, the node checks its local table of $< initiator, id >$ values from recent requests it has received, to determine if it has already seen a request from this same Route Discovery. If it has, the node discards the packet, as in DSR.
The node also checks whether the time interval in the request is valid: that time interval must not be too far in the future, and the key corresponding to it must not have been disclosed yet. If the time interval is not valid, the node discards the packet. Otherwise, the node modifies the request by appending its own address (A) to the node list in the request, replacing the hash chain field with $H[A,$ hash chain], and appending a MAC of the entire REQUEST to the MAC list.
The node uses the TESLA key K_{A_i} to compute the MAC, where i is the index for the time interval specified in the request.
Finally, the node rebroadcasts the modified RREQ, as in DSR. When the target node receives the RREQ, it checks the validity of the request by determining that the keys from the time interval specified have not been disclosed yet, and that the hash chain field is equal to:

$$H[\eta_n, H[\eta_{n-1}, H[\ldots, H[\eta_1, MAC_{K_{S,D}}(initiator, target, id, timeinterval)]\ldots]]]$$

where η_i is the node address at position i of the node list in the request, and where n is the number of nodes in the node list. If the target node determines that the request is valid, it returns a RREP to the initiator, containing eight fields:

$$< \text{ROUTE REPLY}, target, initiator, \text{time interval},$$
$$\text{node list}, \text{MAC list}, \text{target MAC}, \text{key list} >$$

The *target*, *initiator*, time interval, node list, and MAC list fields are set to the corresponding values from the RREQ, the target MAC is set to a MAC computed on the preceding fields in the reply with the key $K_{D,S}$, and the key list is initialized to the empty list.
The RREP is then returned to the initiator of the request along the source route obtained by reversing the sequence of hops in the node list of the request. A node forwarding a RREP waits until it is able to disclose its key from the time interval specified, then it appends its key from that time interval to the key list field in the reply and forwards the packet according to the source route indicated in the packet. Waiting delays the return of the RREP but does not consume extra computational power. When the initiator receives a RREP, it verifies that each key in the key list is valid, that the target MAC is valid, and that each MAC in the MAC list is valid.
If all of these tests succeed, the node accepts the RREP; otherwise, it discards it. In

order to prevent the injection of invalid route errors into the network fabricated by any node other than the one on the sending end of the link specified in the error message, each node that encounters a broken link adds TESLA authentication information to the route error message, such that all nodes on the return path can authenticate the error. However, TESLA authentication is delayed, so all the nodes on the return path buffer the error but do not consider it until it is authenticated. Later, the node that encountered the broken link discloses the key and sends it over the return path, which enables nodes on that path to authenticate the buffered error messages.
ARIADNE is protected also from a flood of RREQ packets that could lead to the cache poisoning attack. Benign nodes can filter out forged or excessive RREQ packets using Route Discovery chains, a mechanism for authenticating route discovery, allowing each node to rate-limit discoveries initiated by any other node. The authors present two different approaches that can be found in [88].
ARIADNE is immune to the wormhole attack only in its advanced version: using an extension called TIK (TESLA with Instant Key disclosure) that requires tight clock synchronization between the nodes, it is possible to detect anomalies caused by a wormhole based on timing discrepancies.

- ARAN

The ARAN secure routing protocol [57] proposed by Dahill, Levine, Royer and Shields is conceived as an on-demand routing protocol that detects and protects against malicious actions carried out by third parties and peers in the ad hoc environment.
ARAN introduces authentication, message integrity and non-repudiation as part of a minimal security policy for the ad hoc environment and consists of a preliminary certification process, a mandatory end-to-end authentication stage and an optional second stage that provides secure shortest paths.
ARAN **requires the use of a trusted certificate server** (T): before entering in the ad hoc network, each node has to request a **certificate** signed by T. The certificate contains the IP address of the node, its public key, a timestamp of when the certificate was created and a time at which the certificate expires along with the signature by T. All nodes are supposed to maintain fresh certificates with the trusted server and must know T's public key.
The goal of the *first stage* of the ARAN protocol is for the source to verify that the intended destination was reached. In this stage, the source trusts the destination to choose the return path. A source node, A, initiates the route discovery process to reach the destination X by broadcasting to its neighbors a route discovery packet called RDP:

$$[RDP; IP_X; cert_A; N_A; t]K_{A-}$$

The RDP includes a packet type identifier ("RDP"), the IP address of the destination (IP_X), A's certificate ($cert_A$), a nonce N_A , and the current time t, all *signed* with A's

private key. Each time A performs route discovery, it monotonically increases the nonce. Each node records the neighbor from which it received the message. It then forwards the message to each of its neighbors, signing the contents of the message. This signature prevents spoofing attacks that may alter the route or form loops. Let A's neighbor be B. It will broadcast the following message:

$$[[RDP; IP_X; cert_A; N_A; t]K_{A-}]K_{B-}; cert_B$$

Nodes do not forward messages for which they have already seen the $(N_A; IP_A)$ tuple. The IP address of A is contained in the certificate, and the monotonically increasing nonce facilitates easy storage of recently-received nonces. Upon receiving the broadcast, B's neighbor C validates the signature with the given certificate. C then rebroadcasts the RDP to its neighbors, first removing B's signature:

$$[[RDP; IP_X; cert_A; N_A; t]K_{A-}]K_{C-}; cert_C$$

Eventually, the message is received by the destination, X, who replies to the first RDP that it receives for a source and a given nonce. There is no guarantee that the first RDP received travelled along the shortest path from the source. The destination unicasts a Reply (REP) packet back along the reverse path to the source. Let the first node that receives the RDP sent by X be node D. X will send to D the following message:

$$[REP; IP_A; cert_X; N_A; t]K_{X-}$$

The REP includes a packet type identifier ("REP"), the IP address of A, the certificate belonging to X, the nonce and associated timestamp sent by A. Nodes that receive the REP forward the packet back to the predecessor from which they received the original RDP. All REPs are signed by the sender. Let D's next hop to the source be node C. D will send to C the following message:

$$[[REP; IP_A; cert_X; N_A; t]K_{X-}]K_{D-}; cert_D$$

C validates D's signature, removes the signature, and then signs the contents of the message before unicasting the following RDP message to B:

$$[[REP; IP_A; cert_X; N_A; t]K_{X-}]K_{C-}; cert_C$$

A node checks the signature of the previous hop as the REP is returned to the source. This avoids attacks where malicious nodes instantiate routes by impersonation and re-

play of X's message. When the source receives the REP, it verifies that the correct nonce was returned by the destination as well as the destination's signature. Only the destination can answer an RDP packet. Other nodes that already have paths to the destination cannot reply for the destination.
While other protocols allow this networking optimization, ARAN removes several possible exploits and cuts down on the reply traffic received by the source by disabling this option.
The *second stage* of the ARAN protocol guarantees in a secure way that the path received by a source initiating a route discovery process is the shortest. Similarly to the first stage of the protocol, the source broadcasts a Shortest Path Confirmation (SPC) message to its neighbors: the SPC message is different from the RDP message only in two additional fields that provide the destination X certificate and the encryption of the entire message with X's public key (which is a costly operation).
The *onion-like* signing of messages combined with the encryption of the data prevents nodes in the middle from changing the path length because doing so would break the integrity of SPC the packet. Also the route maintenance phase of the ARAN protocol is secured by digitally signing the route error packets. However it is extremely difficult to detect when error messages are fabricated for links that are truly active and not broken. Nevertheless, because messages are signed, malicious nodes cannot generate error messages for other nodes. The non-repudiation provided by the signed error message allows a node to be verified as the source of each error message that it sends.
As with any secure system based on cryptographic certificates, the **key revocation issue** has to be addressed in order to make sure that expired or revoked certificates do not allow the holder to access the network. In ARAN, when a certificate needs to be revoked, the trusted certificate server T sends a broadcast message to the ad hoc group that announces the revocation. Any node receiving this message re-broadcast it to its neighbors. Revocation notices need to be stored until the revoked certificate would have expired normally. Any neighbor of the node with the revoked certificate needs to reform routing as necessary to avoid transmission through the now un-trusted node. This method is not failsafe. In some cases, the un-trusted node that is having its certificate revoked may be the sole connection between two parts of the ad hoc network. In this case, the un-trusted node may not forward the notice of revocation for its certificate, resulting in a partition of the network, as nodes that have received the revocation notice will no longer forward messages through the un-trusted node, while all other nodes depend on it to reach the rest of the network. This only lasts as long as the un-trusted node's certificate would have otherwise been valid, or until the un-trusted node is no longer the sole connection between the two partitions. At the time that the revoked certificate should have expired, the un-trusted node is unable to renew the certificate, and routing across that node ceases. Additionally, to detect this situation and to hasten the propagation of revocation notices, when a node meets a new neighbor, it can exchange a summary of its revocation notices with that neighbor; if these summaries do not match, the actual signed notices can be forwarded and re-broadcasted to restart propagation of the notice.
The ARAN protocol protects against exploits using modification, fabrication and impersonation but the use of **asymmetric cryptography makes it a very costly protocol** to use in terms of CPU and energy usage. Furthermore, ARAN is not immune to the wormhole attack.

- SEAD

Hu, Perrig and Johnson present a **proactive** secure routing protocol based on the Destination-Sequenced Distance Vector protocol (DSDV, [42]). In a proactive (or periodic) routing protocol nodes *periodically* exchange routing information with other nodes in attempt to have each node always know a current route to all destinations [89]. Specifically, SEAD is inspired by the DSDV-SQ version of the DSDV protocol. The DSDV-SQ version of the DSDV protocol has been shown to outperform other DSDV versions in previous ad hoc networks simulations [60] [47].
SEAD deals with attackers that modify routing information broadcasted during the update phase of the DSDV-SQ protocol: in particular, routing can be disrupted if the attacker modifies the sequence number and the metric field of a routing table update message. Replay attacks are also taken into account. In order to secure the DSDV-SQ routing protocol, SEAD makes use of *efficient one-way hash chains* rather than relaying on expensive asymmetric cryptography operations.
However, like the other secure protocols presented in this chapter, SEAD **assumes some mechanism** for a node **to distribute an authentic element of the hash chain** that can be used to authenticate all the other elements of the chain. As a traditional approach, the authors suggest to ensure the key distribution relaying on a trusted entity that signs public key certificates for each node; each node can then use its public key to sign a hash chain element and distribute it.
The basic idea of SEAD is to authenticate the sequence number and metric of a routing table update message using hash chains elements. In addition, the receiver of SEAD routing information also authenticates the sender, ensuring that the routing information originates form the correct node.
To create a one-way hash chain, a node chooses a random initial value $x \in \{0,1\}^{\rho}$, where is the length in bits of the output of the hash function, and computes the list of values $h_0, h_1, h_2, h_3, \ldots, h_n$, where $h_0 = x$, and $h_i = H(h_{i-1})$ for $0 < i < n$, for some n.
As an example, given an authenticated h_i value, a node can authenticate h_{i-3} by computing $H(H(H(h_{i-3})))$ and verifying that the resulting value equals h_i.
Each node uses a specific authentic (i.e. signed) element from its hash chain in each routing update that it sends about itself (metric 0). Based on this initial element, the one-way hash chain provides authentication for the lower bound on the metric in other routing updates for that node. The use of a hash value corresponding to the sequence number and metric in a routing update entry prevents any node from advertising a route to some destination claiming a greater sequence number than that destination's own current sequence number. Likewise, a node cannot advertise a route better than those for which it has received an advertisement, since the metric in an existing route cannot be decreased due to the one-way nature of the hash chain.
When a node receives a routing update, it checks the authenticity of the information for each entry in the update using the destination address, the sequence number and the metric of the received entry, together with the latest prior authentic hash value received from that destination's hash chain. Hashing the received elements the correct number of times (according to the prior authentic hash value) assures the authenticity of the received

information if the calculated hash value and the authentic hash value match. The source of each routing update message in SEAD must also be authenticated, since otherwise, an attacker may be able to create routing loops through the impersonation attack. The authors propose two different approaches to provide node authentication: the first is based on a broadcast authentication mechanism such as TESLA, the second is based on the use of Message Authentication Codes, *assuming a shared secret key between each couple of nodes in the network.*
SEAD does not cope with wormhole attacks though the authors propose, as in the ARIADNE protocol, to use the TIK protocol to detect the threat.

3.2.2 Key management proposals

Authentication of peer entities involved in ad hoc routing and the integrity verification of routing exchanges are the two essential building blocks used by secure routing protocols. Both entity authentication and message integrity call on the other hand for a key management mechanism to provide parties involved in authentication and integrity verification with proper keying material. Key management approaches suggested by current secure routing proposals fall in two categories:

- Manual configuration of symmetric (secret) keys: pair-wise secret keys can serve as encryption keys in a point-to-point key exchange protocol towards the establishment of session keys used for authentication and message integrity between communicating parties. If some dedicated infrastructure including a key server can be afforded, automatic distribution of session keys with a key distribution protocol like Kerberos [75] can also be envisioned.

- Public-key based scheme: each node possesses a pair of public and private keys based on an asymmetric algorithm like RSA. Based on this key pair, each node can perform authentication and message integrity operations or further exchange pairwise symmetric keys used for efficient authentication and encryption operations.

Secure routing proposals like SRP assume manual configuration of secure associations based on shared secret keys. Most of other proposals such as ARIADNE rely on a public-key based scheme whereby a well-known trusted third party (TTP) issues public key certificates used for authentication. The requirement for such a public-key infrastructure does not necessarily imply a managed ad hoc network environment and an open environment can be targeted as well. Indeed, it is not necessary for the mobile nodes that form the ad hoc network to be managed by the public-key certification authority. However, the bootstrap phase requires an external infrastructure, *which has to be available also during the lifetime of the ad hoc network* to provide revocation services for certificates that have expired or that have been explicitly revoked. Two interesting proposals presented in the next subsections tackle the complexity of public-key infrastructures in

the ad hoc network environment through self-organization: public-key management based on the concept of web of trust akin to Pretty Good Privacy (PGP, [113]) and a public-key certification mechanism based on polynomial secret sharing. Recent solutions rely on ID-based cryptography and cryptographically generated identities (crypto-based ID): these two approaches try to improve public-key based solutions by reducing the need for a centralized administration at the network bootstrap phase. A symmetric shared key distribution protocol allowing any two nodes to share a common secret key for data confidentiality is also presented. Another interesting approach outlined in the following subsections is based on the distance-bouncing technique and provides a secure encounter tracking service to a MANET.
Table 3.2 summarizes the approaches presented hereafter.

	Identity-Public Key binding				No identity	
	PGP web of trust	**Self organized CA based on secret sharing**	**Identity-based cryptography**	**Crypto-based identity**	**Key pre-distribution**	**Context-aware**
Symmetric cryptography					X	X
Asymmetric cryptography	X	X	X	X		
Public key Certificates	YES	YES	NO	NO	NO	YES
Bootstrap of security associations	PGP-like	Distribution of initial shares	Secure naming for unique identifiers	Self-generated keys	Get globally trusted key-pool	Self generated hash chains
Drawbacks	Initialization Storage Transitivity of trust	Distribution of initial shares Peripheral nodes	ID spoofing Distribution of initial shares KDC knows all secret keys	Bogus Identity generation	Cover-free algorithms Only shared keys No authentication	Viable only for small networks Reliance on tight timing
Advantages	Distributed approach	Distributed approach	No need for certificates Distributed approach	No need for certificates No need for Certification authority	No need for certificates Distributed approach	Self-organized Distributed approach

Table 3.2: Summary of key distribution approaches available in the literature

Public key management

- ID, Public-key binding with certificates

  – Self-Organized Public-Key Management based on PGP

    Capkun, Buttyan and Hubaux propose a fully self-organized public key management system that can be used to support security of ad hoc network routing protocols [25]. The suggested approach is similar to PGP [113] in the sense that users issue certificates for each other based on their personal acquaintances. However, in the proposed system, certificates are stored and distributed by the users themselves, unlike in PGP, where this task is performed by on-line servers (called certificate directories). In the proposed self-organizing public-key management system, each user maintains a local certificate repository.When two users want to verify the public keys of each other, they merge their local certificate repositories and try to find appropriate certificate chains within the merged repository.
    The success of this approach very much depends on the construction of the local certificate repositories and on the characteristics of the certificate graphs. The vertices of a certificate graph represent public-keys of the users and the edges represent public-key certificates issued by the users. The authors investigate several repository construction algorithms and study their performance. The proposed algorithms take into account the characteristics of the certificate graphs in a sense that the choice of the certificates that are stored by each mobile node depends on the connectivity of the node and its neighbors in the certificate graph.
    More precisely, each node stores in its local repository several directed and mutually disjoint paths of certificates. Each path begins at the node itself, and the certificates are added to the path such that a new certificate is chosen among the certificates connected to the last node on the path (initially the node that stores the certificates), such that the new certificate leads to the node that has the highest number of certificates connected to it (i.e., the highest vertex degree). The authors call this algorithm the Maximum Degree Algorithm, as the local repository construction criterion is the degree of the vertices in a certificate graph.
    In a more sophisticated extension called the Shortcut Hunter Algorithm, certificates are stored into the local repositories based on the number of the shortcut certificates connected to the users. The shortcut certificate is a certificate that, when removed from the graph makes the shortest path between two users previously connected by this certificate strictly larger than two.
    When verifying a certificate chain, the node must trust the issuer of the certificates in the chain for correctly checking that the public key in the certificate indeed belongs to the node identification (ID) named in the certificate. When certificates are issued by the mobile nodes of an ad hoc network instead of trusted authorities, this assumption becomes unrealistic. In addition, there

may be malicious nodes who issue false certificates. In order to alleviate these problems, the authors propose the use of authentication metrics [93]: it is not enough to verify a node ID key binding via a single chain of certificates. The authentication metric is a function that accepts two keys (the verifier and the verified node) and a certificate graph and returns a numeric value corresponding to the degree of authenticity of the key that has to be verified: one example of authentication metric is the number of disjoint chains of certificates between two nodes in a certificate graph.
The authors emphasize that before being able to perform key authentication, each node must first build its local certificate repository, which is a complex operation. However this initialization phase must be performed rarely and once the certificate repositories have been built, then any node can perform key authentication using only local information and the information provided by the targeted node. It should also be noted that local repositories become obsolete if a large number of certificate are revoked, as then the certificate chains are no longer valid; the same comment applies in the case when the certificate graph changes significantly. Furthermore, PGP-like schemes are more suitable for small communities because that the authenticity of a key can be assured with a higher degree of trustiness. The authors propose the use of authentication metrics to alleviate this problem: this approach however provides only probabilistic guarantees and is dependent on the characteristics of the certificate graph on which it operates. The authors also carried out a simulation study showing that for the certificate graphs that are likely to emerge in self-organized systems, the proposed approach yields good performances both in terms of the size of the local repository stored in each node and scalability.

– Authentication based on polynomial secret sharing

In [58] Luo and Lu present an authentication service whereby the public-key certificate of each node is cooperatively generated by a set of neighbors based on the behavior of the node as monitored by the neighbors. Using a group signature mechanism based on polynomial secret sharing, the secret digital signature key used to generate public-key certificates is distributed among several nodes. Certification services like issuing, renewal and revocation of certificates thus are distributed among the nodes: a single node holds just a share of the complete certificate signature key. The authors propose a localized trust model to characterize the localized nature of security concerns in large ad hoc wireless networks. When applying such trust model, an entity is trusted if any k trusted entities claim so: these k trusted entities are typically the neighboring nodes of the entity. A locally trusted entity is globally accepted and a locally distrusted entity is regarded untrustworthy all over the network.
In the suggested security architecture, each node carries a certificate signed by the shared certificate signing key SK, while the corresponding public key PK is assumed to be well-known by all the nodes of the network, so that certifi-

cates are globally verifiable. Nodes without valid certificates will be isolated, that is, their packets will not be forwarded by the network. Essentially, any node without a valid certificate is considered a potential intruder. When a mobile node moves to a new location, it exchanges certificates with its new neighbors and goes through mutual authentication process to build trust relationships. Neighboring nodes with such trust relationship help each other to forward and route packets. They also monitor each other to detect possible attacks and break-ins. Specific monitoring algorithms and mechanisms are left to each individual node's choice. When a node requests a signed certificate from a coalition of k nodes, each of the latter checks its records about the requesting node. If the requestor is recorded as a legitimate node, a partial certificate is computed by applying the local node's share of SK and returned to the requestor. Upon collecting k partial certificates, the requesting node combines them to generate the complete certificate of its public-key as if issued by a centralized certification authority.
The multiple signature scheme used to build the certificate is based on a k-threshold polynomial secret sharing mechanism. This technique requires a bootstrapping phase where a "dealer" has to privately send each node its share of the secret signature key SK. The authors propose a scalable initialization mechanism called "self-initialization" whereby the dealer only has to initialize the very first k nodes, regardless of the global network span. The initialized nodes collaboratively initialize other nodes: repeating this procedure, the network progressively self-initializes itself. The same mechanism is applied when new nodes join the network.
Certificate revocation is also handled by the proposed architecture and an original approach to handle roaming adversaries is presented in order to prevent a misbehaving node that moves to a new location from getting a valid certificate. Roaming nodes are defeated with the flooding of "accusation" messages that travel in the network and inform distant nodes about the behavior of a suspect node.
The main drawback of the proposed architecture is the *requirement for a trusted dealer* that initializes the very first k nodes of a coalition to the choice of the system-wide parameter k. To cope with the first problem, the authors propose to use a distributed RSA key pair generation [102] for the very first k nodes.
The other major limitation of the scheme is the strong assumption that every node of the network has at least k trusted neighbors. Moreover, the authors assume that any new node that joins the system already has an initial certificate issued by an off-line authority or by a coalition of k neighbors.

- Implicit ID, Public-key binding without certificates

  – Toward Secure Key Distribution in Truly Ad-Hoc Networks

The mechanism proposed in [52] focuses on the problem of key generation and management for peer-to-peer communication in a truly ad hoc network, i.e. a network in which both an infrastructure and a centralized authority are not available.
The proposal made by Kahlili et al. is based on the concept of ID-based cryptography and avoids the need for users to generate their own public keys and to distribute them throughout the network: the user's identity acts as their public key. This significantly reduces the computation necessary to join the network. Furthermore, as opposed to a CA-based solution [111] where a user is required to propagate both his public key as well as a signature (by the CA) on his public key, in an ID-based system, users need only propagate their identity (which is typically included in every message anyway). This can lead to huge savings in bandwidth.
It is important to notice that the authors do not specify neither the nature of the identity to be used, nor a means for authenticating users' identities before sending them (shares of) their secret key. Initial authentication of user's identities represents an important issue that is common to CA-based solutions: the entire security of the scheme is based on the initial validation of the data (identity) that will be certified for later use. The interested reader should refer to [20] to gather extensive information on ID-based cryptography. In [52] the authors suggest that at the time of network formation, the nodes willing to form the network decide on a mutually acceptable set of security parameters. Any node that is not satisfied by the choice of parameters can choose to refuse to participate in the network. The security parameters might include a threshold t of key service nodes, the number and identity of key service nodes, particular parameters of underlying schemes (e.g., key lengths), and a policy for key issuance. This initial negotiation is independent of the proposed scheme and the authors do not discuss it in any detail. It should be noted, though, that the initial policy negotiation is a potential target for active or byzantine adversaries, and the negotiation protocol should address this issue.
The initial set of nodes can then form a threshold key distribution center (KDC) for an ID-based scheme. These nodes will generate the master secret/public keys in a distributed manner such that fewer than t nodes cannot recover the master secret key. The master public key is given to all members of the network when they join, and the KDC can start issuing personal secret keys to nodes (including themselves) based on their identities and the key issuance policy. An identity can be something usually present in transmitted messages, like a MAC (or other network layer) address. To receive the private key corresponding to some identity, a node presents this identity and any extra material specified by the key issuance policy to t (or more) nodes forming the KDC and receives a share of their personal private key from each of them. With t correct shares, the node can then compute its personal private key within the network's ID-based system. It is important to notice that an efficient local mechanism to check the correctness of the individual shares and the computed private key has to be provided, but the authors to not provide any further information

about which mechanism should be used. Distributing the key generation and the KDC service prevents a single point of failure and resists compromise or insider attack (up to the threshold k). In addition, distributing the KDC in a t-out-of-n fashion makes the scheme flexible when some nodes are unreachable due to ad-hoc conditions (mobility, link breaks, Ĕ) as long as at least k are still reachable.
The authors stress that the proposed scheme makes no assumption about the "security" of users' identities, e.g., that they are set in hardware or cannot be spoofed. However, spoofing only needs to be prevented/detected by the nodes forming the KDC at the time of key issuance (and this can be done by requiring some "unspoofable" supporting material to be presented at the time of a key request). For completeness, the authors recommend a number of possibilities for user identities. One possibility is to use statistically unique cryptographically verifiable (SUCV) addresses [78] (applied to ad-hoc networks by [17]).
The solution presented in [52] offers an interesting alternative to the classical certificate-based systems presented in this section and it should be noted that the authors target a fully independent ad hoc network where nodes act in a self-organized way.
However, the initial establishment of a KDC using threshold cryptography **needs** the presence of **an external authority/management** that decides the threshold system parameters and that elects trusted nodes to participate to the initial set of nodes forming the KDC. This is obviously in contrast with the targeted self-organization characteristic of the network and a fully self-organized network set-up is far from being achieved. It is important to remember that introducing new cryptographic techniques or an original combination of existing techniques could have the sole effect of moving the problem (self-organization) one-step further, only giving the illusion of a security scheme adapted to a truly self-organized network.

– Bootstrapping Security Associations for Routing in Mobile Ad-Hoc Networks

In [17], the authors show how to bootstrap secure associations for the routing protocols of MANETs online, without assuming any trusted authorities or distributed trust-establishment services.
The proposed scheme relies on the use of statistically unique and cryptographically verifiable (SUCV) identifiers [78] and public-secret key pairs generated by the nodes themselves, in much the same way SUCVs are used in MobileIPv6 (MIPv6) to solve the address "ownership" problem [78].
The authors present the bootstrapping solution in the context of the Dynamic Source Routing (DSR) protocol and argue that the solution is applicable to other secure routing protocols, such as SEAD [89] and Ariadne [88].
In [17], in order to produce an IP address that is securely bound to a public key, a node generates a 64-bit pseudo-random value by applying a one-way, collision-resistant hash function to the public key of its (uncertified) public-

private key pair. Then, the IP address is generated as the concatenation of a network specific identifier (64 bits in MIP6) and the hash of the public key (64 bits).
The binding between this IP address and the public key is secure because it is computationally unfeasible for an attacker (1) to create another $< public, private >$ key pair whose hash generates the same IP address (because of the second pre-image resistance of one-way, collision-resistant hash functions), and (2) to discover the secret key, or create a different one, for a given public key (by definition).
Due to the size of the resulting address space, this IP address is also statistically unique. A source node can then use the secure binding to authenticate its IP address and the contents of its packets to an arbitrary destination node as follows.
The node signs a packet with its private key and sends the signed packet to the destination address together with its public key (and IP address). The destination node verifies that the IP address is securely bound to the public key by computing the hash function of the received public key and comparing the result with the lower 64-bit field of the IP address. (Thus, the IP address "certifies" the validity of the public key thereby preventing an attacker from spoofing the source address.) Then, the destination node authenticates the content of the packet by verifying the signature with the public key.
The authors propose in [17] the application of their ideas to the DSR route discovery phase, both in an asymmetric version and in a symmetric version, and analyze the performances and computational overhead of a MANET using cryptographically generated IP addresses. It should be noted that other security solutions [52] for MANET suggest relying on such scheme to protect against spoofing attacks.
However, the absence of a certification authority that validates the information locally generated by every node can have a drastic consequence on the overall security of the scheme. Indeed, there is nothing that prevents a malicious node to generate as many public-private key pairs as needed to produce bogus IP addresses for which it can securely prove possession. The only guarantee that the scheme proposed in [17] offers is that a legitimate node can be assured to be the only owner of an IP address.
As another example, the scheme proposed in [52] [17] could not be used to protect from identity spoofing in a MANET using a cooperation enforcement mechanism based on reputation (see section 3.2.3). Indeed, the scheme would only prevent a misbehaving node to use a legitimate node IP/ID in order to steal her positive reputation rating. It would allow, however, a misbehaving node to get rid of a bad reputation record simply by generating a new (verifiable) IP/ID and impersonate a new node joining the network.

Distribution of Symmetric Keys

- Key pre-distribution

  In [112] the authors propose a **symmetric pair-wise secret keys establishment** protocol for ad hoc networks. Excluding a set-up phase where an *off-line key distribution center is needed*, the main features of the proposed protocol are:

  - It is fully distributed - no on-line key server is required;
  - It is computationally efficient - it relies only on symmetric cryptography;
  - It is storage scalable - the storage requirements per node are independent of the size of the network.

  The basic functioning of the protocol is the following: the pre-key distribution server deterministically selects a set of keys from a common key pool and delivers them to any node that wants to participate to the ad hoc network. The key selection is deterministic and based on the node identity, thus allowing any node to determine intersection/common keys with other nodes of the network.
  When forming the network, any node that needs to establish a secret shared key with any other node in the network has to determine whether there is any *direct path* or *indirect path* to the desired recipient.
  Consider as an example two nodes, u and v that wish to communicate privately. Note that u and v may already share one or more keys from the pool of keys after the key pre-distribution phase. However, these keys are not known *exclusively* to u and v because every key of the key pool may be allocated to multiple nodes; hence, they cannot be used for encrypting any message that is private to u and v.
  The goal of the proposed algorithm is to establish a key, sk, that is known exclusively to u and v. To establish sk, a sender node (say u) splits sk into multiple shares using a threshold secret sharing scheme. It then transmits to the recipient (v) all these shares, using a separate secure *logical path* for each share. The recipient node then reconstructs sk after it receives all (or a certain number of) the shares and acknowledge the correct reception with a ciphered hello message using the common shared key.
  There are *logical paths* between two nodes when (i) the two nodes share one or more keys (*direct path*) and (ii) the two nodes do not share any keys, but through other intermediate nodes (called *proxies*) they can exchange messages securely (*indirect paths*). The protocol always uses any direct paths that exist between nodes in preference to indirect paths, since the use of an indirect path incurs additional computational and communication overhead.
  Logical paths between two nodes can be easily found since the key pre-distribution algorithm is public and deterministic; a node u can independently compute the set of key IDs corresponding to a node v's key set. Therefore, without proactively exchanging the set of its key IDs with others, a node knowing the IDs of its neighbors

can determine not only which neighbors share or do not share keys with it, but also which two neighbors share which keys. The latter knowledge is very valuable when node u does not share any keys with a node v, because node u can use a neighbor (say x) which shares keys with both of them as a proxy. For example, suppose node u shares a key $k_{u,x}$ with node x, node v shares a key $k_{v,x}$ with node x, but no shared key exists between node u and node v. To transmit a message M to node v securely, node u executes the following steps:

$$u \rightarrow x : \{M\}_{k_{ux}}$$

$$x \rightarrow v : \{M\}_{k_{xv}}$$

The node x in the above example is called node u's one-hop proxy to v. More generally, node x is said to be node u's i-hop proxy if x is i hops away from u and x shares a key with both u and v respectively. If u and v do not have any direct paths or one-hop proxies to each other, they can resort to a proxy node of multiple-hop away. Note that it is also possible to establish a logical path with multiple proxies involved. For example, if there is a shared key between u and x, between x and y, between y and v respectively, u and v can establish a logical path as well.
The idea proposed in [112] represents a distributed and efficient way of establishing a shared secret key between two nodes in a MANET but **relies on the presence of an external authority and/or centralized management** to compute and distribute the large amount of initial pre-shared keys. Furthermore, this limitation is accentuated by the need of undeniable and un-spoofable node identities. The overall system security relies indeed on the fact that every node is able to compute a pool of pre-shared keys with her neighbors in order to initiate a logical-path discovery to reach the desired recipient. As an example, by generating fake identities a node could contact the pre-key distribution center and gather enough keying material to decrypt all the shares of a shared-secret key and compromise the confidentiality of communication.

Context-aware key management

- SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks

In [26] Chapkun, Buttyan and Hubaux present a set of mechanisms for the secure verification of the time of encounters between nodes in multi-hop wireless networks with the aim of enabling any node to prove to any other node (or base station) its encounters with other nodes before or at some specific time. SECTOR can be used as a basic mechanism to prevent the wormhole attack, to help securing routing protocols based on history of encounters [36] or to provide topology monitoring both in pure ad hoc systems and in cellular networks.

The system model and the basic assumptions made by the authors require *loose time synchronization* between all the nodes forming the network and that nodes can measure time locally with nanosecond precision. Furthermore, the authors also assume that each node is equipped with a *special hardware module* that can temporarily take over the control of the radio transceiver unit of the node from the CPU. A node can be put in a special state where it is capable of responding to a one-bit challenge with a one-bit response essentially immediately, without the delay imposed by the usual way of processing messages. Lastly, the authors assume that the bits are correctly transmitted, meaning that there are no collisions and no jamming.
Security associations are provided prior to the network formation so that each node shares a symmetric secret key with any other node and a central authority controls the network membership and assigns a unique identity to each node. In order to provide a basis for secure and authenticated proofs of encounters the MAD (Mutual Authentication with Distance-bounding) protocol is introduced. Based on the Brands-Chaum technique [22], the MAD protocol enables any two nodes to determine their mutual distance at the time of encounter and enhances the basic scheme by avoiding the use of digital signature in favor of a more efficient symmetric scheme. Furthermore, the commitments bits that are used in the distance-bounding technique are firstly sent in the initialization phase of the protocol, and then gradually revealed when the distance evaluation takes part: this ensures that the parties cannot send bits too early, and thus, cannot cheat the other party by appearing to be closer than they really are. The MAD protocol has to be used in conjunction with the GTE (Guaranteed Time of Encounter) mechanism that consists in the construction of a particular type of Merkle hash tree [64] in which leafs contain also time information about the disclosure of the hash chain element. The root of the tree is then distributed in an authentic way to every other node of the network.
Any two nodes willing to exchange secure proofs of the time of their encounter, verifiable by any other node of the network, will have to run first the MAD protocol in order to mutually authenticate themselves. Then they will disclose the element of the Merkle tree that corresponds to the actual time (every node is loosely synchronized): after authentication of the disclosed proof, the nodes have to store the released tree element to subsequently prove their encounter. The authors provide a detailed study that focus on possible enhancements of the SECTOR protocol suite to meet specific requirements due to the presence of an infrastructure or to a limited storage capacity of the devices. The interested reader should refer to [26] to seize the different applications of SECTOR.
Though SECTOR provides a first tentative to cope with secure tracking of node encounters in MANET, its application could be narrowed by the limiting requirements imposed by the design. The need for a deployed security infrastructure, pre-established security associations between nodes or, in its advanced version, the use of TESLA keys for broadcasting authentic roots of the Merkle hash trees render SECTOR not adapted to a short-lived, highly mobile and dynamic network. In addition, the security of SECTOR relies on the presence of an external trusted authority that provides unique identities (and certificates) to the nodes of the network.

If such an authority were not available, SECTOR would be vulnerable to attacks such as the forgery of proofs of encounters. Furthermore, as the authors suggest, storage requirements can be very demanding depending on the size of the network: this can be a serious limitation when considering heterogeneous MANET formed by portable devices.

3.2.3 Cooperation enforcement mechanisms

Current cooperation enforcement proposals for MANET can be classified as follows:

- Approaches based on threshold cryptography
- Approaches based on micro-payments
- Approaches based on reputation

Table 3.3 provides a summary of cooperation enforcement schemes with their features.

	Token based	**Nugglets**	**Sprite**	**I-Pass**	**CONFIDANT**
Incentive	Participation Token	Virtual currency	Virtual currency	Auctions	Reputation
Punishment	Impact on route selection	Network utilization prevented	Network utilization prevented	Low bandwith	Impact on route selection
Monitoring Technique	Watchdog	N/A	N/A	N/A	Watchdog
Reintegration	Not provided	N/A	N/A	N/A	Need for specific mechanism
Need for Security Infrastructure	YES	YES Tamper proof hardware	YES Credit Clearance Service	YES Credit Clearance Service	NO
Drawbacks	Need for security infrastructure	Need for security infrastructure Need for tamper proof hardware Credits per route evaluation	Need for security infrastructure Credit Service Low mobility required	Need for security infrastructure Credit Service Complexity of auction scheme	Reliance on watchdog Need for a reintegration mechanism Reputation propagation can be a weakness
Advantages	No need for watchdog	No need for a Credit Service	No need for tamper proof hardware	Integration in real business scenarios	Reputation propagation

Table 3.3: Summary of cooperation enforcement schemes available in the literature

The most significant proposals in each category are outlined in the sequel of this section.

Cooperation enforcement mechanisms based on threshold cryptography

- Self-Organized Network-Layer Security in Mobile Ad Hoc Networks

In [109] Yang, Meng, Lu suggest a mechanism whereby each node of the ad hoc network is required to hold a token in order to participate in the network operations. Tokens are granted to a node collaboratively by its neighbors based on the monitoring of the node's contribution to packet forwarding and routing operations. Upon expiration of the token, each node renews its token through a token renewal exchange with its neighbors: the duration of a token's validity is based on the duration of the node's correct behavior as monitored by the neighbors granting/renewing the token. This mechanism typically allows a well-behaved node to accumulate credit and to renew its token less frequently as time evolves.
The token-based cooperation enforcement mechanism includes four interacting components: neighbor verification through which the local node verifies whether neighboring nodes are legitimate, neighbor monitoring that allows the local node to monitor the behavior of each node in the network and to detect attacks from malicious nodes, intrusion reaction that assures the generation of network alerts and the isolation of attackers, and security enhanced routing protocol that consists of the ad hoc routing protocol including security extensions.
A valid token is constructed using a group signature whereby a mechanism based on polynomial secret sharing [102] assures that at least k neighbors agree to issue or renew the token. The key setup complexity of polynomial secret sharing and the requirement for at least k nodes to sign each token both are incompatible with high mobility and call for a rather large and dense ad hoc network. Furthermore, the duration of a token's validity increases proportionally with the duration of the node's correct behavior as monitored by its neighbors; this feature again calls for low mobility. The token-based cooperation enforcement mechanism is thus suitable for ad hoc networks where node mobility is low. Spoofing attacks through which a node can request more than one token claiming different identity, are not taken into account by the proposal even if the authors suggest that MAC addresses can be sufficient for node authentication purposes.

Cooperation enforcement mechanisms based on micro-payment schemes

- Nuglets

In [45], Buttyan and Hubaux present two important issues targeted specifically at the ad hoc networking environment: first, end-users must be given some incentive to contribute in the network operation (especially to relay packets belonging to other nodes);

second, end-users must be discouraged from overloading the network.
The solution consists of a virtual currency called Nuglet used in every transaction. Two different models are described: the Packet Purse Model and the Packet Trade Model.
In the Packet Purse Model each packet is loaded with nuglets by the source and each forwarding host takes out nuglets for its forwarding service. The advantage of this approach is that it discourages users from flooding the network but the drawback is that the source needs to know exactly how many nuglets it has to include in the packet it sends.
In the Packet Trade Model each packet is traded for nuglets by the intermediate nodes: each intermediate node buys the packet from the previous node on the path. Thus, the destination has to pay for the packet. The direct advantage of this approach is that the source does not need to know how many nuglets need to be loaded into the packet. On the other hand, since the packet generation is not charged, malicious flooding of the network cannot be prevented.
There are some further issues that have to be solved: concerning the Packet Purse Model, the intermediate nodes are able to take out more nuglets than they are supposed to; concerning the Packet Trade Model, the intermediate nodes are able to deny the forwarding service after taking out nuglets from a packet.

- Sprite

As opposed to the Nuglets approach, the proposal presented by Zhong et al. [30] does not require a tamper-proof hardware.
At a high level, the basic scheme of sprite can be described as follows. When receiving a message, a node keeps a signed receipt of the message generated by the source node. Later, when the node has a fast connection to a Credit Clearance Service (CCS), it reports to the CCS the messages that it has received/forwarded by uploading its receipts. The CCS then determines the charge and credit to each node involved in the transmission of the message, depending on the reported receipts. The main objectives of Sprite consist in stimulating node to cooperate and preventing cheating by making it unattractive. The overall system architecture can be described as follows. To identify each node, the authors assume that each node has a public key certificate issued by a scalable certificate authority such as those proposed in [111], [58]. When a node sends its own messages, the node will lose credit (or virtual money) to the network because other nodes incur a cost to forward the messages. On the other hand, when a node forwards others' messages, it should gain credit and therefore be able to send its messages later.
There are two ways for a node to get more credit. First, a node can pay its debit or buy more credit using real money, at a variable rate to the virtual money, based on the current performance of the system. However, the preferred and dominant way to get more credit is by forwarding others' messages. In order to get credit for forwarding others' messages, a node needs to report to the CCS which messages it has helped to forward by presenting a valid message receipt. The choice of charging only the source (and not the destination) of a message derives from the fact that it prevents nodes to flood the network with useless messages.
The authors provide a detailed study of the payment/charging scheme parameters to

make cooperation attractive and a cheating behavior unappealing based on game theory. However, it can be argued that the basic assumption made when providing an analytical study of the solution of the game are somehow misleading: indeed, the authors assume that a node is following the truth-telling strategy when a node truly received a message and reports a valid receipt to the CCS. Unfortunately, the key feature of the Sprite protocol, namely the receipt used as a proof of forwarding, could constitute a weak point of the system. A receipt is generated by the source of a message, signed with the source's secret key and appended to the message that needs to be forwarded. Subsequent nodes on the path just need that receipt that can be gathered both by receiving the message (but not necessarily forward it) and by colluding with other nodes.
The authors took into account this possibility when modelling the problem in game theoretical terms: indeed, system parameters are designed and tuned such that collusion does not pay. However, if the nodes had other means of exchanging receipts other than using their radio interfaces, then collusion and cheating could become attractive.

- IPass

In the following approach, the authors adopt the "pay for service" model of cooperation, and propose an auction-based incentive scheme to enable cooperative packet forwarding behavior in MANET.
In iPass, each router constitutes a "smart market", where an auction process runs continuously to determine who should obtain how much of the bandwidth, and at what price. The bidders are the set of traffic flows currently passing that router. Each flow carries a bid indicating its willingness to pay for the forwarding service. Based on these bids, the router runs a "generalized Vickrey auction" to determine the bandwidth allocation for the flows. A remarkable property of this auction is that, it is incentive compatible for the users to place a bid equal to their true utilities for bandwidth. A user's utility for bandwidth reflects the valuation (or satisfaction) of the user if she is given such bandwidth, which may depend on the application being used. Incentive compatible means that a user has no incentive to deviate from this bidding strategy, because it leads to higher payoff for the user, defined as the user's utility of winning the goods less the price to pay. Therefore, the user's bidding strategy is greatly simplified, and the outcome of the auction is efficient, meaning that it gives bandwidth to those users who need them the most.
From an end-to-end point of view, a flow usually travels through multiple hops (or auction markets) where it is allocated different amounts of bandwidth. This information is carried with each data packet to the receiver, and returned to the sender as feedback, as part of a signaling protocol. The sender should then police its rate in compliance with the allocated bandwidth. Therefore, besides creating incentive for packet forwarding, the iPass scheme also assumes the task of flow control, and possesses certain "differentiated service" capability based on the bids of the flows.
The reader interested on details of the generalized Vickrey auction scheme is advised to refer to [31], [59], [108].
It is important to notice here that the design of the iPass mechanism is incomplete: the

authors suggest referring to [30] in order to find guidelines to complete it with a payment protocol allowing the router node to claim reward for forwarding the packet, for example by saving a "receipt" of the packet, similar to Sprite. Detail integration of the iPass scheme with a secure payment and auditing system remains a future direction.
As opposed to other cooperation enforcement schemes presented in this chapter, iPass appears to be very interesting since its inherent features makes it easy to integrate in real business scenarios.
However, the main drawback of iPass is the general complexity of the auction scheme. It is sometimes important to keep in mind that a cooperation enforcement scheme is designed to cope with node selfishness that derives mainly from a limited quantity of energy available to mobile nodes. The continuous execution of a heavy cooperation enforcement mechanism could have serious consequences on the energy consumption of a node, which in turn would have the tendency to behave selfishly.

Cooperation enforcement mechanisms based on reputation

- CONFIDANT

Buchegger and Le Boudec proposed a technique called CONFIDANT (Cooperation Of Nodes, Fairness In Dynamic Ad-hoc NeTworks) [23], [24] aiming at detecting malicious nodes by means of combined monitoring and reporting and establishing routes by avoiding misbehaving nodes.
CONFIDANT is designed as an extension to a routing protocol such as DSR. CONFIDANT components in each node include a network monitor, reputation records for first-hand and trusted second-hand observations about routing and forwarding behavior of other nodes, trust records to control trust given to received warnings, and a path manager to adapt the behavior of the local node according to reputation and to take action against malicious nodes. The term reputation is used to evaluate routing and forwarding behavior according to the network protocol, whereas the term trust is used to evaluate participation in the CONFIDANT meta-protocol.
The dynamic behavior of CONFIDANT is as follows. Nodes monitor their neighbors and change the reputation accordingly. If they have a reason to believe that a node misbehaves, they can take action in terms of their own routing and forwarding and they can decide to inform other nodes by sending an ALARM message. When a node receives such an ALARM either directly or by promiscuously listening to the network, it evaluates how trustworthy the ALARM is based on the source of the ALARM and the accumulated ALARM messages about the node in question. It can then decide whether to take action against the misbehaved node in the form of excluding routes containing the misbehaved node, re-ranking paths in the path cache, reciprocating by non-cooperation, and forwarding an ALARM about the node.
The first version of CONFIDANT was, despite the filtering of ALARM messages in the trust manager, vulnerable to concerted efforts of spreading wrong accusations. In a recent enhancement of the protocol, this problem has been addressed by the use of Bayesian

statistics for classification and the exclusion of liars.
Simulations with nodes that do not participate in the forwarding function have shown that CONFIDANT can cope well, even if half of the network population acts maliciously. Further simulations concerning the effect of second-hand information and slander have shown that slander can effectively be prevented while still retaining a significant detection speed-up over using merely first-hand information.
The limitations of CONFIDANT lie in the assumptions for detection-based reputation systems. Events have to be observable and classifiable for detection, and reputation can only be meaningful if the identity of each node is persistent, otherwise it is vulnerable to spoofing attacks.

3.3 Research challenges

Despite the lack of maturity of ad hoc networks, security has gathered much attention from the research community due to the novelty of requirements with respect to classical networks.
Research activities in ad hoc network security fall in one of three areas: routing security, key management and cooperation enforcement.
In spite of the large number of solutions, ad hoc routing does not seem to raise any new security requirement with respect to routing in classical networks, apart from key management problems that are addressed by a large class of original work in this specific category.
Key management approaches try to answer the hard question of how to establish security associations with no a-priori knowledge, no a-priori trust and lack of infrastructure. Several original key management schemes based on advanced cryptographic constructs like threshold cryptography and id-based cryptography have been suggested in the literature but they all fall short of meeting the ultimate goal of building a keying infrastructure "from scratch" since they all involve an initial key set-up phase.
Cooperation enforcement is another original requirement raised by ad hoc networks. Several approaches have been suggested to tackle this problem but solutions still are in their infancy with respect to real-life scenarios and their integration with basic networking functions.
In the reminder of this thesis we focus on cooperation enforcement and key management issues. A collaborative cooperation enforcement scheme based on node's reputation and an original key management scheme that enables routing security are proposed. Furthermore, we focus on the validation of these mechanisms. If the properties of the key management scheme can be assessed through classical evaluation methods, i.e. computational and storage requirements as well as robustness in spite of attacks, the validation of the cooperation enforcement scheme is more awkward. Indeed, a proof-of-concept validation of the mechanism is possible through a basic network simulation software, but a specific evaluation of the incentive properties of the scheme can be assessed only if a realistic model of node selfishness is available. Thus, a considerable effort have been made

towards the definition of a model that represents a "rational" behavior of self-interested nodes as well as an analytical framework that serves as a tool for studying the interactions between legitimate and misbehaving nodes of an open ad hoc network.

Relevant publications

R. Molva and P. Michiardi. Security in ad hoc networks. In *Proceedings of Personal Wireless Communications Conference (PWC'03)*, Venice, Italy, September 2003.

P. Michiardi and R. Molva. Ad hoc network security. *ST Microelectronics Journal of System Research*, 2003.

P. Michiardi and R. Molva. *Chapter 12: Ad hoc network Security.* Mobile ad hoc networking. Wiley-IEEE Press, New York, NY, USA, 2004.

P. Michiardi and R. Molva. *Ad hoc network Security.* Handbook of Information Security. Wiley-IEEE Press, New York, NY, USA, 2005.

