On Uncoordinated Wireless Ad-Hoc Networks: Data Dissemination over WIFI and Cross-Layer Optimization for Ultra Wide Band Impulse Radio

THÈSE N^{O} 4388 (2009)

PRÉSENTÉE LE 3 JUILLET 2009

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

LABORATOIRE POUR LES COMMUNICATIONS INFORMATIQUES ET LEURS APPLICATIONS 2

PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Alaeddine EL FAWAL

acceptée sur proposition du jury:

Prof. M. Hasler, président du jury
Prof. J.-Y. Le Boudec, directeur de thèse
Prof. C. Barakat, rapporteur
Dr E. Gauthier, rapporteur
Prof. P. Thiran, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2009

Abstract

Emerging pervasive wireless networks, pocket switched networks, Internet of things, vehicular networks and even sensor networks present very challenging communication circumstances. They might involve up to several hundreds of wireless devices with mobility and intermittent connectivity. Centralized coordination in such networks is practically unfeasible. We deal with these challenge using two potential technologies: WIFI and Ultra Wide Band (UWB) Impulse Radio (IR) for medium and short communication range, respectively. Our main goal is to improve the communication performance and to make these networks sustainable in the absence of a centralized coordination.

With WIFI, the goal is to design an environment-oblivious data dissemination protocol that holds in highly dynamic unpredictable wireless ad-hoc networks. To this end, we propose a complete design for a scope limited, multi-hop broadcast middleware, which is adapted to the variability of the ad-hoc environment and works in unlimited ad-hoc networks such as a crowd in a city, or car passengers in a busy highway system. We address practical problems posed by: the impossibility of setting the TTL correctly at all times, the poor performance of multiple access protocols in broadcast mode, flow control when there is no acknowledgment and scheduling of multiple concurrent broadcasts. Our design, called "Self Limiting Epidemic Forwarding" (SLEF), automatically adapts its behavior from single hop MAC layer broadcast to epidemic forwarding when the environment changes from being extremely dense to sparse, sporadically connected. A main feature of SLEF is a non-classical manipulation of the TTL field, which combines the usual decrement-when-sending to many very small decrements when receiving.

Then, we identify vulnerabilities that are specific to epidemic forwarding. We address broadcast applications over wireless ad-hoc networks. Epidemic forwarding employs several mechanisms such as forwarding factor control and spread control, and each of them can be implemented using alternative methods. Thus, the existence of vulnerabilities is highly dependent on the methods used. We examine the links between them. We classify vulnerabilities into two categories: malicious and rational. We examine the effect of the attacks according to the number of attackers and the different network settings such as density, mobility and congestion. We show that malicious attacks are hard to achieve and their effects are scenario-dependent. In contrast, rational attackers always obtain a significant benefit. The evaluation

is carried out using detailed realistic simulations over networks with up to 1000 nodes. We consider static scenarios, as well as vehicular networks.

In order to validate our simulation results, we build a solid and widely adaptable experimental testbed for wireless networks. It is composed of 57 mobile wireless nodes equipped with WIFI interface. The adopted platform is OpenWrt, a Linux-like firmware, which makes the testbed robust and easily configurable.

With UWB IR, the main problem we deal with is the presence of uncontrolled interference. Indeed, similarly to Code Division Multiple Access (CDMA) systems, signal acquisition with UWB IR signaling requires power control in the presence of interferers, which is very expensive in an uncoordinated system. We solve this problem through a cross-layer optimization: We propose a new signal acquisition method that is independent of the received signal power and we adapt the MAC layer accordingly. Our signal acquisition method is designed to solve the IUI (Inter-User Interference) that occurs in some ad-hoc networks where concurrent transmissions are allowed with heterogeneous power levels. In such scenarios, the conventional detection method, which is based on correlating the received IR signal with a Template Pulse Train (TPT), does not always perform well. The complexity of our proposal is similar to that of the conventional method. We evaluate its performance with the Line Of Sight (LOS) and the Non-LOS (NLOS) office indoor-channel models proposed by the IEEE P802.15.4a study group and find that the improvement is significant. We also investigate the particular case where the concurrent transmissions have the same time-hopping code, and we show that it does not result in collision, such scenarios appear in ad-hoc networks that employ a common code for control or broadcast purposes.

At the MAC level, we focus only on one component of a MAC layer, which is the sleeping mode that could be added to any MAC layer proposal adequate to UWB IR. We are motivated by the low power consumption constraint required by the potential applications. We identify the design elements that should be taken into account for an optimal design for a sleeping protocol for UWB-IR such as the possibility of transmitting concurrently without collision and the power consumption model of the hardware behind which is completely different than with the narrow-band signaling. Then, we design two sleeping protocols for centralized and decentralized ad-hoc networks, respectively. We evaluate their performance analytically with the adopted metric being the average life-time of the wireless nodes.

Keywords

Data dissemination, epidemic forwarding, ad-hoc networks, spot applications, forwarding factor control, spread control, adaptive TTL, congestion control, pseudo-broadcast, SLEF, Haggle, ultra wide band, signal acquisition, inter-user interference, concurrent transmissions, power saving.

Résumé

Les réseaux sans fil omniprésents, les réseaux de poches, l'Internet des objets, les réseaux véhiculaires et même les réseaux de capteurs présentent des conditions de communication très difficiles. Ils pourraient inclure jusqu'à plusieurs centaines de nœuds mobiles sans fil avec une connectivité intermittente. Une coordination centralisée des tels réseaux est pratiquement impossible. Nous traitons ces défis en utilisant deux technologies potentielles: WIFI et la Radio Impulsives (RI) à Bandes Ultra-Large (ULB) pour des portées moyenne et courtes, respectivement. Notre objectif principal est d'améliorer la performance de communication et de rendre ces réseaux viables en l'absence d'une coordination centralisée.

Avec WIFI, l'objectif est de concevoir un protocole de diffusion de données adapté aux réseaux ad-hoc sans fil qui sont très dynamiques et imprévisibles. A cette fin, nous proposons un middleware de diffusion multi-saut d'une portée limitée, qui est adaptée à la variabilité de l'environnement ad-hoc et qui fonctionne dans des réseaux ad-hoc de grande échelle tel qu'une foule dans une ville, ou des passagers des voitures sur l'autoroute. Nous abordons les problèmes posés par l'impossibilité de toujours régler le TTL correctement, la mauvaise performance en mode diffusion des protocoles d'accès multiple, le contrôle de flux en l'absence d'acquittements et l'arrangement des diffusions simultanées. Notre protocol, appelé " Self Limiting Epidemic Forwarding " (SLEF), s'adapte automatiquement d'une diffusion à saut unique au niveau de la couche MAC, à un transfert épidémique lorsque l'environnement passe d'une densité forte à une densité faible avec une connectivité sporadique. L'une des principales caractéristiques de SLEF est une manipulation non-classique du TTL, qui combine la décrémentation habituelle lors de la transmission à des décrémentation beaucoup plus faible lors de la réception.

Ensuite, nous identifions les vulnérabilités qui apparaissent en utilisant le transfert épidémique. Nous considérons des applications de diffusion dans les réseaux ad-hoc sans fil. Le transfert épidémique utilise plusieurs mécanismes, comme le contrôle du facteur de transfert et la contrôle de la portée de propagation, et chacun d'entre eux peut être mis en uvre par des méthodes alternatives. Ainsi, l'existence de vulnérabilités est fortement dépendante des méthodes utilisées. Nous examinons les liens entre les vulnérabilités et les méthodes utilisées. Nous classons les vulnérabilités en deux catégories: les malveillantes et les rationnelles. Nous examinons l'effet des attaques en fonction du nombre d'attaquants et les différents paramètres

du réseau, tels que la densité, la mobilité et la congestion. Nous montrons que les attaques malveillantes sont difficiles à réaliser et leurs effets dépendent du scénario. En revanche, les attaquants rationnels obtiennent toujours un avantage important. L'évaluation est effectuée à travers des simulations réalistes et détaillées des réseaux contenant jusqu'à 1000 nœuds. Nous considérons les scénarios statiques, ainsi que les réseaux véhiculaires.

Afin de valider nos résultats de simulation, nous construisons un testbed expérimental qui est robuste et largement adaptable pour les réseaux sans fil. Il est composé de 57 nœuds sans fil mobiles équipés chacun d'une interface WIFI. La plate-forme adoptée est OpenWrt, un firmware similaire à Linux, ce qui rend le testbed robuste et facilement configurable.

Avec RI ULB, le principal problème que nous traitons est la présence de l'interférence non-contrôlée. En effet, l'acquisition de signal avec RI ULB exige le contrôle de puissance en présence d'interférences, ce qui est très cher dans un système de coordination. Nous résolvons ce problème par le biais d'une optimisation multi-couche: Nous proposons une nouvelle méthode d'acquisition de signal, qui est indépendante de la puissance du signal reçu, et nous adaptons en conséquence la couche MAC. Notre méthode d'acquisition de signal est destinée à résoudre l'interférence entre utilisateurs qui se produit dans certains réseaux ad-hoc où les transmissions simultanées sont permises avec des niveaux de puissance hétérogènes. Dans des tels scénarios, la méthode de détection conventionnelle, qui est fondée sur une corrélation entre le signal du RI reçu avec un modèle de série d'impulsions, ne performe pas toujours bien. La complexité de notre méthode est similaire à celle de la méthode conventionnelle. Nous avons également considéré le cas particulier où les transmissions suméltannées ont le même code de saut de temps, et nous montrons qu'il ne résulte pas en collision, des tels scénarios apparaissent dans des réseaux ad-hoc qui utilisent un code commun pour le contrôle ou la diffusion.

Au niveau de la couche MAC, nous nous concentrons seulement sur un composant d'une couche MAC, qui est le mode d'économie de puissance. Ce composant pourrait être ajouté à toute proposition de la couche MAC adaptée à RI ULB. Notre motivation est la contrainte de la faible consommation d'énergie requise par les applications potentielles. Nous identifions les éléments de conception qui devraient être pris en compte pour une conception optimale d'un protocole d'économie de la puissance pour RI ULB, tels que la possibilité de transmettre simultanément sans collision et le modèle de la consommation d'énergie du matériel éléctronique derrière qui est complètement différent de celui des systèmes à bande étroite. Ensuite, nous

concevons deux protocoles d'économie de puissance pour les réseaux ad-hoc centralisés et décentralisés, respectivement. Nous évaluons leurs performances par analyse, tout en adoptant la durée de vie moyenne des nœuds sans fil comme métrique de mesure.

Mots-Clés

Dissémination de données, diffusion épidémique, réseaux ad-hoc, applications spot, contrôle de facteur de diffusion, contrôle de portée, TTL adaptatif, contrôle de congestion, pseudo-broadcaste, SLEF, Haggle, bande ultra large, acquisition de signal, interférence entre-utilisateurs, transmissions simultanées, économie de puissance.

Acknowledgment

First, I would like to thank my advisor, Professor Jean-Yves Le Boudec, for giving the great chance to make a PhD under his guidance and for making it such an enriching experience. I am deeply grateful to him for his availability and help, not only in research, but also in all other aspects of the PhD. I am also very grateful to him for taking so much care of his student's personal career and promoting their results.

It was a great pleasure and a humbling experience to have Prof. Chadi Barakat, Dr. Eric Gauthier, Prof. Martin Hasler and Prof. Patrick Thiran in my jury. I want to thank them for accepting to review this thesis and for the interest they demonstrated with respect to my work.

Next, I would like to thank all people in my lab for making it a friendly and a lively working place. I am indebted to Adel Aziz for his help and support. I had the chance to have a joint project with him. Beside enriching, working with him was a pleasure. I also want to acknowledge my office mate, Irina Baltcheva, for her kindness, for bearing my mood and for taking care of the plants!!!

I also thank all hagglers with whom I had seminal meetings and discussions, and enjoyable dinners!!!

I am very obliged to the lab's staff, Danielle Alvarez, Holly Cogliati, Angela Devenoge, Patricia Hjelt, Hervé Chabanel, Philippe Chammartin, Jean-Pierre Dupertuis, Yves Lopez and Marc-André Lüthi for making everything work so smoothly.

Finally, I am immensely indebted to my parents for their love and support throughout my everlasting studies, and for the thirst for knowledge they infected me with.

Last but not least, I would like to express my deep gratitude to my fiancée, Sally, for her patience, for sharing with me deepest moments of joy, and for supporting me and encouraging me whenever fear or discouragement where looming over me.

Contents

Chapter 1

Introduction

In this dissertation, we investigated different facets of wireless ad-hoc networks. The main goal is to make wireless networks sustainable in fully self-organized environments. This work was supported by two different projects: Haggle [1] and MICS [8]. Consequently, we had to work with two different technologies: WIFI and Ultra Wide Band (UWB) Impulse Radio (IR). With both projects and technologies, the main challenge to address was the absence of a centralized coordination and being self-organized.

With WIFI, the goal was to come up with an environment-oblivious data dissemination protocol that holds in highly dynamic, unpredictable networks. These networks alternate quickly from connected to disconnected, from dense to sparse or congested to non-congested with intermittent connectivity. In the absence of a centralized coordination, gathering information about network conditions is impractical. Therefore, the data dissemination protocol should be autonomic, that is able to adapt itself according to network changes.

With UWB IR, the main problem we dealt with is the presence of uncontrolled interference. Indeed, similarly to Code Division Multiple Access (CDMA) systems, signal acquisition with UWB IR signaling requires power control in the presence of interferers, which is very expensive in uncoordinated systems. We solve this problem through a cross-layer optimization: We propose a new signal acquisition method that is independent of the received signal power and we adapt the MAC layer accordingly.

This dissertation consists of two parts. The first part describes our contributions on data dissemination on top of WIFI interfaces. The second part describes our cross-layer optimiza-

tion for UWB IR systems.

1.1 Data Dissemination for Spot Applications

Over the last few years, WIFI interfaces have been expanded dramatically. They come with many personnel devices such as laptops, PDAs, mobile phones, video games and even with peripherals and vehicles. WIFI starts to play an important role in everyday life. It is very frequent to find people with more than one WIFI interface, one in the mobile phone, another in a laptop, and may be a third in their vehicle. This expansion dramatically changes the ad-hoc network semantic.

This change in the ad-hoc environment opens the door to new applications and communication paradigms. Within this new ad-hoc environment that we call an open-ended network, we are interested in Spot applications. Such applications aim at making a spot around the source and, in order to disseminate their packets within the spot, require from the underlying layers a multi-hop broadcast service. Hence, the destination of a packet is all the nodes within the application spot. With very limited ad-hoc networks, e.g. in a campus building, the spot could be the entire network. In contrast, the spot is a very small part of a network in an open ended-environment such as the population on the highway. Spot-applications could be a free alternative to costly cellular services. Typical applications are traffic information dissemination, chat on-the-highway, bulletin board, mobiclick or the bootstrapping phases for routing protocols such as spray routing family [67, 68] for Disruption Tolerant ad-hoc Nnetworks (DTNs).

Spot applications are supposed to be deployed in a wide range of network settings ranging from very dense to very sparse, with different traffic loads and types of mobility. Potential users are people with vehicles in rural or urban environments, and people in stadiums, at ski stations, at festivals, in enterprises or on university campuses or simply walking or sitting anywhere in towns. This variation in scenarios exhibits very challenging communication conditions. For instance, intermittent connectivity resides in all the aforementioned scenarios: either because of mobility or because people switch their devices off from time to time to save energy. Once users get in contact with each other, they require receiving all the packets that have circulated before in the network. Also, a node that has the ease of transmitting in a non-congested sparse

network, should be able to adapt its dissemination rate and deal with collisions when it looses this opportunity while moving to a congested network. This is the case of a vehicle that is suddenly stuck in a traffic jam. Therefore, the data dissemination protocol beneath the spot applications must hold in all circumstances.

Throughout this work, we aim at finding an adequate combination of protocol elements and adjusting their parameters in order to ensure the sustainability of spot-applications in all environments. As broadcast functionality is required, we decided on an epidemic-forwarding-like protocol. In the literature, epidemic forwarding is proposed either for point-to-point (pt-2-pt) communication or for data dissemination. With pt-2-pt communication, epidemic forwarding acts as a routing mechanism that routes a packet to its final destination. With data dissemination, the semantic is very close to Spot-applications, the destination is all nodes within the spot of an application. The spot size is fixed; it is either the entire network or limited to a predefined geographical area, as in vehicular networks [17, 49–51, 60]. Two metrics are used to evaluate epidemic forwarding protocols: the delivery ratio and the forwarding factor (which is how many times in average a node forwards a packet). The best epidemic forwarding protocol is the one that has the highest delivery ratio with the smallest forwarding factor, that is the smallest amount of redundancy.

In the Spot-application context, we keep the spirit of the epidemic forwarding, which is the multi-hop broadcast, but have a different goal and use a different communication paradigm. First, we do not consider the delivery ratio as a metric, but we consider the application spot size instead. Indeed, as explained in Sect. 3.2.3, increasing the spot size would be at the expense of the application rate. Therefore, the spot size should be controlled and adapted to the network conditions in order to insure a minimum required rate to the application. Second, the mobility and the intermittent connectivity should be taken into account while managing redundancy and setting the forwarding factor. Forwarding a packet several times does not necessarily result in redundant transmissions. Due to mobility and intermittent connectivity, a node always makes new contacts; and passing packets to these contacts is desirable if it is allowed by the network conditions, whereas a node is allowed to forward a packet once at most with data dissemination work in the literature.

Beside controlling the spot size and the forwarding factor, offering an integral data dissemination service to the Spot applications requires additional protocol elements. In the presence

of multiple sources in a network, fairness is an issue. A local scheduler is needed to apply a fairness policy and arbitrate among the packets existing in the epidemic buffer and belonging to different sources. Also, congestion control mechanism is needed to adapt the application rate to the network conditions. And last, special care should be given to the 802.11 broadcast mechanism because efficiency and reliability are lacking: Indeed, it does not use any mutual exclusion mechanism, which results in a high probability of collision. And, it does not employ any acknowledgment mechanisms. A source cannot know whether its packet is received by other nodes, it undergoes a collision or it is simply transmitted in the vacuum.

1.2 Cross-Layer Optimization for UWB IR Networks

Being impulsive with a low-duty cycle, UWB IR possesses key features that make it viable for high quality, fully mobile, short range, indoor radio systems such as Internet of things, pervasive and sensor networks.

First, the fact that an impulse radio system operates in the lowest possible frequency band that supports its wide transmission bandwidth means that this radio has the best chance of penetrating materials that tend to be more opaque at higher frequencies. Furthermore, the effect of multipath fading, the bane of sinusoidal systems, is much less in IR systems than in the conventional radio systems. In fact, Rayleigh fading, so noticeable in cellular communications, is a continuous wave phenomenon, not an IR communication phenomenon. In UWB IR, the multipath is resolvable down to path differential delays on the order of the pulse width. This reduces significantly the fading effect and, thus, UWB IR is inherently suitable for indoor and harsh environments where multipath propagation is an issue.

Second, the lack of significant multipath fading may considerably reduce fading margins in link budgets and allow for a low transmission-power operation. Low transmission-power and short-range operation with this ultra-wide bandwidth result in an extremely low transmitted power spectral density, which ensures that impulse radios cannot interfere with narrow-band radio systems operating in dedicated bands [13]. This is an important issue as Internet of things and sensor networks might involve hundreds of nodes, which entails an unaffordable interference in the case of a high level of radiated power, and it breaks environmental and health constraints.

Third, manufacturers claim low cost of hardware implementations of UWB IRs with low power consumption. This is of crucial importance for applications such as sensor networks, where tiny nodes operating on batteries are deployed and they are expected to serve for several months.

Forth, UWB IR allows for concurrent transmissions even to the same destination and does not need a mutual exclusion mechanism. This is in a sharp contrast with the narrow-band systems where concurrent transmissions result in collisions. Therefore, UWB IR is adequate for dense scenarios with fully-self-organized and uncontrolled networks, where narrow-band systems lead to a network failure.

For all these reasons, the UWB IR physical layer has been chosen for the IEEE 802.15.4a [44] amendment to IEEE 802.15.4 [22, 43], a standard that targets low data-rate wireless networks with extensive battery life and very low complexity.

However, with UWB IR, we have to deal with two problems to ensure the good performance of UWB IR networks. The first concerns the physical layer due to the inherent characteristic of the physical nature of the UWB IR signal that is impulsive with low duty cycle; this imposes a long synchronization time and makes carrier sensing impossible. The second issue is to define a MAC layer that takes advantage of the benefits of the UWB IR signaling. For instance, the MAC layer should be able to manage concurrent transmissions on the same channel (see Sect. 7.6.3), which is impossible with narrow band signaling.

In the literature, both issues are treated independently from each other, which results in a sub-optimal solution; the first aspect is a physical layer problem whereas the second concerns in particular the MAC (Medium Access Control) and upper layers. Our work differs in that we study the interaction between both aspects in order to eventually come up with an optimal employment of the UWB IR signaling in a network with uncoordinated MAC. The synchronization is the first challenge in our application where a node achieves synchronization each time it receives a packet, unlike centralized networks where a global synchronization is ensured by the access point. Therefore, the time needed to accomplish this synchronization plays an important role in the performance of the network. The bane of a fast synchronization is the extreme Inter-User Interference (IUI) case (near-far problem), when there are multiple interfering transmitters, asynchronous transmissions and heterogeneous power levels. This occurs for example in the presence of multiple interfering piconets, or in purely ad-hoc networks

that allow concurrent transmissions, always at full power [2], [3]. As a typical example, we can imagine a headphone set employing the IR UWB technology to exchange music with some master device such as a laptop. Several people may use several headphones with different masters in an office environment or even in the same room. They may move or exchange places, which creates very harmful interference. Another application could be the sensor networks. We can imagine tens or even hundreds of sensors deployed in a small area communicating with each other in an ad-hoc fashion with a huge amount of interference. Using the conventional synchronization method and in the presence of the IUI, an extremely long synchronization preamble is needed, which constitutes an unaffordable overhead and increases dramatically the synchronization time.

The second challenge concerns the MAC level. We focus only on one component of a MAC layer, which is the sleeping mode that could be added to any MAC layer proposal adequate to UWB IR such as the protocols proposed in [19, 54]. We are motivated by the low power consumption constraint required by the potential applications. Existing work on sleeping protocols concerns narrow-band systems and results in resource wasting and inefficient power saving if applied with UWB-IR. An optimal sleeping protocol design should take into account several elements, some of them are specific to UWB-IR signaling, such as the possibility of transmitting concurrently without collision and the power consumption model of the hardware behind which is completely different than with the narrow-band signaling. For instance, with UWB-IR, transmitting consumes less power than listening to the channel, which is in sharp contrast to the sinusoidal signaling systems (see Sect. 8.2.3).

1.3 Contributions

Multi-Hop Broadcast Middleware:

- **Publications:** One conference paper [32].

We have designed a multi-hop broadcast middleware that we call Self Limiting Epidemic Forwarding (SLEF). SLEF matches the Spot-application requirements. SLEF is a middleware that is functional between the application and sockets (UDP sockets or directly on top of raw sockets). It offers an efficient and reliable data dissemination service to the application, and

it does not need any additional mechanism to be functional. SLEF is environment oblivious: it does not need or exchange any topology information and furthermore, it does not use handshaking among nodes. This is of crucial importance in highly dynamic networks where the contact time is very short. SLEF implements six mechanisms that are indispensable for a data dissemination service:

1. **Forwarding factor control:** We propose a new forwarding factor control mechanism that is adaptive and deals with the intermittent connectivity challenge (see Sect. 4.2.1).

2. **Spread control:** The spread of a packet is the number of nodes that receive a copy of this packet. Thus, controlling the spread is equivalent to controlling the spot-size. We identify the spread-rate trade-off. Increasing the spread happens at the expense of the application rate. We deal with this trade-off by proposing a spread control mechanism that maintains an adequate balance between both elements of this trade-off. The goal is to keep the spread as large as possible while maintaining a minimum required rate to the application. Our mechanism is adaptive and it is obliviously sensitive to the congestion in the network and to node density (see Sect. 3.2.3).

3. **Fairness:** to the best of our knowledge, our work is the first that considers the fairness issue with epidemic forwarding. We propose a scheduler that ensures source-based fairness by serving packets per source ID using a processor sharing approach (see Sect. 3.2.4).

4. **Congestion control:** This issue has not been discussed before with epidemic forwarding. Such a mechanism faces major challenges such as being in ad-hoc mode, the communication being in broadcast and being oblivious. We consider these challenges and others, and we design the first congestion control mechanism proposed for epidemic forwarding (see Sect. 3.2.5).

5. **Buffer management:** It is needed in order to keep space for new incoming packets in case the epidemic buffer is full. It drops packets that are judged useless. As it is shown in Sect. 5.4.2, such a mechanism has a major impact on the performance of epidemic forwarding. For instance, dropping packets that have the smallest TTL results in very poor

performance. Our buffer management adopts the aging scheme defined in Sect. 3.2.3 in order to judge packets useless and to drop them (see Sect. 3.2.6).

6. **Efficient and reliable MAC broadcast:** 802.11 broadcast is known to be unreliable and inefficient. We deal with this issue by proposing enhancing mechanisms, that are the pseudo-broadcast and the presence indicator (see Sect. 3.2.7). Our proposal does not require any changes to the 802.11 driver.

Vulnerabilities in Epidemic Forwarding:

- **Publications:** One conference paper [33].

We identify vulnerabilities that are specific to epidemic forwarding over wireless ad-hoc networks. We classify these vulnerabilities into two categories: malicious and rational. The malicious does not look for a personal benefit but aims to harm other nodes. In contrast, the rational seeks to increase its personal profit from the network. We evaluate their impact according to the number of attackers and the different network settings. We find that the impact of malicious attacks depends on the position of the attacker relative to the victim, the network density, the traffic load and mobility. In static scenarios, we identify the attacks that reduce dramatically the victim spread, whereas the harm of other attacks is reduced due to the adaptive forwarding factor control and the injection rate control. In highly mobile vehicular network, the impact of malicious attacks are minimized due to the spread control. We have studied the rational case in presence of only one attacker in the network. The attacker could achieve considerable profit in all scenarios (see Chap. 4).

SLEF Validation and Experimental Testbed:

1. **SLEF validation:** To prove that SLEF is functional and it does not need assistance from other middlewares, we implemented it and we ported it for four platforms that are Windows XP, Windows Mobile, Linux and OpenWrt. Further, we demonstrate its feasibility on very constrained resource devices such as smartphones and wireless routers with a flash memory of $8MB$, a RAM of $32MB$ and a microprocessor running with a clock of 266MHz (see Sect. 5.2).

2. **Experimental testbed:** We want to go one step further in validating SLEF. In order to stress test SLEF, we build a solid and widely adaptable experimental testbed for wireless networks. It is composed of 57 wireless routers running the 802.11 MAC protocol. Mobility is ensured by adding batteries to nodes. We run SLEF on these devices (see Sect. 5.3). The stress testing consist of running SLEF at a full rate for several hours on 50 resource-constrained devices getting in contact of each other. The stress test was successful. Further, we compare SLEF to fixed TTL based epidemic forwarding. SLEF performs better than fixed TTL for several reasons such as: (1) SLEF is adaptive, whereas fixed TTL needs to find the good buffer size for each scenario, otherwise entailing a huge amount of redundancy, and (2) SLEF uses buffer management based on aging, whereas fixed TTL uses TTL-based buffer management, which performs poorly (see Sect. 5.4.2).

Robust Signal Acquisition in UWB ad hoc networks

- **Publications:** One journal paper [30], one conference paper [36] and one patent [29].

1. **Problem Identification:** We identify and explain the problem that arises using the conventional signal detection method (see Sect. 7.3), which correlates the received UWB Impulse Radio (IR) signal with a Template Pulse Train (TPT) and performs a threshold check on the output of the correlation; we show that the synchronization is either unfeasible or entails an extremely large overhead due to the Inter-User Interference (IUI) in the predescribed scenarios.

2. **A New Signal Detection Method:** In order to solve the extreme IUI case (near-far problem), we propose a new detection method, which we call a Power Independent Detection (PID) method. Our PID method solves the problem without any additional complexity overhead, e.g. for a digital receiver, it employs the same sampling frequency and number of operations as the conventional detection method (see Sect. 7.4).