Chapter 4

CORE: reputation-based incentives for node cooperation in MANET

4.1 Introduction

The simulation study conducted in our laboratory [65] and presented in Chapter 2 showed that the performance of MANET severely degrades in presence of simple node misbehavior. Apart from special cases like military networks whereby an a-priori trust exists between all nodes, the nodes of an ad hoc network cannot be trusted for the correct execution of critical network functions. Essential network operations assuring basic connectivity can be heavily jeopardized by nodes that do not properly execute their *share of the network operations* like routing, packet forwarding, name-to-address mapping, etc. Node misbehavior that affects these operations may range from simple selfishness or lack of collaboration *due to the need for power saving* to active attacks aiming at denial of service and subversion of traffic. Because of their increased vulnerability, ad hoc networks should take into account security problems as a basic requirement regardless of the application scenarios and countermeasures must be integrated with basic networking mechanisms at the early stages of their design.
Another important finding of our simulation study is that the performance degradation due to selfish nodes refraining from packet forwarding is more significant than the impact of selfish behavior simulated by attacks on the DSR routing protocol. We believe that similar results that highlight the inherent sensitivity of MANET to nodes' selfishness can be obtained with network functions other than routing and packet forwarding. Security mechanisms that solely enforce the correctness or integrity of network operations would thus not be sufficient in MANET. A basic requirement for keeping the network operational is to enforce ad hoc nodes' contribution to network operations despite the conflicting tendency of each node towards selfishness as motivated by the scarcity of node power.
In this Chapter we propose a mechanism called CORE to enforce node cooperation based on a distributed monitoring technique, which has been presented in [66]. Our cooperation

enforcement mechanism does not prevent a node from denying cooperation or deviating from a legitimate behavior but ensures that misbehaving entities are **punished** by gradually withholding communication services. CORE is suggested as a generic mechanism that can be integrated with any network function like packet forwarding, route discovery, network management, and location management. In CORE, each network entity keeps track of other entities' collaboration using a cooperation metric called **reputation**. The reputation metric is computed based on data monitored by the local entity and eventually using information provided by other nodes involved in each operation.
Based on reputation, a punishment mechanism is used as a deterrent to inhibit selfish behavior by gradually refusing communication services to misbehaving entities.
As pointed out in the rest of the chapter, the immediate consequence of a CORE-enabled MANET is that legitimate nodes (i.e. nodes that cooperate to the network operation) save energy by not serving nodes that have been detected as selfish. Furthermore, depending on the selfishness model adopted to represent misbehaving nodes, it is possible to show that CORE provides incentives to cooperate.
In the remaining of the Chapter a detailed definition of the security objectives of CORE and a node selfishness model are provided. Fundamental assumptions used in the design of our cooperation enforcement mechanism are discussed in depth. Section 4.4 provides a sketch of CORE operation while section 4.4 focuses on the building blocks that constitute our mechanism. We then provide a discussion on the properties of our scheme with an emphasis on the security evaluation of CORE and conclude with an application example where CORE is used to enforce node participation to the packet forwarding function.

Note on reputation mechanisms

Reputation mechanisms have recently been proposed for use within ad hoc networks to address some of the threats arising from selfish network nodes. These mechanisms are of particular value in addressing threats targeted to the ad hoc network layer because they are inherently distributed thus well adapted to an infrastructure-less environment. In the context of an ad hoc network, these mechanisms seek to dynamically assess the trustworthiness of neighboring nodes, with a view to excluding untrustworthy nodes.
The use of reputation systems in many different areas of IT is increasing, not least because of their widely publicized use in online auctions and product reviews, see, for example eBay and Amazon [94]
Mui et al. [80] give many examples of how reputation systems are used. Reputation systems are used to decide who to trust, and to encourage trustworthy behavior. Resnick and Zeckhauser [95] identify three goals for reputation systems:

1. To provide information to distinguish between a trustworthy principal and an untrustworthy principal;

2. To encourage principals to act in a trustworthy manner;

3. To discourage untrustworthy principals from drawing advantage from services they do not contribute to.

Generic reputation systems rely on principals monitoring sequences of transactions with other principals, and on communications between principals that are willing to take part in the reputation system. Each principal maintains a reputation value for some subset of the other principals in the system: these values may be shared between principals or may be unique for each participant. The precise meaning of the reputation value, how it is calculated and updated, and how it is communicated between parties, are all system-dependent.
In the sequel of this Chapter we show how and to what extent the generic goals of reputation mechanisms are achieved by CORE.

4.2 Security objectives

We now discuss the objectives of the CORE scheme by referring to the security requirements discussed in Chapter 2. CORE is intended for *open ad hoc networks*, in which an a-priori trust base among the nodes is not available: nodes are operated by end-users that do not share any organizational structure (i.e. they belong to different authorities) and that do not pursue a common goal. Furthermore, we assume that a networking infrastructure is not available.
CORE follows a **distributed approach** and do not rely on any centralized service used to coordinate the activity of the nodes or to collect and distribute information on nodes' behavior. In CORE, every node is responsible for the **local detection and punishment** of selfish neighboring nodes. CORE is suggested as a generic mechanism that can be integrated with any network function like packet forwarding, route discovery, network management, and location management. In our design, CORE provides incentives to cooperate **both** to packet forwarding and routing functions. However, we often provide examples of CORE operation when used to mitigate the effects of a non-cooperative forwarding behavior in order to improve the readability of the Chapter.
CORE is a **lightweight mechanism** that do not waste limited energetic resources: nodes that participate in a CORE-enabled ad hoc network should also be able to minimize the energetic burden derived from costly monitoring operations. Furthermore, our aim is also to **minimize traffic overhead** by reducing signalling messages eventually exchanged during CORE execution.
A fundamental objective to keep in mind when designing a security mechanism concerns the **robustness** of the scheme. Although cooperation enforcement mechanisms only deal with selfish nodes that do not perpetrate any active attacks, malicious nodes *cannot exploit CORE* to initiate denial of service attacks by misusing the punishment component. Moreover, **identity spoofing** must be prevented: nodes that have been detected as selfish by the cooperation enforcement mechanism must not be able to modify their identity in order to trick legitimate nodes and persist with a non-cooperative behavior by getting

rid of a bad reputation record.
Following, we discuss on different selfishness models that can be addressed by cooperation enforcement mechanisms and determine which model is more suitable for the design and the subsequent validation of CORE.

4.2.1 Selfishness models

Initially, we used a basic definition of node selfishness that does not give any information on the **behavior dynamics**. A selfish node could **always** be selfish, that is, follow selfishness model 1 or model 2 that have been defined in the simulation study described in Chapter 2. On the other hand, selfish behavior could be **conditioned** by a local measurement of energetic availability, that is, follow the selfishness model 3 described in Chapter 2. If battery recharge is possible, nodes could manifest a selfish behavior only when the end-user that operates the node decides to decrease the cooperation effort for example by selectively disabling the networking functions. Furthermore, a **selective selfish behavior** could arise in networks where end-users discriminate a sub-set of neighboring nodes, by offering networking services only to a limited number of end-users.
Cooperation enforcement mechanisms available in the literature explicitly target the first type of selfishness in which the misbehaving node permanently disables the forwarding function. However, it is clear that a selfish behavior has to be attributed to end-users that operate the nodes: a model that takes into account a degree of **"rationality"** in the decision making process that result in a selfish behavior would be *more realistic*. By rationality we intend the ability of making a decision based on self-interested objectives (i.e. minimize battery consumption) as well as on the knowledge that eventual countermeasures against an egoistic behavior (i.e. a cooperation enforcement mechanism) are operational in the system. The CORE mechanism has been designed to cope with **rational selfish nodes** that can selectively target neighboring nodes. A formal definition of rationality is presented in Chapter 5, where we provide both a simulation-based and a analytical validation of CORE.
The implications of the selfishness model targeted by the cooperation enforcement mechanism have a direct influence on the punishment inflicted to misbehaving entities: a detailed discussion on the choice of the punishment mechanism that we have made during the design of CORE and on possible alternatives is provided in section 4.7.

4.3 Background and assumptions

This section illustrates the assumptions that have been made for the design of our cooperation enforcement scheme. For the sake of simplicity, in this Chapter we focus on the utilization of CORE as an additional component to the layer 3 of the ISO/OSI stack to cope with nodes' lack of cooperation to **network-level** functions.
Since monitoring of nodes' behavior in CORE is performed through an additional feature

of the medium access control (MAC) layer, this section presents the necessary hypothesis that were made on the underlying services that are provided by wireless cards based on the 802.11 protocol standard. Furthermore, the punishment mechanism interacts with the packet forwarding function implemented in the ad hoc routing protocol that has been chosen for the network: we thus provide the assumptions made on the ad hoc network layer.
We finally discuss on a set of assumptions that were made to limit our attention to a network formed by *uniquely identifiable* nodes *equal* in terms of computational power and energetic availability.

- Promiscuous mode operation

Throughout this Chapter we assume bi-directional communication symmetry on every link between the nodes. This means that if a node B is capable of receiving a message from a node A at time t, then node A could instead have received a message from node B at time t. Furthermore, we assume a fixed transmission power, i.e. the radio range is the same for every node of the network.
In addition, we assume wireless interfaces that support promiscuous mode operation. Promiscuous mode means that if a node A is within range of a node B, it can overhear communications to and from B even if those communications do not directly involve A. Promiscuous mode operation entails some weaknesses that are discussed in section 4.5.

- Layer-2 encryption

Throughout the Chapter we assume that communications are **not** ciphered through a Layer-2 encryption. Security services for the MAC layer are out of the scope of this thesis. However, the monitoring technique used by CORE requires content validation of the messages exchanged between nodes: a hop-by-hop encryption would have the negative effect of providing a security hole to attack CORE. Indeed, a malicious node could trick the monitoring mechanism by piggybacking her own data to a packet in transit because of the blinding effect of a hop-by-hop encryption.

- Homogeneous network assumption

We assume a network formed by *homogeneous nodes* that have equal computational power and energetic availability. Every node relies on wireless cards that use omnidirectional antennaes and operate in promiscuous mode.
If the latter requisite is inherent to CORE, we believe necessary to address nodes equipped with the same amount of energy and the same computational power. Even if the homogeneous network assumption is arguable because fairly realistic, security requirements would be considerably augmented in the heterogeneous case. Reputation management of an heterogeneous network would require a differentiated weighting base depending on the node

type. Furthermore, node energetic type (e.g. a PDA, a Laptop, a mobile VoIP-phone) is difficult to estimate. Evaluation of node's energy would require an enhanced monitoring mechanism to analyze traffic generation patterns or require nodes themselves to advertise their type, with the inherent security issues due to the untrustworthy nature of the nodes.

- Node identities

Cooperation enforcement mechanisms call for *uniquely identifiable* nodes: the unreliable environment offered by open ad hoc networks requires identities to be **unspoofable and unforgeable**. In this Chapter we assume an external mechanism (e.g. the key management mechanism) that provides unique and verifiable identities. We also assume that *bogus identity generation* is either impossible or costly in economical terms.
In Chapter 7 we present a key management scheme for open ad hoc networks that provides the secure naming services required by CORE: the need for an integrated approach for supplying a comprehensive set of (ad hoc) network security services is emphasized.

- Communication pattern assumption

We assume a **dense** communication pattern: a large subset of the nodes forming the network, including *both* legitimate and selfish nodes, is involved in data communication. Including also selfish nodes in the communication patterns is of fundamental importance to appreciate the features of CORE. This assumption is directly related to the selfishness model: "rational" selfish nodes fully experience the effects of the punishment mechanism (that is, the gradual denial of communication services) only if communication is valued as highly profitable.
The communication pattern assumption can be relaxed if the selfishness model is extended for example to the application layer: a selfish behavior detected at the network layer could entail a punishment that restricts application capabilities. As a practical example, in a peer-to-peer file sharing application running on the ad hoc network, an effective punishment could be the limitation of query capabilities of the selfish node: the need for a dense communication pattern would be mitigated.

- Routing

Throughout this Chapter, we assume that nodes perform discovery and maintenance of routes using the dynamic source routing (DSR) protocol. We selected DSR as the representative of reactive (or on-demand) ad hoc routing protocols due to its critical cooperation requirements, that also characterize reactive routing protocols in general.
Every time a source node needs to send a packet, a route discovery procedure has to be initiated[1] and can be accomplished only with the support of other nodes in the network.

[1]If the node already has a route towards the intended destination, for example in its cache, the discovery phase can be skipped.

However, as presented in section 2.2.2 of Chapter 2, lack of cooperation during the route discovery phase is particularly advantageous for selfish nodes that attempt to "hide" from being selected as relay nodes on a path from a source to the corresponding destination. As a result, network performance can be severely degraded. Furthermore, if countermeasures are not available to face a selfish routing behavior, lack of cooperation to the route discovery phase has no side effects for the attackers. As opposed to ad hoc networks operated by pro-active routing protocols[2] where lack of cooperation to the routing signalling have a (negative) impact on the routing table evaluation **also** for selfish nodes, in reactive protocols routes are established only when they are needed, often following very simple algorithms[3], and lack of cooperation do not distort the topology discovered by the selfish nodes.
In DSR, route maintenance operations are triggered by the detection of broken links. Broken links can be detected through three different types of acknowledgements: layer-2 acknowledgements, promiscuous listening of neighborhood activities (or layer-3 passive acknowledgments) and "software-level" explicit acknowledgments. Overhearing neighbors activities would constitute a duplicate technique already implemented in the CORE mechanism and the properties of "software" acknowledgments have not been sufficiently explored by the research community: we thus assume that route maintenance operations are based on layer-2 acknowledgements. Only mobility and node failure are taken into account as events that trigger route error messages.

- Routing security

In order to prevent the negative effects of active attacks (presented in section 2.2.1, Chapter 2) perpetrated on the routing protocol, we assume routing operations to be secured through a secure routing mechanism such as the ones presented in Chapter 3 or in Chapter 7.
Active attacks might have an influence on the correct execution of the CORE mechanism: for example, in the case of traffic subversion, a misbehaving node forwarding traffic to the wrong next hop would be tagged as legitimate and the reputation value associated to the node's behavior would erroneously be increased.
In section 4.9, we outline an advanced version of CORE that supports networks lacking routing security mechanism.

[2]In pro-active routing protocols, path selection is performed through periodic signalling and update of routing tables following link-state or distance-vector routing principles. Erroneous or lack of routing information modify the entire topology interpretation for all the nodes.

[3]In DSR there is no shortest path algorithm: route requests are flooded until the destination is reached. The destination then reverses the route listed in the first packet she receives and eventually replies to subsequent route requests in order to optimize the responsiveness of the routing protocol in presence of route failures, through the use of caches.

4.4 General CORE operation

In our scheme, MANET nodes can be thought of as members of a community[4] that share a common resource. The key to solve problems related to node misbehavior derives from the **strong binding** between the utilization of a common resource and the cooperative behavior of the members of the community. Thus, all members of a community that share resources have to contribute to the community life in order to be entitled to use those resources.
However, the members of a community are often unrelated to each other and have no information on one another's behavior. We believe that **reputation** is a good measure of someone's contribution to common network operations. Indeed, reputation can be defined as the amount of trust inspired by a particular member of a community in a specific setting or domain of interest.
Members that have a good reputation, because they helpfully contribute to the community life, can use the resources while for members with a bad reputation, because they refused to cooperate, resource utilization is gradually withhold.
Our research pointed out two possible roles that a node can assume while operating the network: the *requestor* and the *provider* role. We use the notation requestor when referring to a node asking for the execution of a function f and the notation provider when referring to any entity supposed to participate to the execution of f. Finally, we will use the notation *trusted entity* when referring to a network entity with a positive value of reputation. Examples of f can be the packet forwarding (PF) function and the routing function.
Following we describe the basic actions performed by requestors and providers that use the CORE mechanism.

- **The requestor.** The requestor issues a request for the execution of the function f and monitors its execution by the visible providers (i.e. providers that are within the wireless transmission range) through the *detection* technique provided by the **monitoring mechanism** described in section 4.5. The requestor validates the result of the execution of f and, based on the outcome of the validation phase, updates the reputation ratings relative to the monitored providers using the **reputation manager** component described in section 4.6.

- **The provider.** As a provider receives a request for the execution of a function f, based on the reputation rating associated to the requestor it accepts or denies to serve the request. No explicit message is broadcasted by the provider, which simply drops the source traffic generated by a selfish node. The operations performed by providers in presence of selfish nodes are part of the **punishment mechanism** described in detail in section 4.7.

[4]In the reminder of this Chapter, subjects, entities, nodes are used as synonims.

CORE Components

This section provides an overview of CORE components. Interactions between the components and between CORE and networking function is also provided. Furthermore, we provide guidelines for a real implementation of the CORE mechanism by offering an evaluation of storage requirements as well as an analysis of alternative reputation and punishment methodologies.
The general CORE operation described in section 4.4 can be used as a basis in understanding how the components are used by a CORE-enabled mobile node. Figure 4.1 illustrates the building blocks of CORE and their interactions.

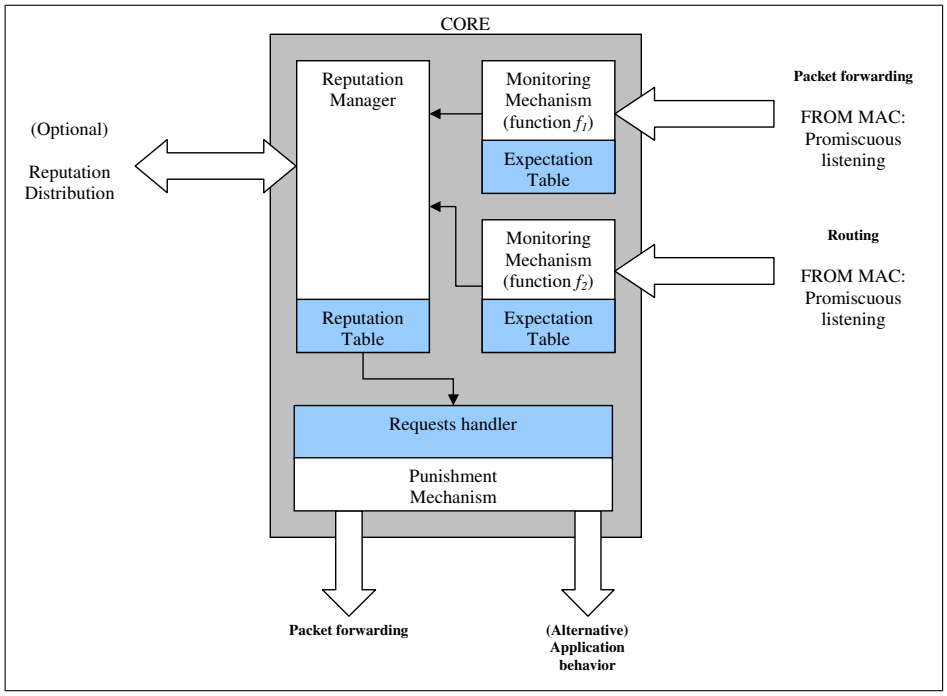


Figure 4.1: CORE architecture.

4.5 Monitoring mechanism

The monitoring mechanism is used to detect selfish behavior of nodes with respect to the execution of function f. A possible implementation of the monitoring mechanism is provided by Marti et al. [62] and is called the watchdog technique. The watchdog technique relies on the promiscuous mode operation of wireless radio cards based on the 802.11 standard and has some weaknesses that are discussed in section 4.5.1. We recall here that promiscuous mode means that if a node A is within range of a node B, it can overhear communications to and from B even if those communications do not directly involve A.

For sake of simplicity, we now switch from the generic notation describing with f an abstract function that needs to be monitored to a specific notation in which the function f is defined as the packet forwarding function. The watchdog (WD) technique implemented in CORE works as follows.

4.5.1 The watchdog technique

Whenever a node needs to send a packet, both as a source of data traffic and as a relay node forwarding a data packet, she adds an entry to the expectation table (see figure 4.1) consisting of the fields showed in table 4.1. An **expected packet** represents a proof

Field	***Description***
UID	Packet unique identifier
IP_S	IP source of the data traffic
IP_D	IP destination of the data traffic
MAC_S	MAC address of the *next hop* on the route
MAC_D	Optional: MAC address of the destination of relayed packet
h(payload)	A message digest (hash functiton) of the data packet that needs to be forwarded

Table 4.1: Expected packet stored in the *Expectation Table*.

of proper execution of the packet forwarding function that *still needs to be validated by an observation*, i.e. the packet that is eventually overheard by promiscuous listening the activities of the next hop on the route (available from the source route stored in every packet) towards the destination.
The first three fields (UID, IP_S, IP_D) uniquely identify the packet: when more than one packet needs to be sent for the same $< IP_S, IP_D >$ pair, the UID is used to distinguish different packets.
The MAC_S represents the layer-2 address of the next hop, which is included by the next hop into the appropriate field of the relayed packet. Optionally, depending on IP-to-MAC resolution services, the layer-2 address of the second next-hop can be included in the entry. This is done in order to prevent a route subversion performed by active attackers that tamper with the MAC fields of a packet in transit, while leaving routing information unaltered.
Finally, a message digest function (such as the popular `MD5` hash function) is used to verify the contents of the relayed packet in order to prevent piggybacking of data information

by a malicious relay node.
Figure 4.2 illustrate a simple network in which a packet is travelling from a source (M) to its destination (X) through the route $< ..., A, B, C, ... >$. In the figure it is possible to see the basic IP and MAC fields of a packet travelling on the wireless links (called relayed packet): an example of expected packet is shown for node A. Every expected packet is

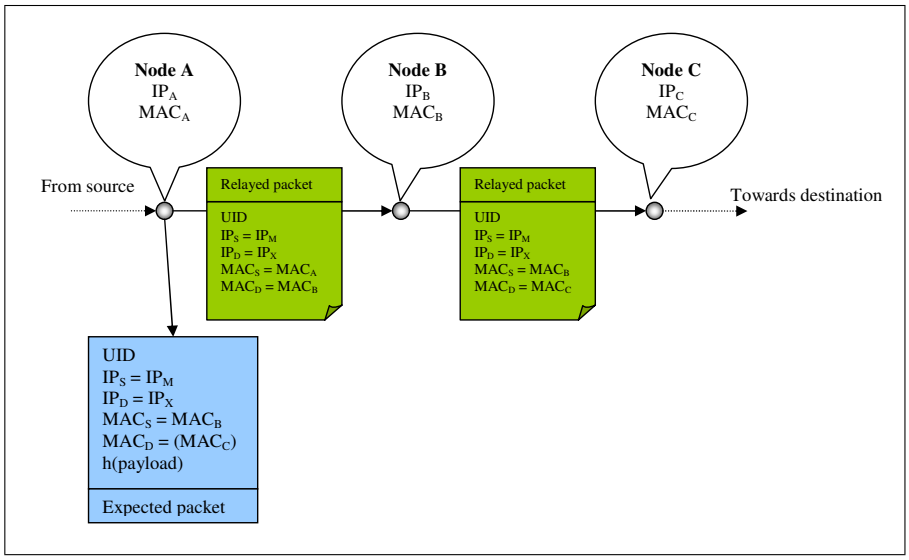


Figure 4.2: Simple network with a expected packet example.

associated to a timer which is set to expire after a defined time-out value WD_{TO}. After setting the timer for the expected packet stored in the expectation table, the monitoring entity enables the promiscuous mode operation and listen to the next-hop activities.
If the next-hop correctly forward the packet, the monitoring node is able to capture the packet which is called **observed packet**. The proof of the correct execution of the packet forwarding function *can be validated*: if all the expected packet's fields match the observed packet's fields, including the message digest function applied to the observed packet's payload, the expected packet entry is **removed** from the expectation table. On the contrary, if the WD_{TO} timer expires and the associated expected packet entry has not been removed from the expectation table, the proof of the next-hop's correct behavior is not validated[5].
We define an observation $\sigma_k \in$ Z as the result of the validation process described above:

$$\sigma_k = \begin{cases} v_+, \text{if expected packet} = \text{observed packet} \\ v_-, \text{if expected packet} \neq \text{observed packet, or } WD_{TO} \text{ expires} \end{cases} \quad (4.1)$$

σ_k represents the *rating factor* given to the *k-th* observation made on the neighbor's behavior and is used by the reputation management component to evaluate the reputation associated to the monitored entity.
v_+ and v_- represent the outcomes associated to the observations: in section 4.6 we will

[5]If the validation process fails, the expected packet entry is removed

show the impact of these values on the reputation metric. As an example, $v_+ = +1$ and $v_- = -1$.

Note: when the monitoring mechanism is used to detect node misbehavior when executing the packet forwarding function, the watchdog mechanism is not set if the monitoring node is the last hop before the destination on the path provided by the routing protocol. Indeed, the last hop before the destination does not expect the relayed packet to be forwarded again by the intended destination.

4.5.2 Storage and energetic requirements

Mobile ad hoc networks are formed by nodes with limited resources: energetic availability and computational power are critical factors that need to be taken into account for the design of security components executed by the nodes. **Storage requirements** represent another important criterium that must be addressed in order to provide security mechanisms that can be used in practice.
In section 4.5.1, the expectation table is loaded with an entry for every packet in transit or generated by the monitoring entity. We minimize the storage requirements by using a hash function applied to the payload of the packet that needs to be monitored. A rough evaluation of the size of an expected packet is provided below (we assume a hash function on $160bit$):

$$sizeof(\text{Expected packet}) = 12\ (UID) + 2 * 32\ (IP_{S,D}) + 2 * 48\ (MAC_{S,D}) + 160\ (h(payload)) = 332bit$$

$332bit$ are required every time a packet needs to be monitored: for example, if 1000 packets need to be monitored, in the worst case (i.e. all expected packets are stored before the validation phase or before the WD_{TO} expires) only $40Kbytes$ are needed.
In the presence of a heavy traffic load, it might be necessary to lower the rate at which packets are monitored. We define the monitoring rate as the **watchdog sampling frequency** (WD_{freq}). As opposed to the basic version of the WD in which every packet in transit or generated by a node needs to be monitored, i.e. when $WD_{freq} = 1$[observations per packet], the sampling frequency can be reduced to $WD_{freq} = 0.2$ [observations per packet], i.e. only 20% of the outgoing traffic is monitored.
The WD sampling frequency has an influence on the **energy consumption** expended during monitoring phase. While in promiscuous mode operation, the nodes receive every packet in transit within the wireless radio range: it is clear that energy consumption is considerably higher than in the normal operation in which only packets directed to the node are processed. In order to reduce energy consumption, the watchdog sampling frequency can be lowered: a discussion on how and when the monitoring frequency can be modified is given in section 4.9. As a practical example, the sampling frequency could be decreased if the average reputation value of the neighborhood is higher than a predefined

threshold: if neighbors show to be trustworthy, it might be un-necessary to constantly monitor their activities.

4.5.3 Alternatives to the watchdog mechanism

The watchdog technique has advantages and weaknesses. The DSR routing protocol enhanced with the watchdog technique has the advantage that it can detect misbehavior at the forwarding level and not just at the link level.
Watchdog's weaknesses (see figure 4.3) are that it might not detect a misbehaving node in the presence of 1) ambiguous collisions, 2) receiver collisions, 3) limited transmission power, and 4) modified transmission range.
The ambiguous collision problem prevents A from overhearing transmissions from B. As

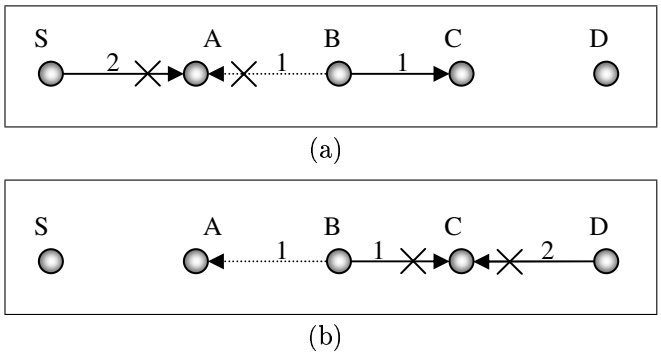


Figure 4.3: Watchdog technique problems: in figure 4.3(a) node A does not hear B forward packet 1 to C, because B's transmission collides at A with packet 2 from the source S. In figure 4.3(b) node A believes that B has forwarded packet 1 on to C, though C never received the packet due to a collision with packet 2.

Figure 4.3(a) illustrates, a packet collision can occur at A while it is listening for B to forward on a packet. A does not know if the collision was caused by B forwarding on a packet as it should or if B never forwarded the packet and the collision was caused by other nodes in A's neighborhood. Because of this uncertainty, A should not immediately accuse B of misbehaving, but should instead continue to watch B over a period of time. If A repeatedly fails to detect B forwarding on packets, then A can assume that B is misbehaving.
In the receiver collision problem, node A can only tell whether B sends the packet to C, but it cannot tell if C receives it (Figure 4.3(b)). If a collision occurs at C when B first forwards the packet, A only sees B forwarding the packet and assumes that C successfully receives it. Thus, B could skip retransmitting the packet and leave A none the wiser. B could also purposefully cause the transmitted packet to collide at C by waiting until C is transmitting and then forwarding on the packet. In the first case, a node could be selfish and not want to waste power with retransmissions. In the latter case, the only reason B would have for taking the actions that it does is because it is malicious. B wastes battery

power and CPU time, so it is not selfish. An overloaded node would not engage in this behavior either, since it wastes badly needed CPU time and bandwidth. Thus, this second case should be a rare occurrence.
Another problem is that a misbehaving node that can control its transmission power can circumvent the watchdog. A node could limit its transmission power such that the signal is strong enough to be overheard by the previous node on the route but too weak to be received by the true recipient. This would require that the misbehaving node know the transmission power required to reach each of its neighboring nodes. Only a node with malicious intent would behave in this manner: selfish nodes have nothing to gain since battery power is wasted and overloaded nodes would not relieve any congestion by doing this. Furthermore, the assumption of a homogeneous network formed by nodes with a fixed radio range protects us from this problem.
For the watchdog to work properly, it must know where a packet should be in two hops. In our implementation, the watchdog has this information because DSR is a source routing protocol. If the watchdog does not have this information (for instance if it were implemented on top of a hop-by-hop routing protocol), then a malicious or broken node could broadcast the packet to a non-existing node and the watchdog would have no way of knowing. Because of this limitation, the watchdog *works best on top of a source routing protocol.*
Due to the limitations presented above, recent studies [49] have proposed alternatives to the watchdog technique that provide detection capabilities to monitoring nodes. An interesting technique presented in [49] is based on *probe packets* that are sent together with real data traffic in order to infer the forwarding behavior of relay nodes on the path from a source to the corresponding destination. It is out of the scope of this thesis to precisely describe the probing technique: for our purposes it is sufficient to focus on the requirements of this alternative technique. The main problem with probe packets is that they must be indistinguishable from real data packets: indeed, a selfish node would simply have to forward probe packets and drop data packets to trick the detection mechanism. In order to apply a "blinding" function to probe packets the authors rely on some cryptographic primitives that in turn need the distribution of keying material to all nodes of the network. Thus, a security infrastructure is needed in order to use this alternative detection technique.
Alternatively, it would be interesting to explore detection techniques based on *acknowledgement (ACK) messages*: it is possible to differentiate ACK messages in two categories. On one hand, end-to-end acknowledgements provide a failure detection method for a packet travelling from a source to its destination. However, it would be impossible for the source of traffic to distinguish which node on the path failed to forward data packets. Furthermore, end-to-end ACK messages would introduce a relatively high traffic overhead in the network.
On the other hand, hop-by-hop ACK messages would help to uniquely identify the selfish node on the path: however, integrity and authentication of hop-by-hop ACK messages is imperative in order to avoid fabrication of ACK messages in the same way that have been described for the fabrication of false routing information (see Chapter 2). Hop-by-hop ACK messages would introduce a high traffic overhead in the network and their generation and transmission would increase energetic consumption.

In conclusion, even if the watchdog technique has some inherent issues due to the 802.11 MAC protocol functioning, it does not introduce any additional traffic overhead in the network and the energetic consumption can be controlled by modifying the watchdog sampling frequency. Furthermore, as explained in the following section 4.6, the effect of eventual errors due to false detection, or due to sporadic node misbehavior because of a temporary failure, can be mitigated by introducing the reputation metric: more than one observation is needed to infer the behavior of a neighbor and assign a corresponding reputation value.

4.6 Reputation Manager

This section describes how reputation is evaluated in CORE. Our scheme is based on three types of reputation and following we show how they are combined. Reputation is formed and updated along time through **direct observations** and, *optionally*, through **indirect information** provided by other entities of the network. Furthermore, we take the stance that reputation is compositional: the overall opinion on an entity that belongs to the network is obtained as a result of the combination of different type of evaluations. We define a **direct reputation**, an **indirect reputation** and a **functional reputation**. The reputation manager component receives a predefined number of past direct observations collected through the monitoring component and stores them in a data structure that we call Reputation Table (RT), as depicted in table 4.6.4. These direct observations are used to calculate the direct reputation ratings that the monitoring node associates to her neighbors.
Reputation information stored in the RT is then made available for queries initiated by the punishment component, which uses reputation ratings to provide or deny service to neighboring nodes, as described in section 4.7.
Following we describe different **reputation types** that are used by CORE.

4.6.1 Direct Reputation

We use the term direct reputation when referring to the reputation calculated directly from a subject's observation through the monitoring mechanism. The direct reputation evaluated by node n_i at (discrete) time k is calculated using a weighted mean of the observations' rating factors, giving more relevance to the past observations[6]. The general formula to calculate a direct reputation is:

$$r_{n_i}^k(n_j|f) = \sum_{l=k-B}^{k} \rho(B,l)\,\sigma_l \tag{4.2}$$

[6]The reason why more relevance is given to past observations is that a sporadical misbehavior in recent observations should have a minimal influence on the evaluation of the final reputation value

where $r_{n_i}^k(n_j|f)$ stands for the direct reputation value calculated at time k by node n_i that represents node n_j's behavior with respect to the function *f*.
σ_l represents the rating factor given to the *l-th* observation as defined in equation (4.1).
B is an important parameter that determines the number of observations needed to evaluate the reputation. Practically, B is the size of the *FIFO* (first in, first out) *buffer* that stores up to B observations:

$$< \sigma_{k-B}, \sigma_{k-B-1}, ..., \sigma_{k-1}, \sigma_k >$$

$\rho(B, l)$ is a time dependent function that gives higher relevance to past values of σ_l.
As explained in section 4.5, a new observation is made for every packet that node n_i sends (as a source or as a relay) to node n_j. When a new observation σ_k is available, the reputation manager updates $r_{n_i}^k(n_j|f)$: the oldest observation in the buffer is erased and the new observation is inserted in the right-most position of the buffer.
Following we provide an example of the direct reputation evaluation process through a MATLAB simulation. The direct reputation component receives as input a "simulated" behavior pattern as if it was generated by the watchdog mechanism: the behavior pattern is formed by a stream of observations made by node n_i on the behavior of node n_j and the observation outcomes can take the values $\{v_+, v_-\}$. In our examples, $v_+ = +1$ and $v_- = -1$.
The output is plotted as the direct reputation $r_{n_i}^k(n_j|f = PF)$ where the monitored function f is, for example, the packet forwarding function (PF).
The reputation value can be implemented by observing that equation (4.2) is nothing but the expression of a finite impulse response (FIR) filter. FIR filters have the distinctive trait that their impulse response lasts for a finite duration of time. The input-output function in the time domain has the following non-recursive expression:

$$a_1 * y[n] = b_1 * x[n] + b_2 * x[n-1] + ... + b_B * x[n-B] \tag{4.3}$$

Expression (4.3) can be re-written in the Z-transform domain as follows:

$$H[z] = \frac{Y[z]}{X[z]} = \frac{b_1 + b_2 z^{-1} + b_3 z^{-2} + ... + b_B z^{-B}}{a_1} \tag{4.4}$$

where B represents again the number of past inputs (i.e. past observations) that are taken into account for the output evaluation (i.e. the reputation value). The (complex) roots of the polynomial in the numerator of expression (4.4) are called **zeros**, while the roots of the denominator are called **poles**. It is out of the scope of this thesis to provide the basis for the understanding of digital filter design, but we provide some notation that will be used in the following. As it is possible to see in equation (4.4), the transfer function contains only zeros and one pole (in the origin): because of this particular form, a well-known theorem in numerical analysis (the *Nyquist theorem*) states that FIR filters are always **stable**. The *number* and the *position* of the poles and zeros of the transfer function when they are represented on the complex plane determines the filter "shape" and the properties of the filtering function.
FIR filters can be of different types: in our case, we want past observations to have more relevance than newest observations. A **low-pass FIR filter** represents an effective way

of evaluating the direct reputation for a node: a sporadic misbehavior take the form of a irregular behavioral pattern (i.e. the vector that contains all the observations collected through time) with *"high-frequency"* variations of the observation values while a persisting misbehavior or legitimate behavior result in *"low-frequency"* variations of the behavioral pattern. A simple low-pass filter **transfer function** can be used to discern continuous misbehavior from a sporadic misbehavior.
In the following diagrams, we present different types of filters by expressing the coefficients of the numerator of expression (4.4). We start with a **first-grade low pass filter** and then add other zeros (increase the filter grade, i.e. increase B) to modify the *"shape of the filter"*. Each of the following figures depicts: a) the simulated behavioral pattern and the (*normalized*) reputation value evaluated with the filter, b) the "shape of the filter", i.e. the magnitude diagram of the filter and c) the *"pole-zero diagram"* which shows the location (i.e. the value) of the zeros on the unitary circle in a complex plane. Observations are made for every presented plot and are based on the same behavioral pattern. Following, we provide some examples of reputation ratings based on the order of the filter that is used to calculate the direct reputation value.

- "Mild" reputation rating

The first FIR filter used to evaluate the direct reputation value calculated by node n_i for node n_j is a **first-order** low pass filter with the following numerator coefficient (refer to expression (4.4)): $[1, -0.9]$.
Diagrams 4.4(b) and 4.4(c) show respectively the filter shape and the position of the only zero of the transfer function.
The behavioral pattern in figure 4.4(a) presents some distinct continuous zones in which a regular behavior has been detected through the monitoring mechanism. The pattern also present some unstable zones in which there are high frequency variations of the observations due for example to a failure of node n_j or problems with the monitoring mechanism.
Figure 4.4(a) shows a reputation rating that has the following characteristics:

1. Reputation "*slowly*" follows a continuous pattern, but the detection of a misbehavior entails a "*rapid*" loss of reputation.

2. On the other side, sporadic misbehavior is treated in a "*soft*" way: the reputation reduction rate is reduced with respect to a sudden (and continuous) misbehavior. As it is possible to see in the graph, the first irregular part of the pattern is not sufficient to drop the reputation value to zero. The second irregular part of the pattern has more influence on the reputation value.

- "Hard" reputation rating

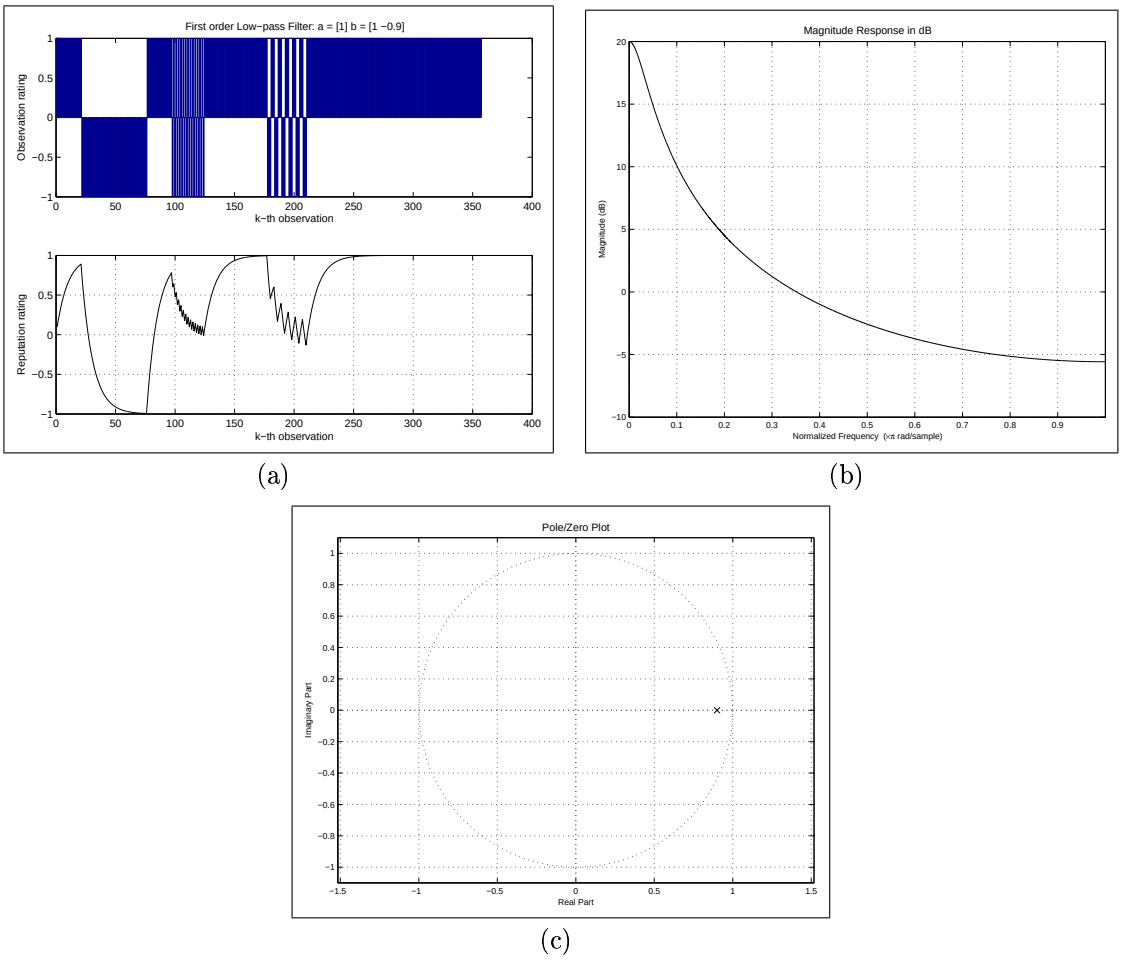


Figure 4.4: **Mild rating**: reputation evaluated through a first-order low pass FIR filter.

The second FIR filter we used is a modified **first-order** low pass filter with the following numerator coefficient: $[1, -0.6]$.
Again, diagrams 4.5(b) and 4.5(c) show respectively the filter shape and the position of the only zero of the transfer function.
Figure 4.5(a) shows a reputation rating that has the following characteristics:

1. Reputation "*rapidly*" follows a continuous pattern. The detection of a legitimate or selfish behavior entails a rapid gain or loss of reputation that "saturates" to the highes or lowest value.

2. On the other side, the reputation rating closely follows sporadic misbehavior and oscillates between negative and positive values.

- "Soft" reputation rating

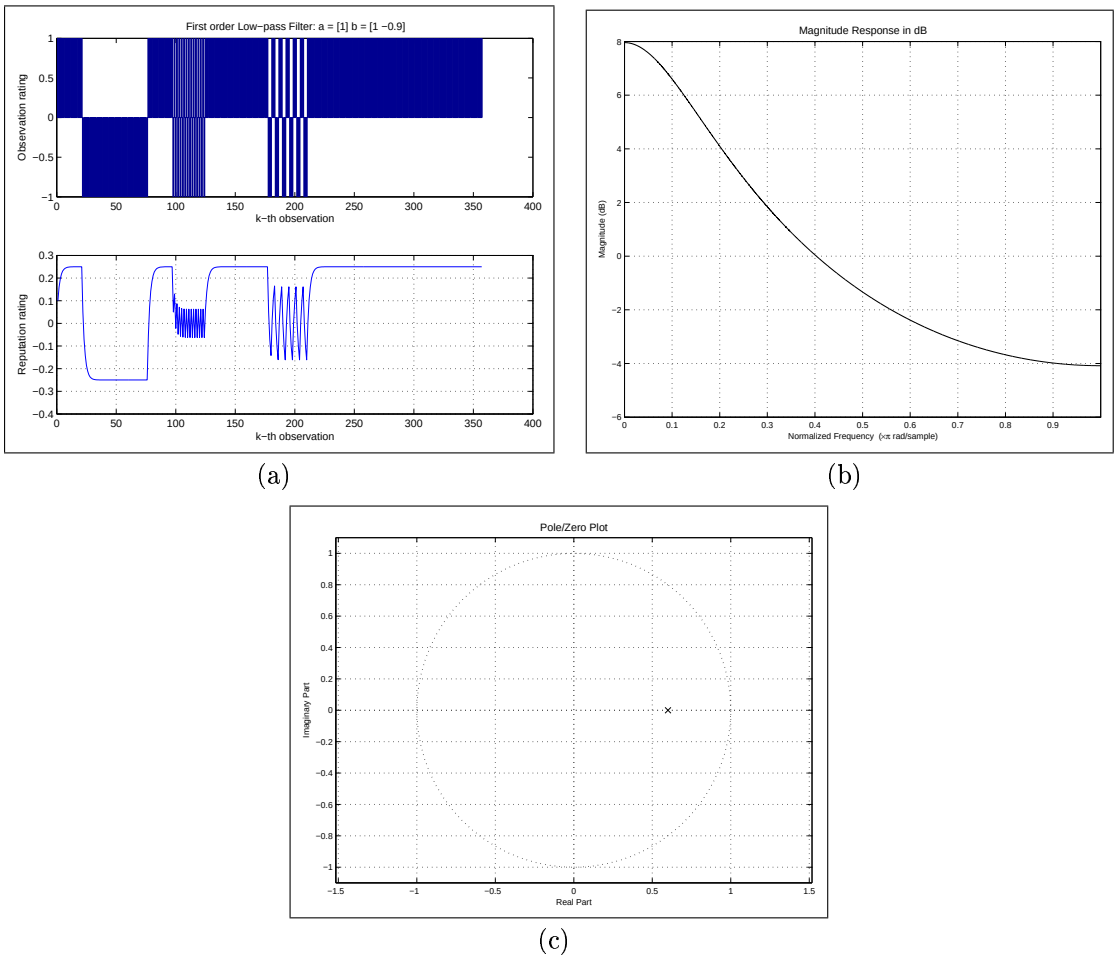


Figure 4.5: **Hard rating**: reputation evaluated through a first-order low pass FIR filter.

The third FIR filter we used is a **first-order** low pass filter with the following numerator coefficient: $[1, -0.97]$.
The filter shape and the position of the only zero of the transfer function are respectively shown in diagrams 4.5(b) and 4.5(c).
Figure 4.6(a) shows a reputation rating that has the following characteristics:

1. Reputation associated to a continuous pattern is hard to build and hard to loose. The detection of a selfish behavior does not immediately entail a negative reputation rating.

2. On the other side, the first sporadic misbehavior present in the behavioral pattern has a low impact on the reputation ratings that are almost constant. The impact of the second sporadic misbehavior has a higher impact on the reputation loss rate.

- "Square" reputation ratings

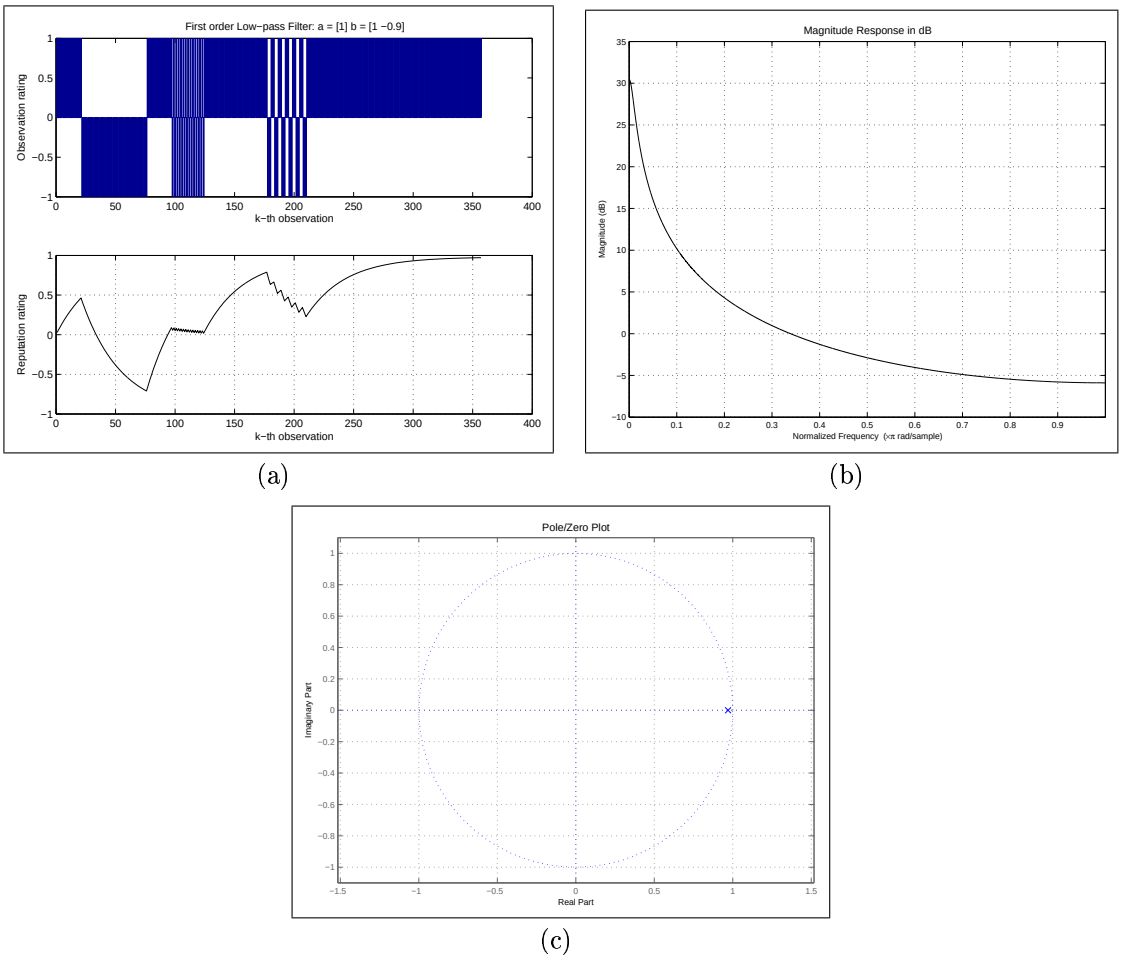


Figure 4.6: **Soft rating**: reputation evaluated through a first-order low pass FIR filter.

We now consider a **second-order** FIR filter that cuts high frequency variations by applying a sharp filter shape to the behavioral pattern. The following numerator coefficient have been used : $[1, -0.8, 0.2]$. The filter shape and the position of the two zeros of the transfer function are respectively shown in diagrams 4.7(b) and 4.7(c).

Figure 4.7(a) shows that the reputation rating associated to the "simulated" pattern is similar to the "hard" reputation case, but exasperated: the reputation slope is almost *square*, and precisely follows the behavioral pattern generated by the monitoring mechanism.

- "Gentle" reputation rating

The last filter we used to calculate the reputation rating associated to the behavioral pattern used in all our examples is a **second-order** low pass filter with the following coefficients: $[1, -0.98, -0.001]$. Diagrams 4.8(b) and 4.8(c) show respectively the filter

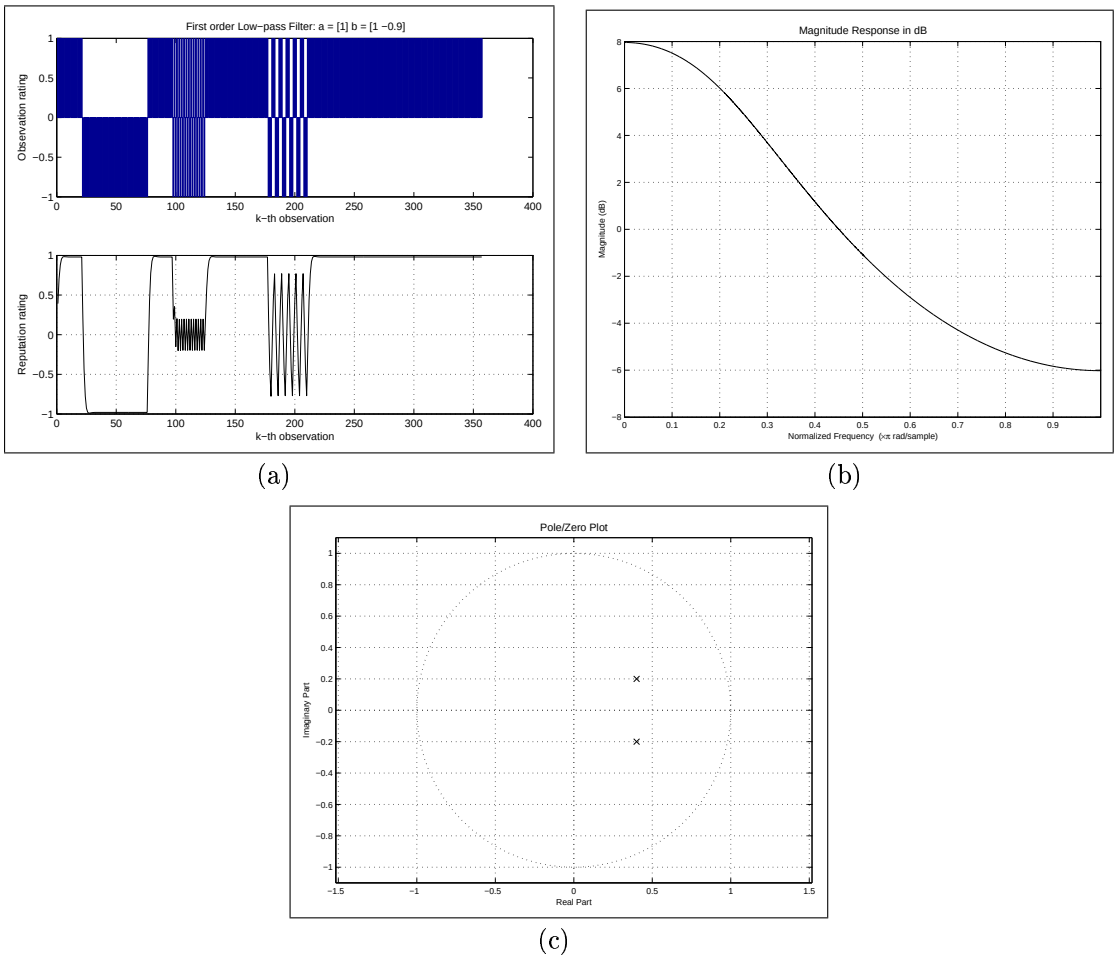


Figure 4.7: **Square rating**: reputation evaluated through a second-order low pass FIR filter.

shape and the position of the two zeros of the FIR filter transfer function.

Figure 4.8(a) shows that the reputation slowly increases (or decreases) with a constant behavior. However, also sudden sporadic misbehavior have a very little impact on the reputation evaluation process: for example, the impact of the first sporadic behavior encountered in the behavioral pattern is mitigated and does not entail a loss in reputation.

Discussion: As we will show in Chapter 5, the choice of the filtering function has an impact on the detection capabilities of CORE. Precisely, the reputation technique exposed in this section has the practical advantage of minimizing the problems due to errors of the CORE monitoring component, which is based on the watchdog technique.
We believe that the "mild", "soft" and "gentle" reputation ratings are better suited for a generic setting in which it is hard to predict phenomena such as collisions, interference and traffic overload. When those phenomena occur, detection errors increase (section 4.5 reports on the issues related to the basic watchdog technique) and the reputation compo-

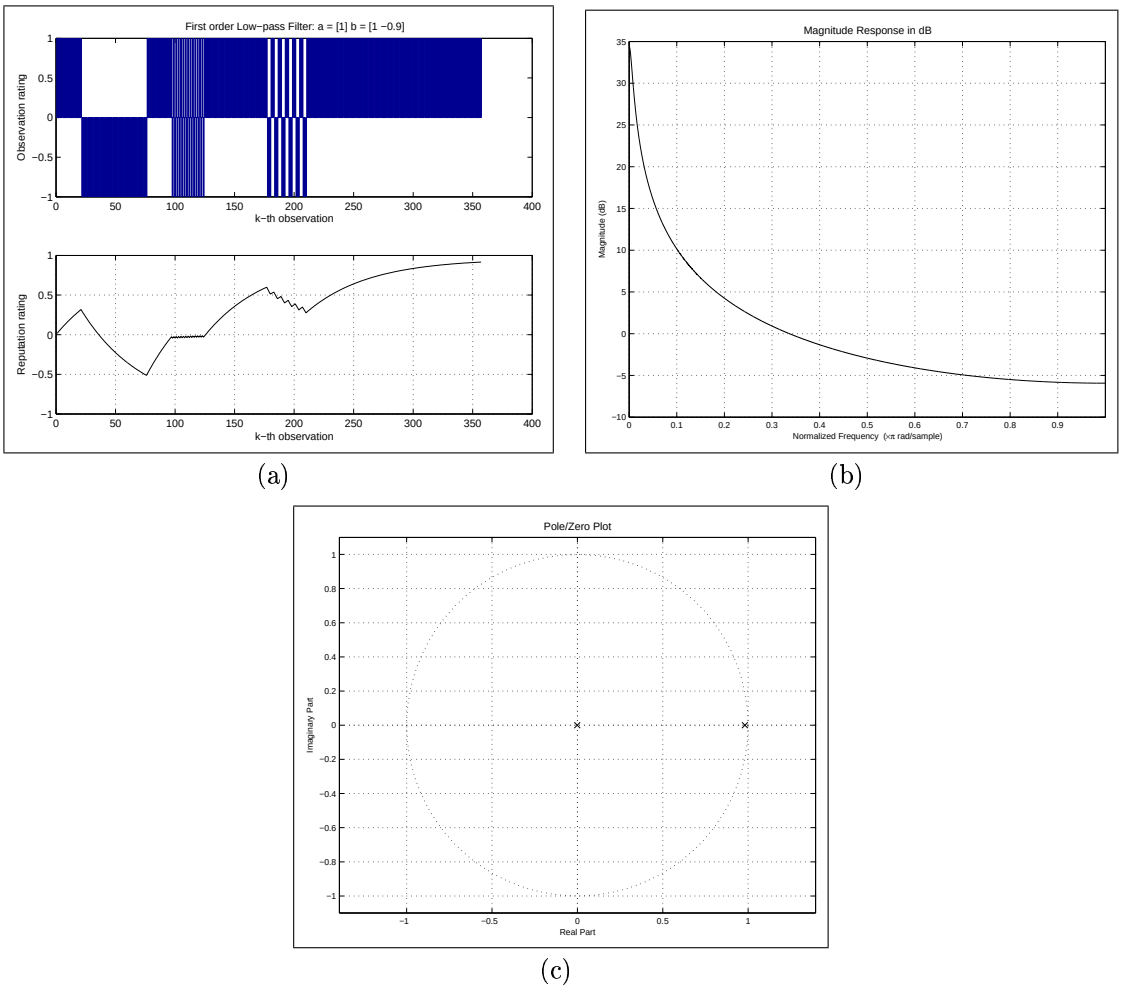


Figure 4.8: **Gentle rating**: reputation evaluated through a second-order low pass FIR filter.

nent must cut off mis-perception errors by applying the appropriate filtering function. Reputation evaluation can also be based on a simple form of filtering function (that we also used in our simulation-based validation of CORE, see Chapter 5), i.e. a moving-average filter with window size equals to the parameter B. In this particular case, only the parameter B, which corresponds to the number of observations needed to calculate the direct reputation, has an influence on the behavior of the reputation component with respect to monitoring errors. Sections 5.2.4 and 6.1.7 in Chapter 5 presents the impact of the parameter B on the performance of CORE.

In a more advanced version of CORE, we envision to consider more complex filtering functions: instead of using low-pass filters to cut off high-frequency variations in node behavior we will explore the impact of using **passband filters** to selectively discern particular behavioral frequencies. We deem this research direction to be strongly coupled with concepts borrowed from traffic analysis theory. Indeed, a more accurate analysis of the traffic overheard by the monitoring component can provide the reputation component

with more information on specific behavioral frequencies that should be attributed to a selfish activity of neighboring nodes.

4.6.2 Indirect Reputation

In our scheme, the direct reputation is evaluated *only* considering the direct interaction between a subject and its neighbors. With the introduction of the indirect reputation measure we add the possibility to reflect in our model a characteristic of complex societies: the final value given to the reputation of a subject is influenced also by the observations made by other members of the community on the same subject.
The reputation manager component implemented in CORE **optionally** use indirect reputation ratings to improve the detection capabilities of the watchdog mechanism, which is able to detect only misbehaving nodes that are within the wireless radio range of the monitoring node. In order to use indirect reputation ratings it is necessary to define a reputation distribution mechanism. Moreover, security issues such as reputation information integrity must be taken into account. In this section we only describe how indirect reputation information is used and combined with direct reputation ratings. In section 4.5 we discuss in more detail the implications and the requirements that are optionally needed if indirect ratings are used.
The general formula used to evaluate the indirect reputation is:

$$r^k_{n_i}(n_l, n_j|f) = r^k_{n_j}(n_l|f) \tag{4.5}$$

where $r^k(n_l, n_j|f)$ denotes the indirect reputation of node n_l evaluated by node n_j at time k for the function f. Note that the indirect reputation value for node n_i over node n_l is equivalent to the direct reputation value for node n_j over node n_l.

Reputation distribution

Direct reputation values evaluated by a node through the watchdog technique could be distributed to other nodes to create a network global view of a node's reputation. Any bad behavior directly experienced by a node will be relayed to a subset of neighboring nodes (1-hop neighborhood, 2-hop neighborhood or even the whole network), so that bad behavior is discouraged more than if reputation remained local knowledge. As presented in Chapter 3, the CONFIDANT mechanism sends reputation messages, reporting any bad behavior which is experienced or observed. In CONFIDANT, indirect information is processed through a complex weighting system called the beta reputation system (initially presented in [16]).
Many of the problems faced by distributed reputation schemes are the same as those faced by any other distributed scheme. For example, reputation messages could be **modified** or **replayed**. Moreover, reputation messages may themselves be accidentally **lost**. As

a result, there is a strong likelihood that serious inconsistencies will arise within a community as to the reputation values for nodes in the network. If these inconsistencies can be exploited by other nodes, then this is a serious vulnerability. Another important issue is the volume of **additional messages** which may be needed to support the distributed system. As bandwidth may be very limited, the priority may be on using the available bandwidth for emergency data rather than for reputation information.
We mention three types of behavior which can give rise to threats when reputation values are distributed throughout an ad hoc network:

- Advertising a false high rating about another node;
- Advertising a false low rating about another node;
- Negative discrimination, where a node refuses services to only some nodes; this can be random or targeted at certain nodes.

These threats cannot be addressed by conventional security mechanisms that provide indirect reputation messages integrity and authentication. If simple attacks performed by malicious nodes who tamper with indirect reputation messages can be prevented by message integrity verification, there is nothing to protect against **false but authentic** advertisements provided by malicious nodes that want to isolate legitimate nodes.
In section 4.6 we suggested to weight indirect reputation advertisements by using local reputation values attributed to the advertising node: the judgment of a trusted node (in the sense that she has a good reputation) weights more than the one coming from a relatively trusted node. Also in CONFIDANT, through the beta reputation system, indirect information is weighted based on the reputation of the source of information. However we claim that this approach is **not sufficient** to assure a valuable reputation evaluation. Indeed, there is nothing that prevents an attacker to behave in a legitimate way and gain reputation in order to diffuse "credible" false information.
In conclusion we believe that reputation information **should not** be distributed, but we designed CORE to be compliant with the eventual utilization of indirect reputation information. We suggest however to explore the properties of a voting system used as a complementary mechanism to validate indirect reputation information. If we assume malicious node collusion to be limited, a simple majority vote over an indirect reputation value could be sufficient to prevent the threats described in this section. Furthermore, in section 4.9 we propose a variation of the monitoring mechanism that is used to detect eventual node misbehavior also when the monitoring node is not directly involved in a requestor/provider relationship (as defined in section 4.4).

4.6.3 Functional Reputation

We use the term functional reputation when referring to direct and indirect reputation information calculated with respect to different functions *f*. With the introduction of this

last type of reputation in our model we add the possibility to calculate a global value of a entity's reputation that takes into account different observation/evaluation criterium. As an example, node n_i can evaluate the direct reputation $r^k_{n_i}(n_j|f(packetforwarding))$ of node n_j with respect to the packet forwarding function and the direct reputation $r^k_{n_i}(n_j|f(routing))$ with respect to the routing function and combine them using different weights to obtain a global reputation value on node n_j.

4.6.4 Reputation combination

Reputation information is combined using the following formula:

$$r^k_{n_i}(n_j) = \sum_{x \in \{PF,R\}} w_x \left\{ r^k_{n_i}(n_j|\, f_x) + \sum_{z \in \{neighborhood\}} \lambda_z r^k_{n_i}(n_j, n_z|\, f_x) \right\} \tag{4.6}$$

where $x \in \{PF, R\}$ takes into account the functions (PF = packet forwarding, R = routing) that needs to be monitored and $z \in \{neighborhood\}$ represent the neighborhood that is used to collect indirect reputation information, as discussed in section 4.6.2.
w_x represents the weight (i.e. the importance) associated to the each functional reputation value.
λ_z is a weight associated to the indirect reputation information provided by node n_z. λ_z is directly related to the reputation value (if present) assigned by node n_i to node n_z. Indirect reputation is weighted based on the reputation rating associated to the informer. A high value of λ_z corresponds to a node with a high reputation value, and viceversa. Indirect information provided by cooperating nodes has an important impact on the calculation of the final reputation value for the node that is being evaluated while information provided by non-cooperating nodes has less influence on the final reputation value.
$r^k_{n_i}(n_j)$ represents a global reputation, that is, the aggregate reputation that combines direct, indirect and functional reputation information.
Besides the global reputation value, it is important to know how reliable is that value. Although there are a lot of elements that can be taken into account to calculate how reliable a global reputation is, we propose two of them: the number of evaluations used to calculate the final reputation value and its variance. However, we consider the use of indirect reputation a delicate problem that need to be discussed in more details. If the global reputation value is calculated using *only direct observations* (over the targeted functions), its reliability is increased: only errors caused by the watchdog mechanism need to be taken into account.

Reputation information is stored in a Reputation Table (RT) that is defined as a local data structure managed by each network entity. Each row of the table includes the reputation data pertaining to a node. Each row consists of four entries: the unique identifier of the entity, a collection of recent direct observations made on that entity's

behavior, a list of the recent indirect reputation values provided by other entities and the value of the reputation evaluated through expression (4.6) for a predefined function. Table 4.6.4 shows a typical RT for a defined function *f*.

Node UID	*Last B observations*	*Indirect Reputation*	*Reputation*
n_j	$\sigma^j_{k-B}, \sigma^j_{k-(B-1)}, ..., \sigma^j_k$	$r^k_{n_i}(n_j, n_l\|f), r^k_{n_i}(n_j, n_t\|f), ...$	$r^k_{n_i}(n_j\|f)$
n_m	$\sigma^m_{k-B}, \sigma^m_{k-(B-1)}, ..., \sigma^m_k$	$r^k_{n_i}(n_m, n_t\|f), r^t_{n_i}(n_m, n_l\|f), ...$	$r^k_{n_i}(n_m\|f)$
...	...	...	...

Table 4.2: Reputation Table stored in node n_i for function f

4.7 Punishment mechanism

This section describes the punishment mechanism used by CORE to selectively castigate a selfish behavior detected through the monitoring mechanism and processed by the reputation management component. The punishment mechanism bases its actions on the information provided by the reputation component: during a generic requestor/provider transaction, as described in section 4.4, the provider node, *i.e.* the node that is asked to execute the function f (the packet forwarding function, for example), verifies the reputation value associated to the requestor. A predefined threshold P_{th} determines whether the reputation earned by the requestor is enough to be entitled with the rights of being served. As an example, if node n_j requests the execution of the packet forwarding function to the provider node n_i, n_i checks wether $r^k_{n_i}(n_j) \geq P_{th}$. In node n_j's reputation is high enough, the provider node n_i do not deviate from a normal behavior, *i.e.* serves the requestor by executing the requested function f. On the contrary, if the reputation rating of node n_j falls below the punishment threshold P_{th}, the provider denies service provision by simply **dropping the request** of the selfish node. For example, the punishment threshold $P_{th} = 0$ activate the punishment mechanism for nodes that have a negative reputation rating.
It is important to point out that only requests **originated** by the *neighboring* selfish node n_j are dropped.
The following example illustrates a punishment inflicted by node n_i to the selfish node n_j (with $r^k_{n_i}(n_j) < P_{th}$) that try to send packets to the destination n_b. As it is possible to see in figure 4.9, only packets that have $IP_S = n_j$ are dropped **without any notification**, while packets originated in node n_a are relayed. This is done in order to take into account the possibility of a **selective selfish behavior** in which the selfish node n_j only misbehave with node n_i but not with node n_l.
The potential threat caused by a selfish node that modifies the IP source field (IP_S) of her own packets must be addressed through the integration of a secure routing mechanism that prevents this type of **modification attacks**.
By using the reputation ratings evaluated by CORE, a node can gradually deny network utilization to neighboring selfish nodes: however, the punishment mechanism have an

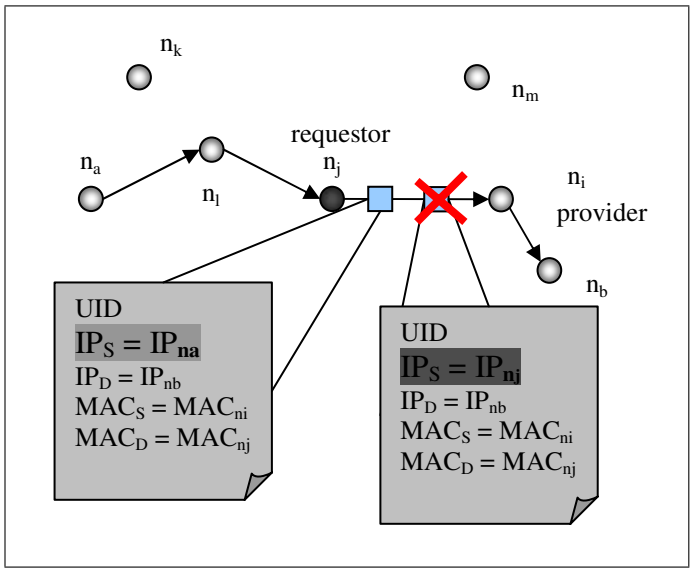


Figure 4.9: Punishment example: provider n_i denies relaying packets originated by node n_j

effect only on *"selfish sources"*. Indeed, selfish nodes can still **receive** packets as destinations.

The punishment mechanism *only affects requests made by neighboring selfish nodes.* The fundamental implication is that *selfish nodes are sill used as providers* by legitimate nodes. Depending on the selfishness model, a selfish behavior can have a limited duration in time: because of the "rationality" of the end-user operating the node, a node previously tagged as misbehaving might change into a legitimate node. The CORE mechanism inherently implements **automatic reintegration** of selfish nodes. Also misbehaving nodes are used in the route discovery phase of the routing protocol and are requested to forward data packets (if they are on the path from the source of the data traffic to the corresponding destination). If the selfish behavior is reverted to a cooperative behavior, nodes have the possibility to rise their reputation value above the punishment threshold P_{th}, thus gaining the rights to be served again.
In the following section, we explore alternative punishment mechanisms that could be implemented in CORE, and discuss their advantages as well as their drawbacks.

4.7.1 Alternative punishments

Communication services can only be partially withhold by the punishment mechanism described in section 4.7: selfish nodes can still receive packets. A complementary punishment mechanism that can be used by legitimate providers together with the one presented

in section 4.7 consists in dropping packets that normally would be forwarded to the selfish destination. By doing so, the selfish node would experience an important drop in her throughput performance. However, punishment of *selfish destinations* requires an additional message sent by legitimate providers to explicitly inform neighboring nodes of the punishment action. Indeed, in case we consider the packet forwarding function, the legitimate node n_k on the path $< ..., n_k, n_i, n_j >$ would erroneously detect a selfish behavior of the legitimate node n_i that denies relaying a packet to the selfish destination n_j. An explicit denial of service message broadcasted by node n_i, or an indirect reputation message advertising the detection by node n_i of node n_j's misbehavior are needed. However, both alternatives are affected by the problems discussed in section 4.6.2.
An alternative punishment mechanism initially proposed in [62] and used also by the CONFIDANT mechanism affects the path selection procedure implemented by legitimate nodes. Routes are ranked by a "path rater" component that selects only paths with no selfish nodes on it. The immediate consequence is that selfish nodes are not used anymore: they are excluded from the network. This approach is suitable when considering the **simple selfishness model** that defines a constant node misbehavior for the whole duration of the network lifetime. By assuming that selfish nodes **never** change behavior, the cleverest thing to do is to completely and definitively exclude them from the network. Network performance are proven to moderately increase (refer to [24]).
However, in the realistic scenario of a "rational" selfish behavior in which node selfishness can suddenly turn into a cooperative behavior, the circumnavigation of selfish nodes could have the opposite effect to the one described when a simple selfishness model is assumed. Network performance could degrade due to the intensive utilization of the remaining legitimate nodes. The overall network lifetime is reduced and bottlenecks are introduce in the network. To cope with this unwanted side-effect, an additional **reintegration mechanism** is required. The reintegration mechanism could for example make a periodic check on the neighborhood behavior to detect changes in the behavior of previously tagged selfish nodes. It is out of the scope of this thesis to define a viable reintegration mechanism: also the authors of CONFIDANT do not address this issue.

4.8 CORE application example

This section presents an application example of CORE. In the following scenarios we assume that a **secure** routing protocol is used to find routes from a source to the corresponding destination and that every node participate to routing operations. On the other hand, we assume that node selfishness affects only the packet forwarding function: we thus show examples that illustrate how CORE detects a selfish forwarding behavior and how reputation is updated. Furthermore, we assume that reputation information is not distributed: only direct reputation is used to discern a legitimate from an uncooperative behavior.
The example scenarios presented in the following have been selected to point out situations that might be critical for the correct operation of the basic version of CORE without

reputation distribution. However, due to the basic definition of node selfishness, *i.e.* a node that permanently do not participate to the forwarding function, we cannot show that CORE provides incentives to collaborate: only node punishment can be represented on the following figures, that is, it is possible to show a legitimate node denying service provision to a selfish neighbor.

- Scenario 1: no selfish nodes, dense traffic

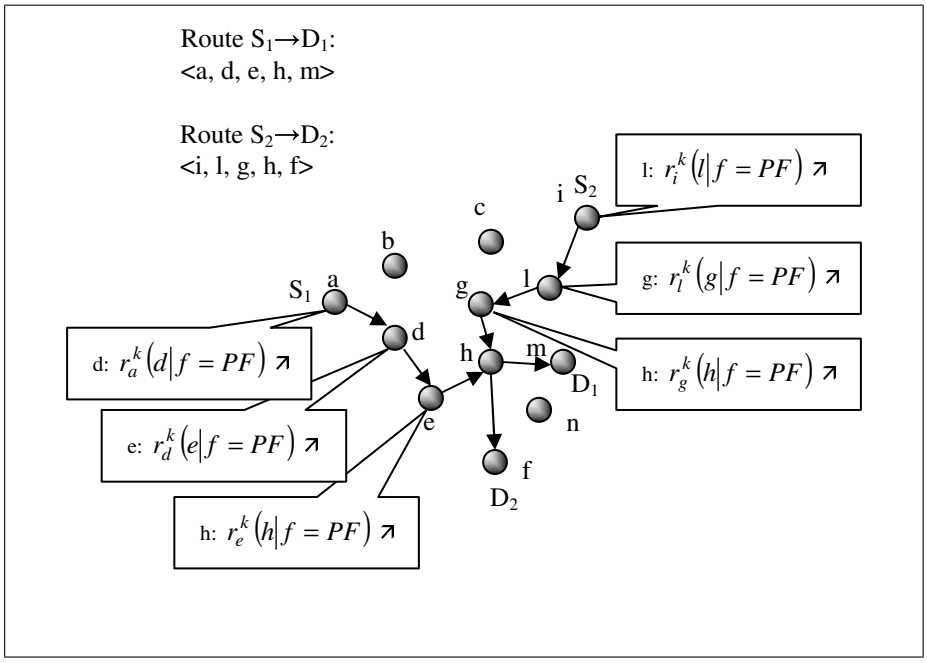


Figure 4.10: Example scenario 1: all nodes cooperate, reputation is updated consequently.

The first scenario shows how reputation is updated when all nodes behave correctly. Figure 4.10 represents two sources (S_1 and S_2) that have discovered a route to the corresponding destinations (D_1 and D_2) through the route discovery phase of the DSR routing protocol: routes are indicated in the figure.
Every node on the path from the source to the destination, except the last hop before the destination, updates the direct reputation value stored in the local Reputation Table: the monitoring mechanism reports a positive observation that is inserted in the observation buffer. Positive observations are collected for every packet that is sent through the corresponding route and boost the reputation value assigned to the next hop on the route.
The more dense the traffic pattern (*i.e.* the number of source,destination pair) the more relevant reputation information is collected by the nodes. If we suppose that every node has to send (as a source or as a relay) packets to every other node of the network and we take into account node mobility, then CORE assures that nodes hold enough observations on the behavior of a large subset of nodes to build a complete Reputation Table.
We argue however that keeping information on the reputation of every node is not necessary; indeed, node "rationality" would produce a variable behavior that might not be

detected when selfish nodes move around in the network.
Moreover, reputation information is relevant only for neighboring nodes. Even in the advanced case in which reputation information is diffused in the network, only indirect reputation on **neighboring nodes** can be used. Indeed, the punishment mechanism can **only punish neighbors** in a reliable way. The third scenario presented in this section is better suited to explain why.

- Scenario 2: one selfish node, dense traffic, no mobility

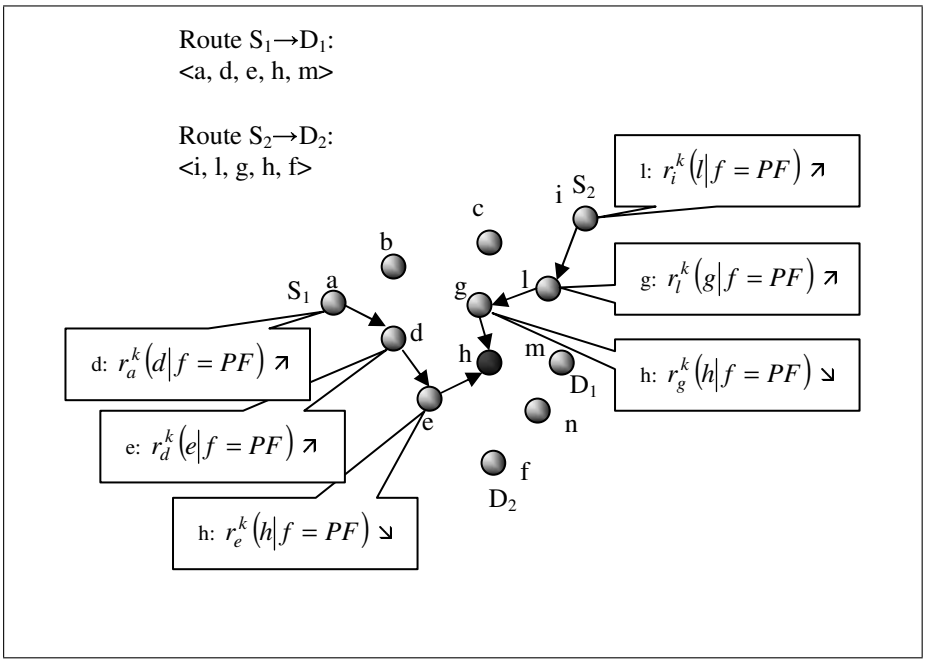


Figure 4.11: Example scenario 2: node h is selfish, reputation is updated consequently.

In figure 4.11, node h is selfish: all data packets that need to be relayed are dropped. Considering the same source,destination pair as in figure 4.10, node e and node g detect node h's misbehavior by monitoring her activities with respect to the packet forwarding function and decrease consequently the reputation value assigned to the selfish neighbor. If node h's reputation decreases below the punishment threshold P_{th}, node e and node g have enough evidence to punish her selfish behavior. For example, let us suppose that node h has the following routes to destination node b: $< h, g, b >$, or alternatively $< h, e, d, b >$. Node h's data packets will be dropped by node g if the first route is used, and by node e if the second route is used.

- Scenario 3: one selfish node (also selective), low traffic, no mobility

Figure 4.12 represents a critical scenario in which traffic density is low. As in the first two scenarios, node a has a route to destination node m. On the route, selfish node h fails

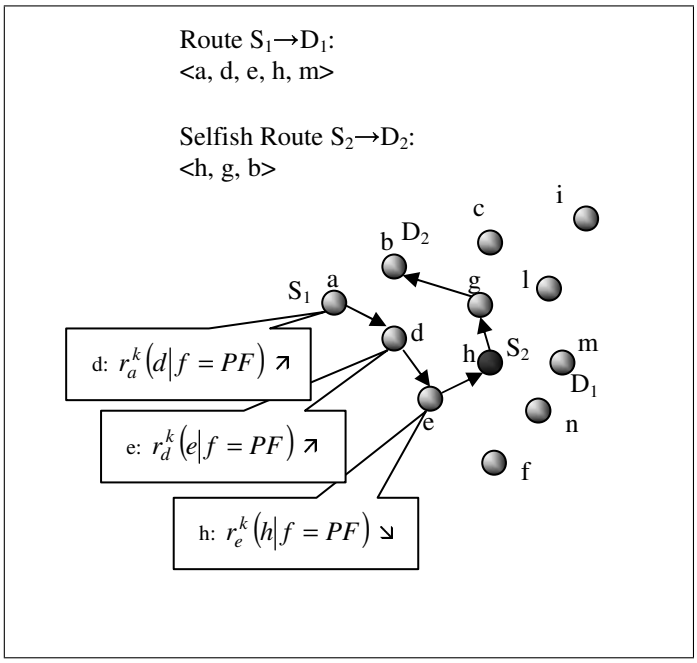


Figure 4.12: Example scenario 3: node h is selfish, reputation information not available for node g.

to forward data packet to the destination: the misbehavior is detected by node e, that eventually drops node h's reputation below the punishment threshold.
However, the other neighboring nodes do not have enough evidence to update the reputation value assigned to node h and cannot be entitled to punish her selfish behavior. Thus, data packets will be forwarded by node g on the route $< h, g, b >$ because of a lack of reputation information.
This example emphasizes the limitations of CORE in the absence of node mobility and when low traffic density is perceived in the network.
To counter lack of reputation information due to scarce node mobility and traffic patterns, indirect reputation ratings could be used. However, reputation information about node h's behavior evaluated by node e could reach node g only through node d, and the "second hand" information would have a relatively reduced effect on node g's perception.
Moreover, indirect reputation information can be used only to punish neighboring nodes. Indeed, suppose now that in figure 4.12 the selfish node h has the following route: $< h, g, d, a >$.
If we assume that node e provided indirect reputation information on node h's misbehavior to node d, and that node h's reputation falls below the punishment threshold, then node d would be entitled to punish node h. If, on the other hand, node g has no information on node h's misbehavior then node d would be erroneously detected by node g as misbehaving.
In conclusion, serious inconsistencies in reputation information diffusion could arise, making the punishment mechanism less effective.
A practical solution that avoids the distribution of reputation information and solves

the problems pointed out by this last scenario is based on an enhanced version of the monitoring mechanism, presented in section 4.9.2.

4.9 Discussion

This section concludes the Chapter and provides a discussion on the validation of the CORE mechanism. Furthermore, we outline an enhanced version of the monitoring mechanism that provides advanced functionalities to a CORE-enabled MANET in which reputation ratings cannot be diffused or when a secure routing infrastructure is not available.

Properties of the scheme

We summarize in this section the properties of CORE with respect to the security objectives exposed in section 4.2.
In the reminder of this section we assume that nodes are operated by "rational" selfish end-users and that every node implements the basic version of CORE, that is, no reputation information is distributed.

- **Distributed approach**. CORE is implemented in each node of the network in a distributed fashion, but do not suffer from the coherence problems relative to typical distributed approaches in which reputation information consistency is an issue.
- **Provide incentives to cooperate both in packet forwarding and routing**. CORE monitors multiple functions: detection of selfish behavior is function specific. Reputation ratings are function dependent and can be combined through a modifiable (for example by the users that operate the nodes) weighting system.
- **Punish misbehaving nodes**. All locally detected selfish nodes can be punished by selecting a punishment threshold P_{th} that determines the reputation needed in order to be entitled to be served. Furthermore, different reputation "profiles" can be selected (for example by end-users operating the nodes) in order to adapt to the surrounding environment: if the neighborhood is mainly composed by trusted nodes, the reputation profile can be more gradual than the opposite situation in which most of the neighboring nodes are misbehaving.
- **Lightweight approach**. CORE is a lightweight mechanism in the sense that energy consumption is limited to the promiscuous mode operation. By modifying the monitoring sampling frequency WD_{freq} energy consumption can be further reduced in case of a legitimate neighborhood. Furthermore CORE does not introduce any traffic overhead since the detection mechanism is passive, and because reputation ratings are not distributed to other nodes but used only locally. Selfish node reintegration is automatic.

- **Energetic efficiency**. Due to its lightweight characteristics, CORE is efficient from the power consumption standpoint. No extra energy is needed for signalling messages. Power consumption due to the promiscuous mode operation can be modulated through the watchdog sampling frequency.

- **Robustness**. CORE cannot be used by malicious nodes to perform denial of service attacks to legitimate nodes. Isolation of cooperative nodes due to the distribution of false reputation ratings is prevented. CORE is robust against valid error messages generated by selfish nodes that have been isolated from the network. Punished nodes cannot use route error messages to invalidate valid routes that use legitimate nodes: the underlying ad hoc routing mechanism (DSR) uses layer-2 acknowledgements to initiate a route maintenance procedure so that only node mobility or failure can be the cause of a route error message generation.

- **Secure**. Identity spoofing is prevented if CORE is coupled with a distributed key management scheme as the one presented in Chapter 7.

4.9.1 CORE validation

Validation of reputation based mechanism rise serious problems depending on the selfishness model used to describe node misbehavior. Alternative cooperation enforcement schemes based on the simple selfishness model in which a node's misbehavior is permanent are easier to validate: for example, simulation-based analysis of network performance when the CONFIDANT mechanism [24] is used are valuable because selfish nodes are permanently detected and ignored by the routing mechanism. It is evident that packet delivery ratio increases while the important metric to measure is the detection rate of node selfishness.
A simulation-based analysis of CORE is more difficult. First of all, because it is difficult to decide which metric is relevant to show CORE performances. Since nodes are not excluded from routing, aggregate network performance will hardly rise. As opposed to other simulation-based studies available in the literature, in our case it is more valuable to show that network performance decreases for selfish nodes, because of the punishment mechanism. Another interesting analysis would be to provide evidence of energetic savings for legitimate nodes that punish misbehaving nodes by not forwarding their packets. However, the major issue in the validation of CORE is due to the node selfishness model we assumed. In a generic way, we assumed nodes to be **rationally selfish**, in the sense that nodes make cooperation decisions based on a self-interested vision of communication capabilities of the network while at the same time taking into account the presence of a cooperation enforcement mechanism that is used to promote cooperation. We solved this issue by introducing a formal definition of node selfishness based on concepts borrowed from game theory and micro-economic modelling. The following Chapter is entirely dedicated to the validation of CORE: we first provide a simulation-based analysis of CORE that has been implemented in the GloMoSim network simulation suite and then offer a detailed study of CORE when the MANET is modelled in game theoretical terms.

4.9.2 CORE enhancements

In this Chapter we have described the CORE cooperation enforcement mechanism and provided examples scenarios to illustrate the operations carried out by its components. A detailed discussion has been provided to cover various aspects of cooperation enforcement mechanisms based on reputation, such as the need for a secure routing service (to prevent traffic subversion), the optional requirement for reputation information distribution and alternatives to the watchdog techniques that rely on the promiscuous mode operation of 802.11 based wireless radio cards.
In this section we outline an enhancement to the CORE monitoring mechanism that improves node detection capabilities in order to cope with specific scenarios in which the presence of a secure routing service cannot be guaranteed. Furthermore, the complementary monitoring component described in the following section 4.9.2 can be used as an alternative tool to a local reputation distribution thus improving detection of neighboring selfish nodes when low traffic density is experienced in the network without the drawbacks of reputation distribution described in section 4.6.2.
The CORE detection capabilities are enriched by introducing a new role beside the requestor/provider role described in section 4.4: **peer nodes** are monitoring entities that are not involved in a requestor/provider transaction, but are situated within the wireless radio range of a $< requestor, provider >$ pair. Peer nodes use a modified watchdog component (based on promiscuous mode operation) that produces observations on the behavior of neighboring nodes: these observations are filtered to evaluate a reputation rating for the monitored entities.

Peer nodes

Peer nodes are needed when a secure routing component is not available and a particular type of traffic subversion is performed by malicious nodes that aim at gaining reputation while disrupting the normal network functioning. Furthermore, a node can enable the peer monitoring component when a low traffic density is detected. A lack of reputation information on neighboring nodes behavior due to infrequent requestor/provider transactions is experienced (and can be detected) when the cardinality of the neighbor set (i.e. the set of neighboring nodes detected through a neighbor discovery mechanism) is smaller than the number of entries in the reputation table. By enabling the peer monitoring component, legitimate nodes can infer information on neighbors behavior through direct observations.
The **peer validation component** is based on the watchdog technique, but the expectation table described in section 4.5 contains entries that have not been generated by the peer node. Instead, by promiscuously listening to the surrounding activities, a peer node create an entry in the expectation table on behalf of other nodes, as a monitoring third party: for example, in figure 4.13, node c and node d are using the peer validation functionality.
We take the example scenario represented in figure 4.13 as a reference to describe the

different options available to peer nodes, when node h is misbehaving.

- Peer validation to detect **selfish** behavior

In this example we consider the packet forwarding function executed by nodes that are on the route $< m, h, g, b >$ from the source S to the corresponding destination D. *Peer node* d falls within the radio range of *requestor node* h and *provider node* g. By promiscuous listening a relay request performed by node h, peer node d adds an entry to the expectation table as described in section 4.5 and waits until the expiration of the timeout WD_{TO}. By promiscuous listening the activities of node g, the peer node can produce an observation that is passed to the reputation manager component and used to evaluate a reputation rating for the neighboring (selfish) node h.
A selfish behavior can thus be detected by node d without being directly involved in the requestor/provider transaction.

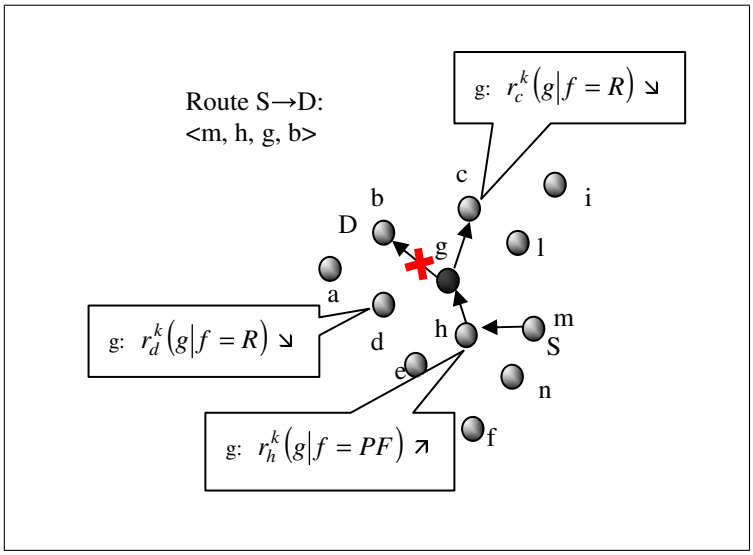


Figure 4.13: Peer nodes example: malicious node h performs a traffic subversion attack.

- Peer validation to detect traffic subversion **attack**

Suppose now that the malicious node g performs a traffic subversion attack. In this attack, the malicious node participates to the routing function and behaves correctly. Node g could rise her reputation by participating to the packet forwarding function, but sending the packet to node c instead: this attack can be done by modifying the MAC destination address of the relayed packet while not tampering with the source route contained in the packet. In this particular attack, node h would erroneously produce a positive observation

for the behavior of node g because she cannot detect the packet modification due to the lack of information on the distant destination node b. However, peer node d can easily detect the misbehavior: the destination MAC address is known and the modification can be discovered. As a consequence, node d decreases the reputation value that concerns the routing behavior of malicious node g.
Furthermore, also node c, by receiving a packet that was not intended for her, can detect (by promiscuous listening) the misbehavior and decrease node g's reputation.

- Peer validation to improve punishment mechanism

The peer validation mechanism can be used also when the punishment mechanism described in section 4.7 is improved by denying service to selfish destinations. Indeed, we mentioned that traffic directed to a neighboring selfish nodes could be *blocked* by explicit denial of service messages broadcasted by legitimate node. However, selfish nodes could explicitly broadcast false denial of service messages in order to legitimately save energy.
In the specific case of figure 4.13 in which peer nodes c and d already have an entry in their reputation table that designate node b as legitimate, explicit denial of service messages sent by node g can be validated, in the case that we assume that *selfish behavior cannot be selective*.
If a false explicit denial of service is overheard by promiscuous listening, peer nodes can detect the inconsistency and decrease the reputation rating related to the misbehaving node: this additional mechanism provides incentives to report only legitimate explicit denial of service messages.

Relevant publication

P. Michiardi and R. Molva. Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proceedings of IFIP Communications and Multimedia Security Conference (CMS'02)*, Portoroz, Slovenia, September 2002.

Chapter 5

Simulation-based validation of CORE

5.1 Introduction

In this Chapter we carry out a simulation-based study of the CORE mechanism, implemented as an add-on component for the Glomosim network simulation suite. The features of CORE are analyzed in terms of simulation metrics that we deem relevant to assess the basic properties of a cooperation enforcement mechanism: the energetic cost beard by CORE-enabled nodes and the efficiency of the detection and punishment mechanisms used in CORE. Similarly to the simulation approaches available in the literature (refer to Chapter 3) we run our experiments for various type of scenarios by taking into account both static and dynamic networks as well as different traffic patterns. In our simulation study, we use a simple model of node selfishness (refer to Chapter 2) whereby misbehaving entities are defined at the beginning of the simulation and node behavior is independent of simulation time. Simulation results are used to understand if and when a mechanism to distribute reputation information could be necessary in order to improve punishment efficiency: reputation distribution is an optional feature of the CORE mechanism and constitutes the main discriminant between CORE and other reputation-based cooperation enforcement mechanisms.
Even if interesting results can be obtained through an accurate simulation-based evaluation of cooperation schemes, we claim that the node selfishness model used in most work available in the literature is not sufficient to grasp the salient features of mechanisms intended to stimulate cooperation among self-interested entities. By assuming a static node misbehavior whereby nodes are defined as selfish for the whole network lifetime, it is arguable that the incentive properties of cooperation schemes can be properly shown. A selfishness model that does not take into account eventual variations in the behavior of the nodes is not appropriate for the validation of a mechanism that is intended to guide selfish nodes (or end-users operating the nodes) towards a more cooperative behavior.
Since a large fraction of existing cooperation enforcement schemes are based on principles akin to decision making and economic modelling, a natural tool that emerged to be suit-

able for the validation of such mechanisms is game theory. In the next Chapter we define two analytical models that describe the MANET environment and the nodes participating to the network operation in game theoretical terms. Our research has been presented in a preliminary work [67], and has been extended in [5–7, 70]. Using game theoretical models to validate a cooperation mechanism allows the definition of a dynamic node behavior that follows a "rational" strategy imposed by the end-user operating the node. A rational strategy represents the behavior of a self-interest user that tries to maximize her profits (in terms of energy consumption) while knowing that other users in the system could do the same or could use a cooperation strategy to enforce cooperation.
Game theoretical models of MANETs are however limited in that they do not realistically represent the underlying mechanisms (and their inherent limitations) that are used to operate the ad hoc network, such as medium access control and routing protocols. In the approaches presented in this chapter we overcome to some of the limitations dictated by a high level representation of nodes' interactions within the network by including in our models the salient features of the mechanisms that are analyzed: for example, in the first model presented in the sequel of this Chapter we take into account the issues related to the watchdog technique that have been exposed in Chapter 4.
By combining the results obtained through the simulation-based analysis and through the game-theoretical analysis of CORE we conclude that our scheme not only meets all the requirements that have been presented in Chapter 2 but also performs better than other cooperation strategies evaluated in the literature when a realistic network model is assumed.

5.2 MANET simulation with CORE-enabled nodes

In this section we present a simulation-based analysis of the CORE mechanism. We provide a description of CORE implementation choices, with an emphasis on the determination of the system parameters that have been presented in Chapter 4. We then describe the simulation set-up by specifying the scenarios that have been chosen to test the features of CORE and the metrics used to judge the efficiency of the detection and punishment of selfish nodes eventually present in the network. We further concentrate on the consequences in terms of energetic consumption when CORE is used by the nodes. A thoughtful discussion on what is it possible to show through our simulations and what cannot be studied because of the inherent limitations of selfishness model is provided.
The validation of the CORE cooperation mechanism is presented through the analysis of simulations results and by further examination of reputation distribution requirements as well as efficiency of the punishment mechanism.