3. **Performance Evaluation:** Based on analysis and simulations, we evaluate the performance of the PID detection method. The simulations were carried out according to the Line Of Site (LOS) and the Non-LOS (NLOS) indoor office channel model proposed by the IEEE P802.15.4a study group [14]. The adopted metrics are (1) the probability of

Missed Detection (P_{MD}) (2) the probability of false alarm (P_{FA0}) and (3) the total error defined as $E_t = P_{MD} + P_{FA0}$. The results show a significant improvement compared to the conventional detection method. Some of our results show that, for a bit energy to noise spectral density ratio, $E_0/N_0 = 15dB$, and in the presence of 10 users transmitting simultaneously, $E_t = 10^{-8}$ with the PID method whereas E_t is almost 1 with the conventional detection method (see Sect. 7.6).

Sleeping Mode for UWB IR Ad-hoc Networks

- **Publications:** One journal paper [31] and one conference paper [55].

1. **Design elements:** We identify the design elements that should be taken into account for an optimal design for a sleeping protocol for UWB-IR (see Sect. 8.2).

2. **Protocol design:** We design two sleeping protocols for centralized and decentralized ad-hoc networks, respectively. We evaluate their performance analytically with the adopted metric being the average lifetime of the wireless nodes. We could show that slotted sleeping is better than unslotted if occasional bursts must be supported. In contrast, unslotted sleeping is better than slotted if occasional maximum latency must be supported (see Sect. 8.3).

Part I

Data Dissemination

Chapter 2

Related Work

2.1 Related Work

Epidemic-style forwarding has been proposed as an approach to achieve system-wide dissemination of messages. With epidemic-style forwarding, the destination of a packet is an entire network, nodes within a few hops away from the source, nodes within a geographical area, or only one node where the epidemic-style forwarding is used to replace or assist routing protocols in Disruption Tolerant ad-hoc Networks (DTNs) or highly mobile networks. It evolves similarly to an infectious disease. An infected node (that has a message) encounters new nodes and may decide to infect them, i.e. to pass them the message.

An unconstrained epidemic forwarding scheme (in which an infected node spreads the messages to all nodes it encounters) is able to achieve minimum delivery delay at the expense of an increased use of resources such as buffer space, bandwidth and transmission power. Variations of epidemic forwarding have been recently proposed in order to exploit the trade-off between delivery delay and resource consumptions. We classify them according to the design elements we consider with SLEF.

2.1.1 pt-2-pt vs. Data Dissemination

We refer to pt-2-pt in the case where the destination is a single node. In contrast, we use the term data dissemination to express the case of multi-destination nodes.

In [56, 59], epidemic forwarding is used for data dissemination to all nodes in the network.

With these schemes, the adopted metric is the delivery ratio defined as the ratio of nodes that receive a packet to all nodes in the network.

Other proposals of epidemic forwarding aim at routing a message to its final destination in a pt-2-pt communication paradigm [16, 20, 24, 40, 41, 52, 53, 71]. Some of them [16, 20, 24, 53, 71] are proposed for DTN or quickly varying environments, where mobility and self-organization make the classic methods based on distributions trees non-practical. They act as routing protocols.

In between, there exists a third category where epidemic forwarding is limited to a part of a network. For instance, this is the case of vehicular networks with safety message dissemination where emergency messages are spread in the geographical area surrounding an accident or an abnormal vehicle [17, 25, 26, 34, 49–51, 60].

2.1.2 Broadcast vs. Unicast

Spreading messages is accomplished either through broadcast or unicast transmissions. With data dissemination, a node forwards a message to all its neighbors, which minimizes the overhead and decreases the delivery delay [17, 25, 26, 34, 49–51, 56, 59, 60]. In contrast, in pt-2-pt communication, where epidemic forwarding acts as a routing protocol, some proposals [40, 41, 52] use broadcast and others unicast [16, 20, 24, 53, 67, 68, 71].

2.1.3 Forwarding-Factor Control and Topology Information

Forwarding-factor control is the key design element in almost all epidemic forwarding mechanisms proposed in the literature. It decides on the forwarding of a packet or not. It aims at minimizing redundancies and thus, at saving resources such as bandwidth, transmission power and buffer space.

Environment-Oblivious Forwarding: Some epidemic forwarding control mechanisms base the forwarding decision on only their local attributes. We refer to them as environment oblivious. For instance, some mechanisms, proposed and discussed in [40, 41, 59], belong to this class. In [40, 41], a node decides with a fixed probability on the forwarding of a packet that it has received. In [59], the forwarding-factor control is adaptive where forwarding a packet is

canceled if it is overheard a fixed number of times. In both work, a packet is forwarded once at most.

Environment-Aware Forwarding: Other proposals are environment aware. They make use of topology information and require distributed mechanisms to gather it. In [40, 41], topology information are assumed to be available to nodes while making a forwarding decision but, the authors do not propose a mechanism to collect this information. In [17, 25, 26, 34, 49–52, 60], the authors assume that each node is equipped with a Global Positioning System (GPS) and thus, it has an access to its position. In [52], knowing the positions of the source and the destination, the nodes in between forward a packet with a fixed predefined probability. With vehicular networks, the authors in [17] build a global overview of the network topology using GPS devices and exchanging node positions. Hence, the furthest node from the sender is selected to forward a packet. In [25, 26, 34, 49–51, 60], a node needs to know its own position and the sender's. Then, it uses a back-off like mechanism where the contention window is inversely-proportional to the distance of the node to the sender. Thus, the furthest node has the highest chance of being the first to forward the packet. Overheard packets are removed from the local buffer, as forwarding them is assumed to be redundant. With vehicular networks [17, 25, 26, 34, 49–51, 60], the data dissemination is unidirectional and it is not suitable for DTNs. In [56], the authors propose a joint framework for clustering and data dissemination. Once clustering is established, data dissemination is optimized. But clustering is not free and entails a huge amount of overhead in dynamic networks, because a node requires two-hop topology information before joining a cluster, and any change in the network requires that a node has d-hop topology information, where d is the cluster diameter. It is clear this work does not support DTNs.

Content-Aware Forwarding: The remaining proposals are about content-aware forwarding. With this class, the communication is pt-2-pt and the transmission is in unicast mode. Forwarding methods within this class builds a prediction model for the likelihood that forwarding to any particular encountered node will result in the delivery of a message to its final destination. A source injects several copies of its message in a network. It chooses the relays based on their attributes so that they maximize the likelihood of delivery to its final destination. Used attributes could be node history or social labels. This class assumes hand-shaking

between nodes before deciding on the forwarding. A relay forwards a message to its final destination or to a node that has a higher chance of encountering the destination. The proposals in [16, 20, 53, 67, 68, 71] belong to this class and are proposed mainly for DTNs.

2.1.4 Spread Limit

With pt-2-pt communication, a packet spread is limited either by fixing the hop-count [24, 71] or by limiting explicitly the number of copies of the packet in a network [67, 68], in other words, by limiting the number of relays that are expected to deliver the packet to its final destination.

Limiting the spread with data dissemination is addressed only with vehicular networks [17, 25, 26, 34, 49–51, 60] where a message spread is limited to a predefined geographical area. For instance, a safety message is valid only within a few kilometers from the source. Each node knows (using GPS) its own position and the source one and, if it is beyond the validity distance, it drops the message when it receives it.

In both cases, limiting the spread is predefined and fixed. It is independent of the network conditions.

2.1.5 Congestion Control and Fairness

To the best of our knowledge, none of the epidemic forwarding mechanisms proposed in the literature has addressed congestion control, i.e. controlling the application rate, or fairness. They all focus on saving resources and on message delivery latency. The study cases are about injecting one message in a network and studying its dissemination. Thus the need for congestion control and fairness has not appeared. But in reality, the presence of several flows by different sources makes crucial both issues to address with epidemic forwarding.

2.1.6 Broadcast Issues

The aforementioned work on data dissemination assumes broadcast service from the MAC layer. Some of them consider reliable and efficient broadcast where all sender neighbors receive the broadcast. In reality, this is not the case. For instance, the 802.11 MAC layer is widely used in wireless networks, but its broadcast mechanism is neither reliable nor efficient. Indeed

it does not involve any mutual exclusion mechanisms to prevent collisions. Further, it does not use an acknowledgment mechanism and thus, a node can not know whether its packet is received by other nodes, it undergoes a collision or simply transmitted in the vacuum. Among the aforementioned work, only one proposal addresses this issue. The authors [49–51] came up with their own MAC layer broadcast that implements a mutual exclusion, similar to the RTS/CTS one adopted by 802.11 with unicast transmissions. But their mechanism is suitable only for unidirectional communications.

However, this issue has been addressed independently of data dissemination. The work in [69, 70], and the references therein, propose improvements to 802.11 broadcast by adding mutual exclusion and by forcing all neighbors to acknowledge broadcast packets. They require modifying the 802.11 driver.

2.1.7 Why SLEF?

Among the aforementioned work on data dissemination, open-ended networks are addressed only with vehicular networks where the proposed mechanisms are not of general use: they are unidirectional and assume all nodes are equipped with a GPS. Further, none of the aforementioned mechanisms are able to deliver a sustainable data dissemination service, as they miss key design elements such as fairness, congestion control and buffer management.

SLEF is designed for open-ended environments with intermittent connectivity. It addresses all required design elements. With SLEF, packets are transmitted in broadcast mode, as the destination is all the sender neighbors. SLEF employs an adaptive forwarding factor control. In a very dense scenario, a packet has little chance to be forwarded by a node. In contrast, a packet is forwarded several times in sparse networks with intermittent connectivity in order to reach more nodes. It deals with the spread-rate trade-off explained in Sect. 3.2.3; it adapts the spot size (the spread of a packet) according to the network conditions in order to ensure a sustainable application rate. It implements a congestion control mechanism. It ensures source-based fairness by serving packets per source ID using a processor sharing approach. It improves the MAC broadcast efficiency and reliability without touching the 802.11 MAC driver. It employs a buffer management mechanism. It is persistent with intermittent connectivity. And finally, it is environment oblivious. It does neither exchange topology information nor use hand-shaking among nodes. By considering all these issues, SLEF delivers a complete

and sustainable data dissemination service to an application.

Chapter 3

Self Limiting Epidemic Forwarding

3.1 Introduction

Broadcast exists inherently in the wireless channel and is used by several communication systems in ad hoc networks. It can either be single hop (the native MAC layer broadcast), or multihop. In static scenarios, multihop broadcast can simply be implemented as follows: The source generates an IP packet with TTL=k and sends it as a MAC layer broadcast; any node that receives it decrements its TTL and if the result is positive, schedules it for a new transmission as a MAC layer broadcast. In Disruption Tolerant ad-hoc Networks (DTNs), multihop broadcast is implemented by some form of epidemic forwarding: Nodes repeat packets they receive with some probability, possibly more than once, in order to extend the spread (number of nodes that receive the packets) while mitigating redundancy. In single-hop or multi-hop forms, broadcast is used to disseminate information in quickly varying environments (e.g. opportunistic networks), where mobility and self-organization make the classical methods based on distribution trees non-practical. Also, it can be used to support routing, resource discovery protocols or in bootstrapping phases for application layer protocols: for instance the "Spray" phase in the Spray-and-Focus protocol [67] is a form of multi-hop broadcast.

We consider open ad-hoc networks, such as a crowd in a city, or car passengers in a busy highway system. A common feature here is that there is no practical bound on the number of users (unlimited network), and contact times may be short and unpredictable. In practice, implementing multihop broadcast in such settings poses a number of practical challenges, which,

if not correctly addressed, may lead to very poor performance. A first issue is how to set the TTL correctly. Consider for example an application that uses multihop broadcast in a vehicular network; the connectivity may range from sparse and sporadic (lightly loaded highway) to very dense (traffic jam, city center). Furthermore, changes from one setting to another may be very sudden. We show in our performance studies that, in a traffic jam with realistic parameters, there are around 200 nodes within range, and any TTL setting $k > 1$ results in congestion collapse. In contrast, in a sparse setting, $k = 1$ results in practically no dissemination. Thus, the TTL, if used in that form, should be set adaptively. A second issue is the absence of acknowledgment in MAC layer broadcasts (e.g. with 802.11). A node cannot know if its transmitted packet is received by someone else, it simply undergoes a collision or it is transmitted in a vacuum. Also, mutual exclusion mechanisms (as CSMA/CA) that manage collisions are usually not implemented in broadcast mode (for example in IEEE 802.11). Therefore, accessing the medium in broadcast mode is similar to ALOHA that performs poorly. A third issue is flow control, i.e. how to control the packet injection rate of the application. This is normally done end-to-end by TCP, but here this probably does not apply. If the injection rate is not adapted to the network conditions, this may result in congestion collapses and failure of the broadcast. A fourth issue is scheduling among competing broadcasts. It is likely that more than one broadcast packets are competing for retransmission at one node, and some form of mechanism is required to know which packet to select next.

In this chapter, we propose a complete design for a scope limited, multi-hop broadcast middleware that is adapted to the ad-hoc environment and addresses all of the above issues. It performs well in DTNs, as well as in other settings, in particular in very dense networks (as in a traffic jam). We call our system "Self Limiting Epidemic Forwarding" (SLEF). SLEF adapts to a rapidly varying environment in a way that is completely transparent to the application. In a very dense environment, SLEF is equivalent to a single hop broadcast; in a sparse environment, to a k-hop broadcast, with k automatically adapted to the network conditions. In DTNs, it performs as an epidemic system, i.e. packets may be re-transmitted more than once if this is required to achieve a good performance. SLEF achieves these goals by a number of mechanisms, described in the next section. A main feature of SLEF is a non-classical manipulation of the TTL field, which combines the usual decrement-when-sending with many very small decrements when receiving.

The SLEF middleware offers the following service to the application. It delivers packets as a limited multi-hop broadcast, without the application having to bother about what the current state of the ad-hoc environment is. The service interface is flow controlled, i.e. the application can send only up to a maximum rate determined by SLEF. This rate is controlled so as to ensure a reasonable balance between spread (number of nodes that receive the broadcast) and rate, in an open, unlimited environment.

3.2 Functions

In order to achieve its goal as a practical broadcast middleware, SLEF has to implement six mandatory functions. The first function is forwarding factor control, which aims at mitigating redundancy. It adapts the forwarding factor (i.e. the number of times a node forwards a packet) based on the send/receive events seen on the same packet: A packet that is seen for the first time, has a high chance to be forwarded, whereas a packet that has seen several send/receive events is considered as well propagated and its chance to be forwarded is lower. The second function is spread control. Recall that the spread is the total number of nodes that receive a packet. Spread control adapts the spread to the network state in order to guarantee a minimum rate for the application when the network scales. It is based on an aging mechanism that decrements a packet TTL field locally based on the receive events seen by the node. The third function is scheduling, which decides which packet to deliver to MAC for transmission. It is likely that more than one packet are competing for transmission at a node. These packets might belong to different sources and might have seen different numbers of send/receive events. In order to apply forwarding factor control, our scheduler has to serve packets according to the numbers send/receive events, and thus it can not be based on naive policies such as First In First Out (FIFO). Moreover, our scheduler considers the packet source Ids in order to achieve source-based fairness. The fourth function is congestion control. It consists of adapting the application injection rate, not only to avoid local buffer overflow, but it goes one step further: If the injection rate is higher than the forwarding capacity of the source neighbors, the source packets will be accumulated in the neighbor buffers and dropped before being forwarded. Therefore, we adapt the injection rate to the forwarding capacity in order to allow packets to propagate in the network. The fifth function is buffer management, which decides when to

drop packets in order to keep space in the buffer for the new incoming packets. In general, our buffer management drops first the packet the most propagated in the network. The sixth function consists of careful use of the MAC broadcast. To compensate for the absence of the mutual exclusion and the acknowledgment in the MAC broadcast (see Sect. intro), we use two mechanisms: pseudo-broadcast and presence indicator. The former implements a CSMA/CA-based mutual exclusion. The latter returns true if some neighbors were present around a node while it was transmitting a packet. In this case, the node considers that the transmission was successful and was not in the vacuum.

All these functions are achieved using only local information to the node and do not need any knowledge about the network topology. In the following, we describe in detail the solutions we propose to achieve them. For ease of understanding, we begin by defining the main variables used in our design and the terminology that will be used in the explanation.

3.2.1 Variable and Terminology Definition

Every node maintains one *epidemic buffer*, used to store received and locally originated packets, with the following attributes:

- `sendCount` : how many times this packet was sent by this node
- `rcvCount` : how many times this packet or a duplicate was received by this node
- `vRate` ("virtual rate"): This attribute is derived from `sendCount` and `rcvCount` , using the method described in Sect. 4.2.1. It is the rate at which this packet would be transmitted if it were alone in the epidemic buffer.
- `age` : combines hop count, real time age (true time to live) and adaptive age, which reflects the amount of competition this packet and its ancestors have encountered so far
- `earliestSendTime` and `pendingSendConfirmation` : see Sect. 3.2.4 and Sect. 3.2.7

We call *clone* the set made of an original packet and its duplicates; all packets in the same clone have the same value for source address (IP address) and the identification field. When a packet is received, it is inserted into the epidemic buffer. If this is the first time a packet of this clone is seen by this node, a new entry is created, otherwise if the existing entry is still present,

it is overwritten (thus there is always at most one packet per clone in the epidemic buffer). The attributes are updated as explained later. We call self-packets the packets originated by the node, and foreign-packets the packets received from other nodes.

3.2.2 Forwarding Factor Control

Forwarding factor control mechanisms aim at preventing nodes from forwarding over-sent or over-received packets in order to minimize redundancy. Different approaches to forwarding factor control are proposed in the literature. They differ in being adaptive or not, and in the information they need about the topology and neighbors.

Our forwarding factor control is adaptive. It allows for forwarding of packets many times to recover from collision or transmitting in the vacuum, as we will see later in Sect. 3.3.2. Note that forwarding packets many times is very useful in highly mobile networks, as several transmissions of the same packet occur in different locations and face different neighbors, and thus, increase the packet spread without adding redundancy.

With our mechanism, a packet in the epidemic buffer is retransmitted with a probability that depends on its `vRate` , which we define as:

$$\texttt{vRate} \leftarrow R_0 a^{\texttt{rcvCount}} \; b^{\texttt{sendCount}}$$

where R_0 is the nominal rate in packets per second of the MAC layer interface and a and b are unit-less constants less than 1. Thus the virtual rate of a packet decreases exponentially with any send/receive event of the same packet. The scheduler (see Sect. 3.2.4) decides which packet is selected next for transmission by the MAC layer; it serves packets with rates not exceeding their virtual rates. Hence, a packet in the epidemic buffer, which has seen many send/receive events, is scheduled at a very low rate, and it is more likely that it will be dropped by the buffer management (see Sect. 3.2.6) mechanism before being transmitted.

3.2.3 Spread Control

We argued in the introduction that spread control is needed to ensure that the transmission rate of any user is satisfactory. Recall that the spread is the total number of nodes that receive a packet. Formally speaking, let λ be the user application rate (generating new information to

forward), FF the forwarding factor, S the spread and R the available transmission rate over the channel, which includes self and foreign packets . In a symmetric network where self-packets are transmitted only once and foreign-packets forwarded FF times we have:

$$\lambda + FF * \lambda * S = R \Leftrightarrow \lambda = \frac{R}{1 + FF * S} \quad . \tag{3.1}$$

So the rate-spread trade-off is obvious.

A natural way to limit the spread is to use the classic TTL, which is the method that comes by default with the Internet Protocol (IP). When a packet is created by a source and placed into the epidemic buffer, it receives a TTL value equal to some positive constant `maxTTL` . When the packet is accepted for transmission by the MAC layer, the TTL field of the *transmitted* packet is equal to the value of the TTL field in the packet in the epidemic buffer, minus 1. The TTL field in the packet stored in the epidemic buffer is unchanged.

When a packet created by some other node is received for the first time at this node, the value of the TTL is screened. If it is equal to 0, it cannot be retransmitted and the packet is discarded. Otherwise (TTL$\geq$ 1), the packet is stored in the epidemic buffer, with TTL equal to the value present in the received packet. When and if the packet is later accepted for transmission by the MAC layer, the transmitted TTL field is equal to the stored TTL minus 1, and the stored TTL is unchanged.

A potential problem with the classic TTL is that it does not adapt to the node connectivity. In a very dense network, we should choose a very small value of `maxTTL` to limit the number of hops and the spread. In contrast, a large value of `maxTTL` is preferable in sparse networks.

We propose an aging-based spread control mechanism that adapts itself to the node density and traffic load. With this mechanism, the `age` attribute is inherited when a packet is received for the first time and is equal to 0 for a newly created clone for a self packet. The `age` stored in the epidemic buffer is a floating point number. It increases depending on the events affecting the packet and the state of the epidemic buffer. When transmitting a packet, the complement to `maxTTL`(=255) of `age` , rounded to an integer, is written in the IPv4 TTL field [resp.IPv6 hop count]. Similarly, when a packet is received for the first time, its `age` is extracted from the TTL/hop count field: `age` $=$ `maxTTL` $- TTL$.

There are three processes that increase the `age` :

- (hop count): `age` is incremented by a constant amount K_0 whenever either this packet

is transmitted or a duplicate is received.

- (real time age): `age` increases at a constant rate $\alpha = 32h^{-1}$. We assume that nodes have free running clocks; there is no need for time synchronization. The constant α is such that a packet lives at most 8 hours, which corresponds to the work cycle.

- (adaptive age): `age` of all packets stored in the epidemic buffer increases by an amount K_1 every time a packet (of an existing or new clone) is received. The adaptive aging constant K_1 is a (possibly non integer) constant less than K_0; its value will be discussed in Sect. 3.3. For self-packets with `sendCount` == 0, this process is valid up to a threshold that we call Self Age Threshold (SAT). When the `age` reaches SAT, it passes immediately to `maxTTL` and it will never be incremented by K_1 until the packet is transmitted. SAT is computed through a density detection mechanism, explained later in this section.

A packet is killed whenever its `age` is too large to be sent, i.e. when `age` $\geq$ `maxTTL` $+1$ (the $+1$ is due to rounding). An exception is made for self-packets with `sendCount` == 0, they are not discarded before being transmitted at least once, even if their `age` exceeds `maxTTL` $+1$. This happens in very congested networks where self-packets have to stay a long time in the epidemic buffer before being transmitted.

When a packet is received for a clone that is present in the epidemic buffer, the TTL of the received packet is ignored and only the increase by K_0 is applied to the `age` of the packet already present in the epidemic buffer. This is to limit the harm of any spurious manipulation of TTL by cheaters [33].

The behavior of hop-count and real-age processes is intuitive, whereas the behavior of the adaptive age needs more explanation. Let N be the number of neighbors a node has, R_0 the MAC nominal capacity in packets/s and γ the channel utilization. In case where each node is running a greedy application (which always has a packet to send), the packet reception rate can be approximated by $\tau_r = \frac{N}{N+1} * \gamma * R_0$, assuming fairness at MAC layer. Thus, the adaptive age increases with a rate equal to $\tau = K_1 * \tau_r$. As we will see in Sect. 3.5, a numerical example could be: $N = 240$ (traffic jam), $R_0 = 83 packets/s$ (MAC rate = 1Mbps and packet size of 1500 bytes), $\gamma = 0.7$ and K_1 = 0.1. As a result, $\tau = 5.8s^{-1}$. That is, a packet can stay at most $\frac{\texttt{maxTTL}}{\tau} = 44s$ in the epidemic buffer before being rejected. Adding the impact of other `age` components, a packet stay will never reach $\frac{\texttt{maxTTL}}{\tau}$.

A density detection mechanism is implemented in order to strictly limit the communication to one hop in very dense networks. It consists of computing SAT as follows:

1. At the beginning: $SAT \leftarrow SAT_0$ ($SAT_0 = 10$ as computed in Sect. 3.3.4):

2. Upon each reception, which is considered as indication of high node density, SAT is decremented by K_1 in order to limit the number of hops:

$$SAT \leftarrow max(SAT_0, SAT - K_1)$$

3. Upon each transmission, which is considered as an indication of a low node density, SAT is incremented by SAT_0 in order to allow more hops:

$$SAT \leftarrow min(\texttt{maxTTL}\ , SAT + SAT_0)$$

We set SAT_0 to 10, which corresponds approximately to a maximal spread of 100 nodes (see Sect. 3.3.4). Thus, SAT will be very close to one of two values in steady state, which is reached in a few seconds. SAT is very close to SAT_0 (=10) in a very dense network where the number of nodes within transmission reach is larger than 100. Hence, a self packet will be transmitted with `age` = `maxTTL` and thus, we ensure only one-hop communication. In contrast, SAT is very close to `maxTTL` in a sparse network that allows several hops communication.

3.2.4 Scheduler

The scheduler decides which packet in the epidemic buffer is selected for transmission (being passed to the MAC layer).

In order to ensure source-based fairness, the scheduler serves packets per source IP address, using a processor sharing approach. Furthermore, every packet should be served at a rate not exceeding its `vRate` (see Sect. 4.2.1).

Every packet in the epidemic buffer has a derived attribute `earliestSendTime` , equal to the last time the `vRate` of this packet was modified, plus $\frac{1}{\texttt{vRate}}$. At any time t, a packet is said to be "eligible" if it has `earliestSendTime` $\leq t$. Eligible packets with the same source IP address are linked in one FIFO per source. Each of these FIFOs has an attribute `sourceClaim` , which keeps track of how much this source can claim to be scheduled. It

is initially 0 and is decremented by 1 when this source is selected for transmission by the scheduler. It is incremented by 1 divided by the number of sources in the epidemic buffer whenever a packet is scheduled for transmission.

The scheduler issues a blocking send function to the MAC layer that returns whenever the packet is accepted by the MAC layer. When this method returns, the scheduler looks for another packet to deliver to MAC. It selects the source with the highest `sourceClaim` that has eligible packets (none-empty FIFO). In case it does not find eligible packet, it waits until one becomes available.

It can be seen that this algorithm allocates the transmission opportunities according to a water-filling algorithm, thus, it approximates an ideal fluid scheduler that would allocate rates to sources in a max-min fair way, subject to the constraint that the rate of a source does not exceed the sum of the `vRates` of the packets of this source.

3.2.5 Congestion Control

Our congestion control consists of adapting the application injection rate to the network conditions.

SLEF allows the existence of at most σ self-packets in the epidemic buffer (we set σ to 2). The application is allowed to inject a new packet in the epidemic buffer in one of three cases.

The first is when the number of self-packets in the epidemic buffer is less than σ. It happens either at the bootstrap of the application or when a self-packet is dropped because it has seen numerous send/receive events and its `age` has reached `maxTTL` . Thus, we assume that this packet has lived enough to proliferate in the network.

The second case is when the epidemic buffer contains σ self-packets but, at least one of them has seen a duplicate forwarded by a neighbor. Indeed, SLEF considers the received duplicate as an implicit acknowledgment (Ack) and that the neighborhood has enough capacity to propagate the packet in the network. In this case, the acknowledged packet is dropped when the application injects a new packet.

The third case is also when the epidemic buffer contains σ self-packets but, this time, at least the `sendCount` of one of them has reached 3, which is an indication that the packet is received by some other node. This is to avoid that the application is blocked for long time in case SLEF has not received any implicit Ack for the self-packets existing in the epidemic

buffer. Indeed, it might happen that the implicit Ack undergoes a collision and it is not received by the source. Therefore, the source continues transmitting the packet, and at the same time, inhibiting its neighborhood from forwarding it (see Sect. 4.2.1), and thus, it might never receive an implicit Ack for this packet. Note that, the `sendCount` is not incremented unless we are sure that, with high probability, a transmission is received by other nodes (as explained later in Sect. 3.2.7).

3.2.6 Buffer Management

The buffer management aims at cleaning the epidemic buffer to save space for new incoming packets. The cleaning process distinguishes between foreign and self-packets. A foreign-packet is dropped when its `age` becomes larger than `maxTTL` . For nodes with very limited buffer size, this may not be sufficient. If an arriving packet requires space to be freed, the foreign packet with the largest `age` is deleted.

As to a self-packet, it is dropped in one of three cases. The first is when its `age` exceeds `maxTTL` and its `sendCount` is strictly positive. The second is when it is implicitly acknowledged and the epidemic buffer contains σ self-packets. This packet is deleted when the application injects a new packet. The third is similar to the second, except that the `sendCount` of the packet reaches 3 instead of being acknowledged.