5.2.1 CORE implementation

The CORE mechanism has been implemented as a plug-in mechanism for the GloMoSim network simulator: by referring to figure 4.1 in Chapter 4 that describes the CORE components, we implemented the detection mechanism that relies on the promiscuous mode operation of 802.11 based radio cards, the reputation management component in which only **local** reputation information is used, and the punishment mechanism that modifies the forwarding behavior of a legitimate node that detected a selfish neighbor.
The *detection mechanism* implemented in each node monitors the behavior of neighboring nodes with respect to the (data) **packet forwarding function** and relies on a FIFO buffer that can store up to B past observations. We recall here that by **observation** we mean the result of the comparison between an expected packet stored in the expectation table (refer to figure 4.6.4 in Chapter 4) and the observed packets overheard by the promiscuous listening of neighborhood operations. We give the value $\sigma_k = +1$ for the successful observation (made at time k) in which the expected packet equals the observed packet, and the value $\sigma_k = -1$ for an unsuccessful observation. The monitoring mechanism is based on the watchdog technique, for which we set a timeout value $WD_{TO} = 50$ms and a sampling frequency $WD_{freq} = 1$ [observation per packet]. The choice of the timeout value has been tailored to meet memory allocation requirements and detection capabilities: a too high value would result in a huge amount of temporary memory wasted to store expected packets that are accumulated in the expectation table at the rate of one expected packet per data packet generated. On the other side, a too low value for the timeout would negatively impact the detection capabilities in a heavily loaded network by triggering false negative observations since eventual collisions at the MAC level or full queuing buffers due to a high number of data packets in transit slow down the forwarding response time of neighboring nodes.
The *reputation management component* has been implemented taking into account only local observations provided by the monitoring mechanism, while reputation information is not distributed in the network. In Chapter 4 section 4.6.2 we argued that the distribution of reputation information is utterly insecure and through our simulation study we want to study in which situations the detection capabilities of CORE would be improved if indirect reputation ratings would be used. For the sake of simplicity, we implemented the simplest reputation evaluation filtering function, that is a **moving average low-pass filter** with a window size of B. A reputation value is calculated for every neighboring node with respect to the packet forwarding function, thus the functional reputation method presented in Chapter 4 is not used in our simulation implementation.
Finally, the *punishment mechanism* has been implemented by temporarily disabling the packet forwarding function for neighboring nodes that have a reputation value that falls below the punishment threshold $P_{th} = 0$, that is, a node n_j is punished at time k by node n_i because considered to be selfish if her reputation $r^k_{n_i}(n_j) < 0$. This implies that selfish nodes cannot send but can eventually receive data packets. In the results section we analyze the effectiveness of the punishment mechanism when constant bit rate (CBR) data flows are defined between pairs of communicating nodes: we show that CBR sessions based on the UDP transport mechanism are blocked for selfish nodes because data packets

originated by selfish sources are not forwarded, even in the case that a valid route has been provided to the selfish node. On the other hand, also HTTP Client-Server sessions based on the TCP transport protocol would be blocked using our punishment mechanism since a selfish node would not be able to send HTTP_Requests to the corresponding servers. Table 5.2.1 summarizes the CORE parameters choices we made in our implementation.

System parameter	*Value*	*Description*
B	5	Window size of the moving average filter, *i.e.* number of past observations used to evaluate *subjective reputation*
WD_{TO}	50ms	Watchdog timeout value, *i.e.* timeout used to validate an expectation
WD_{freq}	$1 \frac{\text{observation}}{\text{packet}}$	Watchdog sampling frequency, *i.e.* number of observations per data packet generated
σ_k	$\{+1, -1\}$	Values associated to a positive or negative observation
P_{th}	0	Punishment threshold, reputation values below the threshold denote selfish nodes
Reputation	Only Local	Reputation value evaluated through the reputation management component

Table 5.1: CORE system parameters.

5.2.2 Simulation set-up

In this section we describe the simulation parameters we used in our study and the different scenarios that have been considered for the evaluation of CORE.
In our simulations we consider an ad hoc network formed by $N = 16$ nodes that use the DSR routing protocol for a simulation time equals to 2000s[1].
Node placement have been manually selected to follow a grid pattern as depicted in figure 5.1 and the distance between two nodes on the grid have been set in order to allow only communications that are on the rows or columns of the gird, whereas diagonal links cannot be established due to the wireless radio range limits. Nodes are free to move (if specified in the scenario) in a 1000m x 1000m area: the mobility model we chose is the *Random Waypoint Model*, in which nodes move to a random destination at a speed uniformly distributed between 1m/s and a maximum of 20m/s. Once they reach this destination, they stay there for as long as specified in the *pause time*, which we will use

[1]The size of the network has been chosen to simplify the discussion of the results, however we also studied the properties of CORE in a larger network obtaining similar results as the one depicted in the plots in section 5.2.4

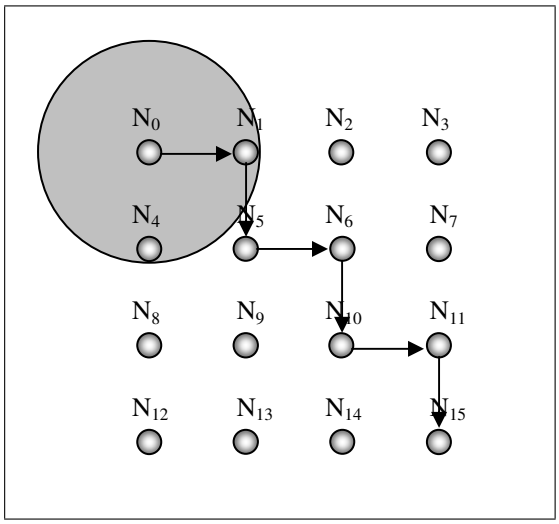


Figure 5.1: 4x4 grid network used in our simulations, with radio range and route example from source node 0 to destination node 15

as a simulation parameter as defined later in this section.
The radio propagation model used in our scenarios is the realistic *two-ray ground reflection model* and we take into account physical phenomena such as signal strength, propagation delay and interference. The radio range has been set to the usual value of 250m.
We have defined two families of simulation scenarios: in the first set of simulations we consider a **static network** while in the second family we use the Random Waypoint mobility model to simulate a **dynamic network** in which we vary the pause time parameter.
To infer the salient characteristics of the CORE mechanism, we completed our simulation scenarios with a further parameter that we call **path diversity**, which is defined for node n_i as:

$$p^t_{D_i} = \frac{l_i}{N_{neighbors_i}}$$

where l_i is the number of incoming (or outgoing) routes to node n_i and $N_{neighbors_i}$ is the number of node n_i's neighboring nodes at simulation time t.
Now, the DSR routing protocol is not a multi-path routing protocol since it does not use multiple paths to reach a destination: however it stores multiple route replies that correspond to different routes from a source to a destination in order to improve responsiveness in the case of link outage. Thus, in our simulations we *"manually"* introduce the path diversity parameter by defining the following traffic patterns that use constant bit rate (CBR) communications based on UDP, and in which 1000 packets of size equals to 64 bytes are sent at a rate of 1 packet per second:

- *High path diversity traffic pattern:* sources and destinations of the CBR traffic are chosen to produce a **fully connected** network in which every possible source has to send data to every possible destination (*excluding all one-hop communications*) in the network, while the beginning time of CBR sessions is uniformly distributed

between 0s and 800s.

- *Low path diversity traffic pattern:* sources and destinations of the CBR traffic are chosen randomly (*excluding all one-hop communications*), as well as the beginning time and the number of distinct CBR sessions.

As we will see in the results section, the path diversity parameter has a important influence on the performance of CORE: however, the "manual" configuration of the path diversity parameter revealed to be difficult to control in the dynamic network scenario. Indeed, when node mobility is high, it is hard to predict if the properties offered by the communication patterns defined at the beginning of the simulation will hold through time. However, path diversity can still be used in the dynamic case to explain some of the phenomena that appears in the result plots.
Furthermore, excluding one-hop communications is important in order to avoid the particular case in which the punishment mechanism implemented in CORE cannot be used. Again, when node mobility is introduced in the network, it is difficult to predict if a route counting more than one hop will hold through time: the metric used to measure the performance of CORE and described in section 5.2.3 excludes one-hop communications eventually formed through the simulation run.
We additionally defined a simulation parameter that takes into account the dimension of the FIFO buffer used to store up to B past observations on neighborhood behavior: this parameter has been used in one specific set of simulations where we study the eventual errors in the punishment mechanism, that we call **false positives**.
Taking as a reference the selfishness model described in section 5.2.2, in the ideal scenario in which nodes cannot temporarily fail, that there are no obstacles and that the monitoring mechanism based on the promiscuous listening produces no errors, one observation would be sufficient to detect and exclude eventual selfish nodes. However, in the more realistic scenario in which temporary misbehavior is due to the presence of detection errors, then a more sophisticated technique such as the one proposed in CORE has to be used. Throughout our simulation study we analyze the impact of the buffer size B on the punishment errors due to failures in the detection mechanism.
For every scenario that is possible to define using the parameters described in this section, we run 20 experiments and took the average of the metrics used in the simulation and described in section 5.2.3. Further, we enriched the simulation scenarios by considering three additional experiment configurations: we consider and ad hoc network in which there are no selfish nodes, and ad hoc network that has a pre-defined percentage of selfish nodes and a CORE-enabled network that is populated with selfish nodes. We took as a reference a network with a low percentage of selfish nodes (6%) and a network in which a rather high percentage (25%) of nodes are selfish.

Selfishness model

The selfishness model used in our simulation study follows the definition that has been given in Chapter 2.
A selfish node is defined as a node that participate to routing operations but that systematically fails in forwarding data packets, by disabling the packet forwarding function at the network layer. As a practical example, a misbehaving end-user that operates a node can easily disable the packet forwarding function by using the `IPTABLES` command in a Linux powered ad hoc node.
In our simulations, during the simulation setup, we define the total number of selfish nodes: selfish nodes drop data packets for the whole duration of the simulation.
In section 5.2.5 we argue that this simple model is not adequate to show how the CORE mechanism is able to promote cooperation of nodes: the necessity for a more complex model that takes into account node "rationality" is discussed, while in section 6 we will explain our proposed solution to overcome the limitations imposed by a simplistic selfishness model.

5.2.3 Simulation metrics

This section describes the metrics we used to evaluate the CORE mechanism: the results presented in section 5.2.4 refer to the metrics defined hereafter.

- Energetic consumption

The GloMoSim network simulator has a built-in energetic model that we use to evaluate the energetic consumption that nodes operating the ad hoc network have to bear. By default, in GloMoSim a node in an IDLE status consume 500mW per hour. We used this value as a basis to infer the further consumption that derives from the network operation and from the application running on the nodes: power consumption statistics are collected through the whole simulation run, whereby a wireless channel-dependent energetic cost is associated to every transmitted or received packet.
In the following graphs, we present the energetic consumption for three set of experiments: an ad hoc network with no selfish nodes, a defenseless ad hoc network with selfish nodes, and a CORE-enabled network with selfish nodes.
Further, we show the **average gain** of a CORE-enabled network with respect to a defenseless network with selfish nodes. The average gain is defined as follows: we take the aggregate average energetic consumption for all the *legitimate nodes* in the defenseless network and compare it to the aggregate average energetic consumption of the *legitimate nodes* in the CORE-enabled network and express the difference in percentage: as an example, an average gain of 10% indicates that, in average, the nodes of a CORE-enabled network saves up to 10% of energy with respect to a defenseless network.

We estimate energetic consumption to be a relevant metric since it represents the additional energetic cost beard by nodes that use a cooperation enforcement mechanism. This additional cost have to be limited since it could be a further source of node selfishness.

- Punishment efficiency

The punishment efficiency metric is defined as follows:

$$p_E = \frac{\sum\limits_{N'} d_p}{\sum\limits_{S'} s - h} \cdot 100\% \tag{5.1}$$

where N' is the subset of legitimate nodes in the network and S' is the subset of selfish nodes in the network.
d_p is the number of data packets originated by selfish nodes and discarded by the punishment mechanism implemented in *all* legitimate nodes.
s is the number of packets originated by each selfish node in the set S' that have a valid route to the intended destination and h represents the number of packets eventually originated by selfish nodes that are on a one-hop route.
For example, suppose that $N' = 15$, $S' = 1$, $d_p = 800$, $s = 1000$ and $h = 100$: $p_E = \frac{800}{1000-100} = 88,89\%$.
The punishment efficiency metric provides an overall metric to judge the effectiveness of the CORE mechanism that takes into account both the detection and the punishment mechanism: indeed, a legitimate node punishes a selfish node when a predefined number of observations have been collected through the monitoring mechanism and processed by the reputation manager component.

- False positives

The false positive metric is similar in its definition to the punishment efficiency:

$$\text{false positives} = \frac{\sum\limits_{N'} d_p}{\sum\limits_{N'} s - h} \cdot 100\% \tag{5.2}$$

With this metric we analyze the percentage of packets that have been erroneously dropped by legitimate nodes and that were originated by other legitimate nodes. Again, we do not take into account one-hop communication patterns.

NOTE: in our simulation-based analysis of CORE we do not show packet delivery ratio variations for legitimate nodes. As opposed to the other cooperation enforcement mechanism available in the literature, CORE does not punish selfish nodes by using the *path rater technique* described in Chapter 2 (see for example the CONFIDANT mechanism). The consequence is that we cannot show through our simulations that the packet

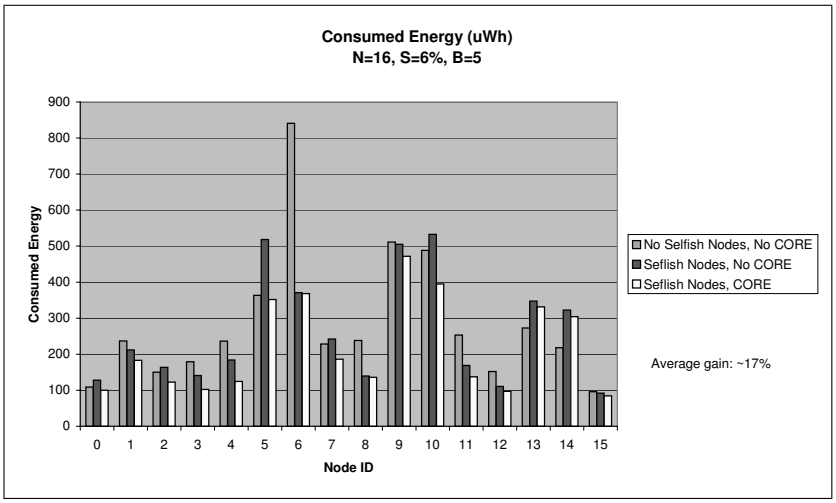
delivery ratio or alternatively the aggregate throughput of the network **increases** when the CORE mechanism is used by the nodes. We already discussed the drawbacks of the path rater technique in Chapter 4: furthermore we do not believe that a performance metric based on throughput is significative to show the salient aspects of a cooperation enforcement mechanism.
Through our simulation study we are able to show the impact of CORE in terms of energetic consumption (which we recall, is the main source of a selfish behavior) and in terms of punishment efficiency, but we claim that the ability of CORE (and other cooperation enforcement mechanisms) to promote cooperation requires the definition of a different selfishness model that allows a selfish node to act *rationally*, i.e. minimizing energy consumption while knowing that other nodes will undertake the appropriate countermeasures whenever a selfish behavior is detected.

5.2.4 Simulation results

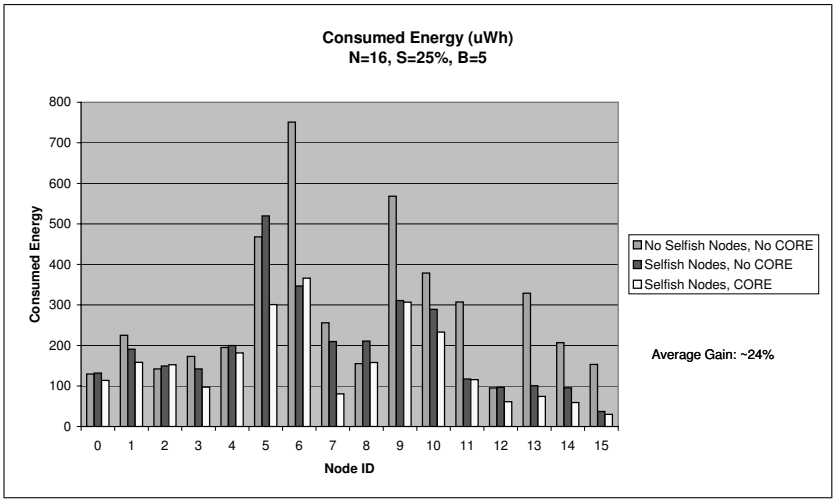
In this section we discuss the results of our simulation study: the plots presented hereafter are organized in two sections, one that shows the nodes' energetic consumption, and one that shows detection and punishment capabilities of a CORE enabled network.
Throughout our simulation study, we were able to assess not only that CORE introduce a low power consumption overhead (also with respect to other cooperation enforcement mechanisms available in the literature), but also that it entails a significant power saving for legitimate nodes that use CORE. Thus, CORE stimulates cooperation of selfish nodes to the network operation **and** introduce an incentive for legitimate nodes to use CORE as a cooperation enforcement mechanism since they save energy.

- Energetic consumption

The following plots (figures 5.2-5.4) present the average energetic consumption (per node) provided by the GloMoSim simulation statistics for different scenarios. We considered both static and dynamic networks and we varied the path diversity parameter (as described in section 5.2.2) and the percentage of selfish nodes. Note that in the following plots, when the percentage of selfish nodes is 6%, only one node misbehave: precisely, in our plots the node with $ID = 6$ is misbehaving. On the other hand, when the percentage of selfish nodes is 25%, there are 4 selfish nodes in the network: the following plots depict a network in which nodes with $ID = 2, 6, 9, 11$ are misbehaving. The first observation that we can make by an overall analysis of the plots derives from a comparison between the power consumption of an ad hoc network without selfish nodes and a defenseless network with a defined percentage of selfish nodes. If it is not surprising to see that selfish nodes save energy by dropping data packets, it is interesting to observe that also legitimate nodes' power consumption is impacted by the presence of selfish nodes: in average, also legitimate nodes might gain in energetic savings when selfish nodes are present in the network.



(a)



(b)

Figure 5.2: Static network, S% of selfish nodes: routes with more than 1 hops and high path diversity.

Indeed, consider for example the following $n+2$-hop path from a legitimate source S to the corresponding destination D:

$$< S, N_1, N_2, N_3, N_4, ..., N_n, D >$$

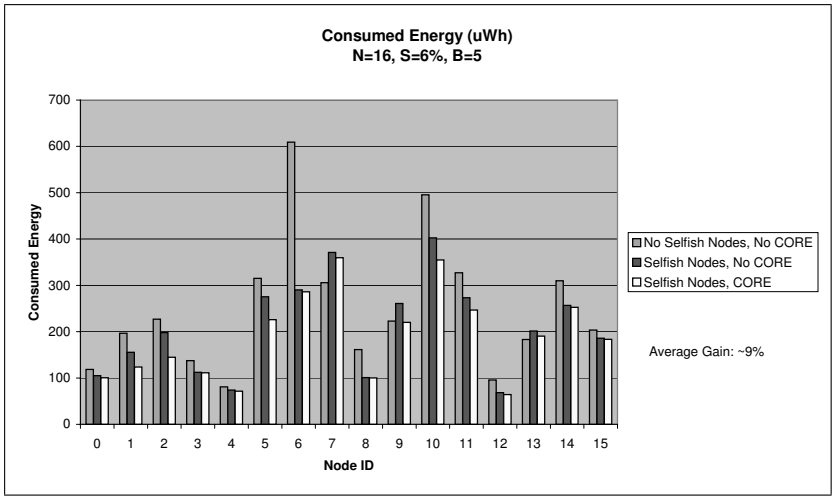
If we consider the extreme scenario in which N_1 is dropping data packets because of a *selfish* behavior, also all the other down-link nodes $\{N_2...N_n\}$ towards the destination will save energy because they will not receive any data packets to forward.
On the other hand, in the opposite scenario in which the last hop before the destination (namely, N_n in our path example) is *selfish*, all the up-link nodes towards the source will waste energy by forwarding data packets that will never reach the destination.
The use of a multi-path routing scheme with a high *path-diversity* would mitigate the waste in terms of energy of legitimate nodes that detected the presence of a selfish node in the route; at the same time, a multi-path routing scheme allows nodes to "probe" multiple routes and eventually increase detection capabilities.

On the other side, when comparing energetic consumption of nodes in a defenseless network as opposed to nodes in a CORE-enabled network, it is possible to see that nodes in a CORE-enabled network save energy by punishing selfish sources that originate data traffic. Consider the following path, in which the source node S has been detected as selfish by (all) her neighbors:

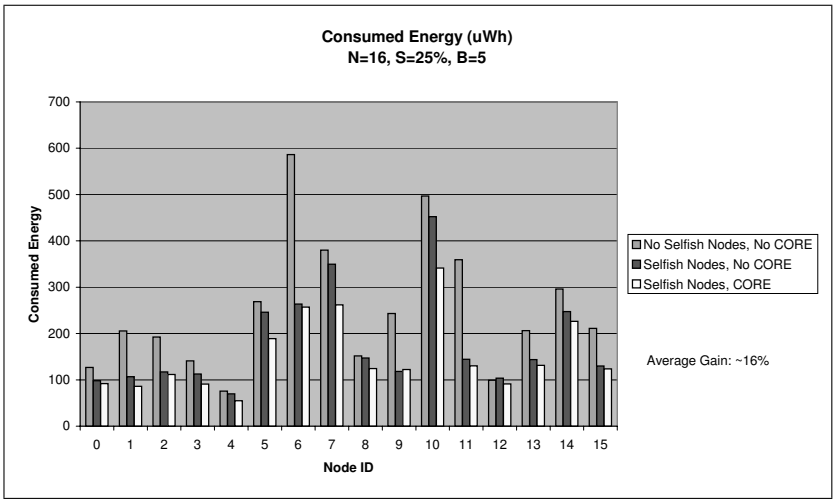
$$< S, N_1, N_2, N_3, N_4, ..., N_n, D >$$

If the network has no countermeasures to cope with node selfishness, every node on the path towards the destination wastes energy by forwarding "illegitimate" traffic generated by a selfish source that does not cooperate to the network operation.
On the contrary, in a CORE-enabled network if node N_1 detected a selfish behavior of the source, she will then punish the selfish source by denying data forwarding. Moreover, all nodes down-link towards the destination will benefit from the punishment of the selfish source performed by N_1 and will save energy.

The plots presented in figures 5.2-5.4 also show the *average gain* in terms of energetic consumption of a CORE-enabled network with respect to a defenseless network, without taking into account the gain of selfish nodes: as it is possible to see in the figures, average gains vary depending on mobility settings as well as percentage of selfish nodes in the network and path diversity paramter. In figure 5.4 we set the *pause time* parameter to 0 seconds in order to tackle with an extreme mobility scenario: nodes constantly move with a speed uniformly distributed between 1m/s and 20m/s.
When comparing plots with the same amount of selfish nodes in the network (figures 5.2(a) and 5.4(a)) but different mobility settings, it is possible to see that average gains in static networks ($\sim 17\%$) are higher than in the dynamic case ($\sim 4-5\%$): indeed, node mobility not only introduces a high amount of traffic overhead due to frequent link breaks but, as it will be shown in the punishment efficiency plots, it has an impact on the detection and punishment mechanisms.
Furthermore, it is possible to deduce from the plots that the average energetic gain is

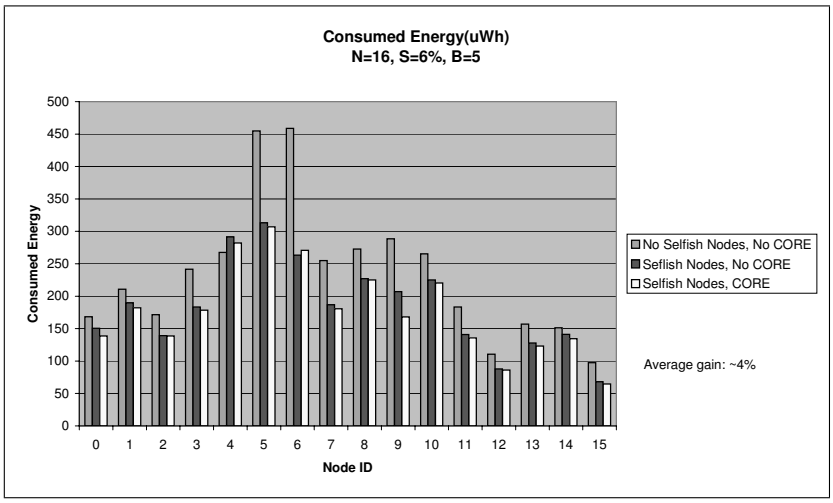


(a)

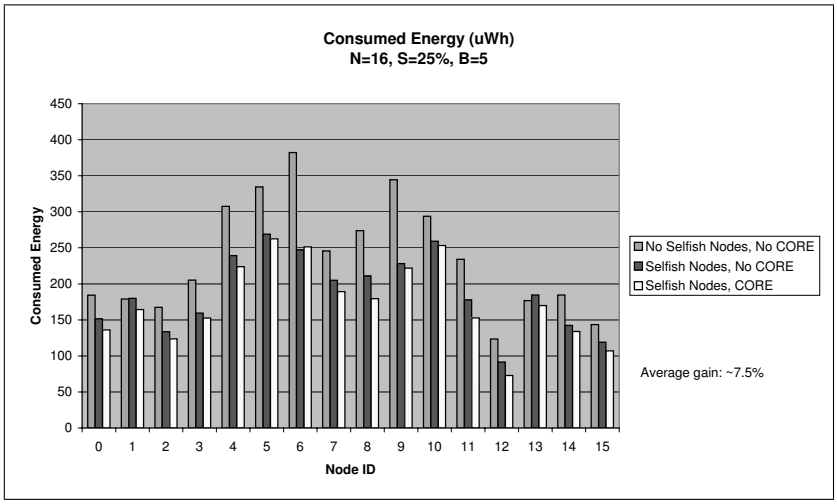


(b)

Figure 5.3: Static network, S% of selfish nodes: routes with more than 1 hop and low path diversity.



(a)



(b)

Figure 5.4: Dynamic network, S% of selfish nodes [pause time = 0].

higher when a larger portion of nodes of the network is set to be selfish: this result follows the discussion on the position (and number) of selfish nodes in a path from a source to a destination and the number of selfish sources in the network, as described in the beginning of this section.
As it will be more clear in the section dedicated to punishment efficiency plots, also the *path diversity* parameter has an influence on the average energetic gain (refer to figures 5.2 and 5.3): detection and punishment capabilities are impacted by the presence of multiple path towards and from a selfish node. A reduced path diversity, that we simulated through a small number of CBR connections in the network, degrade the detection and punishment capabilities of CORE.

- Punishment efficiency

In this section we present the plots for the punishment efficiency (p_E) metric versus the pause time parameter: when we consider a static network, the pause time has no influence on the results, whereas for the dynamic case the impact of node mobility is more important. For every pause time value (i.e. pause time = $\{0, 10, 100, 300, 600, 900\}$ seconds) we run 20 experiments varying the selfish node percentage and the selfish node position (i.e. the selfish node ID) in the network. In the static case, we also varied the path diversity parameter.
Figure 5.5 presents a summary of the punishment efficiency for all simulation scenarios we used, while figures 5.8(a), 5.8(b), 5.7 are enriched with measurement errors for every specific scenario we considered. In figure 5.5 it is possible to observe that, in average,

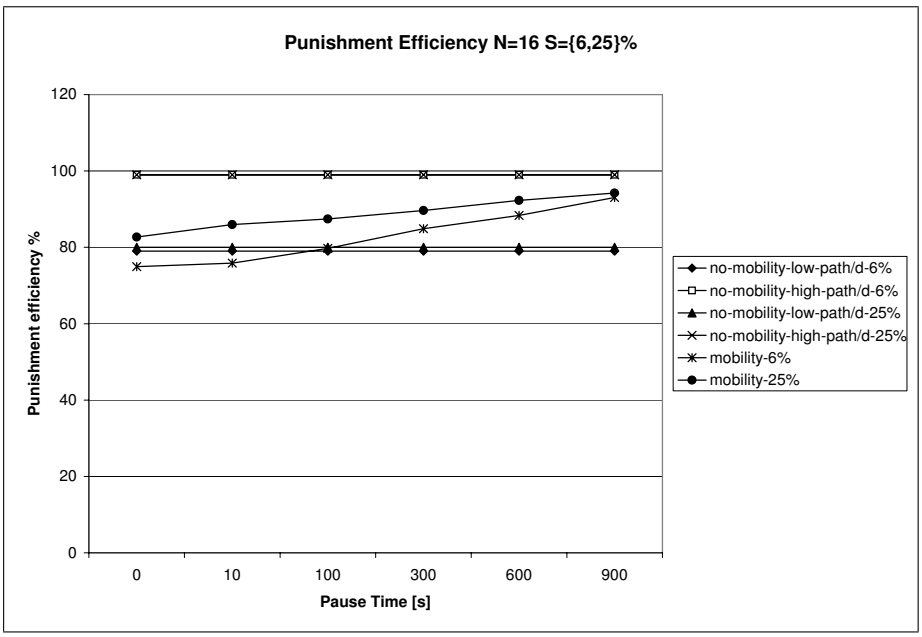


Figure 5.5: Summary of p_E for different simulation scenarios.

punishment efficiency is higher for a static network as compared to a dynamic network. Furthermore, our plots show that the path diversity parameter has a significant influence on the punishment efficiency (and detection capabilities) of a CORE-enabled network. Lastly, as for the energy consumption plots, a higher percentage of selfish nodes in the network produces a higher punishment efficiency.
To explain the behavior of the punishment efficiency plots, we refer now to the following specific scenarios.

* Figure 5.7 - **Static network**

When the nodes of the network do not move, the punishment efficiency depends on the path diversity parameter. Let's give a practical example to explain how path diversity impacts both detection and consequently punishment capabilities of CORE.

The following figure 5.6 depicts the gird network used in our simulation, where node N_6 is selfish. Detection capabilities as a function of path diversity are depicted in figures 5.6(a)

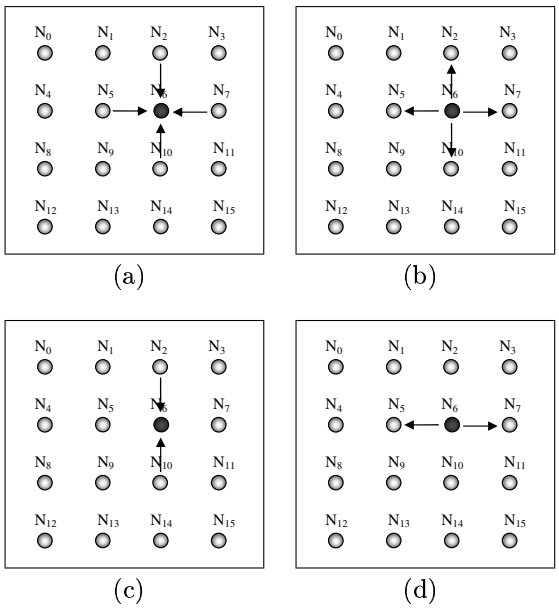


Figure 5.6: Static grid network with 1 selfish node: path diversity example.

and 5.6(c), while punishment capabilities are depicted in figures 5.6(b) and 5.6(d). When path diversity is high (figures 5.6(a),5.6(b)), all neighboring nodes of node N_6 have some routes that use the selfish node as a relay (both because they are sources or because they are relaying packets on behalf of distant nodes): in this case the detection mechanism assures that all neighboring nodes detect and classify node N_6 as selfish. In the same way, when path diversity is high, if node N_6 has routes that use at least one neighboring node, the punishment efficiency reaches almost 100%. Figure 5.7 shows an average value of 99% since our path diversity has been manually imposed through an appropriate traffic pattern setting, which however does not guarantee the same results that we would have obtained by using a true multi-path routing protocol.
On the other side, when path diversity is low (figures 5.6(c), 5.6(d)) only nodes N_2 and N_{10} uses routes that go through the selfish node N_6: only two nodes over four neighbors would detect the selfish behavior. Now, in the non negligible case in which node N_6 uses only routes that go through node N_3 and N_7, the punishment efficiency would be 0%.
This example shows the limitations of CORE: in the absence of a reputation distribution mechanism, legitimate nodes that never directly experienced transmission failures that can be attributed to a selfish behavior would not punish the selfish node. However, this limitation can be overcame in different ways: we already mentioned that a multi-path routing protocol that exploits path diversity would inherently solve the problem. Furthermore, in section 4.9.2 of Chapter 4, we proposed a CORE enhancement that rely on *peer nodes* which are able to observe selfish behavior without being involved in a forwarding transaction. Figure 5.7 shows the average punishment efficiency that we experienced in our simulations: in average, the punishment efficiency is below 100%; however we believe

it sufficient to represent a strong incentive for selfish nodes to participate to the network operation.

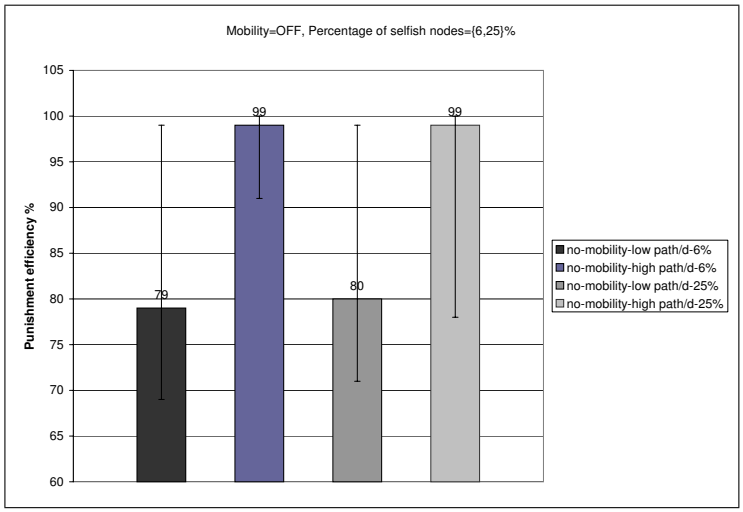
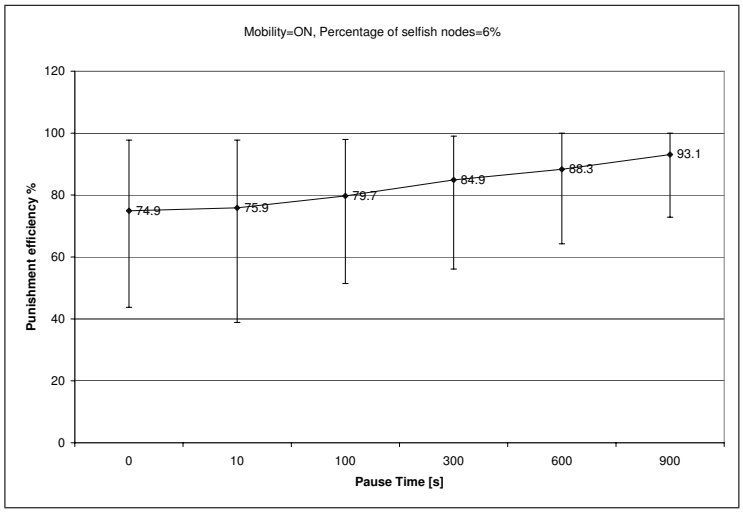


Figure 5.7: p_E for a static network, with measurement errors.

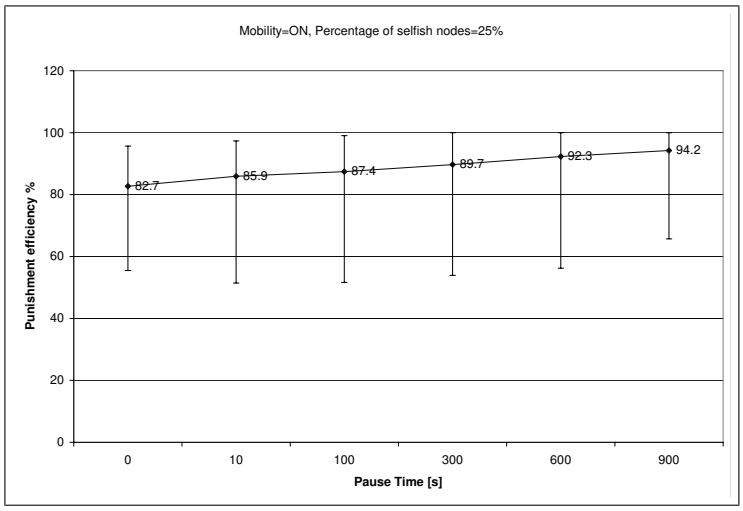
* Figures 5.8(a),5.8(b) - **Dynamic network**

By taking into account node mobility, parameters such as route hop count and path diversity are more difficult to control with respect to the static case. In the plots presented in figures 5.8 and 5.7, we vary the percentage of selfish nodes in the network and we analyze the variation of the punishment efficiency metric with respect to the pause time parameter of the Random Waypoint mobility model. The nodes' initial position is defined using the same grid network showed for the static case, which is then deformed due to node movements.
As it is possible to observe in figures 5.8(a) and 5.8(b), punishment efficiency grows in the pause time, and reaches high efficiency values (i.e. $\sim 100\%$) starting from a pause time of 900s. On the other side, when node mobility is very high, i.e. when nodes continuously move to random destinations, punishment efficiency is comparable to the one evaluated for the static network with low path diversity.
Since the nodes of the network rely on the DSR routing protocol which uses only one path that is reconstructed every time a link break is detected, path diversity is hard to estimate, even by imposing the adequate traffic patterns. We believe that path diversity registers negative variations due to node mobility: for example, neighbors that detected a selfish behavior could move away from the selfish node before she sends packets that would be dropped by the punishment mechanism. Other nodes that never used the selfish node as a relay (*i.e.* they could not detect it as being selfish), could fall within the wireless radio range of the selfish node and thus be used as relays for forwarding selfish traffic.
Even if the punishment efficiency offered by CORE in a dynamic network can be estimated

as an effective incentive for node cooperation, we envision in our future work to use a variation of the DSR protocol that exploits multiple paths and that do not use gratuitous route reply or route error messages, which we deem a source of a low path diversity.



(a)



(b)

Figure 5.8: p_E for a dynamic network, with measurement errors.

- False positives

The last plot we present shows punishment errors due to the shortcomings of the watchdog technique. In the plots, we varied the B parameter, which has always been set to $B = 5$ in all other simulation runs. We believe that the filtering function and the filter parameters used to evaluate reputation metrics for neighboring nodes is the main responsible for errors in the classification of selfish nodes. Assuming that the monitoring mechanism is imperfect, we want to study which is the impact of the parameter B on the false positive

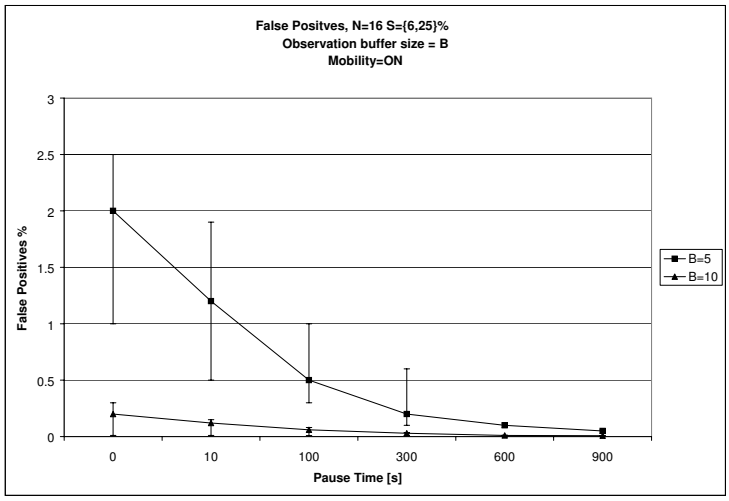


Figure 5.9: False positives with 25% of selfish nodes in a dynamic network.

metric.
As it is possible to see in figure 5.9, the false positive metric is inversely related to the pause time parameter, which we varied as in the punishment efficiency evaluation.
When the observation buffer size is $B = 5$, a non negligible percentage of false positives appears only in the case of a high mobility scenario: however, globally, for every legitimate node in the network, only 2 packets per 100 packets sent by legitimate nodes might be dropped erroneously. Varying the pause time parameter, or by doubling the buffer size $B = 10$, is sufficient to render the percentage of false positives almost negligible.

5.2.5 Discussion

The results section provides a detailed analysis of CORE performance under different networking scenarios: as we observed, the punishment efficiency is for most of the scenarios adequate to promote node cooperation since performance of selfish nodes is drastically reduced in a CORE-enabled network.
Furthermore we demonstrated throughout our simulation study that CORE provides not only **incentives for selfish nodes to cooperate**, but also that CORE is **attractive for legitimate nodes** in that they might save up to 24% of energy which, as a consequence, extends the network lifetime in the case that all nodes use CORE.
In some cases, we noticed that the distribution of reputation information could increase detection capabilities when path diversity is low or when we consider high node mobility scenarios. However we suggest that using a multiple-path routing protocol that exploits

high path diversity would inherently solve the problems pointed out in the results section. Furthermore, by using the extended version of CORE which implements the peer validation technique described in section 4.9.2 in Chapter 4, punishment efficiency can be improved also in the case of a single-path routing protocol.
Our simulation study has also been used as a basis for the design of a test-bed implementation of CORE and might be used for the fine-tuning of CORE parameters.

We believe, however, that the selfishness model implemented in our simulation study (and in other simulation study available in the literature) is not sufficient to grasp the salient aspects of node cooperation when "rational" end-users selfishly operate the nodes of the network. Indeed, a static selfishness model in which a misbehaving node never changes her behavior is not realistic: if changes in the selfish behavior are not allowed, then it is impossible to show in our simulations the capabilities of CORE to stimulate cooperation. In the following Chapter, we present an analytical model of an ad hoc network in which rational selfish agents operate the network maximizing their profits in terms of energy consumption while at the same time knowing that other agents in the network could do the same or could use a cooperation enforcement mechanism like CORE to enforce node cooperation.

Chapter 6

Game theoretical modeling and analysis of CORE

Cooperation enforcement mechanisms have been addressed by the research community in the attempt to cope with the selfish behavior of nodes in mobile ad hoc networks (MANET). As defined in Chapter 2, a node is considered selfish when it does not participate in the basic network operation in order to save energy. As opposed to maliciousness, selfishness is a passive threat that does not involve any intention to damage the operation of networking functions by active attacks like route subversion, tampering with data, etc....

In this Chapter we present two approaches to assess the features of our cooperation enforcement mechanism CORE, described in Chapter 4. Since a large fraction of existing cooperation enforcement schemes are based on principles akin to decision making and economic modelling, a natural tool that emerged to be suitable for the validation of such mechanisms is game theory.

In section 6.1 we present a method based on *non-cooperative game theory* that is used to evince the basic properties of CORE. In this method we use a model that describes the strategy of a self-interested node that has to make a decision about whether to cooperate or not with a randomly chosen neighbor. We then translate the CORE mechanism into a strategy profile that can be compared to other popular strategies. Under the commonly used hypothesis of perfect monitoring, we demonstrate the equivalence between CORE and a wide range of history-based strategies like tit-for-tat. Further, by adopting a more realistic assumption that takes into account **unreliable observations** of nodes' behavior due to communication errors, the non-cooperative model puts in evidence the superiority (in terms of stability and robustness) of CORE over other history-based schemes.

Alternatively, we also defined a method that we will include as an annex to the Thesis, where we extend our model in order to take into account network topology information in the strategy selection phase, i.e. when the node makes a decision about forwarding or dropping data traffic. We also derive an alternative punishment mechanism that provides a milder punishment for nodes that misbehave.

In section 6.2, we explore a different model that takes into account both a node-centric and a network-centric perception of the interactions between nodes that participate in a MANET by using *cooperative game theory*. We first demonstrate the requirement for a cooperation enforcement mechanism in order to promote cooperation between self-interested nodes by showing that in the absence of such a mechanism the best strategy for a node would be to free ride. Moreover, we analyze which would be the size of a coalition of cooperating nodes based on the importance given by a node to the node-centric and network-centric perspective of the game. We finally suggest how the CORE mechanism could be used to stimulate a node to join the coalition of cooperators. The benefit from using cooperative GT derives from the ability of this method to seize the dynamics of large group of players: the strategy chosen by a player does not only depend on a self-interested perception of the game but also takes into account a group-wide policy of the coalition the player belongs to. Although the "cooperative games" approach appears to be appropriate to model the dynamics of large coalitions of nodes forming a MANET, the main limitation of this method is that it is based on a high-level representation of the reputation mechanism that does not take into account the features of CORE. Although the methods described in the following sections focus on CORE as a specific mechanism, some general conclusions can be drawn from this analysis towards the design of cooperation enforcement mechanisms in general.

6.1 Non-cooperative game theory

In our first approach, we investigated on the characteristics of CORE by modelling the interactions between the nodes of a MANET as a non-cooperative game. In this section we describe a game theoretical model that focus on the features and limitations of the monitoring mechanism implemented in CORE, by taking into account the imperfect information provided through the watchdog technique. In the alternative model presented as an Annex to the Thesis, we take into account path and topology information in order to seize the impact of the presence of selfish nodes on a path from a source to a destination of data traffic.

In the following section, we will introduce a specific and well-known game (the prisoner's dilemma, PD) and explain how and why this model is suitable to describe the decision making process that a mobile node would undertake when participating to the ad hoc network operation. Subsequent to the definition of the model that describes the interaction between decision-makers (nodes) involved in the game play, we will extend our analysis to a particular instance of games that goes under the name of repeated games. Repeated (or iterated) games have been exhaustively treated in the game theoretic literature [39, 63, 81, 90, 92, 96] and interesting results concerning the establishment of a

cooperative behavior will be presented. In particular, we will focus on the strategy that a player[1] adopts to determine whether to cooperate or not at each of the moves in the iterated game and describe an important strategy known as tit-for-tat (TFT) which has been considered by a lot of game-theorist to be one of the best strategies not only to promote cooperation. Another important concept that we will consider in the sequel of this Chapter is the idea of evolutionary stable strategies, which we will also call *evolution of cooperation*. An **evolutionary stable strategy** (ESS) is a central concept in game theory: it is based on the idea of a population of organisms playing a certain strategy and that a mutant allele that causes organisms to adopt a different strategy cannot invade the population, but will instead be selected out by natural selection. An ESS depends on the idea of invasion, where a population of strategy-X players is visited by a strategy-Y player. The new player is said to invade if, following strategy Y, he scores better than the average strategy-X player. Assuming players are able to choose and switch strategies, this would induce the indigenous population to start switching to strategy Y. In many cases there are diminishing returns for the later adopters, and what follows is an equilibrium ratio of strategy-X players to strategy-Y players. A strategy X is evolutionary stable if there is no strategy Y that can invade it. That is, anybody bringing a new strategy into a population of strategy-X players will fare no better on average than the X players are already doing.
We will then describe how the CORE cooperation enforcement mechanism can be translated into a strategy for a player and compare it to the TFT strategy to numerically prove the equivalence between CORE and TFT. By further extending the game theoretic concept applied to the classical iterated PD game we will show how the performances of TFT and its derivates (i.e. generous-TFT, GTFT) degrade as noise is introduced in the model. In the following sections we will describe how the introduction of a noise factor allows grasping the undesirable effects of using the promiscuous mode operation of a wireless card as a basis for the monitoring mechanism (the watchdog mechanism) and prove that the CORE strategy outperforms all other known strategies both for promoting cooperation and for the evolution of cooperation. The numerical results obtained through a simulation software designed by [63] are stimulating the more difficult task of providing a formal analysis of the CORE strategy, which is part of our future work.

6.1.1 System model

In order to describe the interaction between nodes of a MANET and the decision making process that results in a cooperative or selfish behavior of the nodes we will use a classical game introduced by A. Tucker [90]. In the classical PD game, two players are both faced with a decision to either cooperate (C) or defect (D). The decision is made simultaneously by the two players with no knowledge of the other player's choice until the choice is made. If both cooperate, they receive some benefit (R). If both defect they receive a specific punishment (P). However, if one defects, and one cooperates, the defecting strategy receives no punishment (T) and the cooperator a punishment (S). The game is often expressed in

[1] In this Chapter we will adopt the word player and node as synonyms

the canonical form in terms of pay-offs, as shown in table 6.1. The PD game is a much

(a)

		Player j	
		Cooperate	Defect
Player i	Cooperate	(R, R)	(S, T)
	Defect	(T, S)	(P, P)

(b)

		Player j	
		Cooperate	Defect
Player i	Cooperate	$(3, 3)$	$(-2, 4)$
	Defect	$(4, -2)$	$(0, 0)$

Table 6.1: Prisoner's Dilemma payoff matrix: (a) general, (b) example

studied problem due to its far-reaching applicability in many domains. In game theory, the prisoner's dilemma can be viewed as a two-players, non-zero-sum, non-cooperative and simultaneous move game. In order to have a dilemma the following expressions must hold:

$$\begin{aligned} &T > R > P > S \\ &R > \tfrac{S+T}{2} \end{aligned} \tag{6.1}$$

In our model, a MANET formed by N nodes is considered as an N-player playground in which randomly, any two nodes can meet. We suppose that every node of the network has some data traffic to be sent through some source route that is the result of the execution of some routing protocol (as an example the DSR protocol). We also suppose that when any two nodes meet, at some time period t, they both need to send some data packets through each other, i.e. using each other as a relay node. Before the actual process of sending a packet, the two nodes have to take the decision whether to cooperate or defect. By cooperating a node will forward one (or more) data packet for the requesting node, whereas by defecting a node will not relay data packets on behalf of the requesting node. Instead of including an accurate description of energetic costs, topology information, possible interference and path information we will base our model on some basic economic modelling. As an illustrative and intuitive example, let us consider two players (nodes) with some letters (data messages) to send. For each letter leaving a player, a stamp (energy cost for sending one data packet) has to be used. When a letter is forwarded towards its destination the player benefit is (arbitrarily) fixed to 5: of course, the benefit for a successful communication should be higher than the (energetic) cost for sending the letter. So, for example, if two players meet and both have a letter to send, if the decision of a player is to cooperate, she will have to spend two stamps (one for her letter, and one for her opponent's letter) and eventually receive a benefit of 5 if her opponent cooperated, leading to a payoff equals to 3 in case the opponent decided to cooperate and to a payoff equals to -2 in case the opponent decided to defect. This situation can be translated in a payoff matrix which matches the one illustrated in table 6.1 of the classical PD

game. Of course, it is arguable that such a simple model can represent a real MANET, but we believe that the limitations imposed by our model are greatly compensated by the consolidated theoretical results available in the literature for the prisoner's dilemma. Furthermore, we plan to extend the model in order to cope with a T-player simultaneous move game where thus taking into account the cooperative strategy of nodes that are part of an entire path from a data source to her selected destination. However, the key of the model presented in this section and any further extensions is the "willingness to communicate" assumption: during every play of the game (both in the basic PD and in the iterated PD, as we will see in the next section) both players engaged in the decision making process (cooperate or not) are supposed to have some data packets to be sent through the opponent player. As we will see later this assumption is necessary in order to implement a punishment mechanism for a non-cooperating node.