Applying Little formula [21] on our `age` based buffer management, we find that the epidemic buffer size is upper bounded by $\frac{\texttt{maxTTL}+1}{K_1}$. To understand this, assume $K_0 = 0$ and a node starts receiving packets, all with `age` = 0. Thus, this node starts dropping packets after receiving $\frac{\texttt{maxTTL}+1}{K_1}$ packets as the `age` of the first packet received is equal now to `maxTTL` $+1$. This node continues dropping packets with a rate equal to the receiving rate and its epidemic buffer size becomes constant equal to $\frac{\texttt{maxTTL}+1}{K_1}$.

3.2.7 Careful Use of MAC Broadcast

We assume that nodes have a MAC layer capable of receiving and sending packets in broadcast mode, at a rate that depends on the network conditions (and is likely to be much less than the peak transmission rate R_0 used above). In practice, if we use the IEEE 802.11 MAC broadcast, there is a performance issue, as it does not use the RTS/CTS exchange and collisions during

transmission go undetected. To avoid this issue, we use the pseudo-broadcast mode proposed in [46], by which a packet is sent to the MAC address of a neighbor (with RTS/CTS), but can be promiscuously copied by all systems within range. This effectively solves much of the performance issue, but may not always be applicable to our case, since we do not want nodes to spend time discovering their neighbors' MAC addresses. Therefore, we use the following method. The MAC layer has a node global MAC state information that says whether the next packet will be sent in pseudo-broadcast, and if so, to which MAC address, or in broadcast mode. The destination MAC address in the pseudo-broadcast mode is the source MAC address of the last received packet. As soon as the node receives one packet, the MAC state is set to pseudo-broadcast. The next packet is thus sent with an RTS. If no CTS is received in response, the MAC layer backs off for a random time (this is the standard operation of 802.11). If during the back-off time a packet is received, the packet is retransmitted (after expiration of the back-off timer) in pseudo-broadcast mode to the MAC address of the newly received packet. Else the MAC state moves to broadcast, and the packet is re-transmitted in broadcast mode.

There remains one issue, however, where a node does not know if a sent packet was received by another; this might become a problem in the desert highway scenario, where a node would repeatedly send a packet in the vacuum, until it ages out. To avoid this, we use two heuristics when sending in broadcast mode: (1) indication of neighbor presence, and (2) implicit acknowledgment by reception of duplicate. (1) consists in building a function around the MAC layer that says whether, shortly before or after a packet transmission in broadcast mode, the carrier is sensed busy. If a packet is sent in the former case, or in pseudo-broadcast mode (some neighbors are around) then `sendCount` is incremented and a flag we call `pendingSendConfirmation` is set to false. Of course, there is no guarantee that a packet sent in these circumstances is actually received by anyone, but the rules for rate adaptation will make it likely for this packet to be retransmitted soon if no duplicate is received (in such a case `vRate` remains large). If in contrast a packet is sent in the latter case (presumably because there is no one around), the flag `pendingSendConfirmation` is set to true for this packet, and the packet is rescheduled at a later date with the same `vRate` . If a packet of the same clone is received while `pendingSendConfirmation` is true, the pending transmission is considered successful. The condition `pendingSendConfirmation` == true can be terminated either by reception of a duplicate or by a subsequent transmission that returns an

indication of presence or is in pseudo-broadcast mode.

3.3 Design Tuning

Throughout this section, we derive simple mathematical equations that describe the behavior of our design and show the interaction among its different components. This analysis allows us to find ranges for our design parameters (a,b,K_0, K_1, and SAT_0) within which, the system performs well in a wide range of settings. We wanted our analysis to be very simple and intuitive for ease of understanding. Nevertheless, the analysis allows for a deep understanding of the impact of each parameter and the interaction among the different SLEF components, and delivers well-tuned parameter ranges.

We consider only two extreme cases, very sparse and very dense, where we find suitable ranges for the parameters. In intermediate cases, SLEF is able to adapt itself without any need to change its parameters. We validate its capacity to adapt later through simulations (see Sect. 3.5).

In both scenarios, we assume that each node is running a greedy application (which always has a packet to send) and that the network is homogeneous. It follows that:

1. The average stay durations of a given packet in each epidemic buffer are the same.

2. All packets have the same average stay duration in a given epidemic buffer.

These two points are ensured by the source-based fairness delivered by the scheduler.

3.3.1 List of Notations

In the following, we are listing the notations used throughout this analysis.

Global Notations

- R_0: Nominal MAC rate [packets/s].

- γ: Channel utilization.

- N: Number of neighbors within the transmission range of a source (excluding the source).
- S: Spread, including the source itself.
- λ: Application rate.
- H: Minimum number of hops in the absence of collision; neighbors within the transmission range are considered as one hop.
- R: MAC effective transmission rate [packets/s]. We approximate it by:

$$R = \frac{1}{N+1}\gamma R_0 \tag{3.2}$$

- τ: Adaptive age increasing rate of a packet in the epidemic buffer caused by receive-events. We approximate it by:

$$\tau = \frac{N}{N+1}\gamma R_0 K_1 \tag{3.3}$$

  where ($\frac{N}{N+1}\gamma R_0 = N * R$) is the packet receiving rate. It is clear that the τ unit is [age units/second].

- D_F: Average delay a FIFO undergoes to be served once. This delay is due to the competition among different FIFOs in the epidemic buffer.
- D_M: Average delay a packet undergoes at MAC layer due to the competition among nodes in accessing the medium. We have:

$$D_M = \frac{1}{R} \tag{3.4}$$

- H_r: Required number of hops; the minimum that we want to ensure in presence of collisions.
- P_c: Probability of collision on one side of a node in the linear grid of the sparse scenario (see Sect. 3.3.2).

SLEF Notations

- A self packet: A packet that is originated by the local node
- A foreign packet: A packet that is received from other nodes
- σ: The maximum number of self packets at a node
- `sendCount` : How many times this packet was sent by the local node
- `rcvCount` : How many times a packet or a duplicate was received by the local node
- a and b: Constants used to compute the virtual rate as follows:

$$\texttt{vRate} \leftarrow R_0 a^{\texttt{rcvCount}} \; b^{\texttt{sendCount}} \tag{3.5}$$

- K_0: Age increment due to the hop-count component (see Sect. 3.2.3)
- K_1: Age increment due to the adaptive-age component (see Sect. 3.2.3)
- SAT: Self Age Threshold (see Sect. 3.2.3)

3.3.2 Sparse Scenario

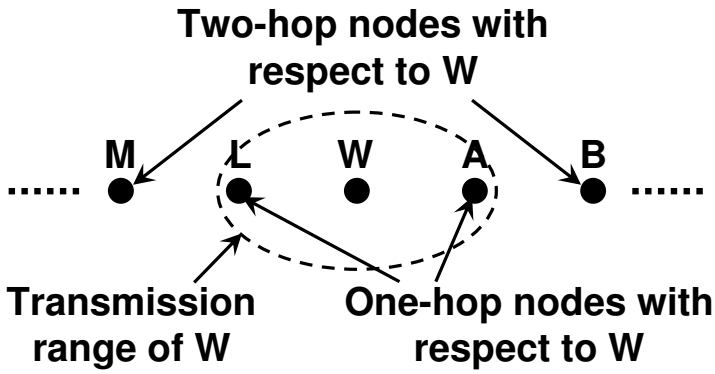


Figure 3.1: The linear grid considered in the sparse scenario. Each node is connected only to its two closest neighbors: for instance, node W is connected only to nodes L and A. Nodes A and L are the one-hop nodes of W. Nodes M and B are the two-hop nodes of W, and so on.

In this scenario, we consider the linear grid in Fig. 3.1, where each node has only two nodes within transmission range, the previous and the next nodes. As this is a sparse scenario, SAT is equal to `maxTTL` and does not appear in the analysis of this section.

In the following, we derive system equations according to two requirements. The first is that we require H-hop broadcast in the absence of collision. The second is that we require at least H_r hops ($H_r < H$) in presence of collision, where packets are forwarded more than once in case of collision. Then, we finish this section with an interpretation of the obtained analysis and with a parameter tuning.

Before beginning the equation derivation according to the two requirements, we show a general constraint that should be respected in the remaining of this section. In order to let packets propagate beyond the one-hop nodes, we should have:

$$b < a \tag{3.6}$$

To understand this, let us consider this example: when W in Fig. 3.1 transmits a self-packet, if it is well received by A, the `vRates` of this packet will be $R_0 b$ and $R_0 a$ at W and A respectively. If b is larger than a, W retransmits the same packet before A and the `vRates` become $R_0 b^2$ and $R_0 a^2$ at W and A respectively, and so on. Thus, if Ineq. 3.6 is not ensured, the first-hop nodes are inhibited by the source itself and packets will never escape from them.

Reaching H Hops in the absence of Collision

To reach at least H hops, the `age` of a packet after (H-1) hops should be smaller than `maxTTL` to be able to do the last (H^{th}) hop. Thus, the following inequation should be satisfied:

$$X_1 + X_2 + X_3 < \texttt{maxTTL} \tag{3.7}$$

where the left hand terms are as follows: X_1 is the `age` increase average of a packet during its stay in the epidemic buffer at the self-node (the node generating this packet) before being delivered for the first time to MAC for transmission. We can write:

$$X_1 \leq \sigma D_F \tau \tag{3.8}$$

We do not consider the delay in MAC, D_M, as the `age` increase during that delay is not transmitted to the next nodes. X_2 is the `age` increase average of a packet during its stay at

foreign nodes (nodes that are not the source of this packet) of the $(H-1)$ hops. We have:

$$X_2 = (H-1)(\frac{1}{R_0 a} + D_F)\tau, \tag{3.9}$$

where $\frac{1}{R_0 a}$ is the inverse of the virtual rate of a packet that has been received only once and has not been transmitted yet. The delay at MAC layer (D_M) is not considered for the same reason as above. X_3 is the `age` increase due to the hop-count component during $(H-1)$ hops. We have:

$$X_3 = (H-1)K_0 \tag{3.10}$$

In the optimal case and in the absence of collision, a node receives packets from its spread, S, and forwards all of them, except those belonging to the two farthest nodes in the spread (as their packets should not go beyond this node), once and only once. Thus, the number of FIFOs in this node that have eligible packets and are competing to be served by the scheduler at a given time is upper bounded by $(S-2)$. Hence, we can write:

$$D_F \leq \frac{1}{R}(S-2) \tag{3.11}$$

with

$$S \geq 2H+1$$

Plugging Ineq. 3.11 in Ineq. 3.8 and Ineq. 3.9, we obtain upper bounds of X_1 and X_2. By using upper bounds of X_1 and X_2, and assuming that we require only H hops (that is $S = 2H+1$), we obtain an upper bound of the lower bound of a, given K_0 and K_1. In this case we have:

$$(\sigma\frac{1}{R}(2H-1) + (H-1)[\frac{1}{R_0 a} + \frac{1}{R}(2H-1)])\tau + (H-1)K_0 < \texttt{maxTTL} \tag{3.12}$$

Reaching at Least H_r hops in the presence of Collision

Consider the two successive neighbors A and B in Fig. 3.1. A forwards a foreign packet for the first time to B. At the beginning, the `vRate` is $R_{1,0} = R_0 a$. Recall that $\frac{1}{\texttt{vRate}}$ is the time a packet waits to be eligible after the last update of its `vRate` (due to a send/receive event on the same packet). After forwarding, the `vRate` of the same packet at A is reduced

to $R_{1,1} = R_0ab$, whereas it is again $R_{1,0}$ at B if it is well received. B will forward the packet before that A forwards it for the second time, as B has larger `vRate` . When A receives the same packet from B, it reduces its `vRate` to $R_{2,1} = R_0a^2b$.

In the absence of collisions, the simplest thing we can do to avoid redundancy is to select b small enough (that is $R_{1,1}$ small enough) so that the packet dies (its `age` reaches `maxTTL`) before being forwarded for the second time by A.

When the probability of collision is high, the above constraint on b does not hold anymore: It is enough that one collision occurs in one side of the source to stop propagating the packet beyond the collision place, which results in a high reduction in the spread, as the probability of collision is high even at the source level because of the hidden node, a well known problem with CSMA/CA systems.

To solve this problem, we play with both, $R_{1,1}$ and $R_{2,1}$. On one hand, (1) we want $R_{1,1}$ large enough to allow A to forward again the packet if no duplicate is received from B, with the constraint that B corresponds to a hop number less or equal than the required number of hops, H_r, which is less than H. On the other hand, (2) we want $R_{2,1}$ small enough so that the packet dies before being forwarded for the third time by A. In the following we develop these constraints on $R_{1,1}$ and $R_{2,1}$ separately.

Constraint on $R_{1,1}$ In the following, we show corresponding equations assuming that the packet escapes from the source on the side we apply the equations and that the probability that two collisions occur on the same packet on the same side of a node is negligible. We write this condition as:

$$Y_1 + Y_2 + Y_3 + Y_4 < \texttt{maxTTL} \tag{3.13}$$

where the left hand terms are as follows: Y_1 is the `age` increase average of a packet during its stay in the epidemic buffer at the self-node until being delivered to MAC for transmission for the first time. We consider only the first transmission by a self-node. Indeed, if W is the self-node and it transmits a self-packet, this transmission faces one of three cases: (1) No collision on both sides and the packet is well received by A and L. (2) A collision occurs on one side, say L-side, then A receives well the packet and forwards it later, which consists an implicit Ack for W, which will drop the packet to allow the application to inject a new one (see Sect. 3.2.5). We neglect the case where the implicit Ack undergoes a collision from A

to W, as this makes appear P_c^2 in the inequation (first, collision from W to L and, then, from A to W) and P_c^2 is too small compared to P_c. (3) The self-packet undergoes a collision on both sides, which happens with negligible probability. Thus, all retransmission possibilities of self-packet are negligible. Therefore, we consider only the first transmission. Similarly to Ineq. 3.8 and Ineq. 3.11, we have:

$$Y_1 \leq \sigma \frac{1}{R}(S-2)\tau \tag{3.14}$$

The delay at MAC layer (D_M) is not considered for the same reason as above. Y_2 is the `age` increase average of a packet during its stay until being delivered to MAC for transmission the first time at foreign nodes of the $(H_r - 1)$ hops. Thus, Y_2 consists of the delay because of the `vRate` , which is $\frac{1}{R_{1,0}}$, and D_F. This latter is always upper bounded by $(S-2)\frac{1}{R}$, as the average length of FIFOs is 1. Indeed, our congestion control mechanism together with the source-based fairness of the scheduler ensure that the waiting time average of a packet in the FIFO is less than or equal to the inter-arrival time average of new packets: The application does not inject a new packet unless the previous one is implicitly acknowledged and all the FIFOs have the same scheduling share in all nodes. Applying Little formula [21], we obtain a FIFOs length average equal to 1. We can write:

$$Y_2 \leq (H_r - 1)(\frac{1}{R_0 a} + \frac{1}{R}(S-2))\tau \tag{3.15}$$

Y_3 is the `age` increase average of a packet during its stay after being delivered to MAC for the first transmission at foreign nodes of the $(H_r - 1)$ hops until being delivered to MAC for the second transmission, because of collision. We can write:

$$Y_3 \leq P_c(H_r - 1)(\frac{1}{R_0 ab} + \frac{1}{R}(S-2) + \frac{1}{R})\tau \tag{3.16}$$

Note that Y_3 considers D_M that corresponds to the first transmission. Y_4 is the `age` increase due to the hop-count component during $(H_r - 1)$ hops. We have:

$$Y_4 = K_0 + (1 + P_c)(H_r - 2)K_0 \tag{3.17}$$

The first right-hand term corresponds to the transmission by a self-node, it does not include P_c because we consider only the first transmission (as discussed above). Finally, by using upper bounds of Y_1, Y_2 and Y_3 and plugging Ineq. 3.14, Ineq. 3.15, Ineq. 3.16 and Ineq. 3.17

in Ineq. 3.13, this gives an upper bound of the lower bound of b, when other parameters are specified. Ineq. 3.13 becomes:

$$\begin{aligned} \sigma\frac{1}{R}(2H-1)\tau &+ \\ (H_r-1)(\frac{1}{R_0 a}+\frac{1}{R}(2H-1))\tau &+ \\ P_c(H_r-1)(\frac{1}{R_0 ab}+\frac{1}{R}(2H-1)+\frac{1}{R})\tau &+ \\ K_0+(1+P_c)(H_r-2)K_0 &< \texttt{maxTTL} \end{aligned} \tag{3.18}$$

Constraint on $R_{2,1}$ If this constraint is satisfied for the one-hop nodes, it is satisfied for farther nodes. Therefore, we consider only the one-hop nodes in the following. We write this condition as:

$$Z_1+Z_2+Z_3+Z_4+Z_5+Z_6 > \texttt{maxTTL} \tag{3.19}$$

where the left-hand terms are as follows: Z_1 is the `age` increase average of a packet during its stay in the epidemic buffer at the self-node (say W in Fig. 3.1) until being delivered to MAC for transmission for the first time. We have:

$$Z_1 = \sigma D_F \tau \tag{3.20}$$

Z_2 is the `age` increase average of a packet during its stay in the one-hop node (say A in Fig. 3.1) until being transmitted for the first time by the same node. We have:

$$Z_2 = (\frac{1}{R_0 a}+D_F+D_M)\tau \tag{3.21}$$

Z_3 is the `age` increase average of a packet during its stay in the one-hop node (A) after being transmitted for the first time at the one-hop node (A) until being transmitted for the first time at the two-hop node (B in Fig. 3.1). We have:

$$Z_3 = Z_2 \tag{3.22}$$

Z_4 is the `age` increase average of a packet during its stay in the one-hop node (A) after being transmitted for the first time at the two-hop node B (and then received by the one-hop node (A)

that updates its `vRate`) until being delivered to MAC for transmission for the second time at the one-hop node (A). We have:

$$Z_4 = (\frac{1}{R_0 a^2 b} + D_F)\tau \tag{3.23}$$

Z_5 is the `age` increase due to the hop-count component because of transmissions at self and one-hop nodes. We have:

$$Z_5 = 2K_0 \tag{3.24}$$

Z_6 is the `age` increase due to the hop-count component when the one-hop node (A) receives the transmission of the two-hop node (B). We have:

$$Z_6 = K_0 \tag{3.25}$$

Plugging Ineq. 3.20 till Ineq. 3.25 in Ineq. 3.19 while neglecting D_F and D_M, gives a lower bound of the upper bound of b when other parameters are specified. That gives:

$$(\frac{1}{R_0 a} + \frac{1}{R_0 a} + \frac{1}{R_0 a^2 b})\tau + 3K_0 > \texttt{maxTTL} \tag{3.26}$$

Analysis Interpretation

In this section, we interpret the three main inequations Ineq. 3.12, Ineq. 3.18 and Ineq. 3.26, and we show how they can be used to find appropriate ranges for the parameters. Replacing τ and R in Ineq. 3.12, Ineq. 3.18 and Ineq. 3.26 by their expressions in Eq. 3.2 and Eq. 3.3, we get respectively:

$$\begin{aligned} \sigma N K_1(2H-1) &+ \\ (H-1)[\frac{\gamma N K_1}{a(N+1)} + NK_1(2H-1)] &+ \\ (H-1)K_0 &< \texttt{maxTTL} \end{aligned} \tag{3.27}$$

$$\begin{aligned} \sigma N K_1(2H-1) &+ \\ (H_r-1)[\frac{\gamma N K_1}{a(N+1)} + NK_1(2H-1)] &+ \\ P_c(H_r-1)[\frac{\gamma N K_1}{ab(N+1)} + NK_1(2H-2)]\tau &+ \\ K_0 + (1+P_c)(H_r-2)K_0 &< \texttt{maxTTL} \end{aligned} \tag{3.28}$$

$$(\frac{2}{a} + \frac{1}{a^2 b})\frac{N}{N+1}\gamma K_1 + 3K_0 > \texttt{maxTTL} \tag{3.29}$$

We notice that Ineq. 3.27, Ineq. 3.28 and Ineq. 3.29 are independent of R_0, and thus this analysis is applicable to any MAC layer with any nominal rate.

Ineq. 3.27 gives a lower bound of a in order to ensure H hops in the absence of collisions. This inequation should be combined with the constraint $0 \leq a \leq 1$. Fig. 3.2-(a) shows this bound according to H for different combinations of K_0 and K_1. The asymptotes in Fig. 3.2-(a) correspond to the maximum number of hops reachable for a given combination (K_0, K_1), which is a decreasing function of K_0 and K_1: when K_0 increases, the maximum number of hops decreases, as it is limited by $\frac{\texttt{maxTTL}}{K_0}$, and when K_1 increases, τ increases and the packets age out faster. In Fig. 3.2-(b), we apply Ineq. 3.27 and show the lower bound of a as a function of K_0 and K_1. We set H to 8, which is reachable with the shown ranges of K_0 and K_1. The lower bound of a is an increasing function of K_0 and K_1. Indeed, with increasing K_0 and K_1, a packet ages faster and we need to increase its `vRate` in order to ensure the same number of hops.

From Fig. 3.2-(a), we notice that $a = 0.1$ corresponds to a H very close to the asymptote with the used ranges of K_0 and K_1. Thus, we fix a to 0.1 in the remaining of the paper. Consequently, H is tuned through the 2 parameters K_0 and K_1 and by applying Ineq. 3.27. This tuning is shown in Fig. 3.2-(c).

Once parameters a, H, K_0 and K_1 are fixed, we apply Ineq. 3.28 and Ineq. 3.29 to find lower and upper bounds of b, respectively. Fig. 3.3 shows that these bounds are increasing functions of K_0 and K_1. Also, as it is expected, these bounds are more sensible to K_1 than K_0. Indeed, b is related to the stay duration of a packet in a node, and it is K_1 that increases the packet `age` during this stay.

3.3.3 Dense Scenario

Through this analysis, we aim at tuning SAT_0. We want to strictly limit the broadcast to one-hop when the number of neighbors within the transmission range exceeds N^*. In this case, a node transmits only self-packets and the application rate in this case, λ^*, is equal to the transmission rate R^*. Thus, the FIFO belonging to self-packets is served on average each $\frac{1}{R^*}$ seconds. During this time, the `age` increase of a packet in the FIFO is $\frac{1}{R^*}\tau = N^*K_1$. Thus, we set SAT_0 to N^*K_1. If $R > R^*$, e.g $N < N^*$, SAT is incremented by $SAT_0 = N^*K_1$ each $\frac{1}{R}$ seconds, but decreased linearly during this time by $\frac{1}{R}\tau = NK_1 < N^*K_1$. Thus, SAT continues increasing on average with a slope equal to $(N^* - N)K_1R$ until it reaches `maxTTL` (see Sect. 3.2.3). In contrast, if $R < R^*$, e.g $N > N^*$, SAT is decremented more than incremented and thus, SAT is kept very close to SAT_0. The behavior of SAT is shown in Fig. 3.4 for $N > N^*$ and $N > N^*$ where it shows a very short transient phase. We require that N^* is equal to 100. Hence, we have: $SAT_0 = 100K_1$.

3.3.4 Default Values

In this section, we give default values to the SLEF parameters based on the above analysis. We set K_0 to 25, as we want to allow at most 10 hops. We set a to 0.1, which corresponds to a number of hops, H, very close to the maximum reachable for a given combination (K_0, K_1) (see Fig. 3.2-(a)). As to b, if it is too small, the second forwarding of a packet in case of collision is largely delayed. In order to avoid this delay, b should be very close to the upper bound in Fig. 3.3-(b). We choose the value 0.01. K_1 is set to 0.1, which is in accordance with the selected value of b (see Fig. 3.3). Furthermore, with $K_1 = 0.1$, the maximum epidemic buffer size needed is $\frac{\texttt{maxTTL}}{K_1} = 2550$ (see Sect. 3.2.6), which is a reasonable size. Finally, for $K_1 = 0.1$, SAT_0 is equal to 10, as discussed in Sect. 3.3.3. An application might need to

change only K_0 and K_1 to adjust the spread-rate balance (see Sect. 3.4) while other parameters are fixed.

3.4 Spread-Rate Balance

One of the main features of SLEF is to limit adaptively the spread in order to keep some balance between rate and spread (see Eq. 3.1). However, an application might need to move this balance in favor of one or the other. For instance, an application might have a very small rate but it requires a very large spread. In order to adjust this balance, SLEF offers to the application two degrees of freedom, which are the two main parameters of the spread control function: K_0 and K_1. These two degrees play complementary roles: One is dominant in some network settings, the other is dominant in those settings just opposite.

K_1 is related to the adaptive age component. This component is incremented during the stay of a packet in the epidemic buffer by K_1 for any receive event. The longer the packet stays before being transmitted (or forwarded), the higher its adaptive age is and the less the spread is. Thus the effect of K_1 is dominant when this stay is long. This happens in two cases: (1) either the traffic load is very high and the number of competing packets in the epidemic buffer is very large, or/and (2) the network is dense and the number of nodes competing to access the medium is large. In both cases, a packet transmission is delayed and the packet might be dropped before being transmitted or, if transmitted, it will have a large age that does not allow it to go very far. Hence, the spread is limited by mainly the adaptive age. Consequently, playing with K_1 in these settings indeed has an impact on the spread-rate balance.

In contrast, K_0 is dominant in sparse networks, in particular with low traffic load. In this case, the spread is mainly limited by the hop-count component of the age. That is, the spread corresponds to a number of hops close to the maximum reachable ($\frac{\texttt{maxTTL}}{K_0}$).

An application that has to adjust the spread-rate balance according to its needs may proceed as follows: Specify the number of hops it needs in a sparse network and set K_0 accordingly ($K_0 \leq \frac{\texttt{maxTTL}}{number\ of\ hops\ needed}$). Then, it has to decrease or increase the default value of K_1 in order to adjust the balance in dense congested networks.

3.5 Design Validation

We validate our design through simulation. Our simulations are carried out through JIST-SWANS [3], an open source simulator for ad hoc networks. The MAC layer is a very accurate implementation of 802.11b in DCF mode with the basic rate of 1 Mbps, as we transmit in broadcast (pseudo-broadcast). As for the radio, we use the capture effect to approach the real WIFI cards, which all implement it [48]. We consider fading channels with free space path-loss.

We applied SLEF to vehicular networks. We use an extension of JIST-SWANS called STRAW [11], which simulates the vehicular traffic and provides a mobility model based on the operation of the real vehicular traffic. We simulate vehicles on an urban road with two lanes in each direction and a speed limit of $80\,Km/h$. Results for other scenarios (static and different mobility models) are omitted, as they show the same behavior.

Throughout our simulations, we adopt the default values set in Sect. 3.3.4, unless it is indicated elsewise. Our results focus mainly on (1) the adaptation of the spread to the rate, (2) the adaptation of the forwarding factor to the density, (3) the need of the pseudo-broadcast and (4) the spread-rate balance.

In order to cover these different aspects, we use the following metrics:

1. Rate: This is the application injection rate in packet/s. We consider a packet size of 1500 bytes.

2. Spread: This is already defined in this chapter. It is the average number of nodes that receives a packet.

3. Forwarding factor: Again, this is already defined in this chapter. It is the number of times a node forward a packet. We compute it as the number of duplicates circulated in the network divided by the spread.

4. Channel utilization: We use this metric only with the traffic jam (very dense network). It is approximated by the receiving rate divided by the nominal transmission rate. Note that, the receiving rate considers only successfully received packets. The channel utilization should include the successful transmission rate. We neglect this as the network is very dense and it is too small compared to the receiving rate.

3.5.1 Adaptation of the Spread to the Rate

In this scenario, we consider a fixed vehicle density of $12.5\,vehicles/Km$. The application is not greedy. It injects packets with a fixed rate that ranges from 0.1 till $0.3\,packets/s$. This range is less than the maximal rate allowed for this scenario by the congestion control mechanism of SLEF, which is around $0.4\,packets/s$ (obtained by simulation). Fig. 3.5 shows the spread according to the rate. As it is expected, SLEF adapts the spread to the traffic load: it decreases the spread with increasing rate. This is equivalent to adapting the TTL.