6.1.2 The iterated Prisoner's dilemma

The iterated version of the PD game, and in general repeated games have been extensively studied in the literature and the interested reader could refer to [92] in order to find a basic yet complete introduction to the theory of games, equilibrium concepts and iterated games. In this section we will not focus on the basic results from game theory applied to the PD (e.g. Nash equilibrium of the one shot PD game) but we will introduce some concepts that will be used in the rest of the Chapter. One surprising feature of many one-shot games (i.e. games that are played only once) including the PD game, is that the Nash equilibrium is non-cooperative: each player would prefer to fink (defect) rather than to cooperate. However, in a more realistic scenario (e.g. a MANET) a particular one shot game can be played more than once; in fact, a realistic game could even be a correlated series of one shot games. In such iterated games an action chosen by a player early on can affect what other players choose to do later on: repeated games can incorporate a phenomena which we believe are important but not captured when restricting our attention to static, one shot games. In particular, we can strive to explain how cooperative behavior can be established as a result of rational behavior. In this section we'll discuss repeated games which are "infinitely repeated". This need not mean that the game never ends, however. We will see that this framework can be appropriate for modelling situations in which the game eventually ends (with probability one) but the players are uncertain about exactly when the last period is (and they always believe there's some chance the game will continue to the next period).

In the reminder of the Chapter, we will often refer to the concept of Nash Equilibrium, for which a definition is given below. The interested reader should refer to [39, 81] for a more formal definition.

Definition: *In game theory, the Nash equilibrium (named after John Nash) is a kind of optimal strategy for games involving two or more players, whereby the players reach*

an outcome to mutual advantage. If there is a set of strategies for a game with the property that no player can benefit by changing his strategy while the other players keep their strategies unchanged, then that set of strategies and the corresponding payoffs constitute a Nash equilibrium.

In the following subsections we'll introduce in a more formal way some basic concepts related to repeated games and infinitely repeated games. We will then show the definition of a strategy for a player and explain how to verify if a (simple) strategy is an equilibrium for a game. A reader who is familiar with game theory is invited to skip the following two sections (6.1.3 and 6.1.3).

6.1.3 Repeated games theory

Consider a game G (which we'll call the **stage game** or the constituent game). Let the player set be $I = 1, ..., n$. In our present repeated-game context it will be clarifying to refer to a player's stage game choices as actions rather than strategies. (We'll reserve "strategy" for choices in the repeated game). So each player has a *pure-action* space A_i. The space of action profiles is $A = X_{i \in I} A_i$.
Each player has a Von Neumann-Morgenstern utility function defined over the outcomes of G, $g_i : A \rightarrow \Re$, that in the particular case of the two players PD game takes the form of a payoff matrix as in table 6.1. Let G be played several times (perhaps an infinite number of times) and award each player a payoff which is the (discounted) sum of the payoffs she got in each period from playing G. Then this sequence of stage games is itself a game: a *repeated game.*
Two statements are implicit when we say that in each period we're playing the same stage game: a) for each player the set of actions available to her in any period in the game G is the same regardless of which period it is and regardless of what actions have taken place in the past and b) the payoffs to the players from the stage game in any period depend only on the action profile for G which was played in that period, and this stage-game payoff to a player for a given action profile for G is independent of which period it is played. Statements a) and b) are saying that the environment for our repeated game is stationary (or, alternatively, independent of time and history). *This does not mean the actions themselves must be chosen independently of time or history.*
We'll limit our attention here to cases in which the stage game is a one-shot, simultaneous-move game. Then we interpret a) and b) above as saying that the payoff matrix is the same in every period. We make the typical "*observable action*" or "*standard private monitoring*" assumption that the play which occurred in each repetition of the stage game is revealed to all the players before the next repetition. Therefore even if the stage game is one of imperfect information (as it is in simultaneous-move games)-so that during the stage game one of the players doesn't know what the others are doing/have done that period-each player does learn what the others did before another round is played. This allows subsequent choices to be conditioned on the past actions of other players. We'll

see later that if we make the assumption of "*imperfect private monitoring*" results can be significantly different.
Before we can talk about equilibrium strategies in repeated games, we need to get precise about what a *strategy* in a repeated game is. We'll find it useful when studying repeated games to consider the semi-extensive form. This is a representation in which we accept the normal-form description of the stage game but still want to retain the temporal structure of the repeated game.
Let the first period be labelled $t = 0$. The last period, if one exists, is period T, so we have a total of $T + 1$ periods in our game. We allow the case where $T = \infty$, i.e. we can have an infinitely repeated game.
We'll refer to the action of the stage game G which player i executes in period t as a_i^t . The action profile played in period t is just the n-tuple of individuals' stage-game actions:

$$a^t = \left(a_1^t, ..., a_n^t\right) \tag{6.2}$$

We want to be able to condition the players' stage-game action choices in later periods upon actions taken earlier by other players. To do this we need the concept of a **history**: a description of all the actions taken up through the previous period. We define the history at time t to be:

$$h^t = \left(a^0, a^1, ..., a^{t-1}\right) \tag{6.3}$$

In other words, the history at time t specifies which stage-game action profile (i.e., combination of individual stage-game actions) was played in each previous period. Note that the specification of h^t includes within it a specification of all previous histories $h^0, h^1, ..., h^{t-1}$. For example, the history h^t is just the concatenation of h^{t-1} with the action profile a^{t-1}; i.e. $h^t = (h^{t-1}; a^{t-1})$. The history of the entire game is $h^{T+1} = (a^0, a^1, ..., a^T)$. Note also that the set of all possible histories h^t at time t is just:

$$A^t = \underset{j=0}{\overset{t-1}{X}} A \tag{6.4}$$

the t-fold Cartesian product of the space of stage-game action profiles A.
To condition our strategies on past events, then, is to make them functions of history. So we write player i's period-t stage-game strategy as the function s_i^t, where $a_i^t = s_i^t(h^t)$ is the stage-game action she would play in period t if the previous play had followed the history h^t. A player's stage-game action in any period and after any history must be drawn from her action space for that period, but because the game is stationary her stage-game action space A_i does not change with time. The period-t stage game strategy profile s^t is:

$$s^t = \left(s_1^t, ..., s_n^t\right) \tag{6.5}$$

So far we have been referring to stage-game strategies for a particular period. Now we can write, using these stage-game entities as building blocks, a specification for a player's strategy for the repeated game. We write player i's strategy for the repeated game as:

$$s_i = \left(s_i^0, s_i^1, ..., s_i^T\right) \tag{6.6}$$

i.e. a $(T + 1)$-tuple of history-contingent player-i stage-game strategies. Each s_i^t takes a history $h^t \in A^t$ as its argument. The space S_i of player-i repeated-game strategies is the

set of all such $(T+1)$-tuples of player-i stage game strategies $s_i^t : A^t \to A_i$.
We can write a strategy profile s for the whole repeated game in two ways. We can write it as the n-tuple profile of players' repeated-game strategies:

$$s = (s_1, ..., s_n) \tag{6.7}$$

as defined in (6.5). Alternatively, we can write the repeated-game strategy profile s as:

$$s = \left(s^0, s^1, ..., s^T\right) \tag{6.8}$$

i.e., as a collection of stage-game strategy profiles, one for each period, as defined in (6.4).
Let's see how this repeated game is played out once every player has specified her repeated-game strategy s_i. It is more convenient at this point to view this repeated-game strategy profile as expressed in (6.8), *i.e.* as a sequence of $T+1$ history-dependent stage-game strategy profiles.
When the game starts, there is no past play, so the history h_0 is degenerate: every player executes her $a_i^0 = s_i^0$ stage-game strategy from (6.5). This zero-*th* period play generates the history $h^1 = (a^0)$, where $a^0 = (a_1^0, ..., a_n^0)$. This history is then revealed (or monitored by the players themselves) to the players so that they can condition their period-1 play upon the period-0 play. Each player then chooses her $t = 1$ stage-game strategy $s_i^1(h^1)$. Consequently, in the $t = 1$ stage game the strategy profile $a^1 = s^1(h^1) = (s_1^1(h^1), ..., s_n^1(h^1))$ is played. In order to form the updated history this stage-game strategy profile is then concatenated onto the previous history: $h^2 = (a^0, a^1)$. This new history is revealed to all the players and they each then choose their period-2 stage-game strategy $s_i^2(h^2)$, and so on. We say that h^{T+1} is the path generated by the repeated-game strategy profile s.

Let us now consider the *payoff function of the repeated game.* We can think of the players as receiving their stage-game payoffs period-by-period. Their repeated game payoffs will be an additively separable function of these stage-game payoffs. Right away we see a potential problem: if the game is played an infinite number of times, there is an infinite number of periods and, hence, of stage-game payoffs to be added up. In order that the players' repeated-game payoffs be well defined we must ensure that this infinite sum does not blow up to infinity. We ensure the finiteness of the repeated-game payoffs by introducing *discounting* of future payoffs relative to earlier payoffs. Such discounting can be an expression of time preference and/or uncertainty about the length of the game. We introduce the average discounted payoff as a convenience which normalizes the repeated-game payoffs to be "on the same scale" as the stage game payoffs.
Infinite repetition can be the key for obtaining behavior in the stage games which could not be equilibrium behavior if the game were played once or a known finite number of times. For example, defection in every period by both players is the unique equilibrium in any finite repetition of the PD[2]. When repeated an infinite number of times, however, cooperation in every period is an equilibrium if the players are "sufficiently patient".
When studying infinitely repeated games we are concerned about a player who receives a

[2]See theorem 4 in "Repeated Games" handouts by J. Ratliff [92].

payoff in each of infinitely many periods. In order to represent her preferences over various infinite payoff streams we want to meaningfully summarize the desirability of such a sequence of payoffs by a single number. A common assumption is that the player wants to maximize a weighted sum of her per-period payoffs, where she weights later periods less than earlier periods. For simplicity this assumption often takes the particular form that the sequence of weights forms a geometric progression: for some fixed $\delta \in (0,1)$, each weighting factor is δ times the previous weight. δ is called her *discount factor*. If in each period t player i receives the payoff u_i^t, we could summarize the desirability of the payoff stream $u_i^0, u_i^1, ...$ by the number:

$$\sum_{t=0}^{\infty} \delta^t u_i^t \tag{6.9}$$

Such an intertemporal preference structure has the desirable property that the infinite sum of the weighted payoffs will be finite (since the stage-game payoffs are bounded). A player would be indifferent between a payoff of x^t at time t and a payoff of $x^{t+\tau}$ received τ periods later if:

$$x^t = \delta^\tau x^{t+\tau} \tag{6.10}$$

A useful formula for computing the finite and infinite discounted sums we will use later in this section is:

$$\sum_{t=T_1}^{T_2} \delta^t = \frac{\delta^{T_1} - \delta^{T_2+1}}{1-\delta} \tag{6.11}$$

which, in particular, is valid for $T_2 = \infty$.
If we adopted the summation (6.9) as our players' repeated-game utility function, and if a player received the same stage-game payoff v_i in every period, her discounted repeated-game payoff, using (6.11), would be $v_i/(1-\delta)$. It is however more convenient to transform the repeated-game payoffs to be "on the same scale" as the stage-game payoffs, by multiplying the discounted payoff sum from (6.9) by $(1-\delta)$. So we define the average discounted value of the payoff stream $u_i^0, u_i^1, ...$ by:

$$(1-\delta)\sum_{t=0}^{\infty} \delta^t u_i^t \tag{6.12}$$

It is often convenient to compute the average discounted value of an infinite payoff stream in terms of a leading finite sum and the sum of a trailing infinite substream. For example, say that the payoffs v_i^t a player receives are some constant payoff v_i' for the first t periods, *i.e.* $0, 1, 2, , t-1$, and thereafter she receives a different constant payoff v_i'' in each period $t, t+1, t+2, ...$ The average discounted value of this payoff stream is:

$$(1-\delta)\sum_{\tau=0}^{\infty} \delta^\tau v_i^\tau = (1-\delta)\left(\sum_{\tau=0}^{t-1} \delta^\tau v_i^\tau + \sum_{\tau=t}^{\infty} \delta^\tau v_i^\tau\right) = (1-\delta)\left(\frac{v_i'(1-\delta^t)}{1-\delta} + \frac{v_i''\delta^t}{1-\delta}\right) = (1-\delta^t)v_i' + \delta^t v_i'' \tag{6.13}$$

It is possible to see that the average discounted value of this stream of bivalued stage-game payoffs is a convex combination of the two stage-game payoffs. We can iterate this procedure in order to evaluate the average discounted value of more complicated payoff

streams. Another useful example is when a player receives v_i' for the first t periods, then receives v_i'' only in period t and receive v_i''' every period thereafter. The average discounted value of the stream beginning in period t (discounted to period t) is: $(1-\delta)v_i'' + \delta v_i'''$. Substituting this for v_i'' in (6.13), we find that the average discounted value of this three-valued payoff stream is:

$$(1-\delta^t)v_i' + \delta^t\left[(1-\delta)v_i'' + \delta v_i'''\right] \tag{6.14}$$

We have now defined all the formalism needed to examine the equilibrium of a (infinitely) repeated PD game and to verify if a predefined strategy constitutes an equilibrium. The various definitions of equilibrium and the related theorems can be found in [92].

Cooperation in the Repeated Prisoner's dilemma

In the one-shot PD, the players cannot avoid choosing their dominant strategy Defect (see table 6.1). In order to make the following analysis simpler, consider the following payoff matrix, presented in table 6.2. It is easy to verify that conditions (6.1) hold.

		Player j	
		Cooperate	Defect
Player i	Cooperate	$(1,1)$	$(-1,2)$
	Defect	$(2,-1)$	$(0,0)$

Table 6.2: Simplified Prisoner's Dilemma payoff matrix.

Even when this game is finitely repeated, because the stage game has a unique Nash equilibrium, the unique subgame-perfect equilibrium has both players defecting in every period. However, when the players are sufficiently patient it is possible to sustain cooperation (i.e. keeping "Cooperate") in every period as a subgame-perfect equilibrium of the infinitely repeated game. First we will see that such cooperation is a Nash equilibrium of the repeated game. We will then show that this cooperation is a subgame-perfect equilibrium.
When an infinitely repeated game is played, each player i has a repeated-game strategy s_i, which is a sequence of history-dependent stage-game strategies s_i^t ; i.e. $s_i = (s_i^0, s_i^1, ...)$, where each $s_i^t : A^t \rightarrow A_i$. The n-tuple of individual repeated-game strategies is the repeated-game strategy profile $s = (s_1, s_2, ..., s_n)$.
As a fundamental example, let us consider a particular strategy that a player could follow and which is sufficient to sustain cooperation. This strategy is also known as the `spiteful` strategy.

Spiteful

- Cooperate in the first period;
- In later periods, cooperate if both players have always cooperated;
- However, if either player has ever defected, defect for the remainder of the game.

More precisely and formally, it is possible to write player i's repeated-game strategy $\bar{s}_i = (\bar{s}_i^0, \bar{s}_i^1, ...)$ as the sequence of history-dependent stage-game strategies such that in period t and after history h^t,

$$\bar{s}_i^t\left(h^t\right) = \begin{cases} C, t = 0 \text{ or } h^t = \left((C,C)^t\right) \\ D, \text{ otherwise} \end{cases} \tag{6.15}$$

First, we will show that for sufficiently "patient players" the strategy profile $\bar{s} = (\bar{s}_1, \bar{s}_2)$ is a Nash equilibrium of the repeated game. Then we will show that for the same required level of patience these strategies are also a subgame-perfect equilibrium.
Now, if both players conform to the alleged equilibrium prescription, they both play "cooperate" at $t = 0$. Therefore at $t = 1$, the history is $h^1 = (C, C)$; so they both play "cooperate" again. Therefore at $t = 2$, the history is $h^2 = ((C, C), (C, C))$, so they both play "cooperate" again. And so on .... The path of s is the infinite sequence of cooperative action profiles $((C, C), (C, C), ...)$. The repeated-game payoff to each player corresponding to this path is trivial to calculate: they each receive a payoff of 1 in each period, therefore the average discounted value of each player's payoff stream is 1.
Can player i gain from deviating from the repeated-game strategy given that player j is faithfully following $\bar{s}_j$?
Let t be the period in which player i first deviates. She receives a payoff of 1 in the first t periods $0, 1, ..., t-1$. In period t, she plays "defect" while her conforming opponent played "cooperate", yielding player i a payoff of 2 in that period. This defection by player i now triggers an open-loop "defect"-always response from player j. Player i's best response to this open-loop strategy is to "defect" in every period herself. Thus she receives zero in every period $t + 1, t + 2, ...$
To calculate the average discounted value of this payoff stream to player i we can refer to (6.14), and substitute $v_i' = 1$, $v_i'' = 2$, and $v_i''' = 0$. This yields player i's repeated-game payoff when she defects in period t in the most advantageous way to be $1 - \delta^t(2\delta - 1)$. This is weakly less than the equilibrium payoff of 1, for any choice of defection period t, as long as $\delta \geq \frac{1}{2}$. Thus we have defined what we meant by "sufficiently patient:" cooperation in this PD game is a Nash equilibrium of the repeated game as long as $\delta \geq \frac{1}{2}$.
To verify that $\bar{s}$ is a subgame-perfect equilibrium of the repeated prisoners' dilemma it is necessary to check that this strategy profile's restriction to each subgame is a Nash equilibrium of that subgame. Consider a subgame, beginning in period τ with some history h^τ. What is the restriction of $\bar{s}_i$ to this subgame? Denoting the restriction by s_i we have:

$$\hat{s}_i^t(\hat{h}^t) = \bar{s}_t^{t+\tau}\left(\left(h^\tau; \hat{h}^t\right)\right) = \begin{cases} C, h^\tau = ((C,C)^\tau) \text{ and } \hat{h}^t = \left((C,C)^t\right) \\ D, \text{ otherwise} \end{cases} \tag{6.16}$$

We can partition the subgames of this game, each identified by a beginning period τ and a history h^τ, into two classes:

A) those in which both players chose "cooperate" in all previous periods, i.e. $h^\tau = ((C,C)^\tau)$,

B) those in which a defection by either player has previously occurred.

For those subgames in class A), the sequence of restrictions $\hat{s}_i^t(\hat{h}^t)$ from (6.16) reduces to the sequence of original stage-game strategies $\bar{s}_i^t(h^t)$from (6.15), i.e. for all and we have:

$$\hat{s}_i^t(\hat{h}^t) = \begin{cases} C, h^\tau = ((C,C)^\tau) \text{ and } \hat{h}^t = \left((C,C)^t\right) \\ D, \text{ otherwise} \end{cases} = \begin{cases} C, \hat{h}^t = \left((C,C)^t\right) \\ D, \text{ otherwise} \end{cases} = \bar{s}_i^t\left(h^t\right) \tag{6.17}$$

Because $\bar{s}$ is a Nash equilibrium strategy profile of the repeated game, for each subgame h^τ in class A), the restriction $\hat{s}$ is a Nash equilibrium strategy profile of the subgame when $\delta \geq \frac{1}{2}$.
For any subgame in class B), $h^\tau \neq ((C,C)^\tau)$.
Therefore the restriction $\hat{s}$ of $\bar{s}$ specifies for all $\hat{s}_i^t = D$. In other words, in any subgame reached by some player having "defected" in the past, each player chooses the open-loop strategy "defect always." Therefore the repeated-game strategy profile $\hat{s}$ played in such a subgame is an open-loop sequence of stage-game Nash equilibria. From Theorem 1 of [92] we know that this is a Nash equilibrium of the repeated game and hence of this subgame. We have shown that for every subgame the restriction of to that subgame is a Nash equilibrium of that subgame for $\delta \geq \frac{1}{2}$. Therefore s is a subgame-perfect equilibrium of the infinitely repeated PD when $\delta \geq \frac{1}{2}$.

6.1.4 Complex strategies in the Iterated Prisoner's Dilemma

In subsection 6.1.3, we detailed the analysis of a particular strategy called *spiteful* that was shown to be an equilibrium strategy (both a Nash equilibrium for the whole repeated game and a subgame perfect equilibrium) for the prisoner's dilemma. Axelrod and Hamilton [9, 10] used a computer tournament to numerically detect strategies that would favor cooperation among players engaged in the iterated PD. In a first round, 14 more or less sophisticated strategies and one totally random strategy competed against each other for the highest average scores in an iterated PD of 200 moves.
Unexpectedly, a very simple strategy did outstandingly well:

TIT-FOR-TAT

- Cooperate on the first period and then copy your opponent's last move for all subsequent periods

This strategy was called Tit-for-tat (TFT) and became the founder of an ever growing amount of successful strategies. To study the behavior of strategies from a numerical point of view, two kinds of computation can be done.

- The first one is a simple round robin tournament, in which each strategy meets all other strategies. Its final score is then the sum (not the discounted sum) of all scores done in each confrontation. At the end, the strategy's strength measurement is given by its range in the tournament.
- The second type of numerical analysis is a simulated ecological evolution, in which at the beginning there is a fixed population including the same quantity of each strategy. A round robin tournament is made and then the population of bad strategies is decreased whereas good strategies obtain new elements. The simulation is repeated until the population has been stabilized, i.e. the population does not change anymore. A good strategy is then a strategy which stays alive in the population for the longest possible time, and in the biggest possible proportion. This kind of evaluation quotes the robustness of strategies.

Before the introduction of CORE as a strategy for the iterated PD, it is important to detail the computation method for ecological evolution, for example involving three strategies. Suppose that, initially, the population is composed of three strategies A, B, C. At generation n each strategy is represented by a certain number of players: $W_n(A)$ using A, $W_n(B)$ using B and $W_n(C)$ using C.
The payoff matrix of two-by-two meeting between A, B and C is computed and is thus known (see table 6.1). $V(A|B)$is the score of A when it meets B, etc... Let us suppose that the total size of the population is fixed and constant. We use the following notationv Π :

$$\forall i \in [1, \infty[, \Pi = W_i(A) + W_i(B) + W_i(C) \tag{6.18}$$

The computation of the score (distributed points) of a player using a fixed strategy at generation n is then:

$$\begin{array}{l} g_n(A) = W_n(A)V(A|A) + W_n(B)V(A|B) + W_n(C)V(A|C) - V(A|A) \\ g_n(B) = W_n(A)V(B|A) + W_n(B)V(B|B) + W_n(C)V(B|C) - V(B|B) \\ g_n(C) = W_n(A)V(C|A) + W_n(B)V(C|B) + W_n(C)V(C|C) - V(C|C) \end{array} \tag{6.19}$$

Note that because of the subtractions the computation of g cannot be simplified. The total points distributed to all involved strategies are:

$$t(n) = W_n(A)g_n(A) + W_n(B)g_n(B) + W_n(C)g_n(C) \tag{6.20}$$

The size of each sub-population at generation $n+1$ is finally:

$$\begin{array}{l} W_{n+1}(A) = \frac{\Pi W_n(A)g_n(A)}{t(n)} \\ W_{n+1}(B) = \frac{\Pi W_n(B)g_n(B)}{t(n)} \\ W_{n+1}(C) = \frac{\Pi W_n(C)g_n(C)}{t(n)} \end{array} \tag{6.21}$$

All division being rounded to the nearest lower integer.
Classical results on the iterated PD, which have been emphasized by Axelrod in [10] show that to bee good a strategy has to:

- Not be the first to defect
- Be reactive
- Forgive
- Be simple

The TFT strategy which satisfies all those criteria, has, since Axelrod's book, been considered to be one of the best strategies not only for cooperation but also for the evolution of cooperation.

6.1.5 CORE as a complex strategy for the Iterated Prisoner's Dilemma

It is now important to define the scope of our analysis. After a brief introduction on the theory behind the study of the iterated PD game, we are focusing on the numerical analysis (through a simulation software [63]) of the features presented by some specific strategies that the players of the iterated PD should follow in order to promote cooperation. Furthermore, we want to compare some of the strategies available in the game theoretic literature and known to be the "best" strategies both from a cooperation point of view and from an evolutionary point of view with the strategy derived from the CORE cooperation enforcement mechanism. We suggest to refer to Chapter 4 in order to grasp the details and the functioning of the CORE mechanism.
Our claim is that the CORE strategy can be considered equivalent to the TFT strategy under certain circumstances (namely when the reputation buffer is of size 1). Furthermore, we will show through the evolutionary simulation outlined in section 6.1.4, that the CORE strategy outperforms over all the other analyzed strategies when the assumption of "perfect private monitoring" is replaced by the "**imperfect private monitoring**" assumption. The `CORE`[3] strategy can be defined as follow:

[3]The reader should be informed that in this Chapter we consider a limited version of the CORE mechanism in which reputation is evaluated through a simple average over the past observations made through the watchdog mechanism. A more faithful definition of the CORE strategy is reserved for our future work.

`CORE`

- Cooperate on the first move;
- In each period, observe the past B opponent's moves and build a vector $\overrightarrow{b} = (b_1, ..., b_k, ..., b_B)$ where each element equals +1 for a cooperation and -1 for a defection;
- Evaluate reputation as $reputation = \frac{1}{B}\sum_k b_k$;
- If $reputation \geq 0$ Cooperate else Defect.

We want to show now that the TFT strategy represents a particular case of the `CORE` strategy. Indeed, if we set $B = 1$ it means that only one observation over the opponent's past moves is taken into account to build the reputation information. This implies that if the opponent cooperated in the last move her reputation will be positive and the player will chose too cooperate. Vice versa, if the last opponent's move was a defection, the reputation would be negative and the response of the player would be to defect. This is exactly what the TFT strategy implies: cooperate on the first move and do what the opponent did in the previous move.
In this Chapter, an analytical result stating that the CORE strategy is an equilibrium strategy will not be presented as the work in this direction is in progress: however we believe that the analysis will be facilitated thanks to the equivalence of the TFT and the `CORE` strategy.
In the following subsections we present some results obtained through evolutionary simulations using the iterated PD software available in [63]. The `CORE` strategy has been coded and added up to the list of available strategies in the software.

6.1.6 Simulations with the "perfect private monitoring" assumption

We present here the results of evolutionary simulations involving three strategies when the standard perfect monitoring assumption is made. As described in section 6.1.4, suppose that, initially, the population is composed of five strategies `tit-for-tat`, `spiteful`, `CORE`, `all-C` (cooperate always) and `all-D` (defect always).
Initially, each strategy is represented by a certain number of players: 100 players using each of the mentioned strategies. As it is possible to see in figure 6.1, after 5 generations the all-D strategy disappears: the three[4] winning strategies are equivalent for promoting cooperation and, more important, for the *evolution of cooperation*. This implies that the winning strategies obtained the same payoff in a two-by-two round robin tournament and can be considered equivalent from an evolutionary point of view.

[4]Note that we are not considering the all-C as a winning strategy because of its history independent nature.

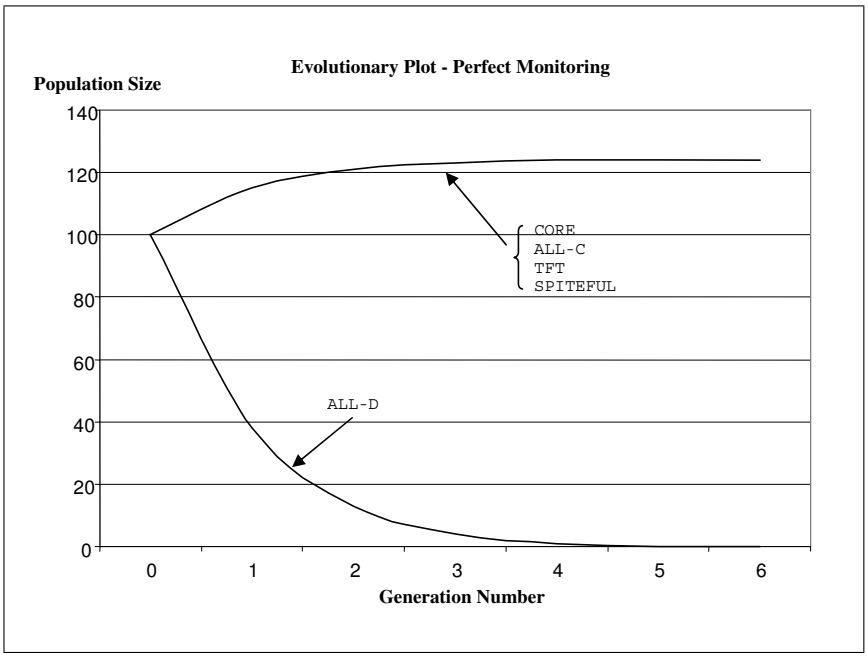


Figure 6.1: Evolutionary simulation of complex strategies for the Iterated PD with **perfect monitoring**.

6.1.7 Simulations with the "imperfect private monitoring" assumption

The majority of work in the iterated prisoner's dilemma has focused on games in a noise-free environment, i.e. there is no danger of a signal being misinterpreted by the opponent or the message being damaged in transit. This assumption of a noise-free environment is not necessarily valid if one is trying to model real-world scenarios. As a specific example, when considering interactions between two nodes in a MANET where the behavior of a node follows the game theoretical model imposed by the prisoner's dilemma, it would be interesting to consider errors due to the watchdog mechanism. The interested reader should refer to Chapter 4 in order to understand intrinsic problems of the watchdog mechanism and the promiscuous mode operation of wireless cards. Specifically, the watchdog mechanism can be thought of as the private monitoring assumption in a two-players iterated prisoner dilemma: it is thanks to the watchdog mechanism (private monitoring) that a node (player) can infer in any period the behavior (opponent's past moves) of her neighbor and decide which actions needs to be taken (strategy). There are different means that can be chosen to introduce noise to the simulation:

- mis-implementation (when the player makes a mistake implementing its choice)
- **mis-perception** (when one player misperceives the other player's signal or choice)

In this section we will concentrate on mis-perception noise as we believe it significantly linked to the problems introduced by the watchdog mechanism. Kahn and Murnighan [50] find that in experiments dealing with prisoner's dilemma in noisy environments, cooperation is more likely when players are sure of each other's payoffs. Miller's experiments in genetic algorithms applied to the prisoner's dilemma results in the conclusion that cooperation is at its greatest when there is no noise in the system and that this cooperation decreases as the noise increases [74]. Some ideas to promote cooperation in noisy environments have been posited by Axelrod; these include genetic kinship, clustering of like strategies, recognition, maintaining closeness when recognition capabilities are limited or absent (e.g limpets in nature), increasing the chance of future interactions (certain social organizations, hierarchies in companies etc.), changing the pay-offs, creating social norms where one learns cooperation. Hoffman [44] reports that results are sensitive to the extent to which players make mistakes either in the execution of their own strategy (mis-implementation noise) or in the perception of opponent choices (mis-perception noise).
In particular, cooperation is vulnerable to noise as it is supported by conditional strategies. For example, in a game between two TFTs, a single error would trigger a series of alternating defection. A number of authors confirm the negative effect of noise of TFT and find that more forgiveness promotes cooperation in noisy environments [79], [14].
As described in section 6.1.6, we executed an evolutionary simulation involving 5 strategies when **mis-perception noise** was taken into account: we decided to set the noise to the value of 10% and we took the average population size over 50 simulation runs. 100 players for each of the following strategies competed in a round-robin tournament as described in section 6.1.4: `tit-for-tat`, `spiteful`, `CORE`, `gradual` and `soft-majo`[5]. As it is possible to observe in figure 6.2, the `CORE` strategy outperforms and results to be the most evolutionary stable and robust strategy among all the population (we believe though that exceptions especially constructed in which performances of `CORE` are not so outstanding are possible but are seldom and not easy to obtain).

The reason why `CORE` performs better than the other strategies when the imperfect monitoring assumption is made can be explained as follows: by adopting the `CORE` strategy, a node base her decision of whether to cooperate or not using a certain amount of observations made on the opponents past moves as defined by the B parameter. Thus, the reputation measure evaluated by the node takes into account more than one observation and is less sensible to any mis-perception noise.
Furthermore, in its advanced version (which has not been implemented in the simulations, though), the `CORE` strategy also weights the past B observations giving more relevance to past observations than recent ones. It is intuitive to realize that a transient misbehavior is filtered out by the reputation mechanism that makes `CORE` more flexible and "*forgiving*" in presence of temporary misbehavior or a momentary high percentage of noise. A specific example of such a situation can be found when thinking of a communication between nodes of a MANET in presence of obstacles or high interference.
In Figure 6.3, two populations of 100 members adopt respectively the `CORE` and the `TFT`

[5]The gradual and soft-majo strategies are described in [63].

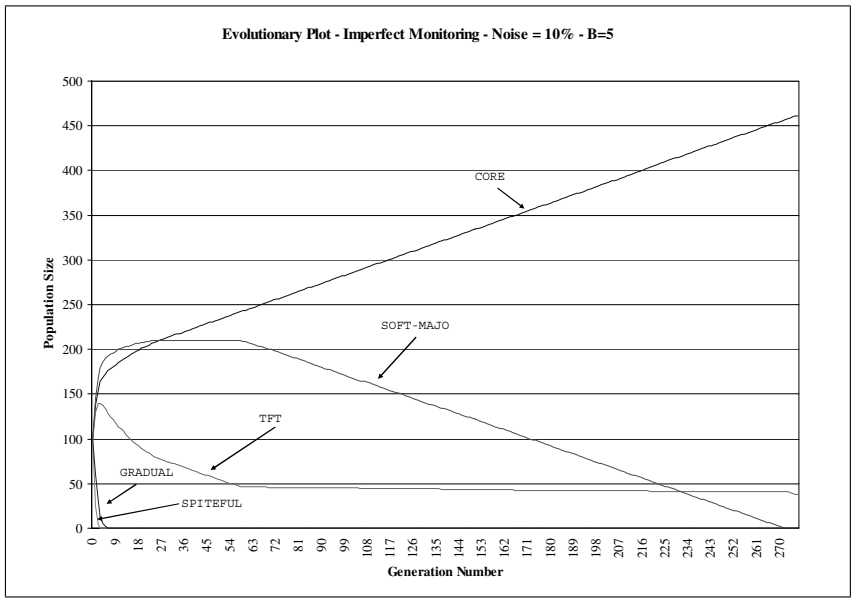


Figure 6.2: Evolutionary simulation of complex strategies for the Iterated PD **with noise**.

strategy. The noise value has been set to 20% and 50 rounds of simulations have been executed in order to take average values of the evolution of population sizes.
It is possible to observe that both strategies are evolutionary stable, in the long run; however, `CORE` is the winning strategy as the population size of players adopting it increases at each new generation, as defined in section 6.1.4. Furthermore, figure 6.3 shows that the reputation buffer size (B) and both the stability condition and population size are directly related.
As B increases, stability is reached at a lower generation number (i.e. earlier) and the population size of players adopting the `CORE` strategy grows faster. We believe however that these interesting results have to be evaluated in an analytical way: the fine-tuning of `CORE` parameters (such as B and the frequency at which observations are made) would require a laborious empirical study if carried out only by means of evolutionary simulations. We plan to analyze the `CORE` strategy in our future work taking as a starting point the analysis of the `SPITEFUL` strategy presented in section 6.1.3.

6.2 Cooperative game theory

In an attempt to explain cooperation and coalition formation, most theoretical models use a two-period structure as introduced in [8, 56]. Players must first decide whether or not to join a coalition. In a second step, both the coalition and the remaining agents choose their behavior non-cooperatively. A coalition is stable if no agent has an incentive to leave[6].

[6]The definition of stability also implies that no agent wants to join the coalition.

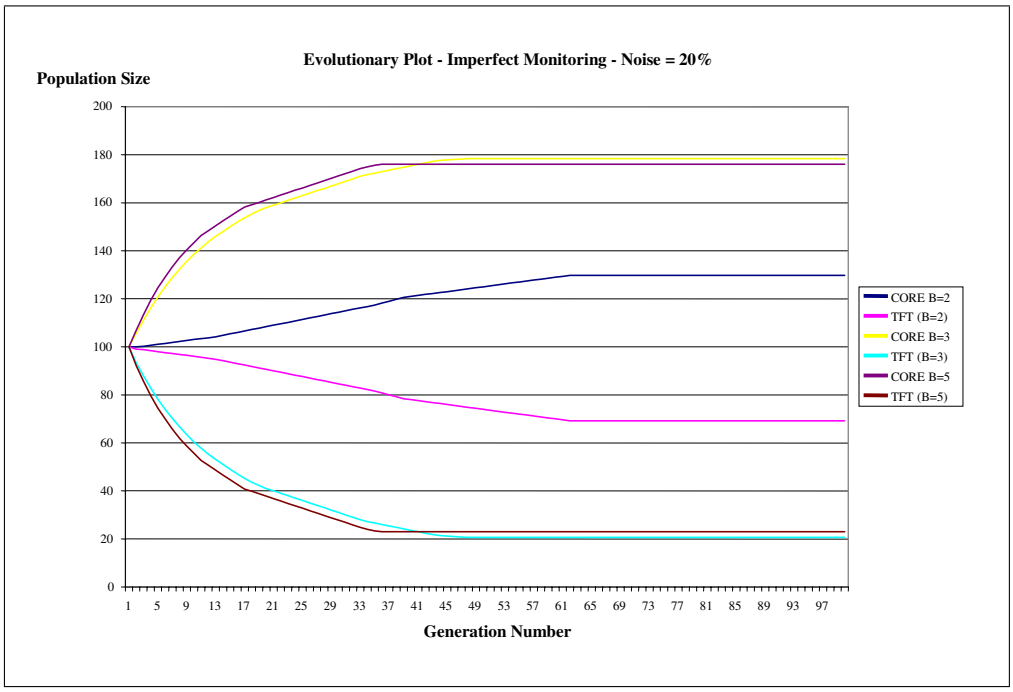


Figure 6.3: Evolutionary simulation of complex strategies for the Iterated PD with noise.

Simulations presented in [11, 27, 28] have shown that, although there is cooperation, the coalition size is rather small.
In this section we suggest an approach based on a preference structure as defined by the ERC-theory [18]. This theory explains most of the behavior of agents observed in diverse experiments but deviates little from the traditional utility concept. The utility of an agent is not solely based on the absolute payoff but also on the relative payoff compared to the overall payoff to all agents. Given a certain relative payoff share, the utility is strictly increasing in the own absolute payoff of the agent. Given a fixed absolute payoff, the agent is best off when receiving just the equal (fair) share. To both sides of this equal share, i.e. when receiving less or more than the fair amount, utility is lower, even if the absolute payoff does not change[7].
In section 6.2.2, we first study a symmetric N-node prisoner's dilemma (PD) game in a non-cooperative setting, in which the agents have only two options available - cooperate or defect. We analyze Nash-equilibrium of the non cooperative game when agents' preferences can be described by ERC, *i.e.* players value both their absolute and their relative payoff. In particular, we look at the number of agents who play cooperatively. We show that non-cooperation is always an equilibrium, since - if no other node cooperates - a node would maximize its absolute payoff and receive the equal share by choosing to defect. Additionally, however, there may be Nash-equilibrium in which nodes cooperate: if, for example, the rest of the agents play cooperatively, a player can get the equal share by choosing to cooperate as well. Hence, if it values its relative payoff being close to the equal share more than its absolute payoff, it will choose to complete the grand coalition.

[7]Note that such a preference for equity is self-centered only and is distinct from altruism.

Clearly, partial cooperation can also occur, whereby some nodes cooperate while others defect. For such equilibrium, we show that the number of cooperating nodes is rather large: since cooperation leads to a lower absolute payoff, for a node to choose to cooperate, playing cooperatively must move it closer to the equal share than defecting would. As we show, this can only be the case if at least half of the nodes cooperate. This result contrasts with the standard result presented in [11] which states that the coalition size is rather small.
Note, however, that in the prisoner's dilemma, the nodes have only the discrete choice of cooperating or defecting, but with respect to the cooperation enforcement problem, the nodes of an ad hoc network might choose their cooperation level[8] continuously. We therefore introduce in section 6.2.3 a symmetric continuous PD-game based on the ERC preference structure. An interesting finding of this analysis is that ERC alone cannot improve upon the non-cooperative Nash-equilibrium with standard preferences in which only the absolute payoff matters to a node. As a further refinement, we propose the cooperative-games approach consisting in a combination between the ERC preference structure and the two-stage coalition formation method [8]. In contrast to the traditional models from the game theory literature, the ERC preference structure allows coalitions to involve a rather large fraction of players. Furthermore, this model allows for a precise characterization of conditions under which even a grand coalition can be obtained.
Finally, in section 6.2.5, we propose a discussion on the relation between a coalition formation process and our cooperation enforcement mechanism CORE, used as an effective complementary tool to impose a specific ERC-type for every node participating in a cooperative setting as an ad hoc network.

6.2.1 The preference structure

Our analysis relies on a preference structure in which players, along with their own absolute payoff, are motivated (non-monotonously) by the relative payoff share they receive, *i.e.* how their standing compares to that of others. We use the ERC model presented in [18] and enhance it with a complete information framework. Let the (non-negative) payoff to node i be denoted by y_i , $i, ..., N$, and the relative share by $\sigma_i = \frac{y_i}{\sum\limits_j y_j}$.

The ERC-utility function is defined as follows: $\alpha_i u(y_i) + \beta_i r(\sigma_i)$ where $\alpha_i, \beta_i \geq 0$ and $u()$ is differentiable, strictly increasing and concave, and $r()$ is differentiable, concave and has its maximum in $\sigma_i = \frac{1}{N}$.
Throughout this Chapter we assume that nodes' disutility from disadvantageous inequality is larger if the node is better off than average, *i.e.* $r(\frac{1}{N} - x) \leq r(\frac{1}{N} + x), \forall x \in \left[0, \frac{1}{N}\right]$. The types of nodes are characterized by the relative weights α_i, β_i.

[8]The definition of cooperation level will be given in section 6.2.3: here it is sufficient to know that cooperation level stands for the fraction of packets (data or routing) that are forwarded by a node of the network playing the cooperation game.

ERC Game Definitions	*Description*
N	Number of players
$\alpha_i u(y_i) + \beta_i r(\sigma_i)$	ERC global utility function for player i
$\alpha_i, \beta_i \geq 0$	ERC-types for player i
y_i	Absolute payoff for player i
$u_i(y_i)$, differentiable, strictly increasing, concave	Absolute utility function for player i
σ_i	Relative payoff for player i
$r(\sigma_i)$, differentiable, concave, maximum in $\sigma_i = \frac{1}{N}$	Relative utility function for player i

Table 6.3: Summarizing table defining the game based on ERC theory.

6.2.2 The prisoner's dilemma with a discrete strategy space

In this section we study a simple symmetric N-node prisoner's dilemma where each mobile node can cooperate, 'c', or defect, 'd': this implies that the strategy set available to each player is discrete and only two actions are allowed. In terms of the node misbehavior problem, this means that the node either correctly executes the network functions or it doesn't.
Let the total number of cooperating nodes be denoted by k. For any given k, the payoff to a node is given by $B(k)$ if the node defects (tries to free-ride). If a node plays cooperatively, it must bear some additional costs $C(k)$. Its payoff is therefore given by $B(k) - C(k)$. We assume decreasing marginal benefits for a node if the number of mobile nodes rises, i.e. $B(k)$ is increasing and concave. Furthermore, the total cost of cooperation, $kC(k)$, increases in k.
In order to generate the standard incentive structure of a PD game, we make the following assumption.

Assumption 1. PD structure:

$$B(k+1) - B(k) < C(k+1) \tag{6.22}$$

Assumption 3. "Individually desirable":

$$B(k+1) - C(k+1) \geq B(k) - C(k) \tag{6.23}$$

Furthermore, we assume that payoffs for both cooperating and defecting nodes are non-negative for all k.

The Nash equilibrium

In the following section we analyze the Nash equilibrium in the one shot PD game under the assumption that all the nodes joining an existing network choose simultaneously.

Assume that k nodes, aside from node i, play cooperatively. We want to study the condition under which node i, which is not part of the set of k cooperating nodes, chooses to cooperate; player i chooses to play 'c' if and only if her utility is higher than when playing 'd', i.e.:

$$\alpha_i u\left[B(k+1)-C(k+1)\right]+\beta_i r\left[\frac{B(k+1)-C(k+1)}{N\cdot B(k+1)-(k+1)C(k+1)}\right]\geq \alpha_i u\left[B(k)\right]+\beta_i r\left[\frac{B(k)}{N\cdot B(k)-kC(k)}\right] \tag{6.24}$$

This is equivalent to node i playing 'c' if and only if:

$$\frac{\alpha_i}{\beta_i}\leq \delta(k) \text{ where } \delta(k)=\frac{r\left[\frac{B(k+1)-C(k+1)}{N\cdot B(k+1)-(k+1)C(k+1)}\right]-r\left[\frac{B(k)}{N\cdot B(k)-kC(k)}\right]}{u\left[B(k)\right]-u\left[B(k+1)-C(k+1)\right]} \tag{6.25}$$

In order to choose 'c' the node must be overcompensated for the loss in absolute gain by moving closer to the average gain. The general conditions for a Nash equilibrium of a ERC-PD game [18] of N nodes whereby the number of cooperating nodes is k^* can be used to study expression (6.25):

$$\frac{\alpha_i}{\beta_i}<\delta(k*-1) \text{ for } k^* \text{ nodes (playing 'c')} \tag{6.26}$$

$$\frac{\alpha_i}{\beta_i}\geq\delta(k*) \text{ for the remaining } N-k^* \text{ nodes (playing 'd')} \tag{6.27}$$

Conditions (6.26) and (6.27) can be used to evaluate the number of nodes k^* that may possibly cooperate in a Nash equilibrium.
On one hand, as long as $\delta(k*-1)<0$, there is no chance of having a coalition of size k^* because $\frac{\alpha_i}{\beta_i}>\delta(k*-1)$for all types and condition (6.26) cannot hold for any node.
On the other hand, the conditions for a Nash equilibrium given by (6.26) and (6.27) imply that if $\delta(k*-1)>0$ then there are types $\left[\left(\frac{\alpha_i}{\beta_i}\right)_{i=1,\ldots,N}\right]$ of nodes such that k^* nodes cooperate and $N-k^*$ nodes free-ride.
Note that for a given distribution of ERC-types, $\delta(k*-1)>0$ is a necessary but not sufficient condition to get a coalition size of k*. For a given payoff structure with $\delta(k*-1)>0$, however, there exist ERC-types such that k^* is the equilibrium for any coalition size.
In order to find feasible coalition sizes, we must therefore study conditions under which $\delta(k)$ is positive.
Note that in (6.25) the denominator of $\delta(k)$ is positive due to **assumption 1**. The sign of the numerator, however, depends on the number k of cooperating nodes.

For $k = 0$ the sign of the numerator is negative, since:

$$r\left(\frac{B(1)-C(1)}{NB(1)-C(1)}\right) = r\left(1-\frac{(N-1)B(1)}{NB(1)-C(1)}\right) < r\left(\frac{B(0)}{NB(0)}\right) = r\left(\frac{1}{N}\right)$$

For $k = N-1$ the sign of the numerator is positive, since

$$r\left(\frac{B(N)-C(N)}{NB(N)-NC(N)}\right) = r\left(\frac{1}{N}\right) >$$

$$r\left(\frac{B(N-1)}{NB(N-1)-(N-1)C(N-1)}\right) = r\left(1-\frac{(N-1)B(N-1)+(N-1)C(N-1)}{NB(N-1)-(N-1)C(N-1)}\right)$$

Therefore, $\delta(0) < 0 < \delta(N-1)$ and no nodes unilaterally cooperate whereas all nodes playing 'c' can establish an equilibrium, provided that all nodes' types $\left(\frac{\alpha_i}{\beta_i}\right)$ are smaller than $\delta(N-1)$.
In general, there are equilibria where only a certain number k^* of nodes cooperate. The crucial point is to find whether or not the numerator is positive. Remember that we previously assumed that:

$$r(\frac{1}{N}-x) \leq r(\frac{1}{N}+x), \forall x \in \left[0, \frac{1}{N}\right]$$

It is necessary, in order to obtain $\delta(k) > 0$, that a node choosing 'd' further deviates from the equal share $(1/N)$ than by playing 'c', i.e.:

$$\frac{1}{N} - \frac{B(k+1)-C(k+1)}{NB(k+1)-(k+1)C(k+1)} > \frac{1}{N} - \frac{B(k)}{NB(k)-kC(k)} \tag{6.28}$$

It is possible to show that inequality (7) is satisfied for $k > N/2$[9] . Assumption (1) and (2) imply that the condition $\delta(k) > 0$ is necessary (but not sufficient) to state that, for any given vector of types, if a node plays 'c' at the equilibrium, then at least half of the nodes cooperate.

Proposition 1. *For any given payoff structure of the PD game with ERC preferences, there is always an equilibrium in which all nodes defect.*

Proposition 2. *Given Assumption 1 and Assumption 2, there is a Nash equilibrium where at least $N/2$ nodes cooperate.*

Based on proposition 2, if there is a coalition of cooperating nodes then it is rather large.

[9]The proof is given in Appendix 1, at the end of the Chapter.

6.2.3 The prisoner's dilemma with a continuous strategy space

In section 6.2.2, we assumed that nodes only have a discrete option as to whether to cooperate or not. Now, we turn to a prisoner's dilemma game where nodes can continuously choose their cooperation levels. As we will see, ERC alone cannot improve upon the non-cooperative Nash-equilibrium with standard preferences whereby only the absolute payoff matters. However, introducing more structure to the game, i.e. if nodes play a coalition game (section 6.2.4), ERC may yield a rather large coalition size or even support the grand coalition.
Let the number of nodes again be denoted by N. We define the cooperation level $q_i \in [0,1]$ as the fraction of packets (both data and routing packets) that node i forwards to its neighboring nodes or to the destination node. Each node must choose its cooperation level q_i $(i = 1, ..., N)$.
Cooperation induces some costs $C(q_i)$ that are assumed to be increasing and convex in the cooperation level $(C'() > 0, C''() > 0)$.
Cooperation also yields some benefit $B(Q)$ in terms of network connectivity and aggregate cooperation effort made available by cooperating nodes, where $Q = \sum_i q_i$ denotes the aggregate cooperation level.
Benefits from cooperation are increasing and concave, $B'() \geq 0, B''() < 0$. The payoff to a node is therefore determined by: $B(Q) - C(q_i)$.

The Nash equilibrium

In order to find the Nash equilibrium point(s) of the game (if one exists) it is necessary to identify the strategy q_i that corresponds to the singularity point(s) of the global ERC utility function, i.e. finding the roots of the first order derivative of the utility function and make sure that those points are maximum of the utility function.
It should be noted that the assumptions made on the convexity of the utility function (see section 6.2.1) alleviate the problem of studying the border conditions in the function domain. In the reminder of this section, we will refer to the *first order condition* when describing the process of finding the Nash equilibrium.
We analyze the Nash equilibrium when nodes act *simultaneously*. Node i chooses q_i to maximize its utility function $\alpha_i u(y_i) + \beta_i r(\sigma_i)$, where:

$$y_i = B\left(\sum_{j \neq i} q_j + q_i\right) - C(q_i) \rightarrow \text{Absolute payoff to player } i$$

$$\sigma_i = \frac{y_i}{\sum_j y_j} \rightarrow \text{Relative payoff to player } i$$

Note that the definition used to express the absolute payoff to node i emphasizes the

strategy space for node i (q_i) as compared to the strategy space available to the other nodes of the network. By choosing q_i, each node determines its own cooperation costs and the benefits from cooperation. The choice of q_i also impacts the payoff of the remaining nodes that in turn is fed back to the node's own utility through the relative payoff. The *first order condition* is therefore given by:

$$\left[\alpha_i u'() + \beta_i r'() \frac{\sum\limits_{j\neq i} y_j}{\sum\limits_j {y_j}^2}\right] [B'(Q) - C'(q_i)] - \beta_i r'() \frac{y_i}{\sum\limits_j {y_j}^2} (N-1) B'(Q) = 0$$

We can rewrite the expression as follows:

$$\alpha_i u'() [B'(Q) - C'(q_i)] + \beta_i r'() \left[\frac{\sum\limits_j y_j - y_i}{\sum\limits_j {y_j}^2} B'(Q) - \frac{\sum\limits_{i\neq j} y_j}{\sum\limits_j {y_j}^2} C'(q_i)\right] = 0 \tag{6.29}$$

or

$$\alpha_i u'() [B'(Q) - C'(q_i)] + \beta_i r'() \left[\frac{1 - N\sigma_i}{\sum\limits_j y_j} B'(Q) - \frac{\sum\limits_{j\neq i} \sigma_j}{\sum\limits_j y_j} C'(q_i)\right] = 0 \tag{6.30}$$

The strategy (q_i) of node i to a given cooperation policy for the rest of the network can be calculated from this first order condition.

Proposition 3. (Continuous game) *In the continuous PD-game based on ERC preferences, the Nash equilibrium is given by solving the following expression:* $B'(Nq^*) - C'(q^*) = 0$. *It is symmetrical as long as at least one node draws utility from its absolute payoff* ($\alpha_i > 0$)[10].

Introducing ERC preferences, therefore, does not increase the cooperation effort chosen by the nodes when playing the PD-game with a continuous action set. It does not even change the equilibrium cooperation levels.
In contrast to the (discrete) prisoner's dilemma, ERC does not add any equilibrium in which there is more cooperation effort. The existence of equilibrium in the PD game that mimics cooperative behavior, therefore, only arises in the presence of discrete action sets. Having a continuous decision variable, ERC does not change the set of equilibrium. The reason is that ERC does not establish a preference for being cooperative, but *for being similar to other nodes* with respect to the payoff.
In this section, however, we used the ERC theory in a classical non-cooperative setting: let us see how the strategy selection of a selfish node change when introducing more structure to the game, i.e. when considering a **cooperative-game setting**.

[10] The proof for this proposition is given in Appendix 2, at the end of this Chapter

6.2.4 Coalition formation: the cooperative-game approach

As a further refinement, we now propose a cooperative-games approach consisting in a combination of the ERC preference structure and the two-stage coalition formation method as introduced in [8].
Let us assume that all nodes are identical with respect to their payoff function (i.e. they use the same definition of utility function). In a first stage, nodes decide whether or not to join the coalition. By the principle of "rationality", each node is assumed to know the decisions of the other nodes. The cooperation levels (i.e. the strategy) that will be chosen in the second stage depend on whether the nodes take part in the coalition or not. The coalition thereby maximizes its collective benefits and plays against the nodes that don't take part in the coalition, which simultaneously maximize their individual utility.
We first study the case of nodes that have **identical ERC-types**. We demonstrate that within the coalition formation game, ERC-preferences can enforce cooperation and even result in the grand coalition. We then look at the case of **heterogeneous ERC-types**. By studying the extreme scenario of nodes that are solely interested either in their absolute payoff or in equity, we will explore the effects of the existence of some equity-oriented nodes in the network.

Coalition of identical ERC-types

We will now solve the coalition formation game backwards, that is, for any coalition size k, we study the *first order conditions* for the choice of the cooperation level inside and outside the coalition. Then, in the second step, the equilibrium coalition size is determined by a *stability condition*. This means that in the equilibrium, k must satisfy the condition that there is no incentive to leave the coalition[11].
For standard preferences (using ERC-preferences this results in the special case $\beta = 0$), the game theory literature shows that the coalition size is rather small. Using ERC preferences, however, the number of nodes within a coalition can be much higher in equilibrium.
Instead of solving the game in general, we will show that if nodes only value the relative payoff high enough, i.e. α/β is below a certain bound, then even the grand coalition can be stable.
The first order condition for nodes outside the coalition (S) is given by (6.31), whereas the cooperation strategy of nodes that take part in the coalition is chosen by maximizing the utility function of a representative member: indeed all nodes within the coalition S select the same strategy q_s since they are assumed to be of the same type. This implies that all members of the coalition have identical absolute payoff ($y_S = B(Q) - C(q_S)$) and relative payoff ($\sigma_S = \frac{y_S}{ky_S + \sum\limits_{j \notin S} y_j}$).

[11] The original work introduced in [63] states that the stability condition is such that there is an incentive to neither leave nor join the coalition.

The first order condition is given by:

$$[\alpha u'() + \beta r'()] \frac{\sum_{j \notin S} \sigma_j}{\sum_j y_j} [kB'(Q) - C'(q_S)] - \beta r'() \frac{\sigma_S}{\sum_j y_j} (N-k) kB'(Q) = 0 \qquad (6.31)$$

$$\alpha u'() [kB'(Q) - C'(q_S)] + \beta r'() \left[-\frac{\sum_{j \notin S} \sigma_j}{\sum_j y_j} C'(q_S) + kB'(Q) \frac{1 - N\sigma_S}{\sum_j y_j} \right] = 0 \qquad (6.32)$$

- For nodes that do not belong to the coalition S we know from section 6.2.4 that if $\sigma_j < (>) 1/N$ for $j \notin S$ then $B'(Q) > (<) C'(q_j)$.
- For the coalition, we obtain from (6.31) and (6.32) that if $\sigma_S < (>) 1/N$ then $kB'(Q) > (<) C'(q_S)$[12]. Since $B'(Q) > kB'(Q)$[13], the first order conditions imply that for nodes within the coalition $\sigma_S \leq 1/N$ and thus: $kB'(Q) \geq C'(q_S)$. To prove that inside the coalition $\sigma_S \leq 1/N$, assume to the contrary that $\sigma_S > 1/N$ and that $\sigma_j < 1/N$ for some nodes j outside the coalition. Inequalities (6.31) and (6.32) imply that $C'(q_j) < B'(Q) < kB'(Q) < C'(q_S)$ which contradicts the assumption of increasing and convex cooperation costs.

Inequalities (6.31) and (6.32) can be used to show the following proposition:

Proposition 4. (Coalition game) *In the symmetric coalition game for identical ERC preferences (type α/β), the grand coalition is stable if α/β is sufficiently small, i.e. nodes are interested enough in being close to the equal share.*

Note first, that within the grand coalition, the cooperation level satisfies the condition $NB'(Nq^*) = C'(q^*)$, independently of the ERC-types and nodes that receive the equal share.
If node i leaves the coalition ($k = N - 1$), then from the first order conditions we obtain:

$$(N-1) B' [(N-1) q_S + q_i] \geq C'(q_S) \geq C'(q_i) > B' [(N-1) q_S + q_i] \qquad (6.33)$$

Let us now look at the cooperation levels that would result if the ERC-type α/β goes to zero. In this case, nodes get more and more interested in getting their equal share, and

[12] Assuming that $\sigma_S > 1/N$ then $r'() < 0$ and (6.32) implies $kB'(Q) < C'(q_S)$
[13] For $k > 2$.

their cooperation levels will converge: in the limit $\tilde{q} = q_S = q_i$. However, in the limit, inequality (6.33) still must hold, *i.e.* $(N-1)\,B'(Nq) \geq C'(q)$.

In the limit the absolute payoff of a node leaving the coalition is smaller than within the grand coalition, whereas the relative payoff is the same, *i.e.* $NB'\left(\tilde{Q}\right) > C'(\tilde{q})$. Therefore, as long as α/β is small enough, the absolute payoff remains lower and the utility derived from the relative payoff is also smaller than in the grand coalition. Thus, no node has an incentive to leave the grand coalition if α/β is small enough.

Coalition of heterogeneous ERC-types

When nodes with heterogeneous ERC-types are allowed to take part in the coalition (S), those nodes that have the largest α_i/β_i will have the greatest interest to leave the coalition in order to obtain a larger absolute payoff. We will now concentrate on the extreme case in which nodes are either interested in their absolute payoff ($\beta_i = 0$) or in equity ($\alpha_i = 0$). The former are referred to as **A-nodes**, the latter as **B-nodes**. In total, there are N_a A-nodes and N_b B-nodes; k_a of these A-nodes and k_b B-nodes form the coalition. The cooperation levels are denoted by q_{as}, q_{bs} for nodes inside S, q_{an} and q_{bn} for nodes outside the coalition.
Let us first look at the behavior of B-nodes. Outside the coalition, any B-nodes can arrive at the equal share by choosing the average cooperation cost level. Thus,

$$C(q_{bn}) = \frac{1}{N_a + k_b}\left[k_a C(q_{as}) + k_b C(q_{bs}) + (N_a - k_a)C(q_{an})\right] \tag{6.34}$$

A B-node inside the coalition has no incentive to leave if it also receives the equal share:

$$C(q_{bs}) = \frac{1}{N_a - k_b}\left[k_a C(q_{as}) + (N_b - k_b)C(q_{bn}) + (N_a - k_a)C(q_{an})\right] \tag{6.35}$$

In equilibrium, all B-nodes choose the same cooperation level, and receive the equal share:

$$C(q_b) = \frac{1}{N_a}\left[k_a C(q_{as}) + (N_a - k_a)C(q_{an})\right] \tag{6.36}$$

A-nodes outside the coalition maximize their absolute payoff, $B(Q) - C(q_{an})$. The first order condition is given by:

$$B'(Q) = C'(q_{an}) \tag{6.37}$$

Within the coalition, the utility of a representative A-type-member is maximized by guaranteeing that the B-members get the equal share, i.e. $C(q_{bs})$. The first order condition for choosing q_{as} is given by:

$$B'(Q)\left[k_a + k_b\frac{\partial q_{bs}}{\partial q_{as}}\right] - C'(q_{as}) = B'(Q)k_a\left[1 + \frac{k_b}{N - k_b}\frac{C'(q_{as})}{C'(q_{bs})}\right] - C'(q_{as}) = 0 \tag{6.38}$$

By construction, for any given k_a and k_b, every B-node is indifferent to being either inside or outside the coalition. For a coalition to be stable, an A-node must not have an incentive to leave the coalition. In general, for any k_b there will be a certain number of A-nodes, k_a, that will join the coalition. We have multiple equilibria.
Inequalities (6.34) - (6.38) can be used to infer the following results:

Result 5. The larger the total number of equity-oriented nodes (N_b), the higher the incentives for A-nodes to join the coalition. Hence, for a given k_b, the number of cooperating A-nodes k_a increases in N_b.

Result 6. The more B-nodes join the coalition, the smaller the incentive for A-nodes to do so. In equilibrium, k_b and k_a are negatively correlated.

Result 7. The total cooperation level increases with the number of B-types outside the coalition. A joining B-node improves the payoffs only if it does not drive out an A-node.

The rationale of results 5 and 6 is the following: if an A-node enters the coalition and the coalition increases its cooperation efforts, B-nodes outside the coalition increase their cooperation activities as well and thereby additionally reward the entering node. If the number of such equity-oriented B-nodes outside the coalition gets larger, this external reward for joining a coalition increases and, therefore, the equilibrium coalition size increases. Analogously, if B-nodes join the coalition, fewer nodes outside the coalition reward the entering A-node by an increase of their cooperation activities. Hence, the incentives for A-nodes to enter the coalition decrease and the number of A-nodes that are inside the coalition in equilibrium gets smaller.
Result 7 reflects the fact that the more nodes cooperate, the higher the efficiency gains are and the closer the aggregate cooperation level is to the efficient one. The impact of A- and B-nodes on the decision of the coalition, however, differs in the following way: a joining A-node is interested in the absolute payoff and, consequently, the re-optimizing coalition increases its cooperation effort because the positive effect on one more node is now taken into account. A joining B-node, however, is not primarily interested in the absolute payoff, but in the equal share. Therefore, the coalition will not increase the total cooperation level that much because the B-node refrains from deviating from the cooperation level of non-cooperating nodes.
Consequently, the efficiency gains are larger if an A-node enters the coalition than if a B-node joins. Therefore, B-nodes are welcome inside a coalition only if their entering does not drive out an A-node.

6.2.5 Discussion: coalition formation process and the cooperation enforcement mechanism CORE

Self-interested, autonomous mobile nodes of an ad hoc network may interact "rationally" to gain and share benefits in stable (temporary) coalitions: this is to save costs by coordinating activities with other nodes of the network. For this purpose, each node determines the utility of its actions in a given environment by an individual utility function.
In section 6.2.1 we introduced a more sophisticated model in which not only self-centered preferences are taken into account to derive the individual payoff of an action but also relative information is used in order to find an extended set of possible equilibrium points. Results obtained with the proposed model are promising: in a dynamic network formed by nodes that follow the definition of utility given by the ERC theory, depending on the node types, it is possible to obtain stable coalitions of a relatively large size and under certain circumstances, even the grand coalition becomes feasible. Node types are determined by the two parameters α and β which represent the key factor of the coalition formation process.
We believe that the *reputation technique implemented in CORE* can be used as an effective mechanism to impose a specific identical ERC type for every node participating in a cooperative setting as an ad hoc network. Indeed, the reputation measure introduced in Chapter 4 is compliant with the incentive structure given by assumption (1) and (2). Cooperation is made attractive from an individual point of view because the cost of participating to the network operation is compensated with a higher reputation value, which is the pre-requisite for a node to establish a communication with other nodes in the network.
On the other hand, when the number of cooperating nodes increases, the cost for participation is compensated by a more connected network that in turn increases the benefit of cooperation. Now, if the two parameters α and β are represented as functions of the reputation r_{n_i} as defined in Chapter 4, then it is possible to enforce a particular value to the α/β ratio.
Specifically it is possible to dynamically adjust the α/β ratio in order to be compatible with **proposition 4**.
Thus, even the grand coalition is stable and every node of the network cooperates bearing the same costs and getting equal benefits by choosing a fair operating point in which no one deviates from the average cooperation level chosen by the coalition. The relation between α, β and r_{n_i} is indirectly proportional: the lower the reputation value (meaning that the past strategy selected by the node has been to reduce the cooperation level) the higher will be factor β and the lower will be factor α thus reducing the α/β ratio, and vice-versa.
The relation between the reputation value and the ERC type of a node becomes more complicated if we allow the presence of nodes with different ERC types: modelling a network that allows different ERC types is interesting when considering mobile nodes with different capabilities such as different battery power and different computational power, i.e. **heterogeneous networks**.
However, in order to provide a formal assessment of the efficiency of the reputation mech-

anism proposed in CORE it is necessary to evaluate the node model presented in the previous sections in a dynamic setting: the reputation value is computed based on the past strategies selected by the nodes of the network and have an influence on those nodes' future actions. Furthermore any variation on the strategy selection phase of a node has an impact on the strategies selected by neighboring nodes: solutions to the dynamic coalition formation process still have to be examined. We believe that the research we have conducted so far has given some interesting results and proposes a useful basis to study the coalition formation process of autonomous self-interested mobile nodes by means of reputation mechanisms which is, to the best of our knowledge, a rather unexplored domain. However, we think that it is possible to express the dynamic coalition formation process using a more elegant and simple methodology, which is a key requirement for studying dynamic games.
The relatively recent literature on the subject states that the models of coalition formation may be classified into two main categories: utility-based models, as it is largely favored by game theory, and complementary-based models. Up to now, most classic methods and protocols for the formation of stable coalitions among rational agents follow the utility-based approach and cover two main activities which may be interleaved: the generation of coalition structures, that is partitioning or covering the set of agents into coalitions, and the distribution of gained benefit among the participants to each of the coalitions.
The future research direction we will take is to prove that reputation mechanisms in general are compliant to the so called *Coalition Formation Algorithm*. Coalition formation algorithms are those mechanisms that provide a feasible solution to a cooperative game in coalitional structure: there are several solution concepts and we will focus on the so called *Kernel-oriented solutions* [104]. Kernel-oriented coalitions are the most suitable for our purpose because the related literature gives precise conditions for a coalition formation algorithm to be kernel-stable with a polynomial complexity, as opposed to other solution/algorithms that are only of theoretical relevance since they have exponential complexity.

6.3 Related work

Recently, much attention has been dedicated to game theoretical models for MANET in general and for cooperation enforcement mechanisms in particular and an increasing number of models have been presented to the research community. It is however out of the scope of this Chapter to propose an extensive state of the art of game theoretical models of cooperation in MANET, thus we will focus on some approaches that we deem related to our setting.
In an interesting approach presented in [105] the authors propose a game theoretical model in which energetic information is taken into account to describe the conflicting interaction between heterogeneous nodes involved in a forwarding game, i.e. a game in which nodes that belong to a path from a source to a destination have to collaboratively relay data packets. The authors study the properties of a well known strategy (generous-tit-for-

tat, G-TFT) and demonstrate that under the energy constraints imposed to the nodes, G-TFT promotes cooperation if every node of the network conforms to it. The model in [105] provides an accurate description of the energetic constraint of a node, which is the main reason for a selfish behavior, but provides only high-level guidelines towards the design of a cooperation enforcement mechanism based on the G-TFT strategy. The main difference between the work presented in this Chapter and the research conducted in [105] is that in our model we take into account a more realistic scenario where the observations made by a node on her neighbors can be affected by errors. The monitoring mechanism is indeed the key feature of a cooperation strategy based on the observation of the opponent's move (such as G-TFT) and we believe that a more accurate description of how these observations are made is fundamental.
Another interesting work towards the definition of a generic game theoretical framework to study cooperation in MANET has been presented in [107]. The authors propose a model that takes into account both the available energy to a node and the traffic generated and/or directed to that node and helps derive some interesting guidelines towards the definition of a cooperation mechanism. The authors not only analyze some existing cooperation mechanisms including CORE but also propose to use the tit-for-tat (TFT) as a cooperation strategy. Similarly to the work presented in [107], with the work presented in this Chapter we are able to accurately describe not only our cooperation strategy CORE but also a wide-range of history-based cooperation strategies (such as TFT). The performance analysis of the TFT strategy presented in [107] is extended in our work and we prove that CORE outperforms all other strategies when the imperfect monitoring assumption is made.
In the Annexe to this Chapter we propose an alternative model for the forwarding behavior of a node that is part of a specific network topology. By using our model, we are able to express the equilibrium forwarding strategy of a selfish node as a function of topology and routing (path length) information. We also propose a punishment mechanism that enforces a cooperative behavior among selfish nodes. Although the results obtained in the Annexe [5] provide a very useful description of the relation between routing, network topology and the cooperative behavior of a node, the proposed punishment mechanism is limited to a specific instance of the considered network topology and does not take into account the imperfect monitoring of the node behavior.
The research presented in [105], [107] in the Annexe of this Chapter ([5]) and in section 6.1, is based on non-cooperative game theory: even when multiple players are considered, the strategy selection phase is always driven by a node-centric perception of the game. As a result, the cooperation strategies obtained through the proposed models take into account only the payoffs obtained by a single player. Hence, in section 6.2 we propose an alternative approach based on a general model using cooperative game theory as a framework to study cooperation as a group initiative rather than a strategy adopted by single players. We believe that the "cooperative games" approach provides an appropriate way of describing the dynamics of group formation in MANET but needs further research in order to introduce in the model a more formal description of cooperation enforcement mechanism.

6.4 Summary

In this Chapter we followed two methodologies for the validation of the CORE mechanism. In the first method we use a discrete parallel simulation tool (Glomosim) enhanced with a simple selfishness model and we implement CORE as an add-on mechanism to the network layer component. Simulations are carried out for different scenarios where we vary parameters such as node mobility, traffic patterns, and percentage of selfish nodes in the network. Results are collected to infer the implications of using CORE in terms of energetic consumption and to assess the basic properties of the detection and punishment mechanisms implemented in CORE. Our results show that nodes that use CORE experience a significant reduction in power consumption as a consequence of the punishment mechanism. Furthermore, we were able to evaluate the efficiency of the punishment mechanism under different scenarios and study whether a mechanism to distribute local reputation information is necessary. We conclude that the gain in detection capabilities offered by the distribution of reputation ratings is not worth the increased sensibility of the network to malicious attackers that modify or fabricate false reputation information. Furthermore, the increased traffic overhead imposed by the distribution of reputation ratings would mitigate the benefits in terms of energy consumption that arise when communication services are withhold for selfish nodes.
We argue however that the results obtained through our simulation study (and those presented in other works available in the literature) can only give a proof-of-concept validation of CORE but are not representative of the incentive features of cooperation schemes. Thus, we introduce an alternative method that we use to validate the CORE mechanism defined as a cooperation strategy in game theoretical terms. Game theory emerged as an useful tool to model the interactions among self-interested users (that operate the nodes of the ad hoc network) that follow a strategy aiming at maximizing their benefits in terms of energetic consumption. In this Chapter we provide two different models based on non-cooperative and cooperative game theory that provide analytical evidence of the need for cooperation enforcement schemes when selfish entities are present in the network. Further, we show that the CORE mechanism can be translated into a cooperation strategy that, if adopted by all nodes of the network, lead to the optimal operating point in which all nodes cooperate. We also defined a simulation environment that borrow its concepts from evolutionary theory and show that CORE represents an evolutionary stable strategy. By taking into account a more realistic scenario in which communication errors and failures of the monitoring component are possible, we are able to show that CORE outperforms with respect to other cooperation strategies available in the literature.

Appendix 1. Proof of proposition 2

We have to show that $\delta(k) > 0$ for $k > N/2$.
Remember that in 6.25 the denominator of $\delta(k)$ is positive due to assumption 1. That is, $\delta(k) > 0$ if the numerator of 6.25 is positive. Remember also that we assumed $r(\frac{1}{N} - x) \leq r(\frac{1}{N} + x), \forall x \in \left[0, \frac{1}{N}\right]$. The numerator in 6.25 is positive if $r(cooperate) > r(defect)$. This is the case when equation 6.28 is satisfied.
Let's proceed by showing that for for $k < N/2 - 1$. It is possible to rewrite equation 6.28 as follows:

$$B(k+1)C(k)Nk + B(k)C(k+1)N(k+1-N) + C(k)C(k+1)\left[Nk - 2k(k+1)\right] < 0$$

or

$$B(k+1)C(k)\frac{N}{k+1} + B(k)C(k+1)N\frac{k+1-N}{k(k+1)} + C(k)C(k+1)\left(\frac{N}{k+1} - 2\right) < 0 \quad (6.39)$$

$$\left[B(k+1)C(k)\frac{N}{k+1} - B(k)C(k+1)\frac{N}{k}\right] + \left[\frac{NB(k)}{k} - C(k)\right]C(k+1)\left(2 - \frac{N}{k+1}\right) < 0 \quad (6.40)$$

Equation 6.40 can also be rewritten as:

$$\left[\frac{B(k+1)}{B(k)} - \frac{(k+1)C(k+1)}{kC(k)}\right] + (k+1)C(k+1)\left[\frac{1}{kC(k)} - \frac{1}{NB(k)}\right]\left(2 - \frac{N}{k+1}\right) < 0 \quad (6.41)$$

Now, from the monotonicity and concavity of $B()$ it follows that $\frac{B(k+1)}{(k+1)} < \frac{B(k)}{k}$.
Furthermore, the total cost of cooperation $kC(k)$ increases in k. Therefore:

$$\frac{B(k+1)}{B(k)} - \frac{(k+1)C(k+1)}{kC(k)} \leq \frac{k+1}{k} - 1 = \frac{1}{k}$$

Since it has also been assumed that payoffs are non negative, $B(k) \geq C(k)$. Thus:

$$(k+1)C(k+1)\left[\frac{1}{kC(k)} - \frac{1}{NB(k)}\right] \geq \frac{(k+1)C(k+1)}{kC(k)}\frac{N-k}{N} \geq \frac{N-k}{N} \quad (6.42)$$

We therefore obtain:

$$\left[\tfrac{B(k+1)}{B(k)} - \tfrac{(k+1)C(k+1)}{kC(k)}\right] + (k+1)C(k+1)\left[\tfrac{1}{kC(k)} - \tfrac{1}{NB(k)}\right]\left(2 - \tfrac{N}{k+1}\right) \leq$$
$$\leq \tfrac{1}{k} + \tfrac{N-k}{N}\left(2 - \tfrac{N}{k+1}\right) = \tfrac{N(k+1)+2(N-k)k(k+1)-(N-k)Nk}{Nk(k+1)}$$

The numerator equals: $-2k^3 + (3N-2)k^2 - N(N-3)k + N$which can be shown to be negative for $1 \leq k < \frac{N}{2} - 1$, as long as $N > 8$.
Hence for $N > 8$ we have that the general conditions for a Nash equilibrium of the ERC-PD game $\delta(k*-1) > 0$ are satisfied for $k > N/2$.
NOTE: the condition $N > 8$ can be removed if we assume that the total cost of cooperation increases more than the total benefits gained by defecting, i.e. :$\frac{(k+1)C(k+1)}{kC(k)} > \frac{NB(k+1)}{NB(k)}$.

Appendix 2. Proof of proposition 3

We have to show that in the cooperation game for ERC preferences, the Nash equilibrium is given by solving the expression $B'(Nq^*) - C'(q^*) = 0$. We then show that the Nash equilibrium point is symmetrical as long as at least one node draws utility from its absolute payoff ($\alpha_i > 0$).

Let us first study the two extreme cases, $\alpha_i = 0$ and $\beta_i = 0$, respectively.

- For $\beta_i = 0$, i.e. a player interested only in her absolute payoff, the first order condition (6.29 or 6.30) reduces to: $B'(Q) - C'(q_i) = 0$
- For $\alpha_i = 0$, the node is solely interested in getting the equal payoff share. Thus, it would choose q_i to satisfy: $NC(q_i) = \sum_j C(q_j)$. Furthermore, when $\alpha_i = 0$, condition 6.29 also reduces to $B'(Q) - C'(q_i) = 0$.
  Indeed, β_i and $r'()$ are positive by definition and the second summand reduces to
  $$\left[\frac{\sum_j y_j - y_i}{\sum_j {y_j}^2} B'(Q) - \frac{\sum_{i \neq j} y_j}{\sum_j {y_j}^2} C'(q_i) \right] = 0$$
  which can be simplified as: $B'(Q) - C'(q_i) = 0$.

For $\alpha_i, \beta_i \neq 0$ the chosen cooperation level is between the levels for those extreme cases: the first order condition must be satisfied for all nodes simultaneously. Since $r'(\sigma_i) = 0$ when $\sigma_i = \frac{1}{N}$ by assumption, it follows that there is a *symmetric equilibrium* where all nodes choose the same cooperation level, i.e. $\sigma_i = 1/N$ for all types α_i/β_i, for $i = 1, ..., N$. The resulting cooperation level q^* is given by solving the condition: $B'(Nq^*) - C'(q^*) = 0$.

Let us prove by contradiction (*reductio ad absurdum*) that there is an *asymmetric equilibrium*, i.e. some nodes receive less, and others more than the equal share. In this case, on the one hand, $\sigma_i < 1/N$ implies that $r'(\sigma_i) > 0$, so from equation 6.29, we obtain:

$$B'(Q) - C'(q_i) > 0 \tag{6.43}$$

On the other hand, for $\sigma_i > 1/N$ we have $r'(\sigma_i) < 0$, and therefore equation 6.30 implies:

$$B'(Q) - C'(q_i) < 0 \tag{6.44}$$

Inequalities 6.43 and 6.44 imply that a node which gets more than the equal share has larger marginal cooperation costs ($C'(q_i)$) than nodes that receive less, which contradicts the assumed payoff distribution.
Hence, only symmetric equilibrium exists. If $\alpha_i > 0$ for at least one node, we get $B'(Nq) - C'(q) = 0$ from equation 6.29.

Relevant publications

P. Michiardi and R. Molva. Report on a working session on security in wireless ad hoc networks. *Mobile Computing and Communication Review*, 6(4), 2002.

P. Michiardi and R. Molva. A game theoretical approach to evaluate cooperation enforcement mechanisms in mobile ad-hoc networks. In *Proceedings of the Workshop: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt'03)*, Sophia Antipolis, France, March 2003.

E. Altman, A. Kherani, P. Michiardi, and R. Molva. Non cooperative forwarding in ad hoc networks. Research Report RR-5116, INRIA Sophia Antipolis, February 2004.

P. Michiardi and R. Molva. Analysis of coalition formation and cooperation strategies in mobile ad hoc networks. *Elsevier - Ad hoc Networks Journal (Special Issue)*, 2004.

E. Altman, A. Kherani, P. Michiardi, and R. Molva. Non cooperative forwarding in ad hoc networks. In *Submitted to IFIP Networking 2005*, 2004.

E. Altman, A. Kherani, P. Michiardi, and R. Molva. Some game-theoretic problems in wireless ad hoc networks. In *Submitted to EURO-NGI 2005*, 2004.

Chapter 7

Key management and authentication in MANET

7.1 Introduction

In this Chapter we focus on key management and authentication services for *open ad hoc networks*. Security solutions for open ad hoc networks are particularly difficult to provide due to the conjunction of several factors. The **lack of a-priori trust** among the nodes of the network that cannot be assumed to be part of any shared organization renders classical security paradigms based on pre-established trust among the parties not applicable. Furthermore, security solutions based on dedicated components with pre-defined roles such as trusted third parties and key servers are not compatible with the basic definition of open ad hoc networks whereby no component has a pre-assigned role and a **networking infrastructure is not available**.
Research on ad hoc network security initially focused on routing protocols since these were deemed the most critical part of network control. However, a close analysis of routing security works (refer to Chapter 3) reveals that the majority of the requirements addressed by the solutions and the suggested mechanisms are not new nor specific to ad hoc networks, apart from a fundamental problem that was often left aside by routing security solutions, that is, key management with no a-priori trust and lack of infrastructure.
In Chapter 3 we propose a detailed survey on secure routing and key management techniques available in the literature that address a variety of ad hoc networking configurations, including open ad hoc networks. The key management approaches that we evaluated in Chapter 3 try to answer the hard question of how to establish security associations with no a-priori knowledge, no a-priori trust and lack of infrastructure. Several original key management schemes based on advanced cryptographic constructs like threshold cryptography and identity based (ID-based) cryptography have been suggested in the literature but they all fall short of meeting the ultimate goal of building a keying infrastructure "from scratch" since they all involve an initial key set-up phase.

In this Chapter we propose an authentication scheme called IDHC and based on an original concept that combines a simple form of identity based cryptography with the Lamport's keyed hash chain method [55]. In our solution [72], users are able to locally generate a chain of authentication material that we call *authentication tickets* using as seed a secret information (that we call a *master authentication ticket*) delivered by a key distribution center (KDC). By removing the reliance on a public key infrastructure, our scheme is particularly suitable for networks with multiple dynamic sources whereas other authentication schemes available in the literature suffer from the limitations imposed by certificate management requirements. In this Chapter, we also describe an interesting application of our scheme: IDHC can be used as a basis to provide a lightweight key distribution mechanism that offers peer authentication services particularly suitable for an infrastructure-less ad hoc network. Indeed, in the proposed solution, there is no need for a network infrastructure and the security bootstrap phase is lightweight: the key distribution center is involved neither in networking operations nor in any further security operations beyond the bootstrap phase.

The reminder of the Chapter is organized as follows: first we provide background concepts needed to understand traditional identity based cryptosystems, then we present the IDHC authentication scheme and focus on the ID-based cryptographic technique used to generate the authentication material. A detailed assessment of our scheme's security properties is provided by analyzing the robustness of IDHC with respect to threats carried out by malicious users in the system. We then focus on the performance analysis of the IDHC scheme both in terms of computational power and storage requirements.
Finally, we describe a possible application of the IDHC scheme for the provision of key-distribution and message authentication services in the challenging scenario offered by ad hoc networks, and explain how IDHC can be used as a substitute for key distribution in an existing secure routing mechanism.

7.2 Background and assumptions

In traditional certificate-based public key cryptosystems, a user's public key is certified with a certificate issued by a certification authority (CA). Any participant who wants to use a public key must first verify the corresponding certificate to check the validity of the public key. When many CAs are involved between two users, trust relationships between those CAs also need to be verified. A public key infrastructure (PKI) can be used to manage the trust relationship between entities in a hierarchical manner. In certificate-based schemes, key revocation is a big issue and requires a large amount of storage and computing. As a consequence, certificate-based public key cryptosystems require a large amount of storage and computing time to verify, and revoke certificates.

The idea of identity based cryptosystem was proposed by Shamir [103] with the original motivation of simplifying certificate management in email systems, thus avoiding the high cost of the public-key management and signature authentication in cryptosystems relying on a public key infrastructure.
ID-based systems *avoid the explicit authentication of public keys* using public-key certificates by offering a cryptosystem wherein *the identity of a user plays the role of his public key*: each entity's public key can be defined by an *arbitrary string*, *i.e.* users may use some well-known information such as email addresses, IP addresses or any other unique identifier as their public key. From a user's identity (which is publicly known and in a standardized form), a TTP (e.g. a key distribution center) computes the corresponding private key and securely transmits it to the user.
The original goal of Shamir was only partially achieved by a few solutions until Boneh-Franklin [21] proposed the first practical identity based encryption scheme based on Weil Pairings on elliptic curves. Since then, several other identity based cryptography schemes [29, 34, 35, 43, 83] have been proposed. The common denominator of ID-based cryptosystems available in the literature is that they provide the same services offered by a PKI without the management costs of a PKI: by contacting a key distribution center (KDC), a user receives a secret key corresponding to the public key derived from the user's identity.
The main issue of such systems is that the KDC possesses all secret keys corresponding to the users of the system, i.e. all proposed schemes have inherent key-escrow properties. This limitation has been addressed, for example, in [35].
Nevertheless, a practical identity based variant built on top of RSA has remained elusive for the simple reason that an RSA modulus n (a product of two large primes) cannot be safely shared among multiple users due to the well known *common modulus attack*.

In this Chapter, we propose an identity based scheme developed atop some widely known RSA-primitives: IDHC blends the features of identity based cryptography and hash-chain based authentication schemes while at the same time offering security comparable to that of the basic RSA cryptosystem.
Using a simple form of RSA encryption, the KDC generates a secret from the user's identity. The user can then use this secret to generate a chain of authentication tickets as with Lamport's keyed hash chain scheme. Authentication tickets can be thought of as symmetric keying material that can be used to generate keyed message authentication codes (MAC). Each party can thus verify the MAC of a message based on the identity of the sender.
As with the general identity based cryptosystem and PKI model, the user who wishes to obtain a master authentication ticket has to authenticate himself to the KDC. Furthermore, users' identities have to be unique and publicly available.
We assume that the KDC cannot be corrupted and that is robust against failures. A typical method to improve the KDC availability and robustness would be to distribute the KDC by using secret sharing techniques: we believe that this direction needs to be explored as part of future research.

7.2.1 Ad hoc networking assumptions

We assume that network links are bidirectional; that is, if a node A is in transmission range of some node B, then B is in transmission range of A. When nodes use equal power levels and identical coding for all link-layer frames (packets) transmitted, and when all nodes use omnidirectional antennas, links are generally bidirectional. Furthermore, many wireless Medium Access Control protocols require bidirectional links, as they exchange of several link-layer frames between a source and destination to help avoid collisions. Medium Access Control protocols are also vulnerable to attacks. For example, in IEEE 802.11, an attacker can paralyze nodes in its neighborhood by sending Clear-To-Send (CTS) frames periodically, setting the "Duration" field of each frame equal to the interval between such frames. Less sophisticated Medium Access Control protocols, such as ALOHA and Slotted ALOHA [3], are not vulnerable to such attacks but have lower efficiency.
In our work, we disregard attacks on Medium Access Control protocols. We assume that the wireless network may drop, corrupt, reorder, or duplicate packets in transmission, for example due to interference effects or packet retransmissions. We further assume that the network includes a mechanism for transmission error detection in packets, such as a checksum or CRC code, and that nodes discard received packets for which this mechanism indicates that the packet has been corrupted. This mechanism, however, is not intended to replace cryptographic integrity checks on received packets.
To allow nodes to choose appropriate transmission time intervals, as described in section 7.3, we assume that each node in the network can estimate the end-to-end transmission time to any other node in the network. This value can be chosen adaptively, and can represent a pessimistic estimate. When this time is chosen to be too large, protocol responsiveness is reduced; when it is chosen to be too small, authentic packets may be rejected as forged, but forged packets are not accepted.
Finally, we assume the network to be operated by nodes that use the dynamic source routing (DSR) protocol.

7.2.2 Node assumptions

The resources of different ad hoc network nodes may vary greatly, from nodes with very minimal computational resources to resource-rich nodes perhaps equivalent in functionality to high-performance workstations. To make our results as general as possible, we design our protocol for nodes with minimal resources.
Most previous work on security for ad hoc networks relies on asymmetric cryptography such as digital signatures. However, computing such signatures on resource-constrained nodes is quite expensive, and we assume that some nodes in the ad hoc network may be so constrained.
We assume that all nodes in the ad hoc network have synchronized clocks within a maximum bound between any two nodes' clocks of Δ. The value of the parameter Δ must be known by all nodes in the network. Though this time synchronization can be maintained

with off-the-shelf hardware based on LORAN-C, WWVB (see [76]), or GPS [32], it is currently not a common part of ad hoc network nodes, and the time synchronization signal itself may be subject to attack. Stable time sources such as microcomputer-compensated crystal oscillators [15] can provide sub-second accuracy for several months. Synchronization to within a few seconds per month can also be achieved using temperature-compensated crystal oscillators. Finally, if normal crystal oscillators are used, Δ can be chosen to be as large as necessary, though a corresponding reduction in protocol responsiveness will result.

7.2.3 Security assumptions

In the reminder of this Chapter, we assume the presence of attackers aiming at disrupting the correct network functioning by perpetrating the threats extensively discussed in Chapter 2, with the exception of the lack of cooperation attack.
Further, we also consider attackers that try to break the cryptosystem proposed in the sequel of this Chapter: these threats are described and analyzed in section 7.4.

7.3 The IDHC scheme

This section presents the IDHC scheme. In our solution, users contact a key distribution center (KDC) and receive a secret called the *master authentication ticket* M which is tightly bound to the users' identity (ID). M is used as a seed to generate a *chain of authentication tickets* as with Lamport's keyed hash chain scheme.
Further, we propose to use the IDHC scheme for a message authentication protocol designed for loosely time-synchronized users. As opposed to other authentication schemes available in the literature, our solution does not rely on any public key infrastructure and, like any ID-based cryptosystem, it does not require public key certificates.

We now describe the stages of the IDHC scheme in this order: key distribution center (KDC) setup and sender setup; we then focus on the sender transmission of authenticated messages, and receiver authentication of messages.

7.3.1 KDC setup

The basic idea behind our identity based hash chain scheme is the use of a single common RSA key pair for all users within a system. Only the public key of this keypair is assumed to be publicly known and the private key is held by the KDC.
As in RSA, the proposed cryptosystem uses computations in Z_n, where n is the product

of two distinct odd primes p and q. For such an integer n, note that $\phi(n) = (p-1)(q-1)$. The formal description of the KDC bootstrap phase is as follows.

Key-distribution Center (KDC) setup:

1. KDC generates: two large random odd primes p and q
2. KDC computes: $n = p \cdot q \rightarrow$ RSA-like modulus (common to all users)
3. KDC selects: **small** $e \in \mathrm{Z}^*_{\phi(n)}$, $k \in \mathrm{N} \rightarrow$ Public values e and k (common to all users)
4. KDC computes: $d = e^{-k} \bmod \phi(n) \rightarrow$ Master Secret Key

Figure 7.1: KDC bootstrap phase.

As sketched in figure 7.1, the KDC uses the RSA modulus to generate a master secret key d that corresponds to a public exponent e^k: this operation is equivalent to the legacy RSA key-pair generation.

We stress that our scheme is not exposed to the well known common modulus attack whereby anyone, based on one's knowledge of a single key-pair, can simply factor the modulus and compute other users' private keys. In the present context, the secret key d is only known to the KDC and kept secret from the users of the system. Our scheme relies on a single keypair of which the private key is only known by the KDC. Further discussion on the common modulus attack is presented in section 7.4.

Master secret key generation

As depicted in figure 7.1, the master secret key used by the KDC to generate master authentication tickets for the users of the system is of the form: $d = e^{-k} \bmod \phi(n)$.
Since the secret key d is generated only once during the system initialization and used to process all user requests, the KDC can afford to run a complex algorithm to generate d. However, an efficient way for calculating d can be derived based on the following observation:

$$d = e^{-k} \bmod \phi(n) = \left(e^{-1}\right)^k \bmod \phi(n)$$

The inverse of the public exponent e can be easily calculated, and then it is sufficient to apply the square and multiply algorithm to compute the exponentiation (see [106] page 170).

7.3.2 Sender setup

In order to produce authenticated packets, the sender needs to contact the KDC that is in charge of issuing a master authentication ticket (refer to figure 7.3). Upon verification of the sender identity ID, the KDC generates and **securely** distributes to the sender the following master authentication ticket:

$$M = (H(ID))^d \bmod n \tag{7.1}$$

where the function $H()$ is a one-way collision resistant function such as the popular `MD5` hash function [97], applied to the user identity ID. Expression (7.1) can be thought of as the KDC's digital signature over the sender identity ID.

Sender setup:

1. Retrieve the public values n, e, k
2. Contact KDC to obtain the master authentication ticket M
3. Generate k time-dependent authentication tickets T_k

Figure 7.2: Sender setup phase.

Next, the sender divides the time into uniform intervals of duration τ_{int}. Time interval 1 starts at time τ_1, time interval 2 at time $\tau_2 = \tau_1 + \tau_{int}$, etc. The sender computes authentication tickets T_i by subsequently encrypting the master authentication ticket M using the public exponent e as shown in figure 7.4. Each authentication ticket is then assigned to a time interval starting with time interval τ_1 and ticket T_k, continuing with time interval τ_2 and ticket T_{k-1} and so on.

The one-way authentication ticket chain is used in the reverse order of generation, so any value of a time interval can be used to derive values of previous time intervals. The sender uses the length k of the one-way chain as obtained from the KDC: this length limits the maximum transmission duration before a new one-way authentication ticket chain must be created[1].

[1] In this Chapter we assume that chains are sufficiently long for the duration of communication.

Distribution of Master Authentication Ticket:

Sender $\rightarrow$ **KDC** : ID
KDC generates: $H(ID) = C = h^k(M)$
KDC $\rightarrow$ **Sender** : $M = C^d \bmod n$ (*securely*)

Figure 7.3: Distribution of Master authentication ticket.

Authentication tickets generation (Sender):
Generation order: $\uparrow$ - Releasing order $\downarrow$

$T_k = M^{e^k} \bmod n$

$T_{k-1} = M^{e^{k-1}} \bmod n$

...

$T_{k-i} = M^{e^{k-i}} \bmod n$, with $i \leq k$

...

$T_1 = M^e \bmod n$

Figure 7.4: Authentication ticket generation.

7.3.3 Transmission of authenticated messages

In this section we outline the main ideas behind the message authentication scheme based on IDHC. Message authentication requires a source of asymmetry, such that the receivers can only verify the authentication information, but not generate valid authentication information. We assume that receivers are all loosely time synchronized with the sender: up to some time synchronization error Δ, all parties agree on the current time.

Here is a sketch of the basic approach:

- Subsequently to the setup phase, the sender assigns each authentication ticket sequentially to the selected time intervals (one ticket per time interval). Note that using the one-way chain of authentication tickets in reverse order of generation renders computationally infeasible for an attacker to forge authentication tickets.

Furthermore, any values of a time interval can be used to derive values of previous time intervals.

- The sender generates a message authentication code (MAC) and attaches it to each packet. The MAC is computed over the contents of the packet that needs to be transmitted. For each packet, the sender determines the time interval and uses the corresponding value from the one-way chain of authentication tickets as a cryptographic key to compute the MAC (see [13] for details). Along with the packet, the sender also sends the authentication ticket it used to generate the MAC in the previous time interval and its unique identifier (ID).

- Upon receipt of a packet, the receiver verifies the authentication ticket contained in the packet and uses it to check the correctness of the MAC of the buffered packet that corresponds to the time interval of the authentication ticket. If the MAC is correct, the receiver accepts the packet.

Each authentication ticket generated using the procedure depicted in figure 7.4 corresponds to a time interval. Every time a sender transmits a message, it appends a MAC to the message, using the authentication ticket corresponding to the current time interval as the key to compute the MAC. The authentication ticket for time interval τ_i remains secret until it is revealed in the packet corresponding to time interval τ_{i+1}.
Figure 7.5 depicts the time intervals and some sample packets transmitted by the sender along time.
Formally, a generic packet sent at time interval τ_i is of the form:

$$P_i = \left\{m_i, MAC_{T_{k-i}}(m_i), T_{k-(i-1)}, ID_{SENDER}\right\} \tag{7.2}$$

where:

- m_i is the data message that the sender needs to transmit,

- $MAC_{T_{k-i}}(m_i)$ is the message authentication code over message m_i generated using the authentication ticket T_{k-i} as the key,

- $T_{k-i} = M^{e^{k-i}} \bmod n$ is the authentication ticket for time interval τ_i derived from the master authentication ticket M as depicted in figure 7.4,

- $T_{k-(i-1)} = M^{e^{k-i-1}} \bmod n$ is the disclosed authentication ticket for time interval τ_{i-1},

- ID_{SENDER} is the unique identifier of the sender.

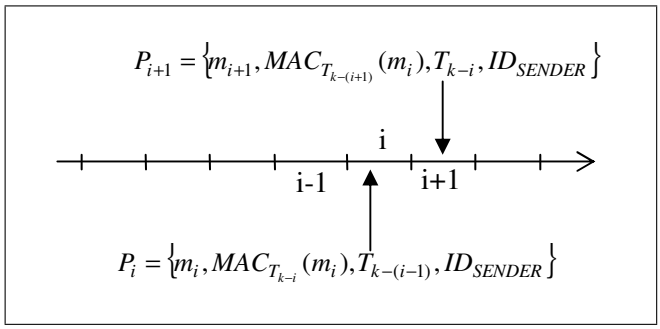


Figure 7.5: Sending authenticated messages.