3.5.2 Adaptation of the Forwarding Factor to the Density

An efficient forwarding factor control mechanism reduces the forwarding factor with increasing node density in order to limit redundancy. This is ensured by SLEF and shown in Fig. 3.6, where each node runs a greedy application and thus, it transmits at the maximal rate allowed by the congestion control mechanism.

3.5.3 Pseudo-Broadcast

	Normal broadcast	Pseudo-broadcast
Channel utilization	0.02	0.7

Table 3.1: Channel utilization in a traffic jam.

In order to show the need of pseudo-broadcast, we have chosen a very challenging scenario: a traffic jam where each vehicle has around 240 others within its transmission range and runs a greedy application. The results are shown in Table 3.1. The pseudo-broadcast solves very efficiently the medium access problem. It achieves a channel utilization of 0.7, whereas it is less than 0.02 with the normal MAC broadcast, which does not implement a mutual exclusion mechanism.

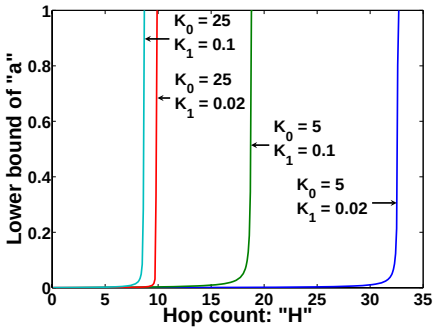
3.5.4 Spread-Rate Balance

In this section, we validate what is discussed in Sect. 3.4 about adjusting the spread-rate balance by showing the impact of K_0 and K_1 in a sample scenario. The scenario is the highway

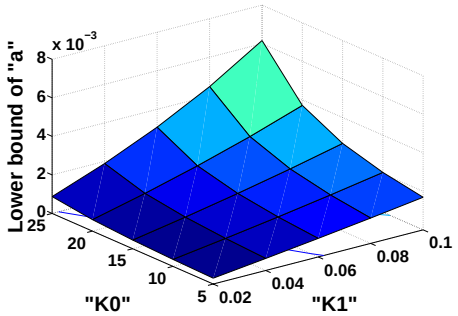
with vehicle density of $12.5\, vehicles/Km$. Each node runs a greedy application. The results are drawn in Fig. 3.7. By increasing K_0 and K_1, the application increases the rate on the expense of the spread.

3.6 Conclusions

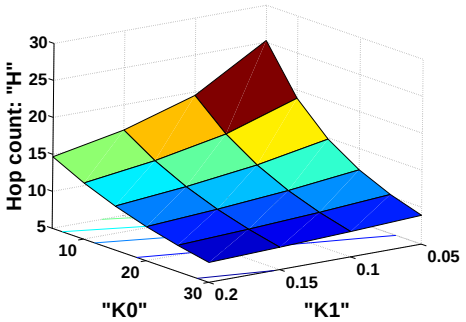
We propose SLEF, a complete practical middleware for multi-hop broadcast in ad hoc networks. It adapts itself to the variability of the ad hoc network environments. This includes the implementation of an adaptive TTL (through the spread control), an adaptive forwarding factor (inhibition) and congestion control. In addition, SLEF achieves buffer management, an efficient use of the MAC broadcast and source-based fairness. All these functions are achieved using only local information to the node and do not need any knowledge about the network topology. We derive simple system equations in order to tune the SLEF parameters, and we deliver default values for them. We validate our design through simulations applied on different vehicular network scenarios ranging from very sparse (DTN like) to very dense (traffic jam). SLEF shows a good adaptation and succeeds in avoiding congestion collapse, even in the extreme scenarios where other multi-hop broadcast schemes fail. Finally, SLEF offers to the application two parameters to adjust the spread-rate balance if it needs to depart from the default values.



(a): The lower bound of a is drawn according to H for different combinations (K_0, K_1). The asymptotes correspond to the maximal value of H reachable for a given combination of (K_0, K_1).

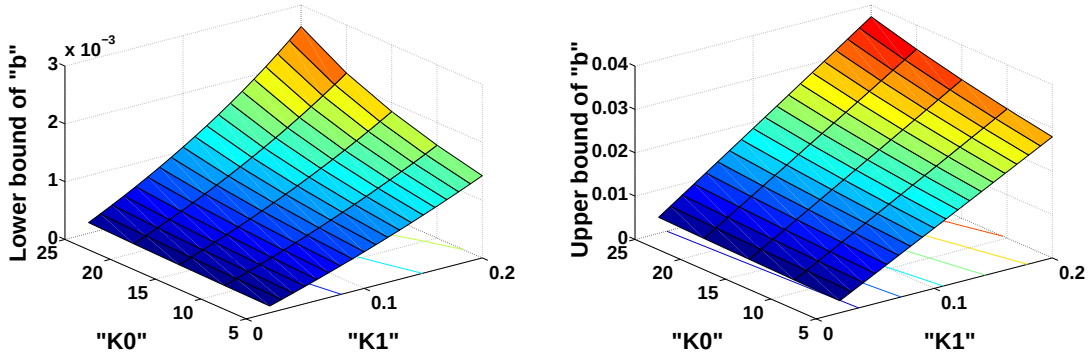


(b): The lower bound of a is drawn as a function of K_0 and K_1. H is set to 8, which is reachable with the used ranges of K_0 and K_1.



(c): H is drawn as a function of K_0 and K_1 for a equal to 0.1 . This value of a corresponds to a H very close to the asymptote (see Fig. 3.2-(a)), which is the maximal H reachable for a given combination of (K_0, K_1).

Figure 3.2: Relations among a, H, K_0 and K_1 based on Ineq. 3.27.



(a) (b)

Figure 3.3: Bounds on b drawn according to K_0 and K_1. These bounds are obtained through Ineq. 3.28 for (a) and Ineq. 3.29 for (b). $a = 0.1$, $H = 8$ and P_c = 0.1 .

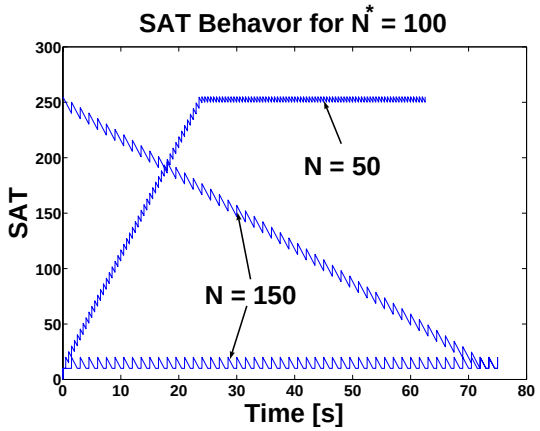


Figure 3.4: SAT behavior according to the density detection mechanism: $N^* = 100$, $K_1 = 0.1$ and $SAT_0 = 10$. We assume 802.11 MAC layer with a nominal rate of 1Mbps and a packet length of 1500 bytes.

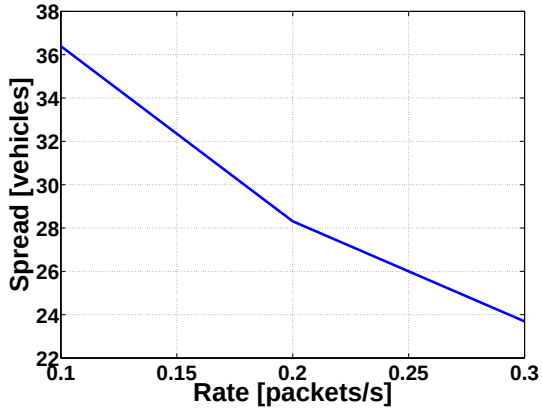


Figure 3.5: SLEF adapts the spread to the rate. This equivalent to an adaptive TTL. The curve corresponds to the highway scenario with a vehicle density of $12.5\, vehicles/Km$.

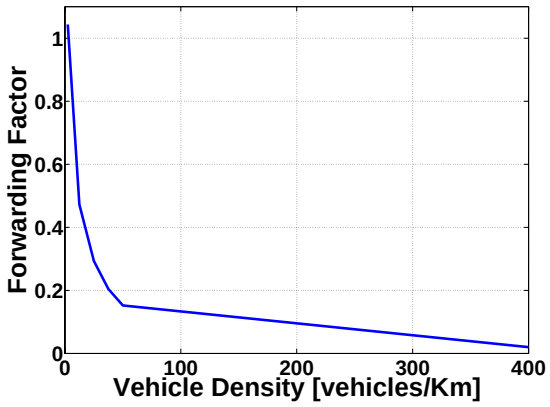
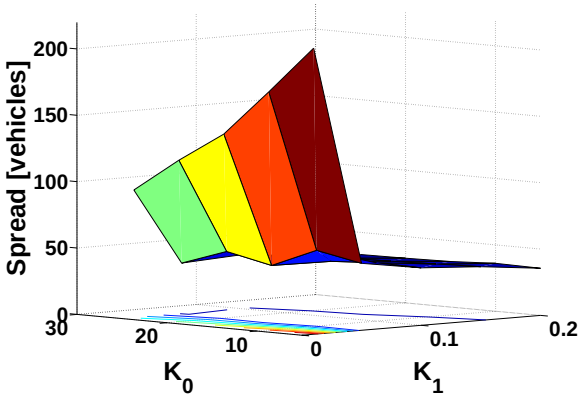
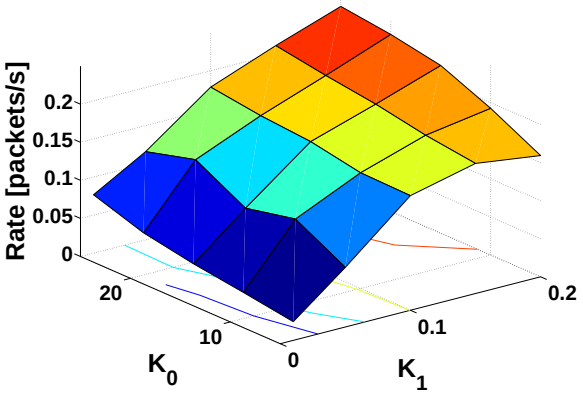


Figure 3.6: Forwarding factor vs vehicle density. SLEF adapts the forwarding factor to the node density in order to mitigate redundancy. This is the achievement of the forwarding factor control function. The curve corresponds to the highway scenario with a vehicles running greedy applications. Thus, the rate corresponds to the maximal allowed by the congestion control mechanism that SLEF implements.



(a)



(b)

Figure 3.7: Spread-rate balance: This balance is adjusted through K_0 and K_1. This is the highway scenario with vehicle density of $12.5\, vehicles/Km$. Each node runs a greedy application.

Chapter 4

Vulnerabilities in Epidemic Forwarding

4.1 Introduction

In this chapter, we identify vulnerabilities in epidemic forwarding over ad-hoc networks. We are interested in broadcast applications such as chatting in a traffic jam or coupon advertisements [38]. The principle of epidemic forwarding is that nodes repeat with some probability the information they hear from others, thus propagating fresh information.

Epidemic forwarding employs several mechanisms such as forwarding factor control, which prevents a node from forwarding over-sent or over-received packets in order to minimize redundancy. Each mechanism can be implemented using different alternative methods. Thus, the existence of vulnerabilities is highly dependent on the mechanisms employed and on the methods adopted to achieve them. We examine the links between these methods and the vulnerabilities.

4.2 Epidemic Forwarding Mechanisms

In this section, we explain the different mechanisms used in epidemic forwarding, in order to understand their vulnerabilities.

4.2.1 Forwarding Factor Control

Forwarding factor control aims at preventing nodes from forwarding over-sent or over-received packets in order to minimize redundancy. We classify them into two sets: rigid and adaptive. With the former set, the mechanisms cannot adapt themselves to different network settings: when the settings change, their parameters need to change. The adaptive mechanisms ensure a good performance in a wide range of settings without changing their parameters.

Rigid Forwarding Factor Control

Within this set we find Gossip-based epidemic forwarding [57] where a node decides to forward a packet with a fixed probability p and drop it with $(1 - p)$. The value of p depends on the setting but Gossip does not involve any mechanism to adapt p.

Adaptive Forwarding Factor Control

Within this set we distinguish between two methods.

Counter Based Forwarding Factor Control This method is essentially the one proposed in [59]. A packet stored in the epidemic buffer has a counter called "Receive Count" incremented by 1 when a duplicate of this packet is received. Initially, i.e. when the packet is created by the application or received for the first time, the counter is set to 0. When the counter reaches a maximum value, the packet is discarded from the epidemic buffer. When a packet is transmitted, the value of Receive Count is lost.

Virtual Rate Based Forwarding Factor Control This is the method we propose with SLEF. With this method, a packet in the epidemic buffer is retransmitted with a probability that depends on its "virtual rate"; it is equal to $c_0 a^R b^S$ where c_0 is a constant (inverse of a time), R [resp. S] is the number of times this packet or a duplicate was received [resp. sent] and a and b are unit-less constants less than 1. Thus the virtual rate of a packet decreases exponentially with any send/receive event of the same packet. A scheduler decides which packet is selected next for transmission by the MAC layer; it serves packets with a rates not exceeding their virtual rates. Hence, a packet in the epidemic buffer, which has seen many send/receive events,

is scheduled at a very low rate and it is more likely that it will be dropped by the used buffer management mechanism before being transmitted (see Sect. 3.2.6). The constant c_0 is equal to the nominal packet rate of the MAC layer.

4.2.2 Spread Control Mechanisms

Spread control mechanisms are essential for epidemic forwarding, as the broadcast capacity does not scale with the population. Spread control can be implemented using one of the following methods.

Classic TTL

This is the method that comes by default with the Internet Protocol (IP). When a packet is created by a source and placed into the epidemic buffer, it receives a TTL value equal to some positive constant "MaxTTL". When the packet is accepted for transmission by the MAC layer, the TTL field of the *transmitted* packet is equal to the value of the TTL field in the packet in the epidemic buffer, minus 1. The TTL field in the packet stored in the epidemic buffer is unchanged.

When a packet created by some other node is received for the first time at this node, the packet is delivered to the application, and the value of the TTL is screened. If it is equal to 0, it cannot be retransmitted and the packet is discarded. Else (TTL≥ 1), the packet is stored in the epidemic buffer, with TTL equal to the value present in the received packet. When and if the packet is later accepted for transmission by the MAC layer, the transmitted TTL field is equal to the stored TTL minus 1, and the stored TTL is unchanged.

Aging

This is the method we propose in Sect. 3.2.3. We present it here in a different but essentially equivalent form. We give here a presentation that combines different options in one single framework. The method uses the TTL field like Classic TTL, but the TTL of a packet may be decremented while it is stored in the epidemic buffer, depending on receive and send events. Formally, every packet in the epidemic buffer has an "age" field, which is a fixed decimal positive number less than 256. When a packet, created by some other node, is received by this

node for the first time, its age is set to the complement to 255 of the received TTL: age$= 255-$ TTL. When a packet is transmitted, its stored age is incremented by a fixed amount K_0 and then its TTL is set to $255-$ age. When a duplicate packet is received, the received TTL is ignored but the stored age is incremented by K_1: age = age$+K_1$. When *any* packet is received, the stored age of *all* packets in the epidemic buffer is incremented by K_2: age = age$+K_2$. The node drops packets with age larger than 255.

4.2.3 Scheduler

Epidemic forwarding needs a scheduler for buffer management. To our knowledge, the only scheduler that is explicitly detailed in the literature is the one we propose in Sect. 3.2.4. It is used with the virtual-rate based forwarding factor control (Sect. 4.2.1). It decides which packet in the epidemic buffer is selected for transmission, i.e. to be passed to the MAC layer. In order to ensure some level of fairness, the scheduler serves packets per source Id, using a processor sharing approach. Moreover, every packet should be served at a rate not exceeding its virtual rate computed in Sect. 4.2.1.

4.2.4 Control of Injection Rate

The only explicitly defined method to achieve control of injection rate is the one proposed with SLEF. It is used together with the aforementioned scheduler. The packets generated by the application at a given node are placed into the epidemic buffer, where they compete with the other packets for transmission (but with a larger virtual rate, having $R = S = 0$). The application rate is controlled by a windowing system : The number of outstanding packets the application is allowed to have in the epidemic buffer at this node is limited to -at most- 2 (see Sect. 3.2.5); a packet is deleted from the epidemic buffer when a duplicate is received, which serves as implicit acknowledgment (Ack).

4.3 Attacks

In this section, we describe the vulnerabilities that are specific to epidemic forwarding. We distinguish between two types of attackers: malicious and rational. The former does not look

for a personal benefit but aims to harm other nodes. In contrast, the latter seeks to increase its personal profit from the network. Most of the attacks are described by drawing (Figs. 4.1 and 4.2) using a generic example where the attacker is M and the victim in malicious case is A.

4.3.1 Malicious Attacks

A malicious attacker aims at decreasing the spread and/or the injection rate of the victim by exploiting vulnerabilities in epidemic forwarding mechanisms. In the following we identify five attacks and map them to their corresponding epidemic forwarding mechanisms.

Artificial High Density (AHD)

In this attack, we exploit the adaptability of the spread control to the congestion and node density. The attacker places itself close to the victim. It acts like any node: it has its self packets (packets that are generated at this node) to send and relays others packets. But it does not forward victim packets. By generating much traffic in the very close surrounding of the victim, the attacker incites the spread-control mechanism at the victim's good neighbors to react negatively and prevent the victim packets from going farther.

Inhibit by Forwarding (IbF) Attack

With IbF (Fig. 4.1), the attacker exploits the adaptive forwarding factor control. It immediately forwards the victim packets a number of times (this number is called Attack-Persistency) to inhibit its neighborhood from forwarding the same packets (see Sect. 4.2.1). With the counter based forwarding factor control (see Sect. 4.2.1), the Attack-Persistency is equal to the maximum value the counter can reach. With the virtual rate based forwarding factor control, this Attack-Persistency should be large enough to make the corresponding virtual rate close to zero (in practice two times are enough).

Inhibit by TTL (IbTTL) Attack

This attack exploits the spread control using TTL. As the attacker receives a victim packet, it forwards it immediately with a TTL = 0. In Fig. 4.1, B and M receive a packet from A with TTL = MaxTTL -1. M forwards it with TTL = 0 instead of (MaxTTL -2). Hence, the

attacker decreases the chance the packet has to travel beyond C as B is inhibited and C drops the packet. Even if B succeeds in forwarding the packet after M, this will change nothing with C. This example considers the use of "classic TTL" (see Sect. 4.2.2). With "aging", the attack is exactly the same. Note that in Fig. 4.1, B applies the first strategy (see Sect. 4.2.2) upon receiving a duplicate of the packet and thus it keeps the old TTL.

Inhibit by Forwarding and TTL (IbFTTL) Attack

This is a combination of IbF and IbTTL (Fig. 4.1). In this case M forwards the victim packets Attack-Persistency times with TTL = 0 to insure that the victim packets at B are well inhibited and thus the packet loses any chance of traveling beyond C.

Send on Behalf of the Victim (SoB) Attack

The attacker exploits the scheduler and the aging mechanism. In Fig. 4.1, M sends fake packets with A's Id. As the scheduler serves packets per source Id, A's packets are delayed in the epidemic buffer and they will be dropped either by the aging mechanism (they become too old) or by buffer overflow.

4.3.2 Rational Attacks

A rational attacker tries to increase its injection rate while maintaining large spread. In the following, we identify two rational attacks.

Do Not Cooperate (DNC) Attack

When a new packet is injected by the application at a given node, it is placed in the epidemic buffer, where it competes with packets received from other nodes. This competition prevents the application from injecting at the full rate allowed by the packet injection control mechanism because of the additional delay in the epidemic buffer. Thus, an attacker decides to not cooperate and to keep only its self packets (packets that are generated at this node) in the epidemic buffer. Note that, if the attacker tries to go beyond the allowed rate, its packets will be accumulated in other nodes, which are not able to serve them at the same rate. Thus, it risks killing its packets for the same reason as in Sect. 4.3.1.

Sybil Attack

We refer to the Sybil attacker as the node that forges multiple identities [28]. This is a well-known attack in networking, but the way it is exploited in this paper is new and very specific to epidemic forwarding. As the scheduler serves packets per source Id, the attacker sends its self packets with different Ids and thus it increases their share of the bandwidth. In Fig. 4.2, we present the scheduler as a process sharing approach where queues are per source Id. In this case, M's packets receive larger bandwidth share than A's packet at B.

4.4 Performance Evaluation

In this section, we evaluate the impact of the aforementioned attacks by simulation. We apply them in static scenarios, as well as in highly mobile networks. We consider vehicular mobility on the highway. Our metrics are based on the spread and the injection rate: a malicious attacker aims at reducing the victim spread and a rational one tries to increase its rate while maintaining large spread.

In our simulation, we consider our SLEF, our multi-hop broadcast middleware. To our knowledge, SLEF is the only complete system proposed for a wide range of settings. Furthermore, SLEF implements all epidemic forwarding mechanisms already discussed in Sect. 4.2: The virtual rate based forwarding factor control, spread control by TTL and aging, injection rate control and the scheduler discussed in Sect. 4.2.3. The parameter values of the virtual rate based forwarding factor control are $a = b = 0.15$, c_0 corresponds to 802.11b basic rate (1Mbps) with a packet length of 1500 bytes. As for the aging, we use $K_0 = K_1 = 25$ and $K_2 = 0.5$.

4.4.1 Settings

With the static scenarios, we simulate from 200 up to 600 nodes uniformly distributed over a square of 500x500 m^2, but, in most cases, we show the results only for 400 nodes as the others are similar. The transmission range is around 50 m (PDA transmission range).

In the case of a malicious attack, the victim is in the middle of the square and attackers take place around it as it is indicated in Fig. 4.3. We want to evaluate the impact of the distance

between attackers and the victim. Therefore, the radius R in Fig. 4.3 can have one of two values: 25m and 100m. With the former, the attackers of the corresponding circle are within the transmission range of the victim and they are outside it with the latter.

In the case of a rational attack, there exists only one attacker, which is in the middle.

The network can be either congested, where all nodes are sources sending at full rate (capacity allowed by the channel) or non-congested, where the victim is the only source in the network and it is sending at full rate. Beside the victim, only attackers can act as sources in the non-congested scenario, based on the attack they want to achieve.

In the following we will use the following notations: "close" [resp. "far"] to indicate that R is equal to 25m [resp. 100m] and "one" [resp. "all"] to indicate that the network is non-congested [resp. congested].

As for the mobile scenario, we simulate 1000 vehicles in an urban two-lane road. The speed limit is 80 km/h. The car density is 12.5 cars/km in each direction. The transmission range is 300m, which is typical for vehicular network.

Our simulations are carried out through JIST-SWANS [3], an open source simulator for ad hoc networks. The MAC layer is a very accurate implementation of 802.11b in DCF mode with the basic rate of 1 Mbps as we transmit in broadcast (pseudo-broadcast, see Sect. 3.2.7). As for the radio, we use the capture effect to approach the real WIFI cards that all implement it [48]. We consider fading channels with free space path-loss. As for the mobile network, we use an extension of JIST-SWANS called STRAW [11], which simulates the vehicular traffic and provides a mobility model based on the operation of real vehicular traffic.

4.4.2 Static Scenarios

Malicious Attacks

AHD The results are shown in Fig. 4.4. Let us begin with the "all" scenario (see Sect. 4.4.1) where the attackers are sources and act as any other node, except that they do not forward victim packets. In the "close" case, the impact of the attack is considerable in both scenarios, "all" and "one", and it increases with the number of attackers. In contrast, in the "far" + "all" scenario, the attackers do not have a major impact; the reason is twofold: (1) the attackers are far from the victim and thus they do not increase the density as much as in the "close"

scenario; (2) the forwarding factor control mechanism is adapted. Indeed, the attackers are numerous and they cooperate in forwarding all packets except the victims, hence they inhibit their neighbors from forwarding packets except those of the victim. Thus, the increase in the victim spread, which we notice for 8 attackers, is due to the fact that victim packets are less inhibited than others. If the forwarding factor control were rigid, we expect that AHD would have more impact on the victim. In the "far" + "one" scenario, attackers are still injecting new packets in the network as before. They reduce considerably the victim spread.

IbF The attackers do not generate fresh packets: their role is merely to forward victim packets as it is explained in Sect. 4.3.1. The results are shown in Fig. 4.4. It is clear that IbF does not achieve its goal. This can be explained as follows: When an attacker receives a new victim packet, it immediately forwards it Attack-Persistency times (if the MAC layer allows). If the same attacker receives another victim packet, before it finishes forwarding the previous packet, it cancels the previous and it starts anew with the newest. Thus, let us consider a scenario that happens frequently. The victim sends a new packet. The attacker forwards it immediately. The victim receives a duplicate of its self packet and considers it as an implicit Ack. Hence, it injects a new self packet that will be received by the attacker before finishing the forwarding process and it will be received by other attackers even before beginning the forwarding process. Thus, all attackers cancel the previous packet, which explains why it is not inhibited.

IbTTL Our implementation of IbTTL is similar to the one of IbF with the difference that it modifies the TTL before forwarding, as it is explained in Sect. 4.3.1. This attack is more harmful than IbF. The attacker needs to forward the packet only once with TTL = 0. Thus, nodes that receive the packet from the attacker for the first time are not able to forward it due to its TTL. This makes the difference with IbF, which needs to forward several times to inhibit the packet in its neighborhood.

IbFTTL This attack has approximately the same impact as IbTTL, which is to be expected as IbF has little effect on the victim.

SoB The attackers send only fake packets at full rate. Fig. 4.4 shows a significant decrease in spread and rate. The spread reduction is due to the fact that victim packets are killed in the

epidemic buffers before being forwarded, which is due to the delay caused by the fake packets (for more explanation see Sect. 4.3.1). Moreover, the decrease in rate is due to the delay of the implicit Ack that controls the injection rate as explained in Sect. 4.2.4.

From what we have seen in this section, we can conclude that the attackers are not able to harm the victim in the presence of mobility for two reasons. The first is that the impact of the attackers is very position-dependent. The second is that, even with the most harmful attack, the attackers could reduce the spread of the victim, but its packets still reach a few tens of nodes. If these nodes are mobile, they will carry the victim packets beyond the barrier imposed by the attacker. This conclusion is well verified later in the vehicular network scenario.

Rational Attacks

DNC We evaluate the impact of DNC only in the "all" scenario, where increasing the injection rate is a challenge. In the "one" scenario, the attacker is the only source in the network and he has the entire network capacity, thus it is meaningless to evaluate its impact in this case. In Fig. 4.6, the performance of a DNC attacker is compared with a well-behaved node that is very close to it and thus they both experience the same network conditions. We show the spread of both nodes and the rate ratio (DNC over well-behaved). The DNC rate is four times larger than a well-behaved node. But, surprisingly, the DNC spread is much larger when the network is very dense (600 nodes). The reason is as follows: The attacker does not forward others' packets. Thus, when it receives others' packets, it drops them without updating the age of its self packets in the epidemic buffer. Hence, the age of its self packets does not increase during their stay in its epidemic buffer by K_2 (see Sect. 4.2.2), which allows them to travel farther.

Sybil We evaluate the impact of Sybil in only "all" scenario for the same reason as with DNC. The attacker uses five different identities. In addition, it does not forward others' packets. So, our implementation is in fact a combination of both attacks, Sybil and DNC, explained in Sect. 4.3. This implementation gives the attacker a much larger advantage than using DNC alone (up to 10 times larger than a well-behaved node and 2.5 larger than the DNC attacker), which explains the impact of Sybil alone.

4.4.3 Vehicular Network Scenario

In this scenario, nodes are highly mobile and the position of the victim is not known. Thus, the attackers are chosen randomly. Beside the attackers, the network contains 1000 well-behaved nodes, all of them are sources ("all" scenario). All nodes cross the same urban road.

Malicious Attacks

Fig. 4.7(a) shows the impact of malicious attacks. The spread of the victim is drawn according to number of attackers. The Attack-Persistency of IbF and IbFTTL is 2. Other values give the same results. As we notice, the effect of the attackers is negligible even in the presence of 100 attackers. In the most harmful case, the IbFTTL attacker reduces the victim spread from 50 to 30 nodes, which is not significant. This can be explained by the presence of the spread control mechanism; the attacker can affect the victim only if their spreads interfere, i.e. there exist common nodes that receive the attacker and the victim packets. And the amount of harm is proportional to the amount of interference. As the spread is limited by the spread control mechanism, this interference is not considerable and does not happen frequently.

Rational Attacks

Contrary to malicious attacks, rational attacks are still powerful even in highly mobile network. The results are shown in Fig. 4.7(b). Sybil still ensures higher gain than DNC.

4.5 State of the Art

To our knowledge, this is the first work that identifies vulnerabilities that are specific to epidemic forwarding, i.e. that use epidemic forwarding mechanisms such as forwarding factor control, spread control, injection rate control and scheduler.

Some of vulnerabilities that we identify could be recovered by cryptographic and authentication methods, if they are available. But all already existing work in the literature on securing wireless network does not apply here as we address broadcast application over wireless ad-hoc networks. In particular, an extensive work assumes the existence of a third trusted part that is the infrastructure [63, 64], which does not exist in what we do. In [61], the authors propose

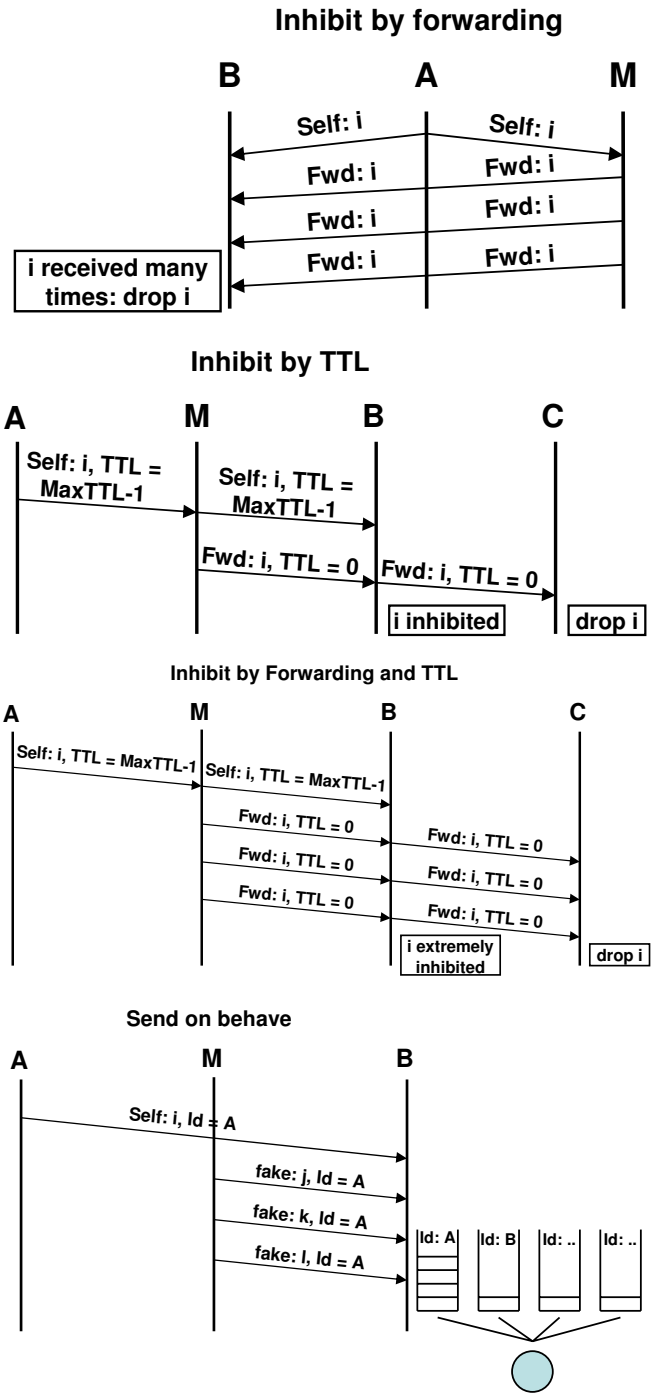
a method based on global synchronization to secure routing. Global synchronization assumes the presence of infrastructure unless mobiles are occupied with GPS (Global Positioning System) devices, which we do not assume in our work. Further, with the broadcast nature of the applications that we address, nodes do not need to know each other to communicate and thus they can not trust each other. Until now and at the best of our knowledge, there does not exist any work proposing an authentication method for this scenario.

4.6 Conclusions

We identify vulnerabilities that are specific to epidemic forwarding over wireless ad-hoc networks. We classify these vulnerabilities into two categories: malicious and rational. We evaluate their impact according to the number of attackers and the different network settings. We find that the impact of malicious attacks depends on the position of the attacker relative to the victim, the network density, the traffic load and mobility. In static scenarios, we identify the attacks that reduce dramatically the victim spread, whereas the harm of other attacks is reduced due to the adaptive forwarding factor control and the injection rate control. In highly mobile vehicular network, the impact of malicious attacks are minimized due to the spread control.

We have studied the rational case in presence of only one attacker in the network. The attacker could achieve considerable profit in all scenarios.

Our work can be extended in different directions. We plan to examine the impact of the presence of several rational attackers on the network. Another extension is to find solutions to recover from these vulnerabilities.



M is the malicious node and A is the victim. We refer by Self to packets generated at the node transmitting them, by Fwd to packets forwarded by the node but generated by others and by Fake to packets that are generated by M but carrying the victim identity (Id = A). We will explain only the "Inhibit by Forwarding and TTL" attack, as other attacks have similar explanation. In "Inhibit by Forwarding and TTL", A sends a Self packet i with TTL = MaxTTL -1, that is received by M and B. M forwards the packet i (Fwd: i) 3 times with TTL = 0, the packet (Fwd: i) is received by B and C. Thus, the forwarding factor control mechanism at B will inhibit packet i, as it is received 4 times, and C will drop the packet, as its TTL = 0.

Figure 4.1: Malicious attacks.

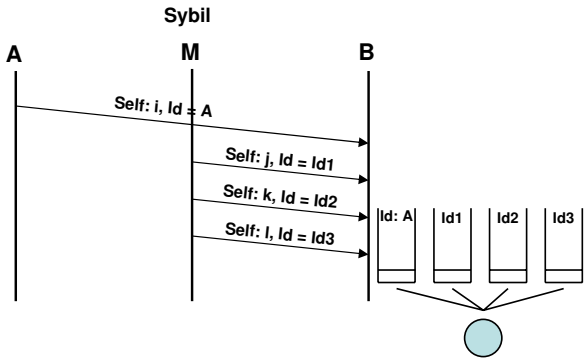
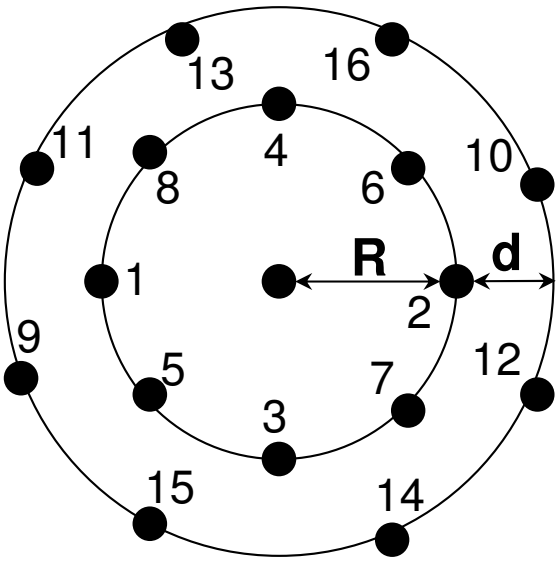


Figure 4.2: Sybil attack: M is the rational node.



The victim is in the middle, the attackers start filling the positions around the victim according to their index in an increasing order: if one attacker, it fills position 1. If 2 attackers, they fill positions 1 and 2, and so on. R can be either 25m or 100m. d = 25m.

Figure 4.3: Malicious attackers positions.

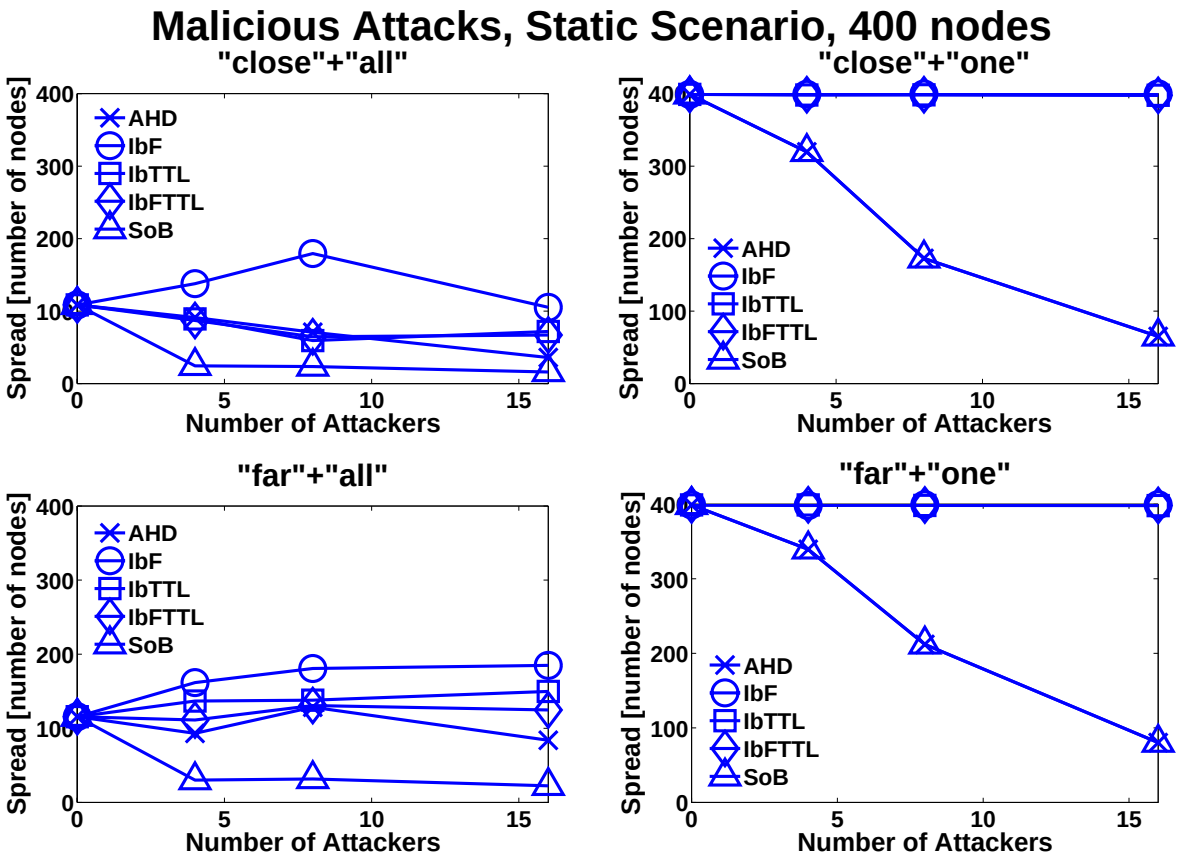


Figure 4.4: Malicious attacks in static scenario.

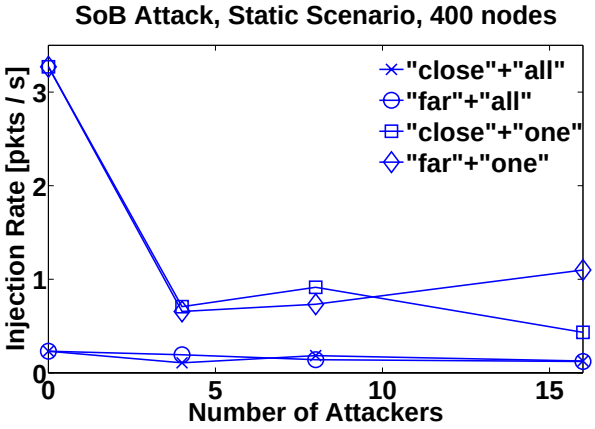


Figure 4.5: Rate of a victim facing SoB attack.

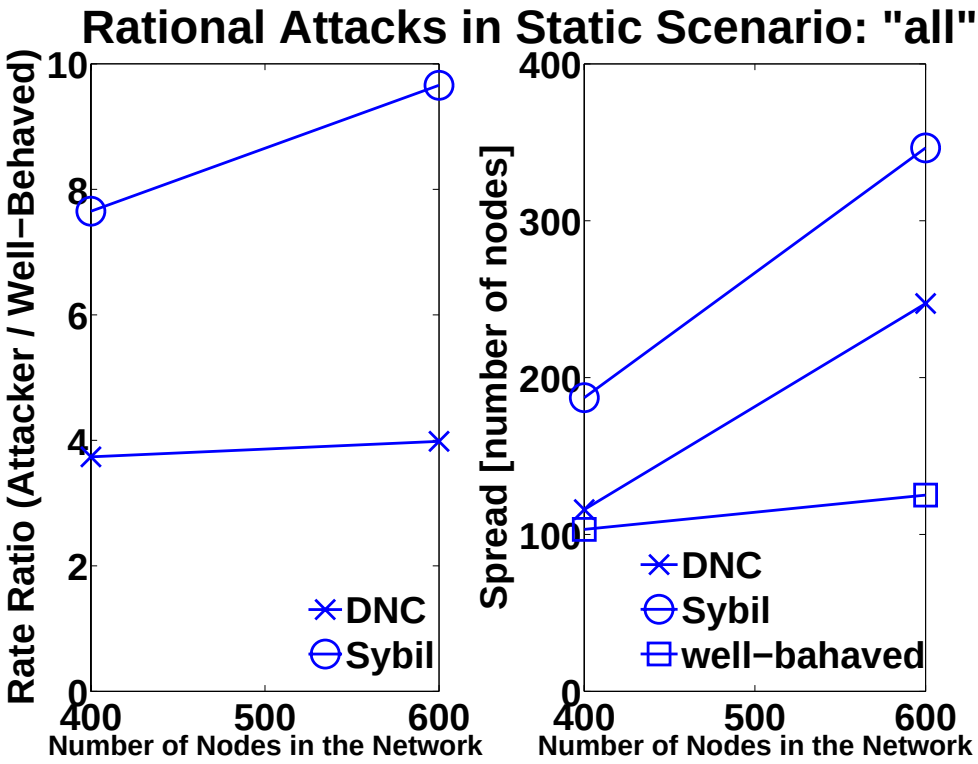
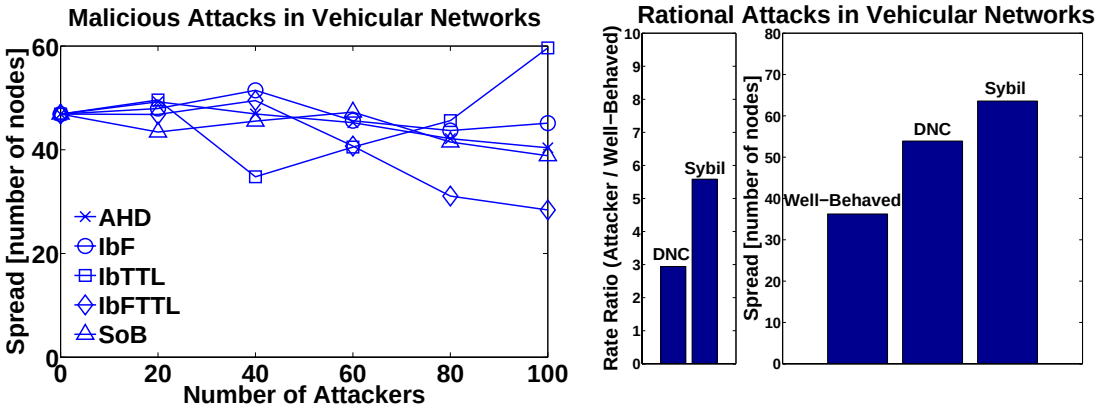


Figure 4.6: Rational attacks in static scenario.



(a) (b)

Figure 4.7: Vehicular Network. Malicious (a) and Rational (b) attacks, “all” scenario.

Chapter 5

Validation and Testbed

5.1 Introduction

Throughout our work on data dissemination, the focus is on practical design that holds in the real world, not only in simulation. Therefore, to validate the claim that SLEF is a complete practical design, an implementation for real platforms and functioning on real devices is needed. Further, this implementation must be able to run on resource-constrained devices such as smartphones.

We validate our claim through successful implementations of SLEF on four different platforms and with different hardware; among the tested hardware there are a smartphone and wireless router. Then, we want to go one step further in validating SLEF. We want to stress test SLEF in real circumstances. Therefore, we build an experimental testbed that involves more than fifty wireless devices. The testbed shows many key features such as the configuration ease of the wireless interface, mobility and robustness. Using this testbed, the stress testing of SLEF is accomplished and, moreover, we compare it to fixed TTL and show that SLEF performs significantly better.

5.2 SLEF Validation

Except for the pseudo-broadcast functionality (see Sect. 3.2.7), SLEF is networking-stack independent. It requires a simple broadcast service from the underlying layer. This service can

be delivered by the IP stack, where SLEF can be implemented at the application layer and use UDP sockets. However, SLEF does not need any IP functionality and can be functional directly on top of MAC layer. For instance, SLEF does not need an IP address, and it can simply use the MAC address as a unique node Id; which is very practical in self-organized ad-hoc networks, where assigning IP addresses dynamically does not scale with the network size.

The only SLEF function that depends on the layer beneath is the pseudo-broadcast. It is proposed to increase the efficiency and the reliability of the 802.11 broadcast, and it should be ignored with other underlying layers. SLEF communicates with the 802.11 layer through raw sockets [6]. In this case, simple Ethernet frames are used in the implementation. The type field of the Ethernet header is different from the one used by IP. Therefore, these frames do not interfere with the packets of IP applications that could be running on the same node or on other nodes in the same network. One advantage of this approach is that IP networking does not have to be initialized on the node, as no IP address is needed.

In our implementations, we consider both architecture: SLEF using UDP sockets and SLEF using raw sockets. Further, we target four platforms: Windows XP, Windows Mobile, Linux and OpenWrt.

5.2.1 Windows XP

With Windows XP, we implemented SLEF in both architectures in two programming languages: Java (J2SE) and C++. With both, implementing SLEF using UDP sockets is straightforward. In contrast, implementing SLEF using raw sockets, uses MAC-level functions such as sending or receiving Ethernet frames and setting parameters of the wireless network card. This requires specific libraries. We address this issue with both languages differently.

J2SE

The required MAC-level functions are implemented using JPCap [4], a library that allows raw ethernet frames from Java to be sent and received. JPCap relies on PCap [12], a C library that provides a high-level interface to packet capture systems. PCap is OS specific but it has already been ported for many OSs. This library is included in standard Linux distributions and an interactive setup for Windows is available. It is necessary to use this library in order to

implement the pseudo broadcast functionality. With standard Java sockets (UDP sockets), it is not possible to receive packets that are addressed to other nodes, as they are discarded either by the network card or by the IP layer. In contrast, pseudo-broadcast assumes that the wireless card is configured in the promiscuous mode and it receives frames addressed to other nodes.

C++

Executing MAC-level functions, such as changing parameters of the network card, sending raw socket Ethernet frames or receiving them, requires the invocation of methods of the driver of the network card. Windows provides a unified interface to communicate with drivers. This interface is called Network Driver Interface Specifications (NDIS).

In NDIS, communication between device drivers and the applications is done through *miniport drivers*. Miniport drivers can be seen as a collection of callbacks that forward messages either to the lower-layer (device driver) or the upper-layer (application). Two well-known miniport drivers are NDISProt and PKTDRV. The former is provided by default with Windows and the latter is a part of WinPCap software [12]. WinPCap also contains a higher-level library called PACKET32, which involves some higher-level functions but still offers the low-level functionalities of PKTDRV. In our implementations, we consider independently both, NDISProt and WinPCap. Both achieve the required functions [47].

5.2.2 Windows Mobile

We have chosen the smartphone HTC S620 for development. It runs on Microsoft Windows Mobile 5.0 platform with 64MB of RAM and 128MB of ROM. Its processor is the Texas Instruments OMAP 850 running at 201MHz. It is equipped with a Wi-Fi interface. It is a non-touch screen device, it offers a compact QWERTY keyboard and a large QVGA display (320x240 with 65,536 colors).

With this platform, we met difficulties while working with raw sockets, therefore our developement was successful only with UDP sockets. We did as with Windows XP, we realized two distributions of SLEF for Windows Mobile: The first is with Java (J2ME) and the second is with C++. With J2ME, we adopted J9 Java Virtual Machine (JVM) [2], as accessing the UDP sockets with other JVM was denied for security issues.

5.2.3 Linux

With Linux, both architectures are implemented successfully in Java (J2SE) and in C++. With Java, the implementation is similar to the Windows XP's one. With C++, accessing raw sockets and MAC-level functions does not require additional libraries, using raw sockets is very similar to UDP sockets.

5.2.4 OpenWrt

OpenWrt is described as a Linux distribution for embedded devices such as wireless routers [9]. More specifically, OpenWrt provides a fully writable filesystem with package management. This frees the user from the application selection and configuration provided by the vendor and allows the user to customize the device through the use of packages to suit any application. For developers, OpenWrt is the framework for building an application without having to build a complete firmware around it; for users this means the ability to fully customize, to use the device in ways never envisioned.

Our development targets the ASUS WL-500G Premium wireless router. It is very resource constrained with 8MB of flash memory, 32MB of RAM and a processor running at 266MHz. It is clear that it is more limited in memory space than the HTC S620 smartphone (see Sect. 5.2.2). We did not succeed in installing a Java virtual machine on this device, therefore we proceeded only with C++. To deal with the memory space limitation issue, SLEF implementation replaces the Standard Template Library (STL) by uSTL, a specific standard template library designed for embedded devices. This library uses far less memory than STL. We successfully tested SLEF with pseudo-broadcast using raw sockets. The other architecture using UDP sockets is not tested, but doing so should be straightforward.

5.3 Testbed

In order to stress test SLEF and validate its performance in the real world, we build an experimental testbed for wireless ad-hoc networks. Our testbed is not limited to SLEF, but it is of general use and it aims to validate theoretical findings and simulation results in the field of wireless communications. For instance, beside evaluating SLEF performance, the testbed

is also currently used for measurements of mesh network protocols [18]. It consists of 57 wireless routers each equipped with a Wi-Fi interface. We accomplish stress testing of SLEF and, moreover, we run a campaign of measurements to understand the testbed behavior and we compare SLEF to a fixed-TTL based epidemic forwarding.

5.3.1 Testbed Features

- **Wireless Router:** The deployed router is ASUS WL-500G Premium v1 wireless router that has a Broadcom 4318 wireless card in miniPCI slot.
- **Technical Specifications:** These devices are very resource constrained. Each device involves a flash memory of 8MB, a RAM of 32MB and a processor running with a clock of 266MHz. Beside the wireless interface and the WAN port, it has two USB 2.0 ports and four RJ45-10/100BaseT ports. USB memory keys can be used for SWAP.
- **Firmware:** We flashed these devices with OpenWrt, a Linux-like firmware that frees the user from the application selection and the configuration provided by the vendor and allows the user to customize the device through the use of packages to suit any application.
- **Configurable Wireless Interface:** To have a fully configurable wireless interface, we use MadWifi, an open source WLAN driver for Linux, developed for Atheros chipsets [7]. Thus, we had to replace the card that comes with the router (broadcom chipset) by an Atheros card [10]. It supports IEEE 802.11a/b/g. Consequently, we were able to access and set almost all the wireless card parameters such as transmission power, promiscuous mode, monitor mode, RTS/CTS threshold, transmission queue length and transmission rate.
- **Mobility:** Mobility is ensured through adding to a router a plumb battery that lasts for more than four hours when transmitting at full rate and max power. Therefore, using this battery, a node is independent of a fixed plug and it can be functional at any place and even while moving.
- **Robustness:** Throughout our measurements, the testbed has shown to be very robust. No unexpected router behavior, such as auto-reset was noticed during the experiments.

5.4 Measurements

Using our testbed, stress testing is accomplished successfully. Then, we run extensive series of measurements to understand the behavior of the testbed. Further, we compare SLEF to a fixed-TTL based epidemic forwarding. The latter employs the same functions as SLEF, except for spread control. With fixed TTL, the TTL of a packet is fixed and thus, is allowed to travel at most through TTL hops. In the case the epidemic buffer is full, the buffer management mechanism drops the packet with the smallest TTL even before the latter TTL expires. Thus, the spread control with fixed TTL uses two parameters, the TTL and the buffer size and, contrary to SLEF, it is not adaptive.

5.4.1 Measurements Design

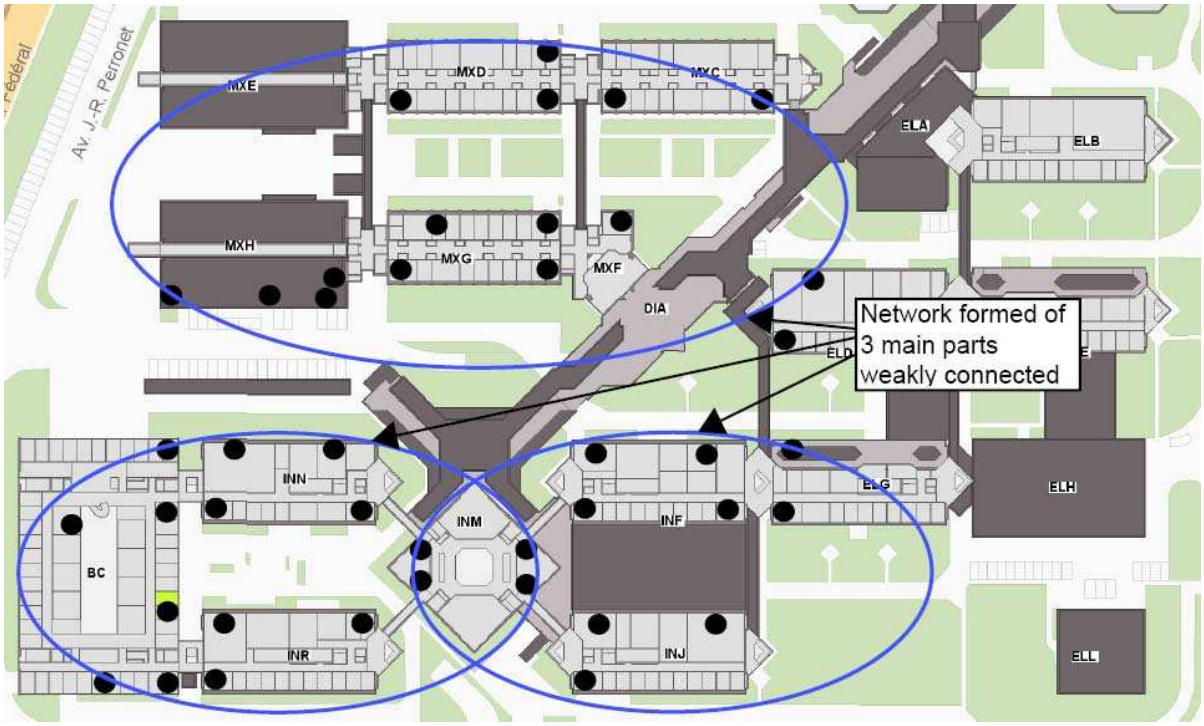


Figure 5.1: The testbed is expanded over 12 buildings in EPFL. Each point corresponds to a wireless node. The network is formed mainly of 3 main parts, that are connected through links that perform poorly in the presence of interference.

We distributed the wireless nodes over 12 buildings at EPFL (see Fif. 5.1). The network is static. The average node degree is around 5. We developed a simple application on top of SLEF. It injects packets at a constant rate if it is allowed by SLEF. In other word, according to the network conditions, the congestion control mechanism of SLEF might reduce the application rate if it can not be supported by the network. All nodes are sources; each one runs only one instance of this application.

During the testing phase, we try to make conclusions based on three metrics. The first is the application rate. The second is the redundancy, reflected by the value of the forwarding factor. The third is the spread at a given percent; the spread at $x\%$ of a node n is the number of nodes that receive $x\%$ of node n packets.

5.4.2 Measurements Results

Impact of Buffer Size

First, we recall that the study shown in this section aims at understanding qualitatively the system behavior and it does not consist of a detailed analysis. In this scenario, the application rate is fixed to 1 packet/s at most. We start with an experiment with SLEF. The parameters are set as follows: $a = 0.1$, $b = 0.01$, $K_0 = 25$, $K_1 = 0.1$ and the buffer size is set to 3000 packets. We compute the buffer occupancy average during this experiment, it is around 620 packets, far less than the buffer size.