7.3.4 Verification of message authentication information at the receiver

Upon reception of packet P_{i+1} the receiver extracts the authentication ticket T_{k-i} that can be used to authenticate the previously received packet P_i. First, the receiver has to verify that the authentication ticket T_{k-i} corresponds to the identity ID_{SENDER} specified in the packet P_i. To that effect, the receiver only has to perform i exponentiations with e that is a small exponent:

$$\begin{aligned}
&(T_{k-i})^{e^i} \bmod n = \left(M^{e^{k-i}}\right)^{e^i} \bmod n = \\
&\left(\left(C^d\right)^{e^{k-i}}\right)^{e^i} \bmod n = \\
&\left(\left(C^{e^{-k}}\right)^{e^{k-i}}\right)^{e^i} \bmod n = \\
&= C^{e^{-k}e^{k-i}e^i} \bmod n = C = H(ID_{SENDER})
\end{aligned} \tag{7.3}$$

If $H(ID_{SENDER})$ obtained in expression (7.3) equals the hash function applied to the ID_{SENDER} specified in the packet P_i, then the authentication ticket is valid and it can be used as a key to verify the MAC for packet P_i.
When a sender discloses an authentication ticket, all parties potentially have access to that ticket and can create a bogus packet and forge a MAC. Therefore, as packets arrive, the receiver must also verify that their MACs are based on safe keys, i.e. a key that is only known by the sender, by checking that the time interval the sender could be in (in the example above, τ_{i+1}) is greater than the time interval corresponding to the disclosed authentication ticket (in the example above, τ_i). Receivers must discard any packet that is not safe, because it may have been forged.

7.4 Security Analysis

In this section we propose a security analysis of the IDHC scheme by assuming that an attacker (internal or external) trying to break the cryptosystem is actually trying to determine the secret master key safely guarded by the key distribution center (KDC) by using disclosed authentication tickets collected over time or by performing a known-plaintext attack.
Further, we consider an attacker who tries to gather a valid authentication ticket by submitting bogus identity information to the KDC or to generate valid authentication tickets from past authentication tickets. In section 7.4.5 we discuss about the choice of the system parameter k that avoids duplicate authentication ticket generation and prevents the re-use of past authentication tickets by an attacker.

7.4.1 Common modulus attack

In a naive setting of RSA-based cryptosystem, to avoid generating a different $n = p \cdot q$ modulus for each user, one could envision to fix n once and for all. The same n could then be used by all users. A trusted central authority could provide user i with a unique pair $< e_i, d_i >$ from which user i would form a public key $< n, e_i >$ and a secret key $< n, d_i >$. However, an observation due to Simmons shows that an RSA modulus should never be used by more than one entity. Indeed, at first glance, a scheme using a common modulus may seem to work: a ciphertext $C = M^{e_A} \bmod n$ intended for Alice cannot be decrypted by Bob, since Bob does not possess d_A. However, this is incorrect, and the resulting system is insecure: Bob can use his own exponents $< e_B, d_B >$ to factor the modulus n. Once n is factored Bob can recover Alice's private key d_A from her public key e_A. The demonstration of how Bob can find the factorization of the common modulus n can be found in [19].
In the IDHC system proposed in this Chapter, however, the common modulus attack is prevented even if all entities of the systems share a common modulus. By carefully analyzing the KDC setup phase (section 7.3.1) and the sender setup phase (section 7.3.2), it is possible to observe that as compared to the typical common modulus attack scenario described above, no (secret) keying material is delivered to the users. Instead, the common modulus n is used to generate a master secret key d that is securely kept by the KDC. The key d is used to encrypt the hashed identity of the user requesting for a master authentication ticket M, unlike with the common modulus attack, and the secret M provided to each user is not a private key but the result of an encryption with the private key d. Thus, the attack detailed in [19] can not be perpetrated against the IDHC system.

7.4.2 Impersonation through blinding

Suppose now that an attacker wishes to impersonate a party known under the identity ID by maliciously gaining access to the master authentication ticket M for identity ID. The attacker knows that the master ticket M is computed by the KDC by encrypting the hashed identity $C = H(ID)$. Now, the attacker randomly chooses g and computes $C^* = g^{e^k} C$. Subsequently, the attacker receives the following master authentication ticket from the KDC: $M^* = (C^*)^d \bmod n$. Based on the definition of C^* we have: $M^* = (g^{e^k} C)^d \bmod n = g^{e^k \cdot d} C^d \bmod n = g \cdot M$. Thus M can be retrieved using $M = \frac{M^*}{g}$.
A simple observation however shows the infeasibility of this attack: finding a bogus identifier ID^* such as $H(ID^*) = g^e C = g^e H(ID)$ requires inverting the one-way hash function $H()$, which is (computationally) infeasible.
As a rule, the study of the impersonation attack suggests to perform the initial authentication of users applying for a master authentication ticket by requesting the full identifier ID of the user rather than a hashed value of the identifier.

7.4.3 Forging authentication tickets

In this section we suppose that an attacker wishes to forge an authentication ticket by using a previously revealed valid authentication ticket.
Suppose that a legitimate sender discloses the authentication ticket: $T_k = M_{ID}^{e^k} \bmod n$, where M_{ID} is the master authentication ticket for the identifier ID. It is straightforward to show that finding M_{ID} is as hard as breaking the RSA cryptosystem. However, we want to show that also forging the authentication ticket T_{k-1} by an attacker holding T_k is as hard as breaking the RSA system.
Since $T_{k-1} = M_{ID_{SENDER}}^{e^{k-1}} \bmod n = \left(M_{ID_{SENDER}}^{e^k}\right)^{e^{-1}} \bmod n$, in order to derive T_{k-1} from T_k, the attacker would have to solve the following equation: $T'_{k-1} = \sqrt[e]{T_k} \bmod n$, which is again equivalent to breaking the RSA system.
On the other hand, suppose an attacker with identity ID^* holds the master authentication ticket $M^* = (C^*)^d \bmod n$. The attacker also knows $C = H(ID)$, where ID indicates the identity of a legitimate user. Let $x = \frac{C^*}{C}$.
Now,

$$T_{k-1} = M_{ID}^{e^{k-1}} \bmod n = \left(C^d\right)^{e^{k-1}} \bmod n = \left(\left(\frac{C^*}{x}\right)^d\right)^{e^{k-1}} \bmod n = \left(\frac{M^*}{x^d}\right)^{e^{k-1}} \bmod n$$

but it is evident that the attacker cannot generate the value x^d that is needed to forge the authentication ticket T_{k-1}. Indeed: $\left(x^d\right)^{e^{k-1}} \bmod n = x^{de^{k-1}} \bmod n = x^{e^{-1}} \bmod n = \sqrt[e]{x} \bmod n$ where $d \cdot e^k = 1 \bmod \phi(n)$.
Again, solving the e-th root of x modulo n is as hard as breaking the RSA system.

7.4.4 Known-plaintext attack

We want now to examine another kind of elementary attack that could be perpetrated by an attacker wishing to determine the secret key d used by the KDC to generate master authentication tickets. The known-plaintext attack is a form of cryptanalysis where the attacker knows both the plaintext and the associated ciphertext. In the IDHC context, an attacker is able to determine the plaintext associated to every master authentication ticket since the KDC generates it from the public identity of the requesting user. However, the master authentication ticket is delivered in a secure way to the corresponding user, which in turn only reveals authentication tickets generated as in section 7.3.2. An attacker needs to know the secret key d in order to extract the master authentication ticket.

It is worth mentioning that the operation carried out by the KDC when delivering master authentication tickets to the users of the system is comparable to the generation of a (RSA) digital signature on a message (in our case, the hashed identity of a user) using the secret key d. Thus, it is possible to affirm that the IDHC system is as secure as the (RSA) digital signature scheme.

7.4.5 Choice of system parameter k

The parameter k determines the number of authentication tickets that can be generated by the user. However, k cannot take on any arbitrary value. A simple observation is sufficient to characterize the choice of k. By construction (see figure 7.4) an authentication ticket takes the following expression: $T_k = M^{e^k} \bmod n$. Now, we would have to find an integer $m \neq k$, such as:

$$M^{e^k} \bmod n = M^{e^m} \bmod n \text{ with } m \neq k \tag{7.4}$$

It is trivial to show that $m = k \bmod \phi(\phi(n))$, so as long as $k < \phi(\phi(n))$ the following implication holds:

$$M^{e^k} \bmod n = M^{e^m} \bmod n \Rightarrow m = k \tag{7.5}$$

By choosing $k < \phi(\phi(n))$ we avoid duplicate authentication tickets.

7.5 Performance analysis

When plain RSA is used for encryption, the public encryption exponent e is typically a small integer with only a few 1-bits. One example is the popular OpenSSL toolkit [2] that uses 65537 as the default public key value for RSA certificates. Encryption with such small exponents can be accelerated with specialized algorithms for modular exponentiation.

In our setting, the secret/public key generation phase (performed by the KDC) is equivalent to an RSA key generation while the master authentication ticket generation (performed by the key distribution server) can be considered equivalent to a RSA signature over the public identity of the requesting node. However, it is critical to evaluate the computation power requirements that a user (a mobile node of the network) has to satisfy in order to generate authentication tickets.
Even by choosing a relatively small exponent e, single nodes have to deal with the generation of k authentication tickets, an operation that can be compared to k RSA encryptions. Finally, the verification performed at the receivers is equivalent to an RSA-signature verification.

We ran some simple tests to assess the cost of IDHC authentication ticket generation/verification for public keys derived from IP addresses. The encryption and verification was tested using OpenSSL cross-compiled for an IPAQ 38xx series with a 400Mhz X-Scale/Arm processor and Linux Familiar operating system [1], [2]. Results are presented in table 7.1.

RSA key length	*Ticket generation [ticket/s]*	*Ticket verification [ticket/s]*
512 bits	121.48	1465.8
1024 bits	26.87	524.75
2048 bits	4.61	157.3
4096 bits	0.7	47.58

Table 7.1: Performance comparison of IDHC Ticket generation/verification with different key-lengths.

Taking as an example an RSA key length of 512 bits, a node can generate 121.48 authentication tickets per second while a potential receiver is able to authenticate 1475.8 packets per second. Results gain more relevance when the IDHC scheme is applied to a specific scenario.

In section 7.6 we present a potential application of the IDHC scheme to secure reactive routing protocols for ad hoc networks and assess the viability of the IDHC scheme.

7.5.1 Storage requirements

If computational power requirements are satisfied by the IDHC authentication scheme, also storage requirements can be a potential issue that has to be taken into account when designing an authentication scheme for mobile devices which have a limited storage capacity. Based on a reference implementation of RSA available in the OpenSSL package, it is straightforward to evaluate the space requirements for a single authentication ticket that needs to be stored in every node of the network. Indeed, the block size of a cipher

text (i.e. an authentication ticket) generated as depicted in figure 7.4 is equal to the key length. For example, by taking a key length of 512-bit, also the authentication ticket would be 512-bit long. To be more precise: the ID used to generate the master authentication ticket consists of 32-bit, since we used as unique identifiers IPv4 addresses. The popular hash function `MD5` applied to the ID results in a 128-bit message digest. The master authentication ticket generated by the KDC for identity ID will be as long as the key length used to generate it: for example using a 512-bit key, the ID's authentication tickets generated by the mobile nodes would also be 512-bit long.

Thus, space requirements for every mobile node is equal to: $k \cdot \text{key\_length}$, where k is the number of elements of the hash chain, i.e. the total number of authentication ticket that need to be generated, as imposed by the system parameter k.

7.6 IDHC for secure routing in mobile ad hoc networks

A particularly challenging requirement for peer authentication is raised by secure routing protocols in the context of mobile ad hoc networking. In this section we describe a lightweight key distribution mechanism based on the IDHC scheme that offers authentication services to an infrastructure-less ad hoc network.
The main features of our solution range from relaxed networking infrastructure requirements to a lightweight security bootstrap phase which is node-oriented as opposed to network-oriented approaches. Indeed, the literature offers key distribution schemes for ad hoc networks in which **all nodes** have to be initialized at the same time of network creation in order to distribute all possible pair-wise authentication material [12]. In our case the initialization phase is only performed for the node joining the network, that might already exist. Furthermore, in our solution there is no need for an organizational structure among peers or between peers and the key distribution center (KDC) which in turn is not involved in any networking operations. Moreover, the KDC is not involved in any further security operations beyond the bootstrap phase.

7.6.1 Authentication ticket distribution

Figure 7.6 shows a typical scenario in which one (or more) KDC offers both naming and authentication services.
We assume the KDC to be initialized as described in section 7.3.1.
During the security bootstrap phase, prior to joining the ad hoc network, a mobile node (for example node N_{ID_9}) that needs authentication services has to contact the closest KDC and provide some initial authentication information. This initial authentication information can take the form of:

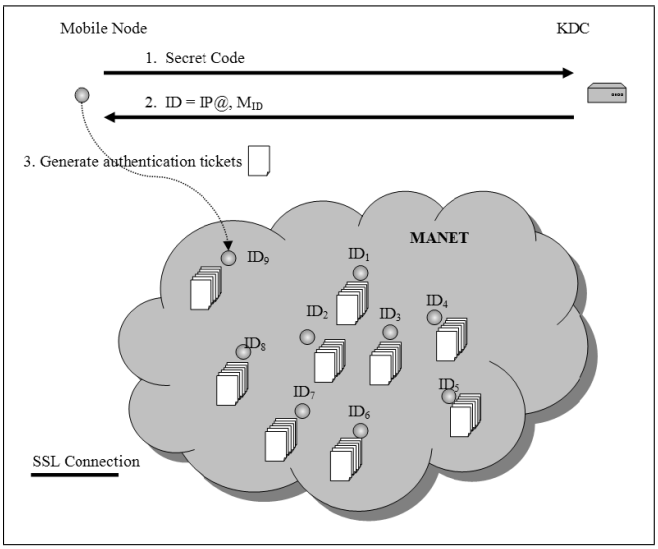


Figure 7.6: Application of IDHC for naming and key management in open ad hoc networks.

- a secret code printed on a **prepaid card** that is delivered by a (automatic) teller,

- a secret code printed on tickets delivered at the entrance of confined areas like shopping malls, airports, conference sites

By providing the initial authentication information to the KDC, the mobile node **securely** receives a unique identifier (that in our case is represented by an IP address for the ad hoc network) and a master authentication ticket M_{ID} generated by the KDC as explained in section 7.3.2 and in figure 7.3. Using IP addresses as node identities allows exploiting existing addressing mechanisms to provide network-wide known and unique node identifiers. However, one might consider the scenario in which nodes' IP addresses might constantly change due for example to hand overs between different ad hoc networks. Moreover, self-organized addressing schemes might be preferable to the addressing scheme proposed in this section. In these situations, an additional overhead for node re-authentication and for the generation of new authentication tickets have to be taken into account. We will address these issues in our future research, were we also plan to use cross-layer information (e.g. pseudonyms used in peer-to-peer applications) to provide a suitable naming service.

Initial user authentication needed to obtain the private cryptographic material used in subsequent authenticated message exchange is critical for various security systems, ranging from PKI to ID-based systems. As a typical example, commercial PKI systems extend the role of the certification authority by introducing the registration authority component: users are demanded to show "real word" credentials such as a passport or

an ID in order to be entitled to perform a certificate request procedure[2]. This is done to prevent **bogus certificate generation** and **identity spoofing**, a problem that affects on-line PKI services such as the one offered by Verisign. Indeed, in such systems the relaxed requirements in terms of initial user authentication, which is performed through an e-mail address validity check, allows an attacker to generate PKI certificates for bogus identities or to spoof identities by tampering with the e-mail systems of their victims.
The IDHC system is robust against impersonation through spoofing, as explained in section 7.4: furthermore, by introducing a monetary overhead prior to the obtention of a master authentication ticket, we make bogus authentication ticket generation an expensive operation.

As depicted in figure 7.6, a mobile node that securely obtained the identifier ID and a master authentication ticket M_{ID}, can proceed with the generation of message authentication tickets as explained in figure 7.4: for the purpose of this Chapter we assume that authentication ticket chains are sufficiently long for the whole duration of the communication in the ad hoc network.
Now that every node joining the network is loaded with a sufficient number of authentication tickets, we describe how this cryptographic material can be used for DSR routing message authentication. The reader should refer to [48] for an authoritative description of the dynamic source routing (DSR) protocol.

7.6.2 Variant of ARIADNE based on the IDHC scheme

A variety of secure routing solutions for ad hoc networks have been proposed in the literature (see for example, [12] chapter 12). In spite of the large number of solutions, ad hoc routing does not seem to raise any new security requirement with respect to routing in classical networks, apart from key management problems that have been often left aside by current solutions available in the literature. Key management approaches try to answer the hard question of how to establish security associations with no a-priori knowledge, no a-priori trust and lack of infrastructure. Several original key management schemes based on advanced cryptographic constructs have been suggested in the literature (see [12] for a literature survey) but they all fall short of meeting the ultimate goal of building a keying infrastructure "from scratch" since they all involve a complex (and often unrealistic) key set-up phase. In section 7.6.1 we described an original approach based on the IDHC scheme that provides authentication material to the nodes of an ad hoc network in a lightweight way. We now propose to use the IDHC scheme coupled with the ARIADNE secure routing mechanism: the basic idea is to use IDHC tickets to compute message authentication codes (MAC) in order to provide *routing* message authentication, as explained in section 7.3.3 and in section 7.3.4.
While we suggest the reader to refer to [88] for an authoritative description of the ARI-

[2]Identity certificates obtained through such a procedure belongs to a defined class of certificates, which is defined in the certification policy of the certification service party

ADNE protocol, in this section we provide an example of the variation of an ARIADNE route discovery phase when the IDHC scheme is used.
In ARIADNE, the basic RREQ mechanism is enhanced by eight additional fields used to provide authentication and integrity to the routing protocol as follows:

$$< \text{ROUTE REQUEST}, initiator, target, id, \text{time interval}, \text{hash chain}, \text{node list}, \text{MAC list} >$$

The *initiator* and *target* are set to the address of the initiator and target nodes, respectively. As in DSR, the initiator sets the *id* to an identifier that it has not recently used in initiating a Route Discovery. The time interval is the IDHC time interval at the pessimistic expected arrival time of the request at the target, accounting for clock skew.
The initiator S of the request initializes the hash chain to

$$MAC_{K_{S,D}}(initiator, target, id, \text{time interval})$$

and the node list and MAC list to empty lists[3].
When any node A receives a RREQ for which it is not the target, the node checks its local table of $< initiator, id >$ values from recent requests it has received, to determine if it has already seen a request from this same Route Discovery. If it has, the node discards the packet, as in DSR.
The node also checks whether the time interval in the request is valid: that time interval must not be too far in the future, and the key corresponding to it must not have been disclosed yet. If the time interval is not valid, the node discards the packet.
Otherwise, the node modifies the request by appending its own address (A) to the node list in the request, replacing the hash chain field with $H[A, \text{hash chain}]$, and appending a MAC of the entire REQUEST to the MAC list.
The node uses the IDHC authentication ticket $T_{k-i}^{(A)}$ as a key to compute the MAC, where i is the index for the time interval specified in the request.
Finally, the node rebroadcasts the modified RREQ, as in DSR. When the target node receives the RREQ, it checks the validity of the request by determining that the keys from the time interval specified have not been disclosed yet, and that the hash chain field is equal to:

$$H[\eta_n, H[\eta_{n-1}, H[\ldots, H[\eta_1, \\ MAC_{K_{S,D}}(initiator, target, id, timeinterval)]\ldots]]]$$

where η_i is the node address at position i of the node list in the request, and where n is the number of nodes in the node list. If the target node determines that the request is valid, it returns a RREP to the initiator, containing eight fields:

$$< \text{ROUTE REPLY}, target, initiator, \text{time interval}, \text{node list}, \\ \text{MAC list}, \text{target MAC}, \textbf{authentication ticket list} >$$

The *target*, *initiator*, time interval, node list, and MAC list fields are set to the corresponding values from the RREQ, the target MAC is set to a MAC computed on the

[3]The meaning of the key $K_{S,D}$ is discussed in the note at the end of the current section.

preceding fields in the reply with the key $K_{D,S}$, and the key list is initialized to the empty list.
The RREP is then returned to the initiator of the request along the source route obtained by reversing the sequence of hops in the node list of the request. A node forwarding a RREP waits until it is able to disclose its authentication ticket from the time interval specified, then it appends the authentication ticket from that time interval to the authentication tickets list field in the reply and forwards the packet according to the source route indicated in the packet. Waiting delays the return of the RREP but does not consume extra computational power. When the initiator receives a RREP, it verifies that each authentication ticket in the list is valid, that the target MAC is valid, and that each MAC in the MAC list is valid. If all of these tests succeed, the node accepts the RREP; otherwise, it discards it. In order to prevent the injection of invalid route errors into the network fabricated by any node other than the one on the sending end of the link specified in the error message, each node that encounters a broken link adds IDHC authentication information to the route error message, such that all nodes on the return path can authenticate the error. However, IDHC ticket authentication is delayed, so all the nodes on the return path buffer the error but do not consider it until it is authenticated. Later, the node that encountered the broken link discloses the key and sends it over the return path, which enables nodes on that path to authenticate the buffered error messages.

In our scheme there is no need for an organizational infrastructure among peers, which can be operated by entities belonging to different organizations. The KDC is not involved in any networking operations, i.e. it must not be on-line during the network operation, and is not involved in any further security operations other than the bootstrap phase or when the authentication ticket pool is exhausted. The initial peer authentication to the KDC is only needed once. Indeed, the mobile node can contact the KDC to renew the master authentication ticket by presenting the last self-generated hash chain element/authentication ticket.

The viability of the IDHC authentication scheme can be assessed by taking the values reported in table 7.1 and comparing them with the average number of control packets sent by all the nodes of an ad hoc network to discover and maintain routes. In [46] the authors provide a simulation-based study of the control overhead generated by three ad hoc routing protocols. Specifically, for the DSR protocol the average control traffic for a typical scenario with 40 mobile nodes in a 4km by 4km area and 20 CBR data flows consists of 3000 packets during all the simulation period (900 seconds). Thus, in average, every node generates five control packets per minute.
Obviously, this value can be relatively higher if we consider critical scenarios with high mobility or dense traffic but for an approximate evaluation of the IDHC authentication scheme we deem sufficient to take an average value. The generation and verification rates of authentication tickets reported in table 7.1 are sufficiently high to support the average control traffic generated by nodes in the simulation scenarios presented in [46] leading to the conclusion that the IDHC scheme is an effective solution, when used in the ARIADNE

secure routing protocol.

NOTE: As in ARIADNE, if the initiator and the target of a route request message need mutual authentication, we assume that every end-to-end communicating source-destination pair of nodes A and B share the symmetric keys $K_{A,B}$ and $K_{B,A}$. These keys can be distributed to the system users by the KDC during the initial authentication phase: however, the distribution of symmetric keys between every source-destination pair limit the flexibility of the system described in section 7.6.1 when a new node join the system. It is possible, however, to further modify the ARIADNE protocol in the following way: instead of using pre-loaded shared symmetric keys between each communicating party, sources and destinations of route request messages can use their IDHC authentication tickets as keys to initialize the *hash chain field*. Source and destination buffers the request or reply. By subsequently disclosing (using the recently constructed route) the next authentication ticket in the appropriate time interval, sources and destinations can mutually authenticate. However, this approach introduces delays in the validation of a route reply and necessitate a thoughtful performance and security assessment, which is part of our future research.

Discussion

By adopting the IDHC scheme, *key management requirements are significantly reduced* with respect to the original TESLA-based ARIADNE since the TESLA [85] scheme must rely on a public key infrastructure (PKI). Indeed, the main drawback of the TESLA authentication protocol is that revealing hash chain elements does not guarantee a proper authentication of the sender: the root of a TESLA hash chain *needs to be certified by a universally trusted third party* (a certification authority for example) in order to be sure that all the hash chain elements belong to the sender with identity ID. Precisely, the hash chain root has to be digitally signed with a secret key belonging to user known under the identity ID. The corresponding public key has to be certified by a certification authority that guarantees the binding between the private/public key pair and the identity ID. A potential receiver has to validate (only once) that the root of the hash chain belongs to the sender that is generating the traffic. This requirement however implies the reliance on some public key infrastructure (PKI) for both *certificate generation and revocation.*
The key idea behind the authentication scheme presented in this Chapter is that the IDHC scheme preserves the main advantages of TESLA but does not rely on any PKI. Indeed, by applying the fundamental principles of ID-based cryptosystems, an authentication ticket that is used for packet authentication is **explicitly related to the identity** ID of the source of traffic and there is no need for a certification authority.
As opposed to a classical PKI client that must have a valid public key certificate of the certification authority that issued all the certificates for the other users, in our scheme the identity of another peer can be verified without the need of the CA's public key certificate. In addition, key revocation is greatly simplified with respect to a classical PKI system:

authentication tickets are limited in number (only k tickets) and their validity can be limited in time or in utilization, by simply appending a validity period or authorization information to the identity used to generate the master authentication ticket.
However, the price to pay for such a simplification in the key management requirements is that the cryptographic primitives used in the IDHC scheme are no longer symmetric, but built on top of the asymmetric RSA cryptosystem. Both storage requirements and computational power are moderately higher than in the TESLA scheme. As an example, a TESLA hash chain element requires 128-bit of space (if the MD5 algorithm is used as hashing function) while an IDHC authentication ticket depends on the key-length used to generate it (typically 512-bit). However, the essential advantage of the IDHC mechanism is that authentication tickets can be pre-computed and their verification is fast due to the small exponent e.

7.7 Summary

This Chapter presents an identity based authentication scheme based on a simple form of identity based cryptography combined with the Lamport's keyed hash chain method. In our solution, users are able to generate a chain of authentication tickets using as seed the secret information delivered by a key distribution center. By removing the reliance on a public key infrastructure, our scheme is particularly suitable for networks with multiple dynamic sources whereas other authentication schemes available in the literature suffer from the limitations imposed by certificate management requirements. In addition, there is no need for any organizational structure among users or between users and the KDC. Our message authentication scheme is designed for loosely time-synchronized users and achieves low communication and computational overhead, scales to large numbers of receivers, and tolerates packet loss. We also provide a detailed security analysis of our scheme and show through various attacks that breaking our scheme is equivalent to breaking the basic RSA algorithm. The viability of the IDHC scheme is verified through a performance analysis of our solution, as well as an evaluation of storage requirements. Our implementation is based on the OpenSSL package and has been cross-compiled to be executed on the ARM/X-Scale platforms such as the IPAQ 38xx series.
Furthermore, we present an interesting application of the IDHC scheme in providing a lightweight key distribution service that offers peer authentication to an infrastructureless ad hoc network. In our scheme, there is no need for a network infrastructure and the security bootstrap phase is lightweight. Further, the key distribution center is involved neither in networking operations nor in any further security operations beyond the bootstrap phase.

Relevant publication

P. Michiardi and R. Molva. Identity based hash chains for message authentication. Research Report RR-04-111, Institut Eurecom, July 2004.

Conclusion and future work

Conclusion

Communication systems based on self-organizing entities that build up the network without the need or reliance for a pre-established infrastructure represent a challenging scenario that will play an important role in society and economy by providing opportunities for the creation of *ad hoc* networks and services. However, in order for these services to be successful, they must rely on a network that is secure. The increased sensibility of mobile ad hoc networks (MANETs) with respect to dedicated networks like the Internet, derives from the lack of nodes with a predefined role that are responsible for the network operation.
Furthermore, in the extreme case of ad hoc networks in which users that operate the network do not belong to the same organization nor are governed by the same authority, the lack of an a-priori trust relationship between the members of the network renders security an essential component to enable a realistic deployment and utilization of such *open* networks.

In this thesis we address the security issues raised by open ad hoc networks. We first investigate on the impact of several threats that have been often neglected by the research community when designing ad hoc routing protocols requiring that participants are *inherently trusted*. If these threats can be considered in line with the experience gathered through the study of a variety of attacks on routing protocols for classical networks, in this thesis we point out and analyze a new type of misbehavior, that we called node selfishness, specific to the highly cooperative environment offered by the ad hoc networking paradigm. A simulation-based analysis conducted in our laboratories revealed that node cooperation is essential because unlike networks using dedicated nodes to support basic functions like packet forwarding, routing, and network management, in ad hoc networks those functions are carried out by all available nodes. However, there is no reason to assume that nodes will participate in the network operation, especially in battery powered environment such as a MANET.
We propose a state of the art of basic security services for ad hoc networks, that range from secure routing protocols to key-management services and we focus on various type of cooperation enforcement mechanisms available in the literature. Our research pointed

out two challenging research directions that we further investigated in the reminder of the thesis: the novelty of cooperation requirements and the difficult task of providing security associations without the support of an external infrastructure.
We thus propose a cooperation enforcement mechanism based on an original reputation system that we called CORE. CORE meets the design requirements that we presented in our preliminary study on cooperation enforcement schemes. Furthermore, we propose a detailed validation of the CORE mechanism using two different methodologies: in the first method, we use a classical network simulation tool to infer the basic properties of CORE. We then extend our work in order to cope with a sophisticated model of node selfishness that takes into account end-users' "rationality" when operating the nodes. Our validation study shows that CORE is an effective and robust mechanism that stimulates cooperation of selfish nodes; furthermore, through the evaluation of nodes' energy consumption, we provide evidence that CORE also offers incentives for legitimate nodes to use it as a cooperation mechanism in that they can save a non-negligible amount of energy.
We conclude the thesis by proposing a novel authentication scheme for mobile ad hoc networks that does not rely on a pre-established network infrastructure and that does not require any shared organization between the users of the network. In our scheme, a lightweight bootstrap phase is necessary for a node entering the network: by contacting an authentication server a node is able to locally generate authentication credentials that are globally verifiable in the network without the need for the distribution of public key certificates. We also propose a practical utilization of our scheme as a complementary key distribution scheme that enables authentication services demanded by secure routing protocols available in the literature.

The following is a list of the original contributions of the thesis:

- Analysis of security exposures in ad hoc routing protocols.
- Definition selfishness related issues in ad hoc networks.
- Simulation-based analysis of node selfishness in terms of network performance.
- Analysis of secure routing proposals, key-management schemes and cooperation enforcement mechanisms for ad hoc networks available in the literature, identification of their weaknesses and identification of challenging research directions.
- Design of a reputation system based on IIR filters.
- Design of a cooperation enforcement scheme for ad hoc networks based on reputation, CORE. Design of CORE enhancements.
- Simulation-based validation of CORE, identification of key features and limitations of the basic version of CORE.
- Analytical validation of CORE based on game theory:

  - Definition of non-cooperative forwarding games in ad hoc networks.
  - Definition of evolutionary simulations for the study of cooperation strategies in ad hoc networks, with and without mis-perception noise.
  - Definition and analysis of coalitions and coalition formation algorithms for ad hoc networks through cooperative game theory

- Design and analysis of an identity-based hash-chain (IDHC) authentication scheme:
  - Application of IDHC for message authentication in ad hoc networks
  - Application of IDHC for routing security in ad hoc networks

We argue that the topics presented in this thesis share many similarities with another growing research area that is overlay networking and peer-to-peer systems. Indeed, ad hoc networking shares many concepts, such as *distribution and cooperation*, with the peer-to-peer (P2P) computing model [101], which constitutes a natural paradigm for the ad hoc computing model. A defining characteristic of P2P systems is their ability to provide efficient, reliable, and resilient routing between their constituent nodes by forming structured "ad hoc" topologies on top of a real network infrastructure. The difference with traditional distributed computing systems is the lack of a central authority controlling the various components; instead, nodes form a dynamically and self-organizing system. The applications best suited for P2P implementation are those where centralization is not possible, relations are transient, and resources are highly distributed [91]. As an example, to find a particular piece of data within the network, P2P systems explicitly or implicitly provide a lookup mechanism or locator function that matches a given string or key to one or more network nodes responsible for the value associated with that key. The common denominator to both *structured* and *un-structured* overlay networks is that they represent a shift from purely commercial content-distribution systems to a "gift economy" in which individual users *offer up* their resources - content, access bandwidth, storage and CPU - *for the benefit of other individuals.*
However, much of the existing work in P2P and overlay networking assumes that users will follow prescribed protocols without deviation. This assumption ignores the user's ability to modify the behavior of an algorithm for self-interested reasons. Here, we argue that **user's "rationality"** is a real issue in P2P and overlay networks as much as it has been shown to be for ad hoc networks and we claim that the natural extension to the validation work presented in this thesis should focus on tools borrowed from micro-economic theory. In particular, we deem mechanism design (MD) theory to be the best candidate as an effective tool to be used for the design and analysis of networks with *rational* nodes.
To support our claims, it is not hard to find evidence of rational behavior in existing P2P networks. There is interesting work documenting the "free rider" problem [4] and the "tragedy of the commons" [41] in a data centric peer to peer setting. In this situation, rational users free ride and consume a resource but do not produce at the same level of their consumption. An additional example can be found in the context of P2P search: consider the rational users in simple file sharing networks who refuse to relay other node queries to conserve their own bandwidth. Although it has not been labeled as such, rational behavior

has occurred in computational P2P settings as well. One example occurred when users of "Seti@Home" modified their client software to make it appear as if they were doing more work than was actually occurring. These users placed a high value on their ranking in a leader board that recorded the "computation units contributed" for the "Seti@Home project". The scoring system did not prevent these rational players from increasing their profit by modifying the behavior of their software [51].
What should be the response to the problems created by user rationality in peer to peer systems? We believe that a good starting point to solve the lack of cooperation problem would be to extend the work on game theoretical modeling presented in this thesis and further explore the use economic modeling as a tool to design (distributed) protocols with self-interest and rationality in mind. The central challenge in turning ideas from economics is to offer incentives for nodes to follow protocols that provide the networks as whole with good system-wide performance.

Directions for future research

Research in security schemes for open ad hoc networks is still in its infancy and a lot of room for creativity is available.

Concerning cooperation enforcement schemes, an interesting direction that needs further research is offered by the analysis of cooperation strategies and coalition formation algorithms through game theory with a more realistic representation of the underlaying networking mechanisms. Indeed, several limitations of our scheme are due to our initial assumptions: as an example, we only consider homogeneous networks whereas it would be more realistic to consider heterogeneous networks. Furthermore, we initially consider interactions among random pairs of nodes in the network: however, we believe that the strategy space of a node could be reduced if we consider multiple-player interactions by taking into account the strategy of all nodes that belong to a route from a source to the corresponding destination.
In our research, we first proposed a solution to provide incentives to cooperate in ad hoc networks, and then analyzed it through game theory. Alternatively, it would interesting to propose a game theoretical framework to infer the guidelines for the design of a generic cooperation scheme while taking into account the peculiarities of the mechanisms used to implement a cooperation strategy.
Cooperation enforcement schemes are based on the promiscuous mode operation of wireless cards and use the watchdog technique. In this thesis we summarized the issues inherent to this technique and propose to use reputation in order to mitigate eventual errors of monitoring components based on the watchdog mechanism. However, it would be interesting to find viable alternatives to the watchdog technique mainly for inter-operability reasons: the experience gathered through the practical evaluation of CORE as part of a test-bed platform currently under development in our laboratories shows that it is unrealistic to assume wireless card to properly (and easily) function in promiscuous mode.

Indeed, most of the off-the-shelf hardware available for commercial use does not support promiscuous mode operation; furthermore, the current trend in the provision of personal wireless connectivity is to integrate 802.11 capabilities directly into CPUs, like for the Centrino technology available from Intel, whereby advanced configuration capabilities of the wireless device are drastically reduced.

Concerning the ultimate goal of designing distributed security services that provides truly self-organized security associations among the nodes of the ad hoc network, most of the solutions proposed in the literature fall short in meetings the requirements imposed by open ad hoc networks. The lack of dedicated components that provide such security services requires the definition of distributed security services that do not require the presence of a trusted third party for their initialization: an interesting area that should be investigated is offered by distributed and secure computation mechanisms that can provide an effective way for the self-initialization of current key-management schemes.

Throughout this thesis we underline the need for a security architecture that defines the interactions and the integration between the security building blocks that cope with network level security exposures. We deem that an interesting topic that needs further research is related to the definition of such an architecture also taking into account cross-layer optimization issues. As an example, we mentioned in this thesis that it would be interesting to study the implications of cross-layer interactions between a layer-3 cooperation enforcement mechanism and the application behavior running on the nodes.

Appendix A

Non cooperative forwarding in ad hoc networks

A.1 Introduction

In order to maintain connectivity in an Ad-hoc network, mobile terminals should not only spend their resources (battery power) to send their own packets, but also for forwarding packets of other mobiles. Since Ad-hoc networks do not have a centralized base-station that coordinates between them, an important question that has been addressed is to know whether we may indeed expect mobiles to collaborate in such forwarding. If mobiles behave selfishly, they might not be interested in spending their precious transmission power in forwarding of other mobile's traffic. A natural framework to study this problem is noncooperative game theory. As already observed in many papers that consider noncooperative behavior in Ad-hoc networks, if we restrict to simplistic policies in which each mobile determines a fixed probability of forwarding a packet, then this gives rise to the most "aggressive" equilibrium in which no one forwards packets, see e.g. [38, Corollary 1], [69], thus preventing the system to behave as a connected network. The phenomenon of aggressive equilibrium that severely affects performance has also been reported in other noncooperative problems in networking, see e.g. [37] for a flow control context (in which the aggressive equilibrium corresponds to all users sending at their maximum rate).

In order to avoid very aggressive equilibria, we propose strategies based on threats of punishments for misbehaving aggressive mobiles, which is in the spirit of a well established design approach for promoting cooperation in Ad-hoc networks, carried on in many previous works [38], [107]. In all these references, the well known "TIT-FOR-TAT" (TFT) strategy was proposed. This is a strategy in which when a misbehaving node is detected then the reaction of other mobiles is to stop completely forwarding packets during some time; it thus prescribes a threat for very "aggressive" punishment, resulting in an enforcement of a fully cooperative equilibrium in which all mobiles forward all packets

they receive (see e.g. [38, Corollary 2]). The authors of [105] also propose use of a variant of TFT in a similar context.

In this work we consider a less agressive punishment policy. We simply assume that if the fraction q' of packets forwarded by a mobile is less than the fraction q forwarded by other mobiles, then this will result in a decrease of the forwarding probability of the other mobiles to the value q'. We shall show that this will indeed lead to non-aggressive equilibria, yet not necessarily to complete cooperation. The reasons for adopting this milder punishment strategy are the following:

1. There has been criticism in the game-theoretical community on the use of aggressive punishments. For example, threats for aggressive punishments have been argued not to be credible threats when the punishing agent may itself loose at the punishing phase. This motivated equilibria based on more credible punishments known as subgame perfect equilibria [99].

2. An individual that adopts an "partially-cooperative" behavior (i.e. forwards packets with probability $0 < q < 1$) need not be considered as an "aggressive" individual, and thus the punishment needs not be "aggressive" either; it is *fair* to respond to such a partially-cooperative behavior with a partially-cooperative reaction, which gives rise to our mild punishment scheme.

3. The TFT policy would lead to complete cooperation at equilibrium. However, our milder punishment seems to us more descriptive of actual behavior in the society in which we do not obtain full cooperation at equilibrium (for example in the behavior of drivers on the road, in the rate of criminality etc.) It may indeed be expected that some degree of non-cooperative behavior by a small number of persons could result in larger and larger portions of the society to react by adopting such a behavior.

As already mentioned, incentive for cooperation in Ad-hoc networks have been studied in several papers, see [38,69,105,107]. Almost all previous papers however only considered utilities related to successful transmission of a mobile's packet to its neighbor. In practice, however, multihop routes may be required for a packet to reach its destination, so the utility corresponding to successful transmission depends on the forwarding behavior of all mobiles along the path. The goal of our paper is therefore to study the forwarding taking into account the multihop topological characteristics of the path.

Most close to our work is the paper [38] which considers a model similar to ours (introduced in Section A.2 below). [38] provides sufficient condition on the network topology under which each node employing the "aggressive" TFT punishment strategy results in a Nash equilibrium. In the present paper, we show that a less aggressive punishment mechanism can also lead to a Nash equilibrium which has a desirable feature that it is less resource consuming in the sense that a node need not accept all the forwarding request. We also provide some results describing the structure of the Nash equilibrium thus obtained (Section A.5). We then provide a distributed algorithm which can be used by

the nodes to compute their equlibrium strategies and enforce the punishment mechanism using only local information (Section A.6). The algorithm is implemented using the MATLAB suite and some numerical results are presented (Section A.7). Scetion A.8 concludes the paper.

A.2 The Model

Consider an Ad-hoc network described by a directed graph $G = (N, V)$. Along with that network, we consider a set of source-destination pairs O and a given routing between each source s and its corresponding destination d, of the form $\pi(s,d) = (s, n_1, n_2, \ldots, n_k, d)$, where $k = k(s,d)$ is the number of intermediate hops and $n_j = n_j(s,d)$ is the jth intermediate node on path $\pi(s,d)$. We assume that mobile j forwards packets (independently from the source of the packet) with a fixed probabilty γ_j. Let $\underline{\gamma}$ be the vector of forwarding probabilities of all mobiles. We assume however that each source s forwards its own packets with probability one. For a given path $\pi(s,d)$, the probability that a transmitted packet reaches its destination is thus:

$$p(s,d;\underline{\gamma}) = \prod_{j=1}^{k(s,d)} \gamma(n_j(s,d)).$$

If i belongs to a path $\pi(s,d)$ we write $i \in \pi(s,d)$. For a given path $\pi(s,d)$ of the form $(s, n_1, n_2, \ldots, n_k, d)$ and a given mobile $n_j \in \pi(s,d)$, define the set of intermediate nodes before n_j to be the set $S(s,d;n_j) = (n_1, ..., n_{j-1})$. The probability that some node $i \in \pi(s,d)$ receives a packet originating from s with d as its destination is then given by

$$p(s,d;i,\underline{\gamma}) = \prod_{j \in S(s,d;i)} \gamma(j).$$

Note that $p(s,d;d,\underline{\gamma}) = p(s,d;\underline{\gamma})$, the probability that node d receives a packet originating from source s and having d as its destination.

Define $O(i)$ to be all the paths in which a mobile i is an intermediate node. Let the rate at which source s creates packets for destination d be given by some constant λ_{sd}. Then the rate at which packets arrive at node i in order to be forwarded there is given by

$$\xi_i(\underline{\gamma}) = \sum_{\pi(s,d) \in O(i)} \lambda_{sd} p(s,d;i,\underline{\gamma}).$$

Let E_f be the total energy needed for forwarding a packet (which includes the energy for its reception and its transmission). Then the utility of mobile i that we consider is

$$\begin{aligned} U_i(\underline{\gamma}) &= \sum_{n:(i,n)\in O} \lambda_{in} f_i(p(i,n;\underline{\gamma})) \\ &\quad + \sum_{n:(n,i)\in O} \lambda_{ni} g_i(p(n,i;\underline{\gamma})) - a E_f \xi_i(\underline{\gamma}), \end{aligned} \tag{A.1}$$

where f_i and g_i are utility functions that depend on the success probabilities associated with node i as a source and as a destination respectively and a is some multiplicative constant. We assume that $f_i(\cdot)$ and $g_i(\cdot)$ are nondecreasing concave in their arguments. The objective of mobile i is to choose γ_i that maximizes $U_i(\underline{\gamma})$. We remark here that similar utility function is also considered in [38] with the difference that node's utility does not include its reward as a destination, i.e., they assume that $g_i(\cdot) \equiv 0$.

Definition: *For any choices of strategy $\underline{\gamma}$ for all mobiles, define $(\gamma_i', \underline{\gamma}^{-i})$ to be the strategy obtained when only player i deviates from γ_i to γ_i' and other mobiles maintain their strategies fixed.*

In a noncooperative framework, the solution concept of the optimization problem faced by all players is the following:

Definition: *A Nash equilibrium, is some strategy set $\underline{\gamma}^*$ for all mobiles such that for each mobile i,*

$$U_i(\underline{\gamma}^*) = \max_{\gamma_i'} U_i(\gamma_i', (\underline{\gamma}^*)^{-i}).$$

We call $argmax_{\gamma_i'} U_i(\gamma_i', \underline{\gamma}^{-i})$ the set of optimal responses of player i against other mobiles policy $\underline{\gamma}^{-i}$ (it may be an empty set or have several elements).

In our setting, it is easy to see that for each mobile i and each fixed strategy $\underline{\gamma}^{-i}$ for other players, the best response of mobile i is $\gamma_i = 0$ (unless $O(i) = \emptyset$ in which case, the best response is the whole interval $[0, 1]$). Thus the only possible equilibrium is that of $\gamma_i = 0$ for all i. To overcome this problem, we consider the following "punishing mechanism". in order to incite mobiles to cooperate.

Definition: *Consider a given set of policies $\underline{\gamma} = (\gamma, \gamma, \gamma, ...)$. If some mobile deviates and uses some $\gamma' < \gamma$, we define the punishing policy $\kappa(\gamma', \gamma)$ as the policy in which all mobiles decrease their forwarding probability to γ'.*

When this punishing mechanism is enforced, then the best strategy of a mobile i when all other mobiles use strategy γ is γ' that achieves

$$J(\gamma) := \max_{\gamma' \leq \gamma} U_i(\underline{\gamma}') \tag{A.2}$$

where $\underline{\gamma}' = (\gamma', \gamma', \gamma', ...)$.

Definition: *If some γ^* achieves the minimum in (A.2) we call the vector $\underline{\gamma}^* = (\gamma^*, \gamma^*, \gamma^*, ...)$ the equilibrium strategy (for the forwarding problem) under threats. $J(\gamma)$ is called the corresponding value.*

Remark: Note that $\gamma^* = 0$ is still a Nash equilibrium, a fact that will be used frequently in Section A.5 where we obtain some structural properties of equilibrium strategy under threats.

A.3 Utilities for Symmetrical Topologies

By symmetrical topology we mean the case where f_i, g_i and ξ_i are independent of i. This implies that for any source-destination pair (s,d), there are two nodes s' and d' such that the source-destination pairs (s',s) and (d,d') are identical to (s,d) in the sense that there view of the network is similar to that of (s,d). This implies that, under the punishment mechanism where all nodes have same forwarding probability, we have $p(s,d;\underline{\gamma}) = p(s',s;\underline{\gamma})$. Thus we can replace the rewards $f_i + g_i$ by another function that we denote $f(\cdot)$.

Consider $\underline{\gamma}$ where all entries are the same and equal to γ, except for that of mobile i. For a path $\pi(s,d)$ containing n intermediate nodes, we have $p(s,d;\underline{\gamma}) = \gamma^n$. Also, if a mobile i is $n+1$ hops away from a source, $n = 1,2,3,...$, and is on the path from this source to a destination (but is not itself the destination), then $p(s,d;i,\underline{\gamma}) = \gamma^n$. We call the source an "effective source" for forwarding to mobile i since it potentially has packets to be forwarded by mobile i. Let $h(n)$ be the rate at which all effective sources located $n+1$ hops away from mobile i transmit packets that should use mobile i for forwarding (we assume that h is the same for all nodes). Let $\lambda^{(n)}$ denote the rate at which a source s creates packets to all destinations that are $n+1$ hops away from it. Then we have

$$U_i(\underline{\gamma}) = \sum_{n=1}^{\infty} \lambda^{(n)} f(\gamma^n) - aE_f \sum_{n=1}^{\infty} h(n)\gamma^n. \tag{A.3}$$

The equilibrium strategy under threat is then the value of γ that maximizes the r.h.s.

Remark: If we denote by $\Lambda(z) = \sum_{n=1}^{\infty} z^n \lambda^{(n)}$ the generating function of $\lambda^{(n)}$ and $H(z) := \sum_{n=1}^{\infty} z^n h(n)$ the generating function of h. Then

$$\max_{\gamma} \Big(\Lambda(\gamma) - aE_f H(\gamma) \Big)$$

is the value of the problem with threats in the case that f is the identity function.

A.4 Examples

In this section we present, by means of two examples, the effect of imposing the proposed punishment mechanism.

A.4.1 An Asymmetric Network

Consider the network shown in Figure A.1. For this case nodes 1 and 4 have no traffic to forward. Note also that if we assume that $g_3(\cdot) \equiv 0$ in Equation A.1 then node 3 has

no incentive even to invoke the punishment mechanism for node 2. This will result in no cooperation in the network. Assume for the time being that $f_2(x) = g_3(x) = x$, i.e., f_2 and g_3 are identity functions. In this case it is seen that the utility functions for nodes 2 and 3 are, assuming $\lambda_{13} = \lambda_{24} = 1$, $U_2(\gamma_2, \gamma_3) = \gamma_3 - aE_f\gamma_2$ and $U_3(\gamma_2, \gamma_3) = \gamma_2 - aE_f\gamma_3$. When we impose the punishment mechanism, it turns out that the equilibrium strategy for the two nodes is to always cooperate, i.e., $\gamma_2 = \gamma_3$. This is to be compared with the TFT strategy of [38] which would imply $\gamma_2 = \gamma_3 = 0$.

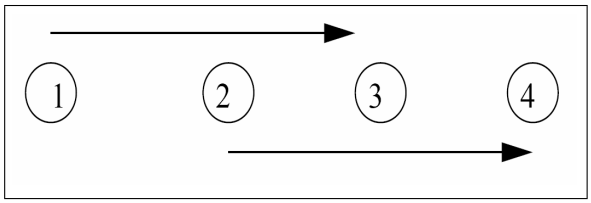


Figure A.1: An asymmetric network.

A.4.2 A Symmetric Network: Circular Network with Fixed Length of Paths

We consider here equally spaced mobile nodes on a circle and assume that each node i is a source of traffic to a node located L hops to the right, i.e. to the node $i + L$.