Then, we repeat the same experiment but with fixed-TTL and while setting the buffer size to the buffer occupancy obtained with SLEF. Finally, we repeated the experiment with fixed-TTL but with a very large buffer size of 10000 packets.

Fig. 5.2 shows the spread of node 1 with SLEF. It could inject 3100 packets/hour. With fixed-TTL with small buffer size, node "1" could inject slightly more packets (6% more), but at the expense of the spread: with SLEF, the furthest nodes receive more than 40% of node "1" packets, whereas this fraction is only 26% with fixed TTL (see Fig. 5.3). With SLEF, we notice that the fraction of node "1" packets received by a given node decreases smoothly with increasing hop-count, whereas it shows a sharp transition with fixed-TTL for the sixth hop, e.g. while moving from node "34" to "35". This is because the buffer management mechanism drops the packets based on their TTL and not on their age. Hense, a packet with

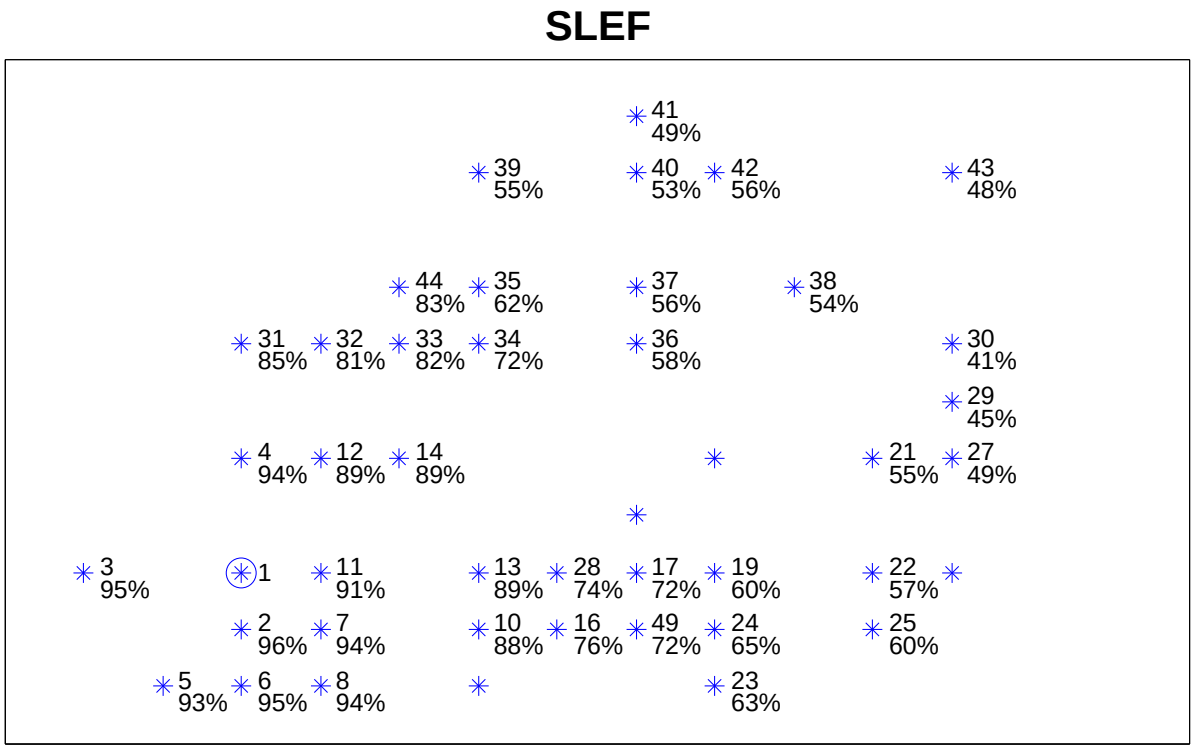


Figure 5.2: Each star corresponds to a wireless node. The positions of the stars correspond to the real positions of nodes in the network. Next to each star, it is indicated its number and the fraction of node ”1” packets received at this node. This later decreases smoothly with increasing hop count.

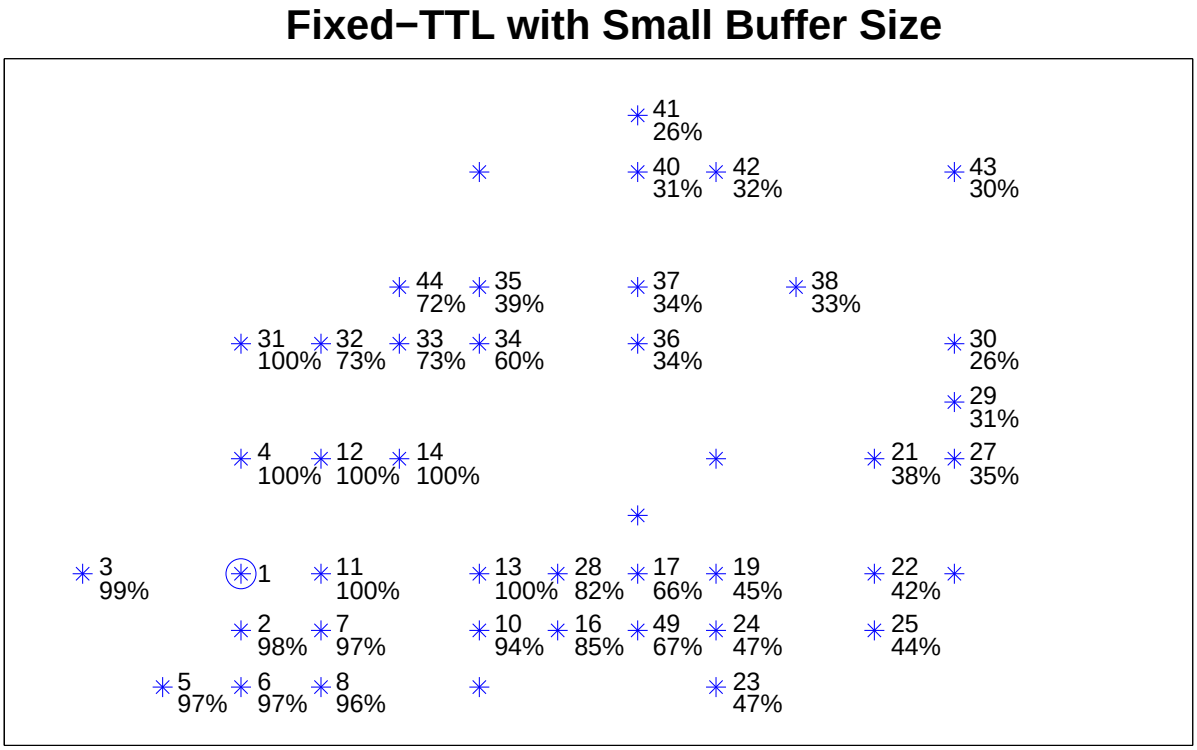


Figure 5.3: The fraction of node "1" packets received at other node shows a sharp transition at the 6^{th} hop, e.g. from node "34" to node "36".

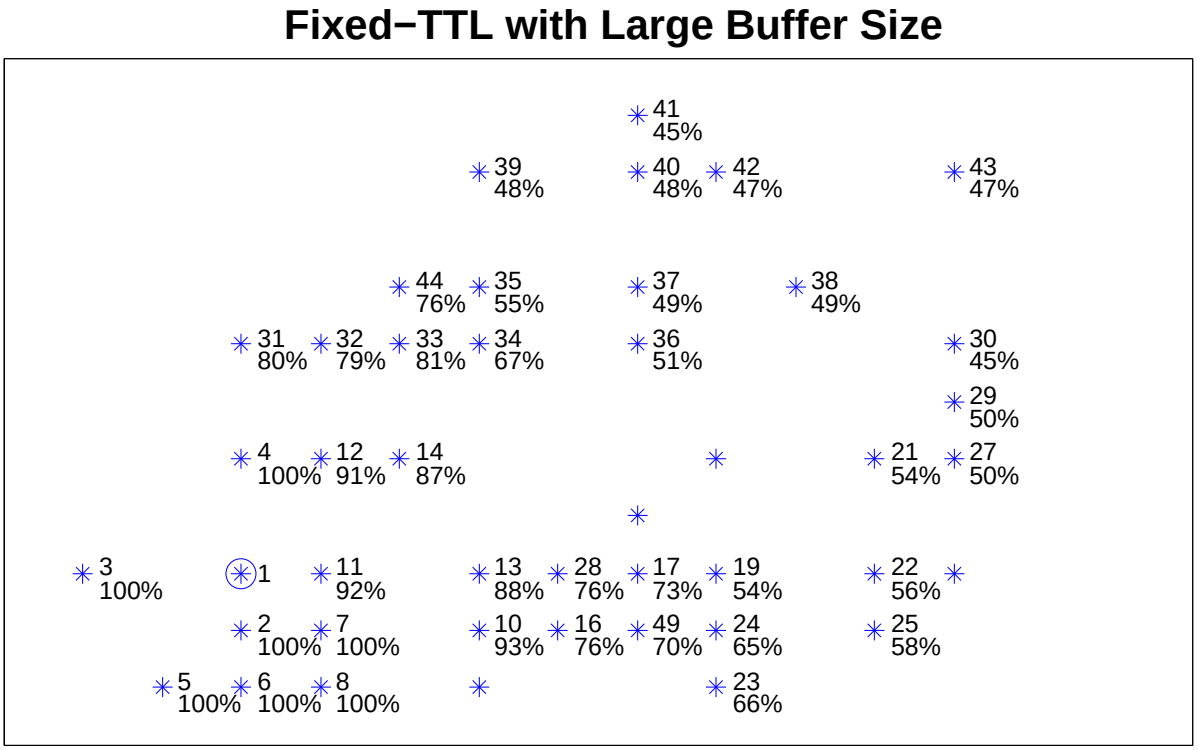


Figure 5.4: The fraction of node ”1” packets received at other node decreases smoothly with increasing hop count.

large TTL lives for long time in a node epidemic buffer independently of the number of the send/receive events it sees during its stay, whereas a packet with small TTL is dropped as soon as it arrives into the epidemic buffer in case there is no more space for it. The epidemic-buffer size defines the TTL decrement threshold at which the sharp transition in the fraction of node "1" received packets appears. In our case, this threshold is around 6.

Increasing the buffer size increases this threshold. In the case of fixed TTL with the large buffer size (see Fig. 5.4), the sharp transition disappears, as the threshold is larger than the network size, and the spread is very close to the one obtained with SLEF (Fig. 5.2). But our measurements show a huge amount of redundancy compared to SLEF, which causes a decrease in the application rate of more than 30%.

Pseudo-Broadcast

In our case, implementing the pseudo-broadcast is very simple: We configure the wireless card in promiscuous mode, and then packets at a given node are sent to the source MAC address of the last received packets. In order to activate the RTS/CTS mutual exclusion, the RTS/CTS threshold in the wireless card driver is set to "0". Fig. 5.5 shows that, without pseudo-broadcast, the spread at 80% of node "1" is reduced to 25%

Link Quality Effect

To explore the effect of link quality on limiting the spread, we ran SLEF in the presence of IP traffic and compared it with the same SLEF traffic load alone. The results are depicted in Fig. 5.6. In the presence of IP traffic, the spread at 80% of node "1" is reduced to 35%. Indeed, as it is shown in Fig. 5.1, this network is formed of three main parts that are connected through weak links. Increasing the traffic load in the network increases the interference and the collision rate. Therefore, the weak links perform poorly. Consequently, the three parts of the network become almost isolated. Thus, in the presence of a high IP traffic load, the spread is reduced due to the bad quality of the links and not by the spread control mechanism, as both scenarios, with and without IP traffic, exhibit the same SLEF load. Note that the spread control mechanism does not consider anything other than SLEF traffic.

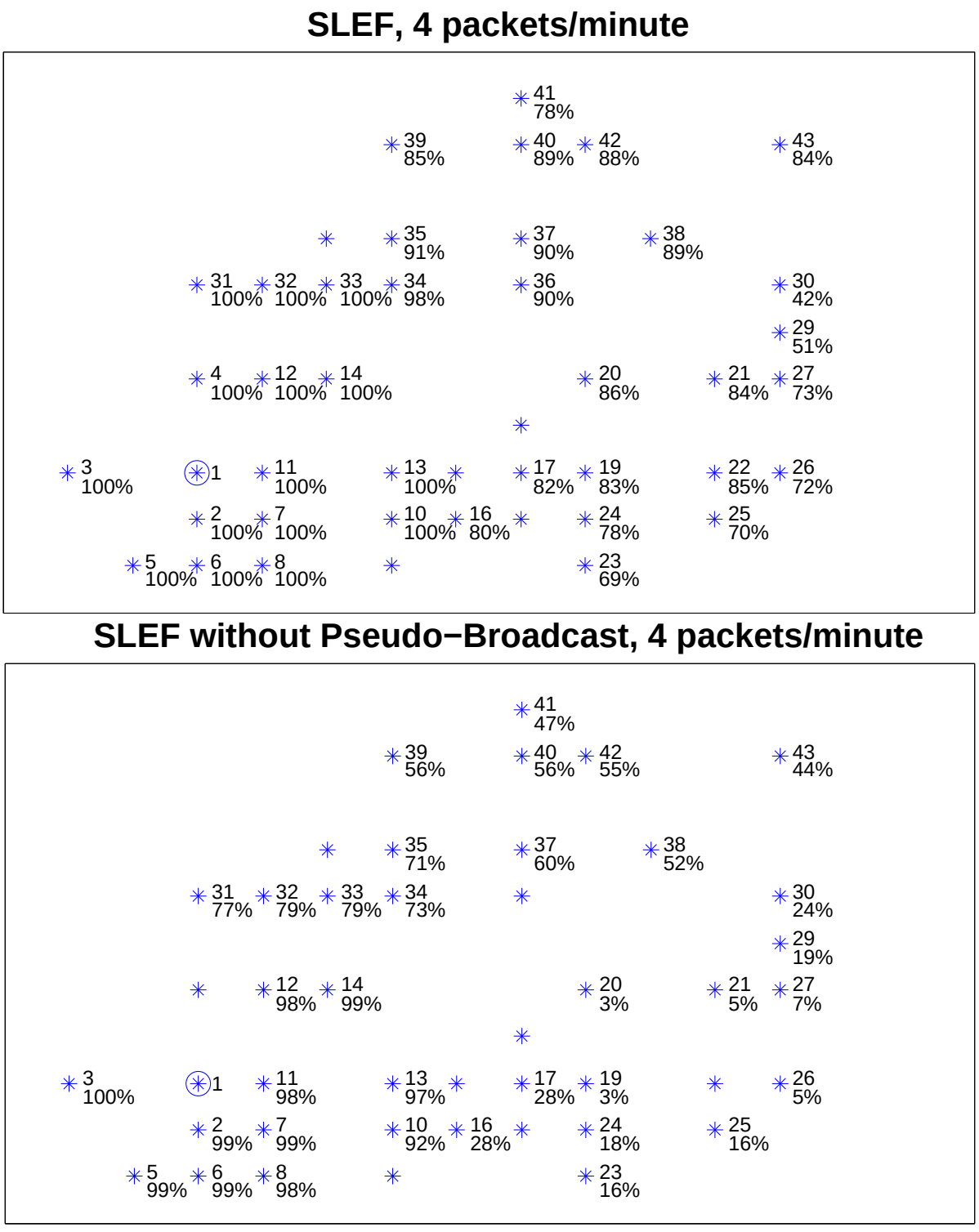


Figure 5.5: Without pseudo-broadcast, the spread at 80% of node 1 is reduced to 25%.

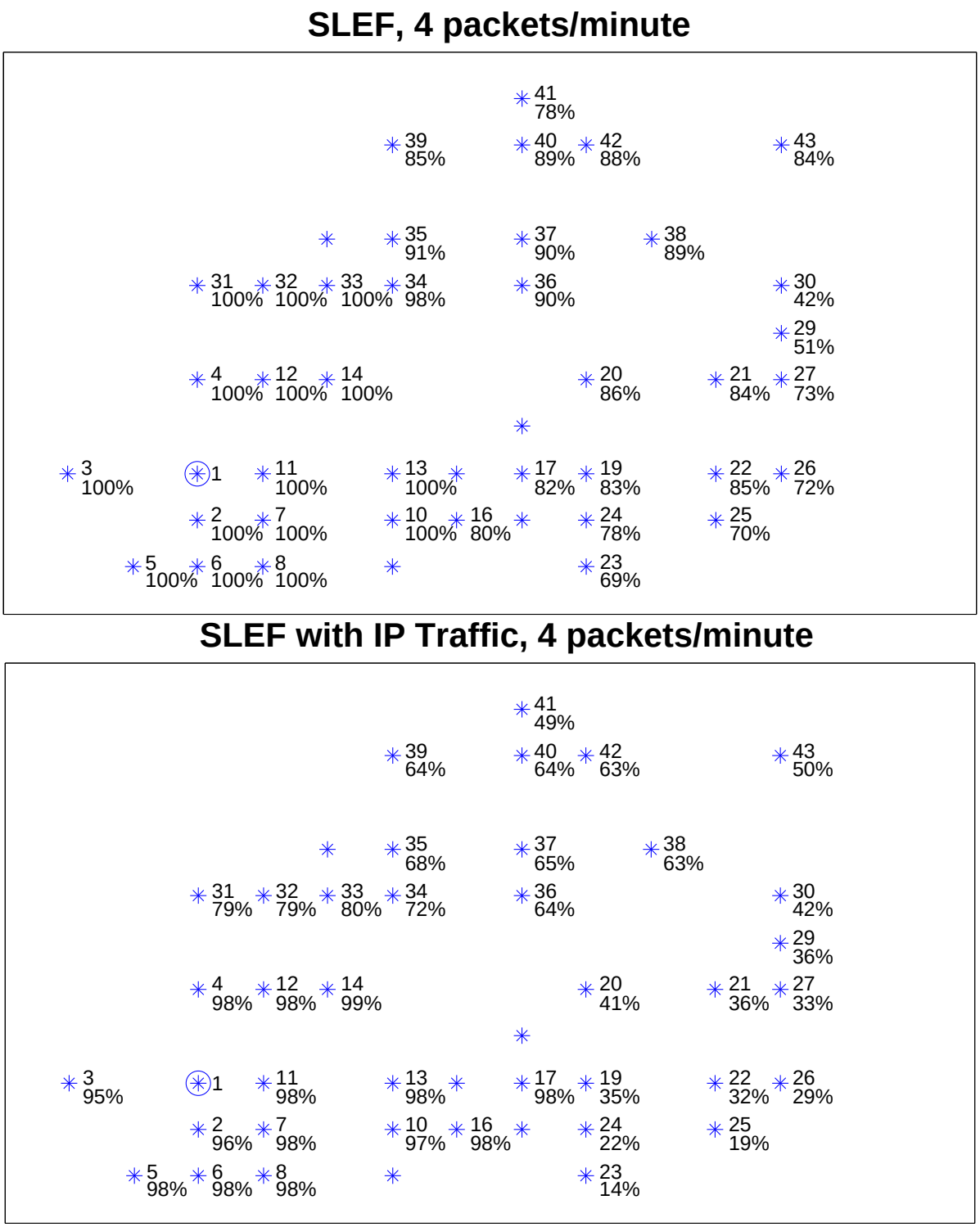


Figure 5.6: In both scenarios, we have the same SLEF traffic load with an application rate of 4 packets/minute. In the presence of high IP traffic load, the links among the 3 main parts of the networks (see Fig.5.1) perform poorly. Therefore, the spread at 80% of node "1" is reduced to 35% as packets have little chance to escape from one part to another.

5.5 Conclusions

In order to prove that SLEF is a practical design, we implemented SLEF targeting four different platforms: Windows XP, Windows Mobile, Linux and OpenWrt. Our implementations validate our claim and prove that SLEF does not need any additional mechanism to be functional. Some of the adopted hardware for our development are the HTC S620 smartphone and the ASUS WL-500G Premium wireless router.

Then, we want to stress test SLEF in the real world, far from the simulation simplifications. Therefore, we build a solid and widely adaptable experimental testbed for wireless networks. It is composed of 57 wireless routers. We show the feasibility of running SLEF on such devices that are very resource-constrained. Then, we show measurement results that aim at explaining the behavior of the testbed and we compare SLEF to fixed TTL. One of these results is that a large buffer size with fixed-TTL based epidemic forwarding results in a huge amount of redundancy, which reduces the application rate by more than 35% compared to SLEF. Then, we show that, without pseudo-broadcast, the spread is reduced to 25% due to collisions. Finally, we show that, in the presence of high traffic load, the spread might be reduced, not only due to the spread control mechanism, but also due to bad link quality. Indeed, a high traffic load increases significantly the interference, which breaks the poor links.

After stress testing SLEF, our perspective is to begin a campaign of measurements of SLEF. We aim at evaluating the performance of SLEF through a factorial analysis applied on the measurement results.

Part II

Cross-Layer Optimization for Ultra-Wide Band Impulse Radio

Chapter 6

Physical Layer Model and Assumptions

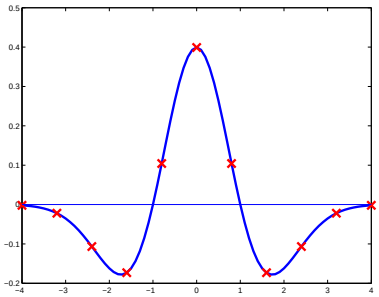


Figure 6.1: Second derivative of the Gaussian pulse. Adopted through out our evaluation.

An IR signal consists of trains of very short pulses to the order of a nanosecond or even a sub-nanosecond with low duty cycle. We consider the second derivative of the Gaussian pulse Fig. 6.1. We chose this pulse because we assume that the transmitter generates the Gaussian pulse, which undergoes 2 derivations at the transmitter and receiver antennas respectively. Thus, to simplify the notation we consider only the pulse received at the correlator, which is the second derivative of the Gaussian pulse, instead of using three different pulses. This simplification does not change the results of this study.

In the literature, two physical layers are proposed in order to organize these pulses and manage their modulation and the channelization of the medium. The first is proposed by Win-Scholtz [73], and we refer to it as the classic physical layer. The second is the 802.15.4a physical layer [44]. Both proposals adopt a time hopping scheme to transmit pulses. With both, a packet consists of a synchronization preamble followed by a data part, but they differ

in the data part. We are concerned only in the preamble, as we are interested only in synchronizing a sender to a receiver. The preamble serves merely for synchronization purposes and does not carry data, therefore the pulses inside are not modulated. UWB-IR modulation concerns only the data part of a packet and it is out of the scope of this dissertation.

6.1 UWB-IR with Time Hopping

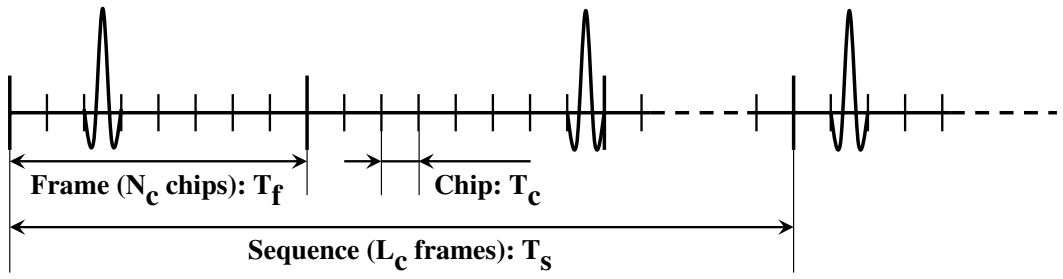


Figure 6.2: UWB-IR physical layer with TH.

With TH, time is slotted in chips of very short duration, T_c, of the order of the pulse width but generally larger; chips are organized in frames of length $T_f = N_c \times T_c$, where N_c is the number of chips in one frame (Fig. 6.2). A node transmits one pulse in one chip per frame, and it uses a pseudo-random code to determine in which chip to transmit. TH codes permit different sources to share the channel, i.e., source destination pairs use independent pseudo random code. Furthermore, they avoid energy peaks in the frequency domain. Frames are organized in sequence of duration of $T_s = T_f \times L_c$, where L_c is the code length. The sequence is repeated over all the preamble duration. Thus, the transmitted signal of the m^{th} user is:

$$s^{(m)}(t) = \mathscr{E}^{(m)} \sum_{j} \sum_{k=1}^{L_c} p\left(t - (c_k^{(m)} - 1)T_c - (k-1)T_f - jT_s - \tau_X^{(m)}\right) \tag{6.1}$$

where $p(t)$ is the second derivative of the Gaussian pulse (Fig. 6.1), $\mathscr{E}^{(m)}$ indicates the signal amplitude, $c_k^{(m)}$ is the k^{th} element of the m^{th} user code, i.e. the number of the chip that corresponds to the pulse position in the k^{th} frame of a m^{th} user sequence and $\tau_X^{(m)}$ is the transmission start time. We assume that the pulse width and the chip duration are equal.

6.2 Channel Impulse Response

We consider the Saleh-Valenzuela (SV) channel model [65] adopted in [14]. For simplicity, we express its impulse response using the well-known tapped delay line expression:

$$h(t) = \sum_{l=1}^{L} a_l \delta(t - t_l) \tag{6.2}$$

where $\delta(t)$ denotes the Dirac impulse, t_l the signal delay along the l^{th} path and a_l is a real propagation coefficient that includes the channel attenuation and the polarity of the signal along the l^{th} path.

This is a relatively simple model that ignores frequency and path dependent effects on the shape of the transmitted pulse $p(t)$ [27]. Also, this model is time-invariant as a_l and t_l do not depend on t. Typically, this assumption corresponds to a packet based network where the channel impulse response is considered to be invariant along the duration of a packet transmission. A channel impulse response is then sampled from a given distribution for each packet to be transmitted. There exists a large body of work regarding the characterization of propagation environments and of distributions for channel impulse responses [42, 65], and in particular for UWB channels [15, 23, 27, 39, 58].

In this dissertation, we consider the indoor office environment defined by IEEE P802.15.41 study group [14] and we evaluate our detection method (see Chap. 7) in both cases, the line-of-sight (LOS) and the non-LOS (NLOS).

Fig. 6.3 illustrates one realization of the channel impulse response in the LOS case. The closest path differential delay is of the order of a few ns, it is larger than the pulse width, which is of $0.2ns$ in our case. This shows the multipath resolvability of UWB-IR: multipath pulses do not overlap, which avoids the destructive interference.

6.3 Concurrent Transmissions

Being impulsive with low duty cycle, allows for UWB-IR concurrent transmissions without collision. Indeed, two UWB-IR signals collide if all their pulses and their multipath replicas overlap. This requires that both signals have the same TH code, the same arrival time at the

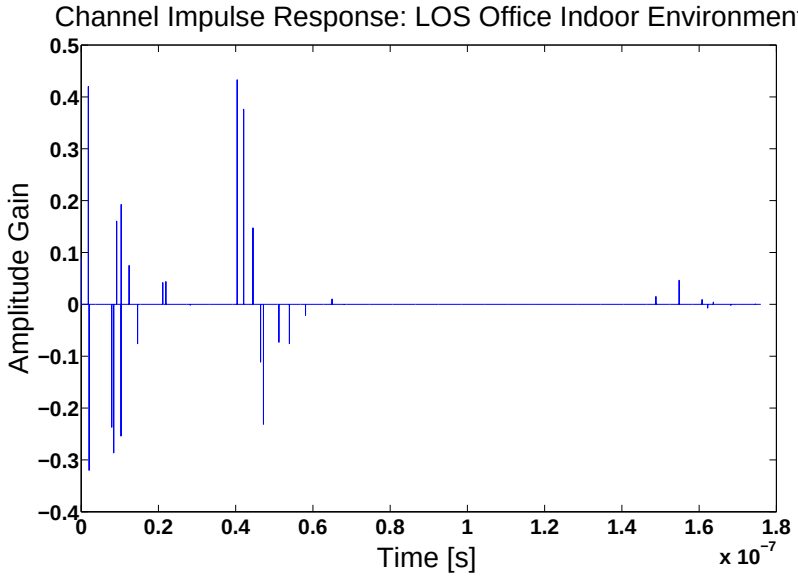


Figure 6.3: A realization of the channel impulse response. With a pulse width of $0.2ns$, all multipath components are resolvable.

receiver and the same channel impulse response, which has no chance to happen. Further, using different TH codes for different users further reduces the probability of collision.

With M concurrent transmitters, the received signal can be obtained from Eq. 6.1 and Eq. 6.2 and it is:

$$r(t) = \sum_{m=1}^{M} \sum_{l=1}^{L^{(m)}} a_l^{(m)} s^{(m)} \left(t - t_l^{(m)} \right) + n(t) \tag{6.3}$$

where $n(t)$ is the thermal noise. Eq. 6.3 and Eq. 6.1 show that two parameters have a major effect on the inter-user-interference (IUI). The first is T_f, which should be very large to ensure low duty cycle. The second is L. Indeed, the share of each transmitter of the medium is $\frac{L \times T_c}{T_f}$. Thus, the larger T_f compared to $L \times T_c$ is, the lower the IUI is and the larger the number of successful concurrent transmissions is.

6.4 Synchronization and Signal Acquisition

The first step toward a correct reception of a packet is that the receiver has to detect the signal and get synchronized with it. This is the purpose of using a synchronization preamble at the beginning of each packet. Two nodes communicating with each other need to agree on a TH code on which the receiver is listening. In the literature, several works address the distribution of the TH code [66], but this is out of the scope of this dissertation.

Assume that a receiver is interested in detecting the signal sent by the first user. Then, the objective of the synchronization process is to detect whether the first user is transmitting or not, and if he is transmitting, it finds the arrival time of one sequence in the preamble of the first user's signal. The receiver can get synchronized with any multipath component that has enough energy. Thus, the receiver needs to find one value of $\left\{ \left(\tau_X^{(1)} + jT_s + t_l^{(1)} \right),\ l = 1, \ldots, L,\ j = 0,\ 1,\ \ldots \right\}$, let τ_0 be the found value. Furthermore, it detects the sign of the corresponding a_l. In the remainder of this dissertation, the terms synchronization and signal acquisition are used interchangeably to designate this procedure.

In our simulation, we consider a frame time, T_f, larger than the delay spread of the channel in order to minimize the inter-symbol interference. We assume the channel is stationary during the synchronization phase. We do not make any assumption about the separation of the channel taps, so pulses might or might not overlap after convolution with the channel impulse response. However, this overlapping happens rarely because we adopt a very short pulse of 0.2 ns. Moreover, overlapping may deteriorate the signals and thus non-overlapped pulses have more chance of being detected.

Chapter 7

A Robust Signal Detection Method

7.1 Introduction

We propose a novel detection method, called PID (Power Independent Detection) method, for non-coherent synchronization in multi-access Ultra Wide Band (UWB) Impulse Radio (IR) networks. To understand what we mean by detection method, let us define the following terminology. We consider the synchronization of one receiver to one sender (also called signal acquisition). We are interested in methods based on the correlation of the IR signal with a Template Pulse Train (TPT). Such methods involve two ingredients: (1) the detection, which correlates the received signal with a TPT, we refer to this detection method as the conventional detection method, and (2) the search algorithm, which shifts the TPT. We focus on detection. Our proposal aims at solving the extreme Inter-User Interference (IUI) case (near-far problem), when there are multiple interfering transmitters, asynchronous transmissions and heterogeneous power levels. This occurs, for example, in the presence of multiple interfering piconets, or in purely ad-hoc networks that allow concurrent transmissions, always at full power [19,54]. As a typical example, we can imagine a headphone set employing the IR UWB technology to exchange music with some master device such as a laptop. Several people may use these headphones with different masters in an office environment or even in the same room. They may move or exchange places, which creates very harmful interference. Another application could be the sensor networks where the IR UWB technology is a potential candidate because of its low power consumption. We can imagine tens or even hundreds of sensors deployed in a small

area communicating with each other in an ad-hoc fashion with a huge amount of interference. In such scenarios the conventional detection method faces a certain failure in the absence of power control, which may entail a prohibitive overhead. Further, according to [62] the optimal scheduling is to allow sending at full power and to apply rate adaptation but not power control.

7.2 List of Global Notation

In the following, we are listing the notations used globally throughout this chapter.

General Notations

- FA: False Alarm
- IR: Impulse Radio
- IUI: Inter-User Interference
- LOS: Line Of Sight
- NLOS: Non Line Of Sight
- PID: Power Independent Detection
- UWB: Ultra Wide Band
- TPT: Template Pulse Train

Physical Layer Parameters

- L_c: code length
- T_c: chip duration
- T_f: frame duration
- T_s: sequence duration
- $\mathscr{E}$: signal amplitude

- E_0/N_0: the bit energy to noise spectral density (one pulse is sent per bit)
- τ_0: detected arrival time

Different Signals

- $p(t)$: second derivative of the Gaussian pulse
- $r(t)$: received signal

Conventional Detection Method Parameters

- α_i: the output of the i_{th} elementary correlation
- β: sum of the elementary correlation outputs with the conventional detection method
- γ: the threshold with the conventional detection method

PID Method Parameters

- α_i: the output of the i_{th} elementary correlation
- θ: the elementary threshold with the PID method
- χ: sum of the elementary threshold check outputs with the PID method
- φ: the main threshold with the PID method

Complete Synchronization Method Parameters

- χ_{max}: the largest χ obtained during the first phase or one iteration of the second phase of the PID synchronization method
- $\varphi^{(1)}$: the value of φ during the first phase
- $\varphi^{(2)}$: the value of φ during the second phase
- A: number of iterations in the second phase

- B: minimum number of succeeded threshold checks during the second phase so that detection is declared
- SB: the signal bin detected in the first phase of the complete synchronization method
- V: a predefined neighborhood of SB used for the search in the second phase

Probability Notations

- P_{MD}: probability of Missed Detection
- P_{FA0}: probability of False Alarm in the absence of the true sequence
- E_t: total error, $E_t = P_{MD} + P_{FA0}$
- P_1: the probability of good detection during the first phase in the presence of the true sequence
- P_2: the probability of a bad detection during the first phase in the absence of the true sequence
- P_3: the probability that one threshold check succeeds during the second phase, given that the first phase has resulted in a good detection in the presence of the true sequence
- P_4: the probability that one threshold check succeeds during the second phase, given that the first phase has resulted in a bad detection in the absence of the true sequence

7.3 Conventional Detection Method

7.3.1 Description

As it is explained in Sect. 7.1, we consider synchronization methods that involve two ingredients: the detection and the search algorithm. With the conventional detection method, the received IR signal is correlated with a TPT, which is a replica of the sequence used by the first user and which is given by:

$$s_{TPT}(t) = \sum_{k=1}^{L_c} p\left(t - (c_k^{(1)} - 1)T_c - (k-1)T_f\right) \tag{7.1}$$

The idea behind the correlation is to compare the TPT with the received impulse radio signal, which may or may not have the identical pattern of pulses as the TPT. Then a threshold check is performed on the output of the correlation (β in (7.2)) to detect whether there is a match (an alignment) between the TPT and the received IR signal.

The role of the search algorithm is to shift the TPT, with predefined time offsets, so that the TPT is placed at various locations in time, as compared to the received impulse radio signal, until a match is obtained between them, i.e. they are aligned. The output of the cross-correlator is:

$$\beta = \int_{\sum_{i=1}^{n} \text{Offset}_i}^{\sum_{i=1}^{n} \text{Offset}_{i+T_s}} r(t) s_{TPT}\left(t - \sum_{i=1}^{n} \text{Offset}_i\right) \mathrm{d}t \tag{7.2}$$

where n is the current shift number and Offset_i is the time offset at the i^{th} shift of the TPT. (7.2) is known in the literature as a coherent integration, but in this paper we refer to it as a correlation between the TPT and the received IR signal (note that we do not assume that the receiver knows the channel). The receiver gets synchronized with the transmitter at the n^{th} offset if $\sum_{i=1}^{n} \text{Offset}_i$ is equal to one value of the set $\left\{\left(\tau_X^{(1)} + jT_s + t_l^{(1)}\right),\ l = 1, \ldots, L,\ j = 0,\ 1,\ \ldots\right\}$, and thus $\tau_0 = \sum_{i=1}^{n} \text{Offset}_i$. Notice that, according to (7.1), (7.2) can be interpreted as L_c elementary correlations $\{(\alpha_k)\}$, $k = 1, \ldots,\ L_c$. α_k is the output of the elementary correlation k that corresponds to the k^{th} pulse in the TPT. We can write:

$$\beta = \sum_{k=1}^{L_c} \alpha_k\ , \tag{7.3}$$

where:

$$\begin{aligned} \alpha_k \quad = \quad & \int_{(c_k^{(1)}-1)T_c+(k-1)T_f+\sum_{i=1}^{n} \text{Offset}_i}^{(c_k^{(1)}-1)T_c+(k-1)T_f+\sum_{i=1}^{n} \text{Offset}_{i+T_c}} p\Big(t - (c_k^{(1)} - 1)T_c \\ & - (k-1)T_f - \sum_{i=1}^{n} \text{Offset}_i\Big) r(t) \mathrm{d}t \end{aligned} \tag{7.4}$$

These L_c elementary correlations correspond to the L_c correlations of the TPT *pulses* and their corresponding intervals of the IR signal. In Fig. 7.1, the L_c elementary correlations are presented by the blocks indexed from 1 to L_c. β is the input of the decision block, which in turn performs a threshold check. Hence, a match between the TPT and the IR signal is

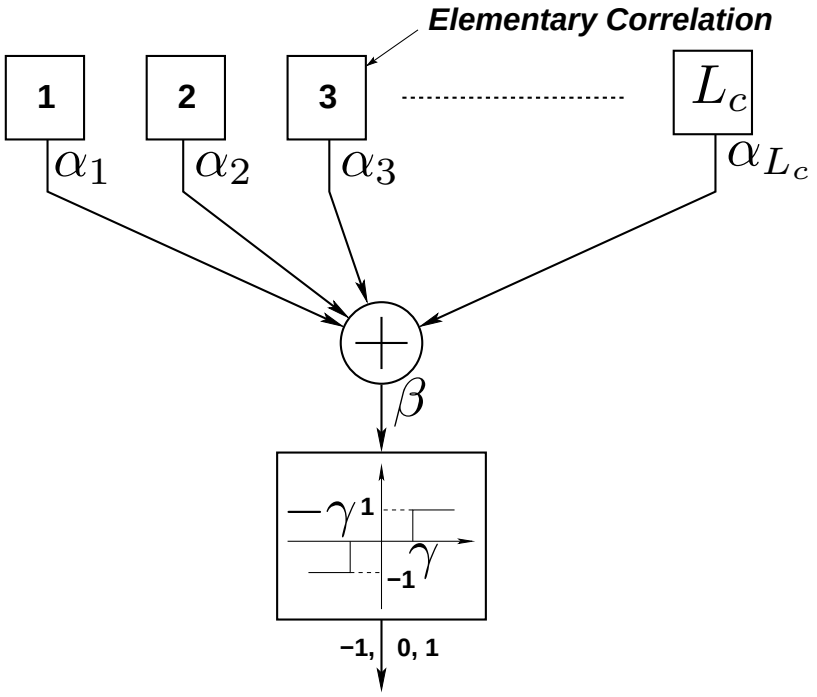


Figure 7.1: The conventional detection method can be interpreted as L_c elementary cross-correlations. Block i, $i = 1, \ldots, L_c$, presents the correlation of the i^{th} pulse in the TPT with its corresponding interval.

declared if the absolute value of β exceeds a certain threshold γ. Note that a (-1) output of the decision block means that a match is declared but the signal is inverted due to reflection, i.e. the corresponding a_l is negative (see previous section).

7.3.2 The Problem with the Conventional Detection Method

To show the inefficiency of the conventional detection method, we present one scenario, for an indoor environment, based on the measurement made by M. Win and R. Scholtz in [72]. Consider a source (user 1) that is 10 m from the receiver. The measurement in [72] gives that the amplitude of the strongest source pulse seen by the receiver is in the order of 0.03V. Assume now that there is an interferer (user 2) that is 1m from the receiver. The measured amplitude of the interfering pulse is 1V, 33 times higher than the source pulse. $\mathscr{E}_r^{(1)}$ ($\mathscr{E}_r^{(2)}$ respectively) referring to the source (respectively interferer) signal amplitude at the receiver,

we have $\mathscr{E}_r^{(2)} \approx 33\mathscr{E}_r^{(1)}$. Let $\alpha_0^{(1)}$ ($\alpha_0^{(2)}$ respectively) be the output of the correlation between one source (interferer respectively) pulse and one TPT pulse when they are aligned, we can write:

$$\alpha_0^{(1)} = \mathscr{E}_r^{(1)} \int_0^{T_c} p^2(t)\mathrm{d}t \;=\; \frac{\mathscr{E}_r^{(2)}}{33} \int_0^{T_c} p^2(t)\mathrm{d}t \;\approx \frac{\alpha_0^{(2)}}{33} \tag{7.5}$$

$\alpha_0^{(2)}$ is 33 times larger than $\alpha_0^{(1)}$. Note that when the source signal and the TPT are perfectly aligned, neglecting the interference and noise effects, β is equal to $L_c \times \alpha_0^{(1)}$. Consequently, γ cannot be larger than $L_c \times \alpha_0^{(1)}$, otherwise the source signal cannot be detected. If $L_c \leq 33$, it is sufficient to have one interfering pulse aligned with one TPT pulse to get a False Alarm (FA).

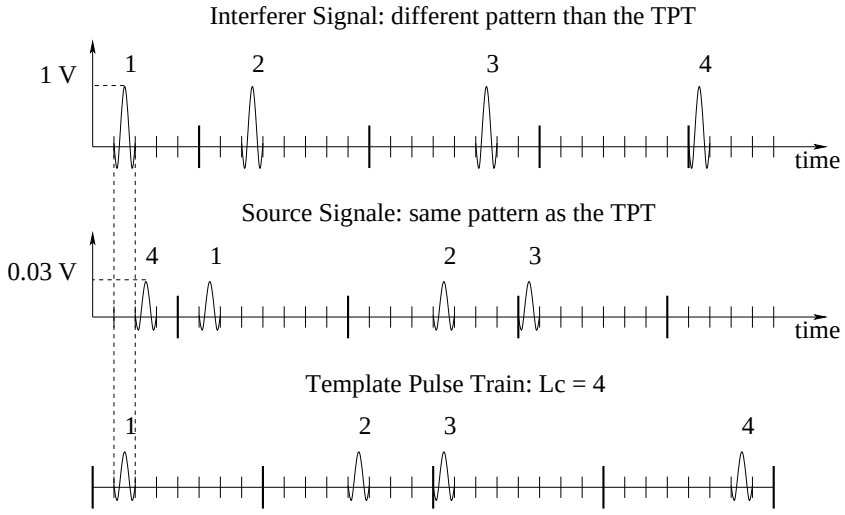


Figure 7.2: A scenario showing the problem with the conventional synchronization method. The Source signal has the same pattern as the TPT, but it is shifted in time. The interferer signal, which is 33 times stronger than the source signal, has one pulse aligned with one pulse of the TPT.

Fig. 7.2 illustrates this scenario with $L_c = 4$. The source signal has the same pattern as the TPT, but it is shifted in time. Corresponding pulses in the TPT and the source signal carry the same number. As we notice, there is one interferer pulse (pulse number 1 of the interferer signal) that is aligned with pulse number 1 of the TPT. In this case, an FA will occur because

the code length L_c is very small compared to the ratio between the source and the interferer signals.

To avoid this FA, but still using the conventional detection method, L_c must be much larger than 33, which would be an extremely unaffordable overhead in term of synchronization time, as the synchronization time is proportional to the code length L_c [35]. Note that, when the number of concurrent transmissions increases, the situation becomes worse.

To summarize this example, the synchronization is either unfeasible or entails an extremely large overhead using the conventional detection method in non-power control IR networks when concurrent transmissions are allowed.

7.4 Our Proposal: Power-Independent Detection Method

The idea behind the cross-correlation between the TPT and the IR signal is to detect a match between them. We need to find in the IR signal L_c pulses that have the same pattern as the TPT. But the conventional detection method does not allow us to do this. It looks at the energy captured by the correlation between the TPT and the received IR signal, which is indicated by β in Fig. 7.1, regardless of its distribution over the L_c elementary correlations. So, if this energy, β, is larger than the threshold, we say that the synchronization is achieved. But what happens if all the energy comes from one elementary correlation, e.g. $\beta = \alpha_1$ and $\alpha_k = 0$, $k = 2, \ldots, L_c$? This is the challenge in the scenario shown in section 7.3.2 in the case where $L_c \leq 33$. Unlike the conventional detection method, our PID method solves the problem by looking at the individual energy captured by each elementary correlation separately, i.e. by looking at each α_k separately, $k = 1, \ldots, L_c$. Fig. 7.3 describes the architecture of our proposal; the output of each elementary correlation α_k, $k = 1, \ldots, L_c$, passes through an elementary decision block that performs an elementary threshold check. If the absolute value of α_k is larger than the elementary threshold θ, then a pulse is detected and the output of the elementary decision block k will be 1 or -1 depending on the sign of α_k (-1 means the detected pulse has negative polarity). Otherwise it will be 0. Let χ be the sum of the L_c *Elementary Decision* block outputs, we have:

$$\chi = \sum_{k=1}^{L_c} \left(\mathbf{1}_{\{\alpha_k \geq \theta\}} - \mathbf{1}_{\{\alpha_k \leq -\theta\}} \right) \in \{-L_c, \ldots, 0, \ldots, L_c\} \tag{7.6}$$

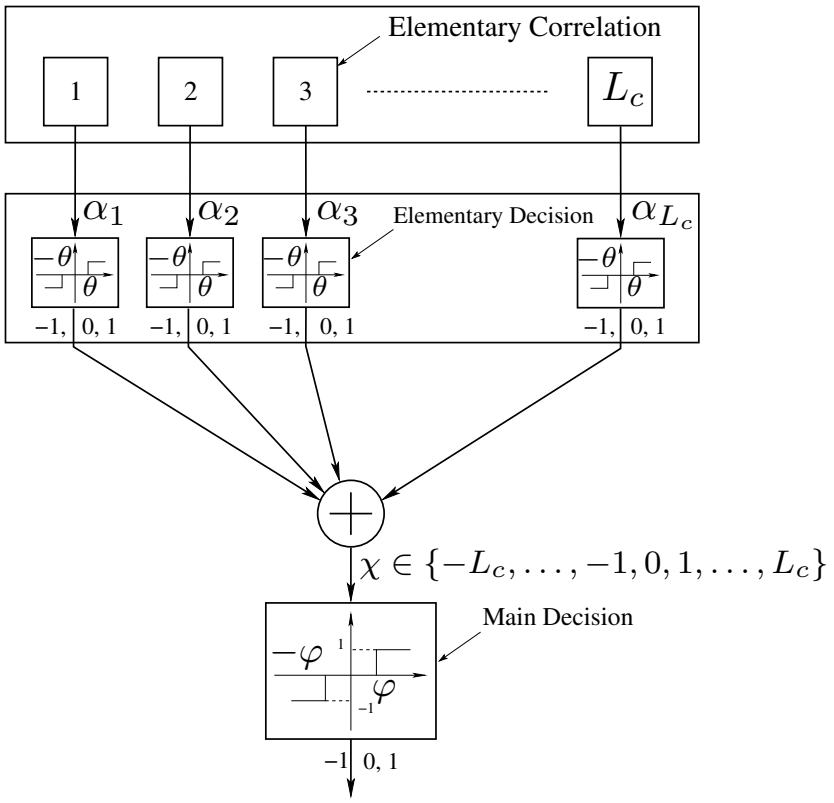


Figure 7.3: PID method: each pulse is detected based on an elementary decision block. The final detection decision is based on the number of pulses detected.

Thus, χ is an integer that gives the number of detected pulses, unlike β in Fig. 7.1 which is a real number indicating the gathered energy. If the absolute value of χ is larger than the main threshold φ, the output of the main decision block will be 1 or -1 (detected path is with negative polarity) and thus a match will be declared between the IR signal and the TPT. In the opposite case the output of the main decision block will be 0. φ should be a positive integer less than L_c, unlike γ in Fig. 7.1 which can be a real number indicating the minimum of the required energy for detection.

It is intuitively clear that this new method should solve the problem described in section 7.3.2; it is designed for an environment without power control because it is sensitive to the existence of a pulse not to its power (assuming it has enough energy to be detected). So we call our proposal "Power-Independent Detection".

7.5 Performance Evaluation Method

We evaluate the performance of PID and compare it to the conventional detection method.

7.5.1 How to Evaluate the Performance

For a meaningful performance evaluation of the conventional detection and the PID methods, we embedded them in a complete synchronization method, which consists of an identification phase, followed by a verification phase. Each phase uses the two aforementioned ingredients of detection and search algorithm iteratively. For the latter, we adopted a serial search. This is because we aim to evaluate the performance of the PID method independently of the effect of optimizations that use coarse synchronization.

The Complete Synchronization Method

When the complete synchronization method uses PID, we call it "PID synchronization method"; when it uses conventional detection, we call it "conventional synchronization method".

Let N be the number of the search bins [1]; let "true sequence" be the sequence to be detected in the received IR signal; it has the same pattern as the TPT at the receiver.

The PID synchronization method consists of two phases. Figs. 7.4 and 7.5 illustrate the flowcharts of the first and the second phases respectively.

In the first phase, the procedure in Fig. 7.3 without the main decision block, i.e. block D, is repeated for the N search bins according to the serial search algorithm; we start with bin 1, then bin 2 up until bin N. The largest χ, χ_{max}, is memorized, as well as its corresponding search bin. Then χ_{max} is compared to a first mean threshold, $\varphi^{(1)}$. If the absolute value of χ_{max} is strictly above $\varphi^{(1)}$, the bin that corresponds to χ_{max} is considered as a signal bin[2], SB, and we move to the second phase. Otherwise, the procedure of the first phase starts anew.

[1] In all conventional synchronization methods, the sequence is divided into N search bins. The bin width is equal to a small fraction of the pulse width. If σ is the bin width, we have $N = L_c \times N_c \times T_c/\sigma \; bins$. The TPT shift offset is a multiple of the bin width and it determines which bin to be searched, i.e to which bin the TPT is shifted. In another word, the bin width gives the shift resolution. After each shift, L_c elementary correlations are done, each one over the whole pulse width T_c and not the bin width (see (7.4)).

[2] We refer by a signal bin to the bin that corresponds to a match between the TPT and the received IR signal.

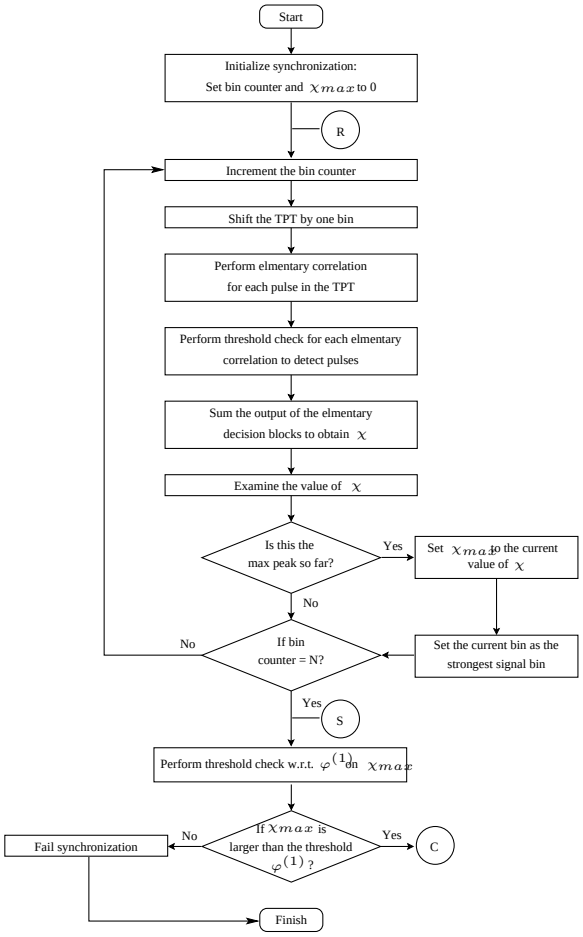


Figure 7.4: The first phase of the PID synchronization method that finds the signal bin, SB, which corresponds to the highest value of χ above the first main threshold $\varphi^{(1)}$.

In the second phase we aim to verify the detection of the first phase. It consists of A iterations, where in each one the procedure is the same as in the first phase. But, in the second phase, it is carried out on a predefined neighborhood of SB, V, including SB, instead of the whole N bins, and with a second mean threshold, $\varphi^{(2)}$, that is larger than $\varphi^{(1)}$. If at least B threshold checks among A succeed, the detection is confirmed, otherwise the detection of the first phase is canceled and the procedure of the first phase starts anew.

The conventional synchronization method is similar to the PID synchronization method with the difference that it does not perform a threshold check on the elementary correlation outputs.

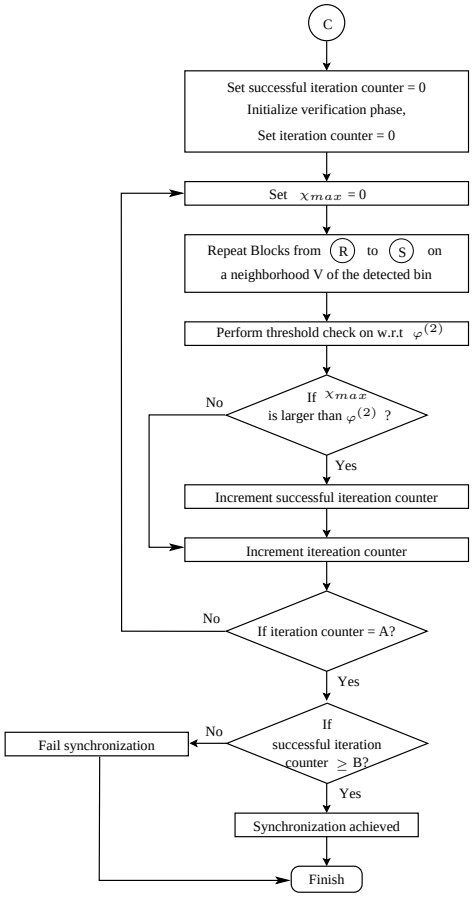


Figure 7.5: The second phase of the PID synchronization method that verifies the detection declared in the first phase. It consists of A iterations similar to the first phase, but it is applied on a predefined neighborhood of SB and with a higher second main threshold $\varphi^{(2)}$.

Performance Metrics

We measure the performance of each procedure by the following metrics, applied to the synchronization method: (1) the probability of Missed Detection (P_{MD}) in the presence of the true sequence in the received IR signal (2) the probability of False Alarm, P_{FA0}, in the absence of the true sequence in the received IR signal and (3) the total error defined as $E_t = P_{MD} + P_{FA0}$. Note that P_{MD} includes both errors that can occur in the presence of the true sequence: the probability of false alarm and the probability of no detection.

7.5.2 Computation of Metric Using Hybrid Method: Analysis + Simulation

Analysis

The goal of the analysis is to express the metrics as functions of other probabilities that we obtain by simulation. The probabilities are as follows: During the first phase we have P_1, the probability of good detection when the received IR signal contains the true sequence; and P_2, the probability of a bad detection when the received IR signal does not contain the true sequence. During the second phase we define P_3 as the probability that a threshold check succeeds, given that the first phase has resulted in a good detection in the presence of the true sequence and P_4 as the probability that a threshold check succeeds, given that the first phase has resulted in a bad detection in the absence of the true sequence.

The Analysis presented in Appendix A.1 gives:

$$P_{MD} = 1 - P_1 \sum_{i=B}^{A} \binom{A}{i} P_3^i \left(1 - P_3\right)^{(A-i)} \tag{7.7}$$

$$P_{FA0} = P_2 \sum_{i=B}^{A} \binom{A}{i} P_4^i \left(1 - P_4\right)^{(A-i)} \tag{7.8}$$

Simulation

In order to compute the metrics, we ran extensive series of simulations to estimate the probabilities P_i, $i = 1, \ldots, 4$. The simulations were carried out using Matlab. We tried to make the simulated scenario as realistic as possible by choosing a real multipath fading channel model and by adjusting all simulation parameters, e.g. the bit energy to noise spectral density ratio E_0/N_0 (one bit corresponds to one pulse), the physical layer parameters, the transmission power levels, and the number of users.