Let the rate of traffic generated from a source be λ. For this case, $h(n) = \lambda I_{\{n \leq L-1\}}$. Also, $\lambda^{(n)} = \lambda I_{\{n=L\}}$, for some λ. It follows from Equation A.3 that the utility function for mobile i is

$$U_i(\underline{\gamma}) = \lambda f(\gamma^{L-1}) - aE_f\lambda \sum_{n=0}^{L-2} \gamma^n.$$

For $f(\cdot)$ an identity function, we see that $U_i(\underline{\gamma}) = \lambda \left[\gamma^{L-1} - aE_f(\gamma^{L-2} + \gamma^{L-3} + \ldots + \gamma + 1)\right]$. Note that if $L = 2$ and $a = \frac{1}{E_f}$, the utility function is independent of γ hence in this case the equilibrium strategy is any value of forwarding probability. Also, if $aE_f \geq 1$, the equilibrium strategy is $\gamma = 0$. We will have more to say on this in th next section where we study the structure of equilibrium strategy for symmetric network.

A.5 Structure of Equilibrium Strategy for Symmetric Network

In this section we undertake the study of dependence of the equilibrium strategy on the various system parameters. We restrict ourselves to the case of symmetric topologies.

Symmetry of the problem along with the imposed punishment mechanism implies that the equilibrium strategy (the forwarding probabilities) will be same for all the nodes in the network. We denote this probability by γ^*.

This is to be understood as follows. When a node i computes its equilibrium strategy γ_i, it must consider the fact that the other nodes will respond with a punishing mechanism to its strategy. Thus, the problem faced by node i is *not* that of optimizing Equation A.3 with respect to γ_i considering γ^{-i} fixed (which will lead to the trivial solution of $\gamma_i = 0$ as seen before). Owing to the punishment mechanism, node i should apriori assume that all the forwarding probabilities are same, i.e., $\gamma^{-i} = (\gamma_i, \ldots, \gamma_i)$. This makes the problem faced by node i a single variable optimization problem.

Though $f(\cdot)$ is concave in its argument ($p(\gamma)$, which is a polynomial in γ), $f(p(\gamma))$ may not be concave as a function of γ. For example, in the case of circular network above, $f(p(\gamma)) = p(\gamma) = \gamma^{L-1}$, convex in γ. Thus obtaining a direct structural result for γ^* seems to be hard for general $f(\cdot)$ and $p(\cdot)$. We can get some interesting insights using some approximations; this is the aim of present section. In particular, we study how γ^* depends on the *system paramaters*, L, $f(\cdot)$, $p(\cdot)$, a and E_f.

It is clear from the expression of the utility function that γ^* will depend on a and E_f only through their product. Let us introduce the notation $K := aE_f$.

It is also clear from the definition of utility function ($U_i(\gamma)$) that if either K or L is *large*, the equilibrium strategy of the game is at *smaller* γ. It is also intuitive that for *small* values of K (or L), a node may forward most of the requesting packets. In the following we characterize what value of K or L can be considered as *large* or *small*. Clearly this characterization will depend on $f(\cdot)$ and the network, i.e., $p(\cdot)$ and $H(\cdot)$.

A.5.1 Dependence of γ^* on K

The case of General Network and $f(\cdot)$

Consider the line starting at $\gamma = 0$ with a value $f(0)$ and having a slope $f'(0)p'(1)$ for $0 \leq \gamma \leq 1$. This slope is an upper bound on the slope of $f(p(\gamma)$ in the stability region since $p(\cdot)$ is convex and $f(\cdot)$ is concave. Thus the line constructed above is an upper bound to $f(p(\gamma))$ for $0 \leq \gamma \leq 1$. Consider also the line starting from $f(0)$ and having a slope $f'(1)p'(0)$ (this slope is clearly a lower bound on the slope of $f(p(\gamma))$. It is thus true that f lies in *between* these two lines. Thus we can get conditions under which the equilibrium strategy is less than (or equal to) the maximal possible value, i.e., the extreme point $\gamma = 1$.

Result A.5.1 *If the network topology and $f(\cdot)$ satisfy $f'(0)p'(1) \leq Kh(1)$, then the equi-*

librium strategy is $\gamma^* = 0$.

Proof: Follows from the construction of bounds above.

Remark: Above result shows that if K, i.e., the energy spent in reception and transmission is larger than a threshold (say $K^* = \frac{f'(0)p'(1)}{h(1)}$), it is best for the nodes to not forward packets at all *even under the punishing mechanism*. This is to be compared with the fact proved below that $\gamma = 0$ is *always* a *local* maximum for $L > 2$ for the circular network. Thus the above result gives a criteria when $\gamma = 0$ is also a *global* maximum.

The Case of the Circular Network with $f(p(\gamma)) = p(\gamma)$

For the case of circular network and $f(\cdot)$ being an identity function, we can say more about the dependence of γ^* on K. It is seen that for this particular case, $U_i(0) = -K$.

For the optimum, the derivative of $U_i(\gamma)$ with respect to γ should be zero, i.e.,

$$\frac{L-1}{K}\gamma^{L-2} = (L-1)\gamma^{L-3} + \ldots + 1.$$

So, if $K > L-1$, the above requirement is not possible for $\gamma < 1$. Thus the only solution is either $\gamma = 0$ or $\gamma = 1$. But, as will be seen in later section, $\gamma = 0$ is always a local maximum. Thus, in absence of any other critical point (where $U_i'(\gamma) = 0$), the global maximum is at $\gamma^* = 0$. Thus, for $K > L-1$, we have that $\gamma^* = 0$. This is, as argued before, reasonable, as for K *large*, a node will spend more energy to forward packets, hence will not do it for any packet it gets for forwarding; we thus obtain the expression for energy requirement that can be characterised as *large*, i.e., $K^* := L-1$.

In getting the above bound on K we assumed that $f(x) = x$. The above reasoning is however true for any $f(\cdot)$ whenever $f'(0) \leq \frac{K}{L-1}$ for the circular networks, i.e., if $f'(\cdot) \geq \frac{L-1}{K}$ since $f(\cdot)$ is concave. Thus, for a fixed set of K and L, we get that $\gamma^* = 0$ for any $f(\cdot)$ that has slope $\leq \frac{K}{L-1}$. Looking at it from other point, we again see that if $K \geq \frac{L-1}{f'(0)}$ then $\gamma^* = 0$.

A.5.2 Dependence of γ^* on L

In the above we saw that in the case of a symmetric network and for a fixed L, if $K > K^*$, we get that $\gamma^* = 0$. The intuition for this result is that a node will have to spend more energy in forwarding the packets as compared to the reward it gets by its own packet forwarded by other nodes. However, if we fix K and increase the hop-length L, it is intuitive that γ^* will eventually start decreasing as a function of L. This is established in the following. Let $\lambda = 1$ and drop the subscript i from the utility function. The idea

is presented by a detailed analysis of the case of circular network, the results though are easily seen to be valid for any symmetric network.

$$\begin{aligned} U(\gamma) &= f(p(\gamma)) - K\sum_{j=0}^{L-2}\gamma^j \\ \frac{d}{d\gamma}U(\gamma) &= \frac{d}{dx}f(x)|_{x=p(\gamma)}\frac{d}{d\gamma}p(\gamma) - K\sum_{j=1}^{L-2}j\gamma^{j-1} \end{aligned}$$

Since $p(\gamma) = \gamma^{L-1}$, it follows that for $L > 2$, $U'(0) = -K$ *irrespective of the function* $f(\cdot)$. Thus we see that $\gamma = 0$ is *always a local maximum for* $L > 2$.

This, along with the fact that $U(\cdot)$ is continuous in γ implies that the first positive solution of the equation

$$U'(\gamma) = f'(p(\gamma))(L-1)\gamma^{L-2} - K\sum_{j=1}^{L-2}j\gamma^{j-1} = 0,$$

corresponds to a local minimum of $U(\cdot)$. By "first positive solution" we mean $\gamma^+ := \lim_{\epsilon\to 0}\inf\{\gamma > \epsilon : U'(\gamma) = 0\}$. It is seen that either $1 > \gamma^* \geq \gamma^+$ or $\gamma^* = 0$. Now, if we assume that $f'(\cdot)$ is bounded by a constant, it follows that

$$\begin{aligned} U'(\gamma) &= f'(p(\gamma))(L-1)\gamma^{L-2} - K\sum_{j=1}^{L-2}j\gamma^{j-1} \\ &\to_{L\to\infty} -K\sum_{j=1}^{L-2}j\gamma^{j-1} \\ &= \frac{-K}{(1-\gamma)^2} \end{aligned}$$

thus, $\gamma^+ \to_{L\to\infty} \infty$ thus $\gamma^* = 0$ as $L \to \infty$. Note that this conclusion requires that γ^+ is nondecreasing; we prove this below.

For a fixed L, to find γ^*, we need to solve the equation

$$\theta_L(\gamma) := U'(\gamma) = f'(p(\gamma))(L-1)\gamma^{L-2} - K\sum_{j=1}^{L-2}j\gamma^{j-1} = 0.$$

Since $\theta_L(0) = -K$, the zero of $\theta_L(\cdot)$ with minimum nonnegative argument γ will see a transition of $\theta_L(\cdot)$ from a negative value to a positive value in direction of increasing γ. So, if we can show that $\theta_L(\gamma)$ is a nondecreasing function of L, then since $\theta_L(0) = -K$

for all $L > 2$, we will have shown that γ^+ is nonincreasing in L. Let $p_L(\gamma) := \gamma^{L-1}$. Then

$$\begin{aligned}\theta_L(\gamma) &= f'(p_L(\gamma))(L-1)\gamma^{L-2} - K\sum_{j=1}^{L-2} j\gamma^{j-1} \\ \theta_{L+1}(\gamma) &= f'(p_{L+1}(\gamma))L\gamma^{L-1} - K\sum_{j=1}^{L-1} j\gamma^{j-1}\end{aligned}$$

hus,

$$\begin{aligned}\theta_L(\gamma) - \theta_{L+1}(\gamma) &= K(L-1)\gamma^{L-2} + f'(p_L(\gamma))(L-1)\gamma^{L-2} - f'(p_{L+1}(\gamma))L\gamma^{L-1} \\ &= \gamma^{L-2}\left[L\left\{K + f'(\gamma^{L-1}) - \gamma f'(\gamma^L)\right\} - f'(\gamma^{L-1}) - K\right] \\ &> \gamma^{L-2}\left[L\left\{K + f'(\gamma^{L-1}) - f'(\gamma^L)\right\} - f'(0) - K\right],\end{aligned}$$

where we have used the fact that $f'(\cdot)$ is decreasing and is bounded by a *finite* value which can be taken to be $f'(0)$.

Fix a $1 > \gamma > 0$ and consider now, $f'(\gamma^{L-1}) - f'(\gamma^L)$. Since $f'(0)$ is finite and $f'(\cdot)$ is assumed to be continuous, it follows that $f'(\gamma^L) \to f'(0)$. It follows that, given an $\epsilon > 0$, $\exists L_\epsilon$ such that for all $L > L_\epsilon$,

$$0 > f'(\gamma^{L-1}) - f'(\gamma^L) > -\epsilon.$$

Take $\epsilon = \frac{K}{2}$. Thus, $\exists L^*$ such that $L > L^*$ implies that

$$K + f'(\gamma^{L-1}) - f'(\gamma^L) > \frac{K}{2}.$$

Since $f'(0)$ is finite, it follows that, for L sufficiently large compared to L^*,

$$\theta_L(\gamma) - \theta_{L+1}(\gamma) \geq \gamma^{L-2}L\frac{K}{2}$$

Now we show that $\theta_L(\gamma) - \theta_{L+1}(\gamma)$ is also bounded above by similar exponentially decaying function.

For a fixed L, to find γ^*, we need to solve the equation

$$\theta_L(\gamma) = U'(\gamma) = f'(p(\gamma))(L-1)\gamma^{L-2} - K\sum_{j=1}^{L-2} j\gamma^{j-1} = 0.$$

Now, since $p_{L+1}(\gamma) \leq p_L(\gamma)$ and $f(\cdot)$ is concave nondecreasing, $f'(p_{L+1}(\gamma)) \geq f'(p_L(\gamma))$. Thus,

$$\begin{aligned}\theta_{L+1}(\gamma) &\geq f'(p_L(\gamma))L\gamma^{L-1} - K\sum_{j=1}^{L} j\gamma^{j-1} \\ &\geq f'(p_L(\gamma))(L-1)\gamma^{L-1} - K\sum_{j=1}^{L} j\gamma^{j-1} \\ \Rightarrow \theta_{L+1}(\gamma) &\geq \gamma\theta_L(\gamma) - KL\gamma^{L-1}, \quad \forall\gamma \leq \gamma_L^*,\end{aligned}$$

along with $\theta_{L+1}(0) = \theta_L(0)$. Now, since $\theta_L(\gamma) < 0$ for $\gamma < \gamma_L^+$, we see that, since now $\gamma\theta_L(\gamma) \geq \theta_L(\gamma)$ for $\theta_L(\gamma) < 0$ and $\gamma < 1$,

$$\theta_{L+1}(\gamma) \;\; \geq \;\; \theta_L(\gamma) - KL\gamma^{L-1}, \quad \forall \gamma \leq \gamma_L^+.$$

Letting $L \to \infty$, we get the desired result, i.e., γ^* is nonincreasing function of L as $L \to \infty$.

A.6 Algorithm for Computing the Equilibrium Strategy in a Distributed Manner

It is interesting to design distributed algorithms which can be used by the mobiles to compute the equilibrium strategy and simultaneously enforce the proposed punishment mechanism. The obvious desirable features of such an algorithm are that it should be decentralised, distributed scalability and should be able to adapt to changes in network.

We propose such an algorithm in this section. We present it, for ease of notation, for the case of symmetric network. Assume for the moment that $f(\cdot)$ is the identity function. In this case each node has to solve the equation (recall the notation of Section A.3)

$$U'(\gamma) = \Lambda'(\gamma) - KH'(\gamma) = 0, \tag{A.4}$$

where the primes denote the derivatives with respect to γ. In general this equation will be nontrivial to solve directly. For the case of more general network, one needs to compute the derivative of the utility function of Equation A.1, the rest of procedure that follows is similar.

Note that in the above expression we first assume that the forwarding probabilities of all the nodes in the network are same (say γ) and then compute the derivative with respect to this common γ. This is because in the node must take the effect of punishment mechanism into account while computing its own optimal forwarding probability, i.e., a node should assume that all the other nodes will use the same forwarding probability that it computes.

Thus, solving Equation A.4 is reduced to a single variable optimization problem. Since the actual problem from which we get Equation A.4 is a maximization problem, a node does a gradient *ascent* to compute its optimal forwarding probability. Thus, in its n^{th} computation, a node i uses the iteration

$$\gamma_i^{(n+1)} = \gamma_i^{(n)} + a(n)(\Lambda'(\gamma_i^{(n)}) - KH'(\gamma_i^{(n)})), \tag{A.5}$$

where $a(n)$ is a sequence of positive numbers satisfying the usual conditions imposed on the learning parameters in stochastic approximation algorithms [54], i.e.,

$$\sum_n a(n) = \infty \text{ and } \sum_n a(n)^2 < \infty.$$

The relation to stochastic approximation algorithm here is seen as follows: the network topology can be randomly changing with time owing to node failures/mobility et cetera. Thus a node needs to appropriately modify the functions $\Lambda(\cdot)$ and $H(\cdot)$ based on its most recent view of the network (this dependence of $\Lambda(\cdot)$ and $H(\cdot)$ on n is suppressed in the above expression).

It is a matter of choice when a node should update its estimate of its forwarding probability, i.e., does the computations mentioned above. One possibility, that we use, is to invoke the above iteration whenever the node receives a packet that is meant for it.

Though the above is a simple stochastic approximation algorithm, it requires a node to know the topology of the part of network around itself. This information is actually trivially available to a node since it can extract the required information from the packets requesting forwarding or using a neighbour discovery mechanism. However, in case of any change in the network, there will typically be some delay till a node completely recognizes the change. This transient error in a node's knowledge about the network whenever the network changes is ensured to die out ultimately owing to the assumption of finite second moment for the learning parameters.

It is known by the o.d.e. approach to stochastic approximation algorithm that the above algorithm will asymptotically track the o.d.e. [54]:

$$\dot{\gamma}_i(t) = \Lambda'(\gamma_i(t)) - KH'(\gamma_i(t)), \tag{A.6}$$

and will converge to one of the *stable* critical points of o.d.e. of Equation A.6. It is easily seen that a local maximum of the utility function forms a stable critical point of Equation A.6 while any local minimum forms an unstable critical point. Thus the above algorithm inherently makes the system converge to a local maximum and avoids a local minimum.

However, it is possible that different nodes settle to different local maxima (we have already seen that there can be multiple maxima). The imposed punishment mechanism then ensures that all the nodes settle to the one which corresponds to the lowest values of γ. This is a desirable feature of the algorithm that it inherently avoids multiple simultaneous operating points. An implementation of the punishment mechanism is described next.

A.6.1 Distributed Implementation of the punishment mechanism

An implementation of punishment mechanism proposed in Section A.2 requires, in general, a node to know about the misbehaving node in the network, if any. Here we propose a simple implementation of the punishment mechanism which requires only local information for its implementation.

Let $\mathcal{N}(i)$ be the set of neighbours of node i. Every node computes its forwarding policy

in a distrubuted manner using the above mentioned stochastic approximation algorithm. However, as soon as a neighboring node is detected to misbehave by a node, the node computes its forwarding policy as follows:

$$\gamma_i^* = \min\{\gamma_i, \min_{j \in \mathcal{N}(i)} \hat{\gamma}_j\} \tag{A.7}$$

where γ_i and $\hat{\gamma}_j$ represents, respectively, the forwarding policy adopted by node i and the estimate of node j's forwarding probability available to node i. γ_i^* represents the new policy selected by node i. Note here that γ_i is still computed using iteration of Equation A.5. We are also assuming here that a node can differentiate between a misbehaving neighbouring node and the failure/mobility of a neighbouring node.

This punishment propagates in the network until all the nodes in the network settle to the common forwarding probability (corresponding to that of the misbehaving node). In particular, the effect of this punishment will be seen by the misbehaving ndoe as a degradation in its own utility. Suppose now that the misbehaving node, say n_i, decides to change to a cooperative behavior: at that point, it will detect and punish its neighbors because of the propagation of the punishment that induced its neighbouring nodes to decrease their forwarding policy. Thus, the intial punishment introduces a negative loop and the forwarding policy of every node of the network collapses to the forwarding policy selected by the misbehaving node. Since now every node in the network has same value of forwarding probability, none of the nodes will be able to increase its forwarding probability even if none of the node is misbehaving now.

An example of this phenomenon can be seen from the network of Figure A.1. Assume that $\gamma_2 = \gamma_3 = \gamma$ and now node 2 reduces γ_2 to a smaller value γ'. Owing to the punishment mechanism, node 3 will respond with $\gamma_3 = \gamma'$. This will result in a reduced utility for node 2 which would then like to increase γ_2. But, since $\gamma_3 = \gamma'$, the punishing mechanism would imply that $\gamma_2 = \gamma'$ as well. This *lock-in* problem is avoided by the solution proposed below.

We modify our algorithm to account for the above mentioned effect. Our solution is based on timers of a fixed duration. When a node enters in the punishing phase (starts punishing some of its neighbour) the local timer for that node is set and the forwarding policy is selected as in equation A.7. When the timer expires, the punishing node evaluates its forwarding policy as if there were no misbehaving nodes, then uses some of standard mechanism to detect any persistent misbehavior (this also helps distinguishing between a misbehaving node and a failed/moved node). In the case no misbehaviors are detected, depending on the choice of the learning parameter of the stochastic apporximation algotithm, the forwarding policy of the network eventually returns to the optimal value for the network. If the neighboring node continues to misbehave, the timer is set again and the punishment mechanism is re-iterated. We assume that the sequence of learning parameters by a node is restarted each time the timer is set.

Remark: It is interesting to see that the proposed implementation of the punishing mechanism is actually having a storage complexity for a node that grows only with the

number of its neighbouring nodes (Equation A.7). Computational complexity is also not large as it depends only on the distance (hops) from a node to its farthest destination (Equation A.5).

A.7 Numerical results

In this section we present numerical results from a MATLAB implementation of the proposed algorithm. We consider a circular network with N mobiles where each mobile is a source of a traffic stream having as its destination the mobile that is L hops away. The numerical results are meant to validate the structural results obtained in the paper and also to indicate the possibility of practical implementation of the proposed punishment scheme. The particular choice of function $f(\cdot)$ are motivated by need to facilitate a better visual presentation. The plots of this section show the value of γ^* computed in the n^{th} iteration of the algorithm vs. the iteration number. The sequence of learning parameters is chosen to be $a(n) = \frac{1}{n \ln(n)}$ (we used various other options also, the results were similar). In the simulations we assume that each node has a lower bound $\gamma_{min} > 0$ on the value of forwarding probability. We take $\gamma_{min} = 0.01$ in results reported here.

A.7.1 Structural Results

We now validate against simulations the structural results obtained in the paper. Figure A.2 depicts the distributed computation of γ^* using the proposed algorithm. We use $f(x) = \ln(100x + 1)$ and $L = 3$. The various curves are obtained by varying the value of K. We see that γ^* is indeed a dereasing function of K. Similar observation is made from Figure A.3 where we fix $K = 0.2$ and $f(x) = (x + 0.0005)^{0.2}$ and vary L. It is seen that γ^* decreases with increasing L and for large value of $L(= 11)$, $\gamma^* = \gamma_{min}$.

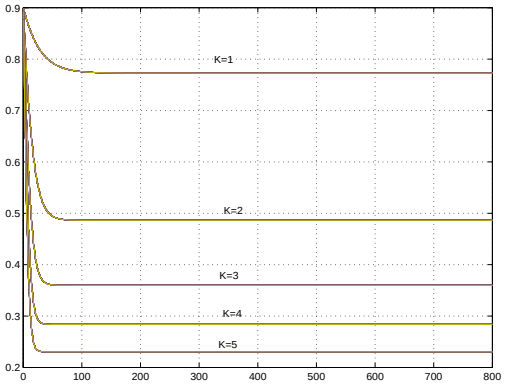


Figure A.2: $L = 3$ and K is varied. The function $f(x) = \ln(100x + 1)$.

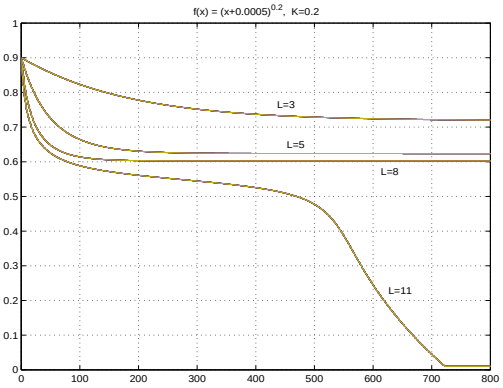


Figure A.3: $K = 0.2$ and L is varied. The function $f(x) = (x + 0.0005)^{0.2}$.

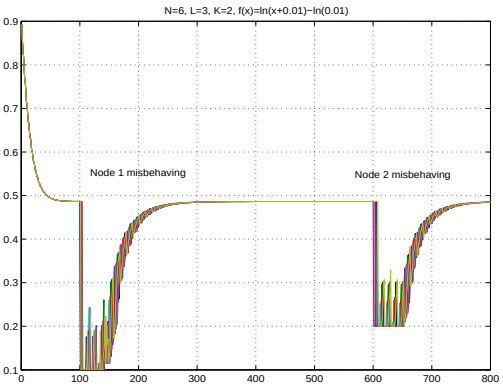


Figure A.4: $N = 6$, $K = 2$, $L = 3$, $f(x) = \ln(100x + 1)$. Node 1 and 2 misbehave in nonoverlapping intervals.

A.7.2 Results with Punishment Mechanism Invoked

Figure A.4 gives the evolution of the estimated of γ^* of various nodes for the circular network with $N = 6$, $L = 3$, $K = 2$ and $f(x) = \ln(100x + 1)$. Node 1 starts misbehaving at iteration number 100 and continues to misbehave till iteration number 130; during this period node 1 keeps a forwarding probability of 0.1. Note that during this period all the other nodes decrease their individual forwarding probability to the value used by node 1, i.e., 0.1. The jumps in the forwarding probabilities during these period are because of the implementation of timer for punishment in the individual nodes (detailed in previous section). The timer value was set to 10 simulation slots. Soon after node 1 stops misbehaving, the forwarding probabilities of all the other nodes increase to settle to the optimal value 0.48. Note here that the convergence of the gradient algorithm is fast enough and that the nodes restart the learning sequence $a(n)$ after each timer expiry. Also shown in the figure is that node 2 misbehaves in the period between iteration 600 and 650.

A.7.3 The Asymmetric Network of Figure A.1

We now consider the case of asymmetric network of Figure A.1. We assume that $f_2(x) = g_3(x) = \sqrt{(}x)$ and that $g_3(\cdot) \equiv 0$. In this case it is seen that the utility functions for nodes 2 and 3 are, assuming $\lambda_{13} = \lambda_{24} = 1$, $U_2(\gamma_2, \gamma_3) = \sqrt{(}\gamma_3) - aE_f\gamma_2$ and $U_3(\gamma_2, \gamma_3) = \sqrt{(}\gamma_2) - aE_f\gamma_3$. On imposing the ponushment mechanism for this case it is seen that, the optimal value of forwarding probabilities of node 2 and node 3 are $\gamma_2 = \gamma_3 = \frac{1}{\sqrt{2aE_f}}$. Figure A.5 gives the evolution of the estimates of γ_2 and γ_3 for this network for different

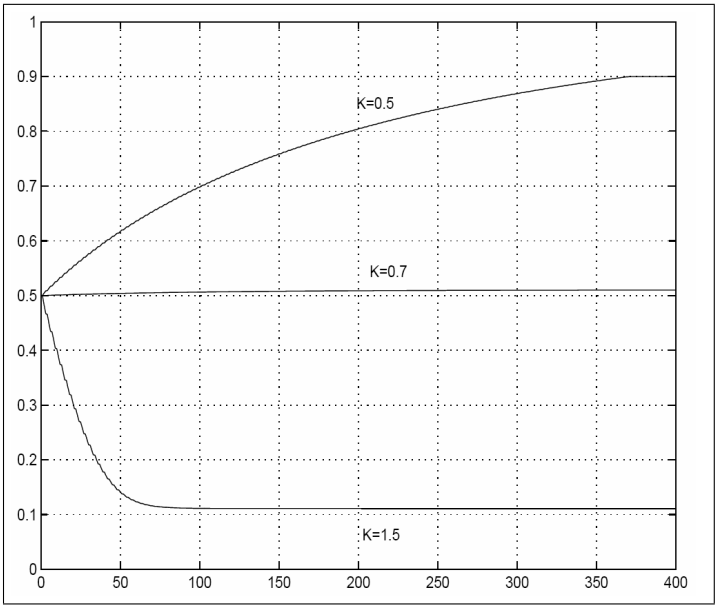


Figure A.5: Evolution of the estimates of γ_2 and γ_3 for the network of Figure A.1. Simulation assumes $f_2(x) = g_3(x) = \sqrt{(}x)$ and that $g_2(\cdot) = f_3(\cdot) \equiv 0$. Value of K is varied.

values of $K = aE_f$. The minimum value of γ_i was taken to be 0.1 and the maximum value was 0.9. Note from the figure that for $K = 1$, the equilibrium strategy of $\frac{K}{2}$ as obtained above is achieved. For large (resp. small) value of K, the equilibrium strategy is at γ_{max} (resp. γ_{min}).

A.8 Conclusion

We use the framework of non-cooperative game theory to provide incentives for collaboration in the case of wireless Ad-hoc networks. The incentive proposed in the paper is based on a simple punishment mechanism that can be implemented in a completely distributed manner with very small computational complexity. The advantage of the proposed strategy is that it results in a less “aggressive" equilibrium in the sense that it does not result

in a degenerate scenario where a node either forwards all the requested traffic or does not forward any of the request.

Some structural results relating the equlibrium strategy to the system parameters were also presented and were verified using an implementation of the punishing mechanism.

Bibliography

[1] Linux familiar distribution, available from http://www.handhelds.org.

[2] Openssl, available from http://www.openssl.org.

[3] N. Abramson. The aloha system. In *Computer-Communication Networks*, pages 501–518. Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

[4] E. Adar and B. Huberman. Free riding on gnutella. *First Monday*, 5(10), May 2000.

[5] E. Altman, A. Kherani, P. Michiardi, and R. Molva. Non cooperative forwarding in ad hoc networks. Research Report RR-5116, INRIA Sophia Antipolis, February 2004.

[6] E. Altman, A. Kherani, P. Michiardi, and R. Molva. Non cooperative forwarding in ad hoc networks. In *Submitted to IFIP Networking*, 2005.

[7] E. Altman, A. Kherani, P. Michiardi, and R. Molva. Some game-theoretic problems in wireless ad hoc networks. In *Submitted to EURO-NGI*, 2005.

[8] R.J. Aumann and J.H. Dreze. Cooperative games with coalition structure. *International Journal of Game Theory*, pages 217–237, 1974.

[9] R. Axelrod. *The Evolution of Cooperation.* Basic Books, New York, 1984.

[10] R. Axelrod. The evolution of strategies in the iterated prisoner's dilemma. *Journal of Genetic Algorithms and Simulated Annealing*, pages 32–41, 1987.

[11] S. Barrett. International environmental agreements in two-level games. *Journal of Conflicts and cooperation in managing environmental resources*, 28:11–37.

[12] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic. *Mobile ad hoc networking.* IEEE Press, Wiley and Sons, US, 2004.

[13] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. *Lecture Notes in Computer Science*, 1109, 1996.

[14] J. Bendor, R.M. Kramer, and S. Stout. When in doubt cooperation in a noisy prisoner's dilemma. *Journal of Conflict Resolution*, 35:691–719, 1991.

[15] A. Benjaminson and S.C. Stallings. A microcomputer-compensated crystal oscillator using a dual-mode resonator. In *Proceedings of the 43-rd Annual Symposium on Frequency Control*, pages 20–26. ePress, 1989.

[16] J.O. Berger. *Statistical Decision Theory and Bayesian analysis.* Springer-Verlag, 1993.

[17] R.B. Bobba, L. Eschenauer, V. Gligor, and W. Arbaugh. Bootstrapping security associations for routing in mobile ad-hoc networks. ISR Technical Report 2002-44, Institute for Systems Research, May 2002.

[18] G.E. Bolton and A. Ockenfels. ERC: a theory of equity, reciprocity, and competition. *The American Economic Review*, 90:166–193, 2000.

[19] D. Boneh. Twenty years of attacks on the RSA cryptosystem. *Notices of the American Mathematical Society (AMS)*, 46(2):203–213, 1999.

[20] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Proceedings of CRYPTO*, 2001.

[21] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. *In Lecture Notes in Computer Science*, 2139, 2001.

[22] S. Brands and D. Chaum. Distance-bounding protocols (extended abstract). In *Advances in Cryptology—EUROCRYPT 93*, volume 765 of *Lecture Notes in Computer Science*, pages 344–359. Springer-Verlag, 1994, 23–27 May 1993.

[23] S. Buchegger and J.Y. Le Boudec. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. In *Proceedings of Tenth Euromicro PDP (Parallel, Distributed and Network-based Processing)*, pages 403 – 410, Gran Canaria, January 2002.

[24] S. Buchegger and J.Y. Le Boudec. Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes — Fairness In Dynamic Ad-hoc NeTworks. In *Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Lausanne, CH, June 2002. IEEE.

[25] S. Capkun, L. Buttyan, and J. Hubaux. Self-organized public-key management for mobile ad hoc networks. In *Proceedings of ACM International Workshop on Wireless Security (WISE)*, 2002.

[26] S. Capkun, L. Buttyan, and J.P. Hubaux. Sector: Secure tracking of node encounters in multi-hop wireless networks. In *Proceedings of the 1-st ACM Workshop on Security of Ad Hoc and Sensor Networks (SANS)*, 2003.

[27] C. Carraro and D. Siniscalco. International environmental conventions: the case of uniform reductions of emissions. *Journal of Environmental and Resource Economics*, 2:141–159.

[28] C. Carraro and D. Siniscalco. Strategies for the international protection of the environment. *Journal of Public Economics*, 52:309–328.

[29] J.C. Cha and J.H. Cheon. An identity-based signature from gap diffie-hellman groups. In *Proceedgins of the International Workshop on Practice and Theory in Public Key Cryptography, PKC*. LNCS, 2003.

[30] J. Chen, S. Zhong, and R. Yang. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *Proceedings of the 22-nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOMM)*, 2003.

[31] K. Chen and K. Nahrstedt. iPass: an Incentive Compatible Auction Scheme to Enable Packet Forwarding. In *Proceedings 24-th International Conference on Distributed Computing Systems, ICDCS*, 2004.

[32] T. Clark. Tom clark's totally accurate clock ftp site. available at ftp://aleph.gsfc.nasa.gov/GPS/totally.accurate.clock/.

[33] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized link state routing protocol. In *Proceedings of the 5-th IEEE International Multitopic Conference (INMIC)*, 2001.

[34] C. Cocks. An identity based encryption scheme based on quadratic residues. *Lecture Notes in Computer Science*, 2260, 2001.

[35] X. Ding and G. Tsudik. Simple identity-based cryptography with mediated RSA. In *CTRSA: CT-RSA, The Cryptographers' Track at RSA Conference, LNCS*, 2003.

[36] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Age matters: Efficient route discovery in mobile ad hoc networks using encounter ages. In *Proceedings of the 4-th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 257–266, New York, June 1–3 2003. ACM Press.

[37] D. Dutta, A. Goel, and J. Heidemann. Oblivious aqm and nash equilibria. In *Proceedings of The 22-nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOMM)*, San Francisco (CA), USA, Jan 2003.

[38] M. Félegyházi, L. Buttyán, and J. P. Hubaux. Equilibrium analysis of packet forwarding strategies in wireless ad-hoc entworks – the static case. In *Proceedings of the IFIP Personal Wireless Communications Conference (PWC)*, Venice, Italy, Sept. 2003.

[39] D. Fudenberg and J. Tirole. *Game Theory*. The MIT Press, Cambridge, Massachusetts, 1991.

[40] M. Grossglauser and D.N.C. Tse. Mobility increases the capacity of ad-hoc wireless networks. In *Proceedings of the 20-th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOMM)*, 2001.

[41] G. Hardin. The tragedy of the commons. *Science*, 162:1243–1248, 1968.

[42] G. He. Destination-sequenced distance vector (DSDV) protocol. Technical report, May 06 2002.

[43] F. Hess. Efficient identity based signature schemes based on pairings. In *Selected Areas in Cryptography, SAC*, pages 310–324. Springer-Verlag, February 2002.

[44] R. Hoffman. Twenty years on: The evolution of cooperation revisited. *Journal of Artificial Societies and Simulation*, 3, 2000.

[45] J.P. Hubaux and L. Buttyan. Nuglets: a virtual currency to stimulate cooperation in self-organized mobile ad hoc networks. Technical Report DSC/2001/001, EPFL, January 2001.

[46] H. Jiang and J.J. Garcia-Luna-Aceves. Performance comparison of three routing protocols for ad hoc netwokrs. In *Proceedings of the 14-th International Conference on Computer Communications and Networks (ICCCN)*, 2001.

[47] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks. In *Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking*, August 15-19, 1999.

[48] D.B. Johnson, D.A. Maltz, , and J. Broch. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. In *Ad Hoc Networking, edited by Charles E. Perkins, Chapter 5, pp. 139-172, Addison-Wesley*, 2001.

[49] M. Just, E. Kranakis, and T. Wan. Resisting malicious packet dropping in wireless ad-hoc networks using distributed probing. In *Proceedings of 2-nd Annual Conference on Adhoc Networks and Wireless (ADHOCNOW)*, pages 151–163, Montreal, Canada, October 2003.

[50] L.M. Kahn and J.K Murnighan. Conjecture, uncertainty, and cooperation in prisoner's dilemma games: Some experimental evidence. *Journal of Economic Behaviour and Organisation*, 22:91–117, 1993.

[51] L. Kahney. Cheaters Bow to Peer Pressure. *Wired online, http://www.wired.com/news/technology/0,1282,41838,00.html*, 2001.

[52] A. Khalili, J. Katz, and W. Arbaugh. Toward secure key distribution in truly ad-hoc networks. In *Proceedings of the IEEE Workshop on Security and Assurance in Ad-Hoc Networks (SAINT)*, 2003.

[53] R. Knopp and P.A. Humblet. Information capacity and power control in single-cell multiuser communications. In *Proceedings of International Conference of Communications (ICC)*, 1995.

[54] H.J. Kushner and G. Yin. Stochastic approximation algorithms and applications. *Springer-Verlag*, 1997.

[55] L. Lamport. Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–772, November 1981.

[56] A. Lange and C. Vogt. Cooperation in international environmental negotiations due to a preference for equity. *Journal of Public Economics*, 2002.

[57] B.N. Levine, B. Dahill, C. Shields, E. Royer, and K. Sanzgiri. A Secure Routing Protocol for Ad Hoc Networks. In *Proceedings of the 10-th IEEE International Conference on Network Protocols (ICNP)*, August 31 2002.

[58] H. Luo and S. Lu. Ubiquitous and robust authentication services for ad hoc wireless networks. Technical Report TR-200030, Dept. of Computer Science, UCLA, 2000.

[59] J.K. MacKie-Mason and H.R. Varian. Pricing the Internet. In *Public Access to the Internet, JFK School of Government, May 26–27, 1993*, page 37, April 1993.

[60] D.A. Maltz, D.B. Johnson, J. Jetcheva, J. Broch, and Y. Hu. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the 4-th ACM Annual International Conference on Mobile Computing and Networking, (MOBICOM)*, 1998.

[61] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6-th IEEE/ACM International Conference on Mobile Computing and Networking*, 2000.

[62] S. Marti, T.J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehaviour in mobile ad hoc networks. In *Proceedings of the 6-th annual International Conference on Mobile Computing and Networking, (MOBICOM)*, pages 255–265, Boston MA, USA, August 2000.

[63] P. Matheiu, B. Beaufils, and J.P. Delahaye. Iterated Prisoner's Dilemma Simulation Software, available at http://www.lifl.fr/IPD.

[64] R. Merkle. Protocols for public key cryptosystems. In *SIMMONS: Secure Communications and Asymmetric Cryptosystems*, 1982.

[65] P. Michiardi and R. Molva. Simulation-based analysis of security exposures in mobile ad hoc networks. In *Proceedings of the European Wireless Conference*, Florence, Italy, February 2001.

[66] P. Michiardi and R. Molva. Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proceedings of IFIP Communications and Multimedia Security Conference (CMS)*, Portoroz, Slovenia, September 2002.

[67] P. Michiardi and R. Molva. Report on a working session on security in wireless ad hoc networks. *Mobile Computing and Communication Review*, 6(4), 2002.

[68] P. Michiardi and R. Molva. Ad hoc network security. *ST Microelectronics Journal of System Research*, 2003.

[69] P. Michiardi and R. Molva. A game theoretical approach to evaluate cooperation enforcement mechanisms in mobile ad-hoc networks. In *Proceedings of the Workshop: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WIOPT)*, Sophia Antipolis, France, March 2003.

[70] P. Michiardi and R. Molva. Analysis of coalition formation and cooperation strategies in mobile ad hoc networks. *Elsevier - Ad hoc Networks Journal (Special Issue)*, 2004.

[71] P. Michiardi and R. Molva. *Chapter 12: Ad hoc network Security.* Mobile ad hoc networking. Wiley-IEEE Press, New York, NY, USA, 2004.

[72] P. Michiardi and R. Molva. Identity based hash chains for message authentication. Research Report RR-04-111, Institut Eurecom, July 2004.

[73] P. Michiardi and R. Molva. *Ad hoc network Security.* Handbook of Information Security. Wiley-IEEE Press, New York, NY, USA, 2005.

[74] J. Miller. The coevolution of automata in the repeated prisoner's dilemma. Working Paper 89-003, Santa Fe Institute, 1989.

[75] S.P. Miller, B.C. Neuman, J.I. Schiller, and J.H. Saltzer. Kerberos authentication and authorization system. Technical report, 1987.

[76] D.L. Mills. Precision synchronization of computer network clocks. *ACM Computer Communication Review*, May 1992.

[77] R. Molva and P. Michiardi. Security in ad hoc networks. In *Proceedings of Personal Wireless Communications Conference (PWC)*, Venice, Italy, September 2003.

[78] G. Montenegro and C. Castelluccia. Statistically unique and cryptographically verifiable (SUCV) identifiers and addresses. In *Proceedings of the Symposium on Network and Distributed Systems Security (NDSS)*, San Diego, CA, February 2002. Internet Society.

[79] U. Mueller. Optimal retaliation for optimal cooperation. *Journal of Conflict Resolution*, 31:692–724, 1988.

[80] L. Mui, M. Mohtashemi, and A. Halberstadt. Notions of reputation in multi-agents systems: a review. In *Proceedings of the 1-st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 280–287. ACM Press, July 2002.

[81] M.J. Osborne and A. Rubenstein. *A Course in Game Theory.* The MIT Press, Cambridge, Massachusetts, 1994.

[82] P. Papadimitratos and Z.J. Haas. Secure rotuing for mobile ad hoc networks. In *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*, January 2002.

[83] K.G. Paterson. ID-based signatures from pairings on elliptic curves. Report 2002/004, Cryptology ePrint Archive, January 2002.

[84] C.E. Perkins and E.M. Royer. Ad hoc on-demand distance vector (AODV) routing. In *Proceedings of the 2-nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, February 1999.

[85] A. Perrig, R. Canetti, D. Tygar, and D.X. Song. The TESLA Broadcast Authentication Protocol. *Cryptobytes (RSA Laboratories)*, 5(2), 2002.

[86] A. Perrig, R. Canetti, J.D. Tygar, and D.X. Song. Efficient Authentication and Signing of Multicast Streams over Lossy Channels. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 56–73, 2000.

[87] A. Perrig, Y. Hu, and D.B. Johnson. Wormhole protection in wireless ad hoc networks. Technical Report TR01-384, Dep. Of Computer Science, Rice University, 2001.

[88] A. Perrig, D.B. Johnson, and Y. Hu. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *Proceedings of The 8-th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, August 06 2002.

[89] A. Perrig, D.B. Johnson, and Y. Hu. SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks. In *Proceedings of the 4-th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, 2002.

[90] W. Poundstone. *Prisoner's Dilemma*. Oxford University Press, 1993. Doubleday, NY, '92.

[91] I. Pratt and J. Crowcroft. Peer-to-peer systems: Architectures and performance. In *Proceedings of Networking 2002 Workshops*, Pisa, Italy, May 2002.

[92] J. Ratliff. Game theory handouts, available at http://virtualperfection.com/gametheory.

[93] M.K. Reiter and S.G. Stubblebine. Authentication metric analysis and design. *ACM Journal: Transactions on Information and System Security*, 2(2):138–158, 1999.

[94] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, December 2000.

[95] P. Resnick and R. Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. *Advances in Applied Microeconomics: The Economics of the Internet end E-Commerce*, 11:127–157, November 2002.

[96] C.O Riordan and S. Bradish. The Voter's Paradox? Evolution of Cooperation in N-player games. Technical report, November 30 2000.

[97] R. Rivest. The MD5 Message-Digest Algorithm, RFC 1321.

[98] R. Rivest, A. Shamir, and L. Adelman. A method for obtaining digital signatures and public key cryptosystems. *Communications of ACM*, 21:120–126, 1978.

[99] L. Samuelson. Subgame perfection: An introduction. *Recent Developments in Game Theory*, pages 1–42, 1992.

[100] S. Saroiu, P.K. Gummadi, and S.D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proceedings of Multimedia Computing and Networking (MMCN)*, San Jose (CA), USA, Jan 2002.

[101] R. Schollmeier, I. Gruber, and M. Finkenzeller. Routing in mobile ad hoc and peer-to-peer networks. a comparison. In *Proceedings of Networking 2002 Workshops*, Pisa, Italy, May 2002.

[102] A. Shamir. HOW TO SHARE A SECRET. Technical Memo MIT/LCS/TM-134, Massachusetts Institute of Technology, Laboratory for Computer Science, May 1979.

[103] A. Shamir. Identity based cryptosystems and signature schemes. In *Advances in Cryptography*. ePress, 1984.

[104] L.S. Shapley. A value for n-person games. *Annals of Mathematical Studies*, 28:307–317.

[105] V. Srinivasan, P. Nuggehalli, C.F. Chiasserini, and R.R. Rao. Cooperation in wireless ad-hoc networks. In *Proceedings of The 22-nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOMM)*, San Francisco (CA), USA, Jan 2003.

[106] D.R. Stinson. *Cryptography: Theory and Practice*. Chapman and Hall/CRC, London, 2002.

[107] A. Urpi, M. Bonuccelli, and S. Giodano. Modelling cooperation in mobile ad hoc networks: A formal description of selfishness. In *Proceedings of the Workshop: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WIOPT)*, Sophia Antipolis, France, March 2003.

[108] H.R. Varian and J.K. Mackie-mason. Generalized vickrey auctions. Working paper, July 1994.

[109] H. Yang, X. Meng, and S. Lu. Self-organized network-layer security in mobile ad hoc networks. In *Proceedings of the ACM Workshop on Wireless Security (WISE)*, pages 11–20, New York, September 28 2002. ACM Press.

[110] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *Proceedings of The 22-nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOMM)*, San Francisco (CA), USA, Jan 2003.

[111] L. Zhou and Z.J. Haas. Securing ad hoc networks. *IEEE Network*, 13(6):24–30, 1999.

[112] S. Zhu, S. Xu, S. Setia, and S. Jajodia. Establishing pair-wise keys for secure communication in ad hoc networks: A probabilistic approach. Technical Rerport ISE-TR-03-01, George Mason University, March 2003.

[113] P. Zimmerman. *The Official PGP User's Guide.* prz@acm.org, 1994. The MIT Press In press.

Curriculum Vitae

Name:	Pietro Michiardi
Date of birth:	January, 24, 1975
Place of birth:	Torino, Italy
Languages:	English, French, Italian

Education

2001-2004	Ph.D. Student in Computer Science and Research Engineer Institut Eurecom, Corporate Communication Dept. Sophia Antipolis, France
2000-2001	Research Engineer Institut Eurecom, Corporate Communication Dept. Sophia Antipolis, France
1998-2000	M.Sc. double diploma in Advanced Telecommunication Systems Institut Eurecom, Multimedia Communication Dept. Sophia Antipolis, France
1995-2001	M.Sc. in Electrical Engineering Politecnico di Torino Tornio, Italy

Publications

- International conferences

P. Michiardi and R. Molva. Inter-domain authorization and delegation for business-to-business e-commerce. In *Proceedings of the eBusiness and eWork Conference and Exhibition*, Future Centre, Venice, Italy, October 2001.

P. Michiardi and R. Molva. Simulation-based analysis of security exposures in mobile ad hoc networks. In *Proceedings of the European Wireless Conference*, Florence, Italy, February 2001.

P. Michiardi and R. Molva. Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proceedings of IFIP Communications and Multimedia Security Conference (CMS'02)*, Portoroz, Slovenia, September 2002.

R. Molva and P. Michiardi. Security in ad hoc networks. In *Proceedings of Personal Wireless Communications Conference (PWC'03)*, Venice, Italy, September 2003.

P. Michiardi and R. Molva. A game theoretical approach to evaluate cooperation enforcement mechanisms in mobile ad-hoc networks. In *Proceedings of the Workshop: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt'03)*, Sophia Antipolis, France, March 2003.

- Research Reports

E. Altman, A. Kherani, P. Michiardi, and R. Molva. Non cooperative forwarding in ad hoc networks. Research Report RR-5116, INRIA Sophia Antipolis, February 2004.

P. Michiardi and R. Molva. Identity based hash chains for message authentication. Research Report RR-04-111, Institut Eurecom, July 2004.

- International Journals

P. Michiardi and R. Molva. Report on a working session on security in wireless ad hoc networks. *Mobile Computing and Communication Review*, 6(4), 2002.

P. Michiardi and R. Molva. Ad hoc network security. *ST Microelectronics Journal of System Research*, 2003.

P. Michiardi and R. Molva. Analysis of coalition formation and cooperation strategies in mobile ad hoc networks. *Elsevier - Ad hoc Networks Journal (Special Issue)*, 2004.

- Book Chapters

P. Michiardi and R. Molva. *Chapter 12: Ad hoc network Security.* Mobile ad hoc networking. Wiley-IEEE Press, New York, NY, USA, 2004.

P. Michiardi and R. Molva. *Ad hoc network Security.* Handbook of Information Security. Wiley-IEEE Press, New York, NY, USA, 2005.

- Submitted papers

E. Altman, A. Kherani, P. Michiardi, and R. Molva. Non cooperative forwarding in ad hoc networks. In *Submitted to IFIP Networking 2005*, 2004.

E. Altman, A. Kherani, P. Michiardi, and R. Molva. Some game-theoretic problems in wireless ad hoc networks. In *Submitted to EURO-NGI 2005*, 2004.

Awards

Best student paper award for the paper:
P. Michiardi and R. Molva. A game theoretical approach to evaluate cooperation enforcement mechanisms in mobile ad-hoc networks. In *Proceedings of the Workshop: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt'03)*, Sophia Antipolis, France, March 2003.

Patents

European Patent - EP015 "Process for providing Non Repudiation of Receipt (NRR) in an electronic transaction environment" Refik Molva, Pietro Michiardi