Channel Model: In our simulations, we consider the indoor office environment defined by the IEEE P802.15.4a study group [14] and we studied the LOS and the NLOS cases. We adopted the LOS channel model as it is proposed in [14]. As to the NLOS, only the pathloss and small scale fading parameters are available in [14] and not the delay profile parameters.

As the available parameters are extracted from [45], we also filled the missing parameters from the same work in [45] so that the model is totally coherent and compatible.

Although the measurements made for these models were using an UWB IR signal, the models are generalized to be used by any carrier modulation system. Thus, the phase of a multipath component is considered as uniformly distributed over [0,2π] which is meaningless in an UWB IR baseband transmission. We solve this problem by relaxing this hypothesis and replacing it by the one adopted in [13], which is appropriate for IR baseband transmission. Then, the phase of a multipath component will be $0/\pi$ with an equal probability for representing pulse inversion due to the reflection from different surfaces.

For simplicity, we assume that the distribution of the small scale fading is Rayleigh instead of Nakagami since the mean value in dB of the "m" parameter of the Nakagami distribution in the adopted model is very close to zero, which corresponds to the particular case of the Rayleigh distribution.

Simulation Parameters: We consider that all users are sending non-modulated IR signals, an assumption that does not affect our results because the interferer signals are already random with respect to the receiver and using data modulation will add one more random variable with zero mean. We have $T_c = 0.2ns$. N_c is set, in the LOS case, to 200 chips/frame that corresponds to $40ns$, which is sufficient to minimize the inter-symbol interference due to the multipath delay spread [37, 45]. In the NLOS case, N_c is set to 400 chips/frame because the multipath delay spread is larger [37, 45]. Further, the cardinality of the code is set to 100, i.e. a source can place a pulse in only the first 100 chips of a frame. The sampling frequency is 50 GHz, much larger than the Nyquist sampling frequency, to simulate an analog receiver, as the impact of the sampling frequency is out of the scope of this study. The elementary threshold (θ in Fig. 7.3) is set to $0.5 \times \alpha_0^{(1)}$ (see 7.4) with $\mathscr{E}_r^{(1)}$ corresponds to the highest multipath components.

Each simulated scenario contains several transmitters, we refer to them as users: the one that is transmitting the true sequence is called the source and others are the interferers. In our simulations, we consider a rather pessimistic scenario where all interferers have at least the same transmit power as the source. The source signal power observed by the receiver is set to $-30dBm$, whereas the interferer signal powers observed by the receiver are uniformly

distributed over [-30dBm, -10dBm]. Then the largest value of signal power that an interferer can have is 20 dB larger than the source signal power. Indeed, according to the pathloss model used in [14], this difference in power corresponds to a communication range of 17 m approximately in the LOS case and to 4.5 m in the NLOS case (see Appendix A.2) where all users are transmitting at the same power and the source is the farthest. Such a communication range is typical for an indoor environment and the adopted channel model of [14] is still valid as it is based on measurements that cover a range from 3 m to 28 m.

As we assume a stationary channel during the synchronization phase, we consider a small neighborhood V of 2 pulses width ($V = 2T_c = 0.4ns$).

In all simulated scenarios, E_0/N_0, which is the bit energy to noise spectral density ratio, is computed with respect to the source signal power. In our simulations, we consider that one pulse corresponds to one bit, i.e. a transmitter sends one pulse per bit.

7.6 Performance Evaluation Results

In this section, we study the behavior of the PID synchronization method according to $\varphi^{(1)}$ and $\varphi^{(2)}$ and define an optimal working point. The behavior study of the conventional synchronization method is omitted because it is similar and can be deducted by analogy. Next, we compare the PID synchronization method with the conventional one. In the end, we evaluate the special case of concurrent transmissions with the same code.

The probabilities P_i, $i = 1, \ldots, 4$, are obtained by simulation. P_1 and P_2 are computed by averaging the results of 200 independent runs for each simulated scenario. A different independent noise realization is computed per run and, within the same run, a different channel realization is computed per user.

To compute P_3 and P_4, the stationarity of the channel during the synchronization should be taken into account. Thus, the computation of P_3 and P_4 is different and more complicated. We proceed as follows: for each run of the 200 runs above, if a detection is declared, 9 other runs are done with the same channel realization for each user but with different noise realization. Then, for a given scenario, if all the 200 runs above result in a detection declaration, we will have 9 additional runs for each run and thus 1800 runs for this scenario.

7.6.1 The PID Synchronization Method

We run simulations for E_0/N_0 values between 0 dB and 20 dB, L_c values between 8 and 30, and number of users between 5 and 20 users. In the extreme scenarios with low E_0/N_0 ($< 10dB$), short L_c (< 16) and large number of users (20 users) the performance is not so good due to a huge amount of interference and noise. But, starting from $E_0/N_0 = 10dB$ and $L_c = 16$, the performance is acceptable and the results seem to be similar. For lack of space, we show only one scenario in order to explain the behavior of the PID synchronization method and to show the optimal working point.

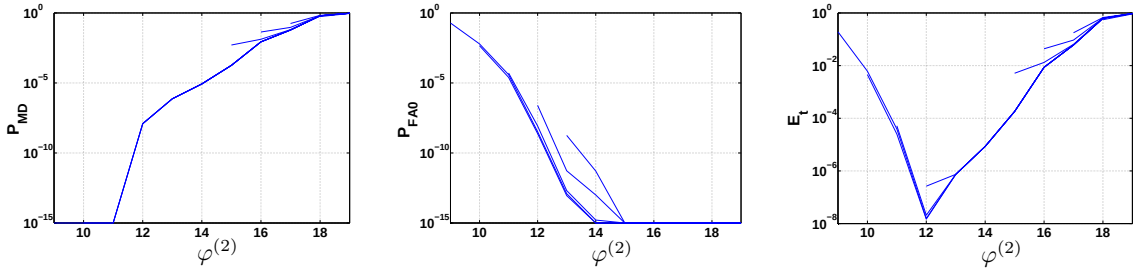
Fig. 7.6 (a), (b) and (c) show the metrics P_{MD}, P_{FA0} and E_t in the LOS case (see the legend for details). For Fig. 7.6 (a), the interpretation is as follows:

For a given $\varphi^{(1)}$, P_1 does not change with $\varphi^{(2)}$ because it is independent of the second phase. As for P_3, it is obvious that it is a decreasing function of the threshold $\varphi^{(2)}$ because, when this latter gets high, it becomes more difficult to succeed the main threshold check in the second phase. According to (7.7), P_{MD} is decreasing with increasing P_3. Consequently, P_{MD} is an increasing function of $\varphi^{(2)}$.

For a given $\varphi^{(2)}$, on one hand, P_1 is a decreasing function of the threshold $\varphi^{(1)}$ because, when this latter gets high, it becomes more difficult to succeed the main threshold check in the first phase. On the other hand, P_3 is an increasing function of $\varphi^{(1)}$. Indeed, P_3 is a conditional probability that χ_{max} in an iteration of the second phase is above $\varphi^{(2)}$ given that χ_{max} in the first phase is above $\varphi^{(1)}$. Thus the smaller the difference between $\varphi^{(1)}$ and $\varphi^{(2)}$, the smaller P_3. This difference decreases when $\varphi^{(1)}$ increases for a given $\varphi^{(2)}$. Consequently, P_3 increases with $\varphi^{(1)}$. According to (7.7), P_{MD} is a decreasing function of both P_1 and P_3. Therefore, given $\varphi^{(2)}$, it is not obvious how P_{MD} varies according to $\varphi^{(1)}$ because P_1 and P_3 vary in opposite directions. Moreover the values of A and B influence the effect of the variation of P_3. For $\varphi^{(2)}$ = 18, P_{MD} increases with $\varphi^{(1)}$ when $\varphi^{(1)}$ goes from 12 to 17, but it decreases when $\varphi^{(1)}$ passes from 17 to 18.

Fig. 7.6 (b) shows the probability P_{FA0}. To understand the trends of the curves, a similar interpretation can be made as above. For instance, for a given $\varphi^{(1)}$, P_2 is independent of $\varphi^{(2)}$ and P_4 is a decreasing function of $\varphi^{(2)}$. Thus, P_{FA0} decreases with $\varphi^{(2)}$ for a fixed $\varphi^{(1)}$. In contrast, for a fixed $\varphi^{(2)}$, P_2 is decreasing with $\varphi^{(1)}$ whereas P_4 is increasing.

Fig. 7.6 (c) shows E_t. The optimal working point for this scenario is for ($\varphi^{(1)}$;$\varphi^{(2)}$) =



(a) (b) (c)

Figure 7.6: Performance of PID in the LOS case for various values of the two main thresholds $\varphi^{(1)}$ and $\varphi^{(2)}$ defined in Section 7.5.1; $\varphi^{(2)}$ is on the x-axis. Each figure shows several curves that correspond to several values of $\varphi^{(1)}$. The values of $\varphi^{(1)}$ for a given curve is the x-value of its leftmost point. To understand these figures, consider the curve whole leftmost point has a x-value equal to 15 in (a). It corresponds to $\varphi^{(1)}$ = 15. As $\varphi^{(2)}$ is always larger than $\varphi^{(1)}$, this curve can not have points with x-values less than 15, which explains the different x-values of the leftmost point of these curves. The y-axis shows: (a) P_{MD} (the probability of Missed Detection), (b) P_{FA0} (the probability of False Alarm) and (c) $E_t = P_{MD} + P_{FA0}$ (the total error). E_0/N_0 = 15 dB, L_c = 20, 10 users, A = 10 and B = 7. In (a), the curves that correspond to $\varphi^{(1)}$ ranging from 9 to 14 coincide. Thus, we see only one curve instead of five. Note that we set to 10^{-15} all values that are below 10^{-15} because Matlab was not able to plot them.

(10;12) where E_t is minimized. On the left hand of the optimal working point, P_{FA0} is dominant and the curves imitate those of P_{FA0} in Fig. 7.6 (b). In contrast, P_{MD} becomes dominant on the right hand of the optimal working point and the curves at this side are similar to those of P_{MD} in Fig. 7.6 (a).

In the NLOS case, the curves have similar trends and we omit them.

In conclusion, using the PID synchronization method, an optimal working point can be obtained by minimizing E_t. For this specific example, the optimal working point is $\varphi^{(1)} = 10$, $\varphi^{(2)} = 12$.

7.6.2 The PID Method vs. the Conventional Method

LOS Case

Now, we compare the PID synchronization method with the conventional one in the LOS case. The NLOS case is left for the next subsection. The results that we show correspond to the optimal working point defined in 7.6.1.

Fig. 7.7 (a) shows P_{MD} according to E_0/N_0 for both synchronization methods. Corresponding values of E_t are shown in Fig. 7.7 (b). Recall that the upper bound of E_t is 2 (see 7.5.1). The simulated scenario corresponds to 10 users, $L_c = 20$ and $N_c = 200$ chips/frames. As for the parameters A and B, they concern the verification phase, which aims at eliminating any false alarm due to random interference or noise. Thus, B can be seen as a threshold. If the number of iterations that result in a detection exceeds B, we consider that this is due to a good detection and not to a random effect. Hence, to ensure good performance, A and B should be well tuned. We tried several values for them and we had good performances when $A = 10$ and $B = 7$. We keep these values for all the following results. As we can notice, The PID synchronization method P_{MD} is decreasing with increasing E_0/N_0 and it is very small when E_0/N_0 becomes larger than 10 dB. In contrast, with the conventional synchronization method, P_{MD} is very high and it is close to 1 even when $E_0/N_0 = 20\,dB$. E_t in Fig. 7.7 (b) is a decreasing function of E_0/N_0, it reaches 10^{-8} for $E_0/N_0 = 15\,dB$ with the PID synchronization method, whereas it is very close to 2 with the conventional one.

NLOS Case

It is obvious that the NLOS case is more challenging as it has larger delay spread and its cluster and ray arrival rates are much higher [45]. Thus, to have an acceptable performance we reduced the number of users to 5, instead of 10 with the LOS case. Also, we used a larger $N_c = 400$ chips/frame in order to compensate the larger delay spread. Fig. 7.8 (a) and (b) show comparison results for P_{MD} and E_t respectively. The performance is not as good as in the LOS case, but our method still performs much better than the conventional one. With the PID synchronization method, E_t is around 10^{-2}, whereas it is very close to 2 with the conventional one.

To summarize this section, we have shown that, with the PID synchronization method, the

synchronization is achieved in the presence of the IUI with a minimal total error. In contrast, the total error is very close to 2 with the conventional synchronization method, which means a certain failure.

7.6.3 Concurrent Transmissions with the Same Code

It is often thought that concurrent transmissions with the same code result always in collision. This is not true. Let us consider first asynchronous concurrent transmissions. In this case, the different transmitted signals[3] with the same code have different arrival times at a given receiver. As the pulse width is very short and the transmissions are asynchronized, not all the multipath components of the transmitted signals overlap at the receiver. Thus, the receiver can solve, with high probability, at least one multipath component that arbitrarily belongs to one of the transmitted signals. We assume that we are not in a very dense multipath environment where even transmitting with different codes results in collision. Now let us look at the extreme case where the different transmissions are synchronized and all the transmitted signals have almost the same arrival time, i.e. the first multipath components of all the transmitted signals overlap at the receiver, which happens with a negligible probability. As each transmitted signal has a different channel impulse response, not all multipath components of the transmitted signals overlap. Therefore, the receiver can, with high probability, solve one multipath component belonging to an arbitrary transmitted signal. In the following, we show results for the asynchronous case because the other case happens very rarely and its results can be deducted from what we show here. In the simulated scenarios, E_0/N_0, L_c, $\varphi^{(1)}$ and $\varphi^{(2)}$ correspond to the optimal working point of Fig. 7.6. All the transmitted signals in a given run have the same code but with different arrival times and different powers chosen randomly in [-30 , -10] dBm. E_0/N_0 is computed according to a -30 dBm power signal. We did two evaluations using two different methods. The first is what we used before. The second is similar to the first but it takes into account the collisions. In the following, only P_{MD} is shown. P_{FA0} cannot be computed because the true sequence is always present in the received signal, which is the superposition of all the transmitted signals and it is expressed in (6.3).

[3]Recall that a transmitted signal is that expressed in (6.1)

First Evaluation

When we applied the evaluation method used before (Section 7.5) for a number of users up to 10 in the LOS case and to 5 in the NLOS case, P_{MD} was exactly equal to zero with the PID and the conventional complete methods. This is not surprising, because all the users transmit the true sequence. Furthermore, the transmission power levels are now higher than the extreme case of Fig. 7.7 and Fig. 7.8, which even give too small $P_{MD}s$. This result does not take into account the collisions as it is explained next section.

Second Evaluation

When all the transmitted signals have the same code, the collisions become more harmful than when only one transmitted signal carries the true sequence. Indeed, as all the transmitted signals carry the same code, different multipath components of different transmitted signals may overlap in all their pulses, which do not occur in the case of different codes. Therefore, we changed p_1 to be the probability of (1) a good detection during the first phase when the received signal contains the true sequence and (2) the detected signal does not collide with another. In other words, when we detect some signal and we find that it overlaps with another, we consider that this is an FA and the detection is canceled.

Fig. 7.9 (a) and (b) show P_{MD} according to the number of users in the LOS and the NLOS cases respectively. Due to the collisions, P_{MD} is now around 10^{-1} (instead of 0 in the first evaluation). Thus, a good detection without collision is obtained with a probability of 90%. It is clear that our method performs better than the conventional one in the NLOS case, which is more challenging. This can be explained by its immunity to the constructive interference. Indeed, when two multipath components belonging to different signals have almost the same timing and thus overlap almost completely, if they have the same polarity, the resultant signal has a higher amplitude that is the sum of the two multipath component amplitudes. Consequently, the probability of detecting the resultant signal, which results in a collision, increases with the conventional method as it has a higher power. With our method, this probability does not change because it is power independent. We observed in our simulations that, with the PID synchronization method, all transmitted signals have the same chance to be detected independently of their power levels, which is another consequence of power independence. Also, we observed that the conventional synchronization method always detects the transmitted signal

of the highest power level.

To summarize this section, concurrent transmissions with the same code do not result always in collision. Thus, the channel employing a common code, such as in [19], cannot be modeled as an Aloha channel. This modeling considers that concurrent transmissions result always in collision and, hence, it does not hold anymore.

7.7 Conclusions

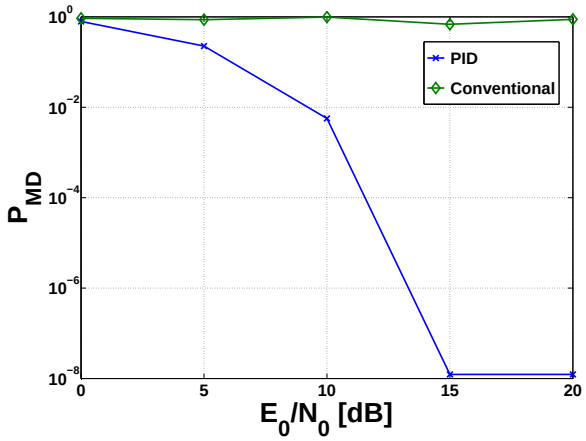
Our paper addresses non-coherent synchronization (signal acquisition) in the presence of asynchronous concurrent transmissions with heterogeneous power levels. This occurs, for example, in the presence of multiple interfering piconets, or in purely ad-hoc networks. This is the first work that identifies the problem that arises when using the conventional detection method, which correlates the received UWB Impulse Radio (IR) signal with a Template Pulse Train (TPT) and performs a threshold check on the output of the correlation. We show that the synchronization is either unfeasible or entails an extremely large overhead due to the Inter-User Interference (IUI) in these scenarios. In order to solve the extreme IUI case (near-far problem), we propose a new detection method, which we call Power Independent Detection (PID) method; it splits the correlation into elementary correlations. Each one corresponds to one pulse in the TPT. Then, two threshold checks are performed. The first is to detect pulses, whereas the second is to detect the signal based on the number of detected pulses. Our PID method solves the problem without any additional complexity overhead, e.g. for a digital receiver, it employs the same sampling frequency and number of operations as the conventional detection method.

We evaluated the performance of the PID detection method based on analysis and simulations. The simulations were carried out according to the Line Of Site (LOS) and the Non-LOS (NLOS) indoor office channel models proposed by the IEEE P802.15.4a study group [14]. The adopted metrics are (1) the probability of Missed Detection (P_{MD}) when the received IR signal contains the synchronization sequence to be detected (2) the probability of false alarm (P_{FA0}) when the received IR signal does not contain the synchronization sequence to be detected and (3) the total error defined as $E_t = P_{MD} + P_{FA0}$. The results presented in this paper show a significant improvement compared to the conventional detection method. Moreover, we define

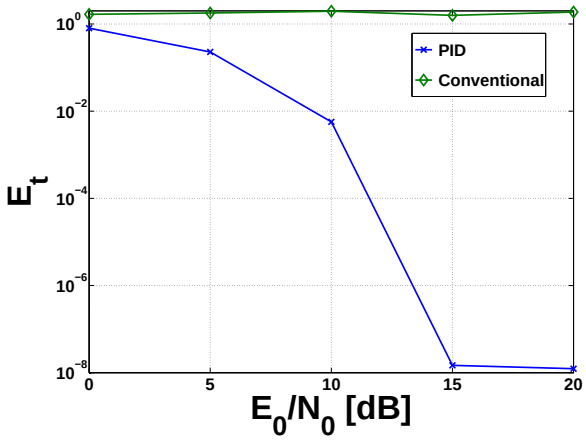
an optimal working point that corresponds to the least total error E_t. Some of the results are that, for $E_0/N_0 = 15dB$ and in the presence of 10 users transmitting simultaneously in the LOS case, $E_t = 10^{-8}$ with the PID method at the optimal working point whereas E_t is very close to 2 with the conventional detection method.

We also investigate the particular case where all concurrent transmissions have the same code: this is the case of broadcast or control channel in ad hoc networks and it may occur even in the presence of multiple interfering piconets. Our results show that, with high probability, no collision occurs and we are still able to detect one of the transmitted signals. Thus, the Aloha model does not hold anymore for these kinds of channels. Further, with the conventional method, the user with the highest power is most likely to be the one that is detected, whereas with our method, all users within the detectability range have the same probability of being detected. Moreover, we show that the immunity of the PID synchronization method to the constructive interference makes its performance better than the conventional one in the presence of collision.

An extension of this work is to investigate how to determine the thresholds θ, $\varphi^{(1)}$ and $\varphi^{(2)}$ for these environments of concurrent transmissions with heterogeneous power levels.

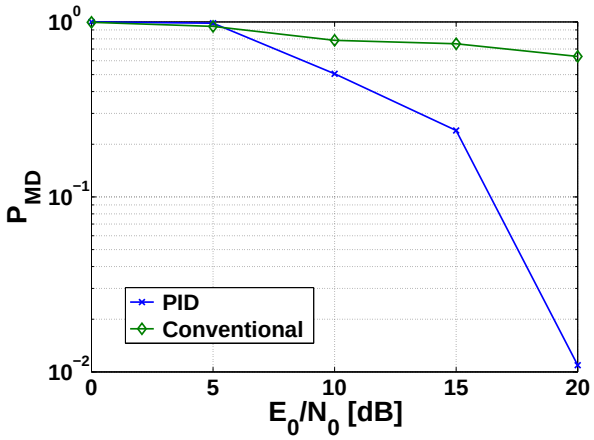


(a)

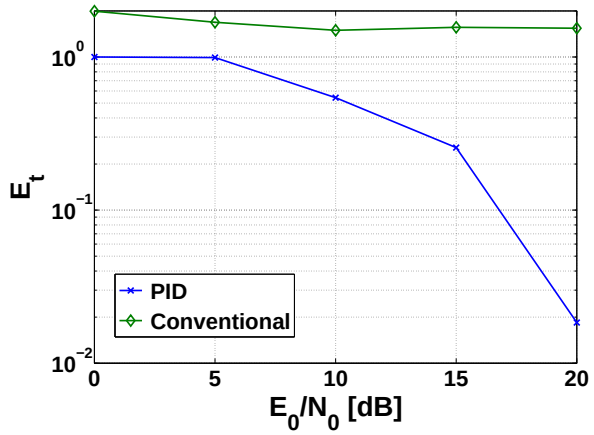


(b)

Figure 7.7: A comparison between the PID and the conventional synchronization methods at the optimal working points in the LOS case. (a) P_{MD} (the probability of misdetection) and (b) $E_t = P_{MD} + P_{FA0}$ (the total error). N_c = 200 chips/frame, L_c = 20, 10 users, A = 10 and B = 7.

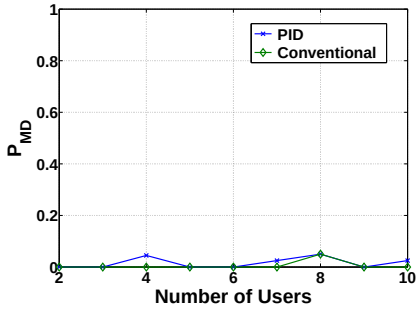


(a)

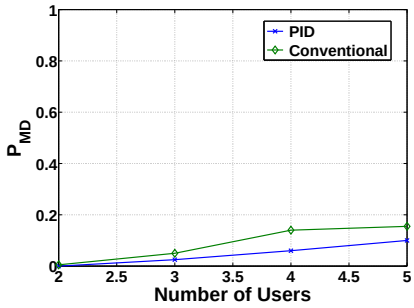


(b)

Figure 7.8: A comparison between the PID and the conventional synchronization methods at the optimal working points in the NLOS case. (a) P_{MD} (the probability of Missed Detection) and (b) $E_t = P_{MD} + P_{FA0}$ (the total error). N_c = 400 chips/frame, L_c = 20, 5 users, A = 10 and B = 7.



(a)



(b)

Figure 7.9: P_{MD} in the case of concurrent transmissions with the same code in both cases the LOS (a) with N_c = 200 chips/frame and the NLOS (b) with N_c = 400 chips/frame. $E_0/N_0 = 15dB$, L_c = 20, A = 10 and B = 7.

Chapter 8

Sleeping Protocols

8.1 Introduction

Emerging pervasive networks assume the deployment of large number of wireless nodes, embedded in everyday life objects. In these types of networks, the focus is more on minimizing energy consumption than maximizing rate. As these networks are characterized by occasional Low Data Rate (LDR) communication, letting nodes sleep is the most effective way to conserve energy and thus maximizing the lifetime. Existing sleeping protocols are dedicated to narrow-band systems and they perform poorly if applied with UWB-IR systems. An optimal sleeping protocol should take into account several design elements, some of them are specific to this kind of signaling, such as the possibility of transmitting concurrently without collision and the power consumption model of the hardware behind which is completely different than with the narrow-band signaling.

8.2 Design Elements

Several elements should be considered when designing a sleeping protocol. Some of them are specific to the kind of signaling used and others are generally related to the communication patterns and rate constraints.

8.2.1 Concurrent Transmissions

Concurrent transmissions using different TH codes do not result in collision as long as the MUI is affordable. Hence, the received signal is still decodable. Furthermore, the study done in [62] shows that the optimal design choice is to avoid mutual exclusion and to allow for concurrent transmissions if interference mitigation is applied. Thus, a sleeping protocol for UWB-IR should take this into account and permit to nodes transmit simultaneously.

8.2.2 Multi-Access to the Same Destination

First, we assume that each node is identified by a unique TH code on which it is listening. A sender uses the receiver TH code to start a communication with it. In this section, we consider accessing the same receiver simultaneously by several senders. In other word, we consider the simultaneous presence of different signals with the same TH code at the receiver. In Chap. 7, we explain and show that this does not result in collision and the receiver acquires one of the received signals independently of its power.

8.2.3 Hardware Power Consumption

First, with narrow-band systems, generating the sinusoidal carrier is very energy consuming. In contrast, transmitting pulses consumes much less energy with UWB-IR systems. Second, due to the carrier sensing with narrow-band systems, the signal acquisition phase starts only after detecting the carrier, that is the output of the pass-band filter shows a sufficient level of energy. In contrast, due to the absence of a carrier with UWB-IR, a receiver is always in the signal acquisition phase when it is listening to the channel, even in the absence of any signal. The signal acquisition phase is very energy consuming as it consists of correlating the received signal with the TPT and searching all possible combinations. Therefore, a sleeping protocol for UWB-IR should minimize the period of listening to the channel, even if it was at the expense of transmitting.

8.2.4 Slotted versus Unslotted

If a global synchronization comes for free, as in a centralized network where it is maintained by the coordinator, a slotted sleeping protocol might be advantageous. In contrast, in a fully-decentralized network, where global synchronization might be costly, unslotted sleeping is an alternative.

8.2.5 Sleeping Cycle - Traffic Load Trade-Off

There exists an important trade-off between long sleep cycles that permit efficient energy savings and short cycles that facilitate communication and improve responsiveness.

8.3 Sleeping Protocol Designs

Letting nodes sleep from time to time is the most effective way to conserve energy in a wireless network and thus maximize the lifetime. However, this requires a mechanism that allows nodes to be contacted when they are awake. In the following, we propose two sleeping protocols for slotted and unslotted networks, respectively.

8.3.1 Slotted

The slotted sleeping uses a periodic beacon sent by the coordinator. This beacon provides a coarse-level synchronization and denotes the start of a superframe. As depicted in Fig. 8.1, a superframe has two parts, a reservation window and a data transmission window. Both windows consist of S_A slots. Transmission requests are carried out by sending an RTS in a reservation slot on the TH code of the receiver (hence concurrent reservations for different receivers are possible). The receiver replies with a CTS if it accepts the reservation. If a reservation is successful, the actual data transmission occurs in the corresponding data slot and is followed by an ACK. In order to minimize the MUI, the sender randomly chooses the reservation slot for its RTS. If the transmission request fails, the sender retries in another random slot until a successful reservation occurs. During the reservation window, nodes are either transmitting requests or waiting for requests. In the same reservation slot, several senders

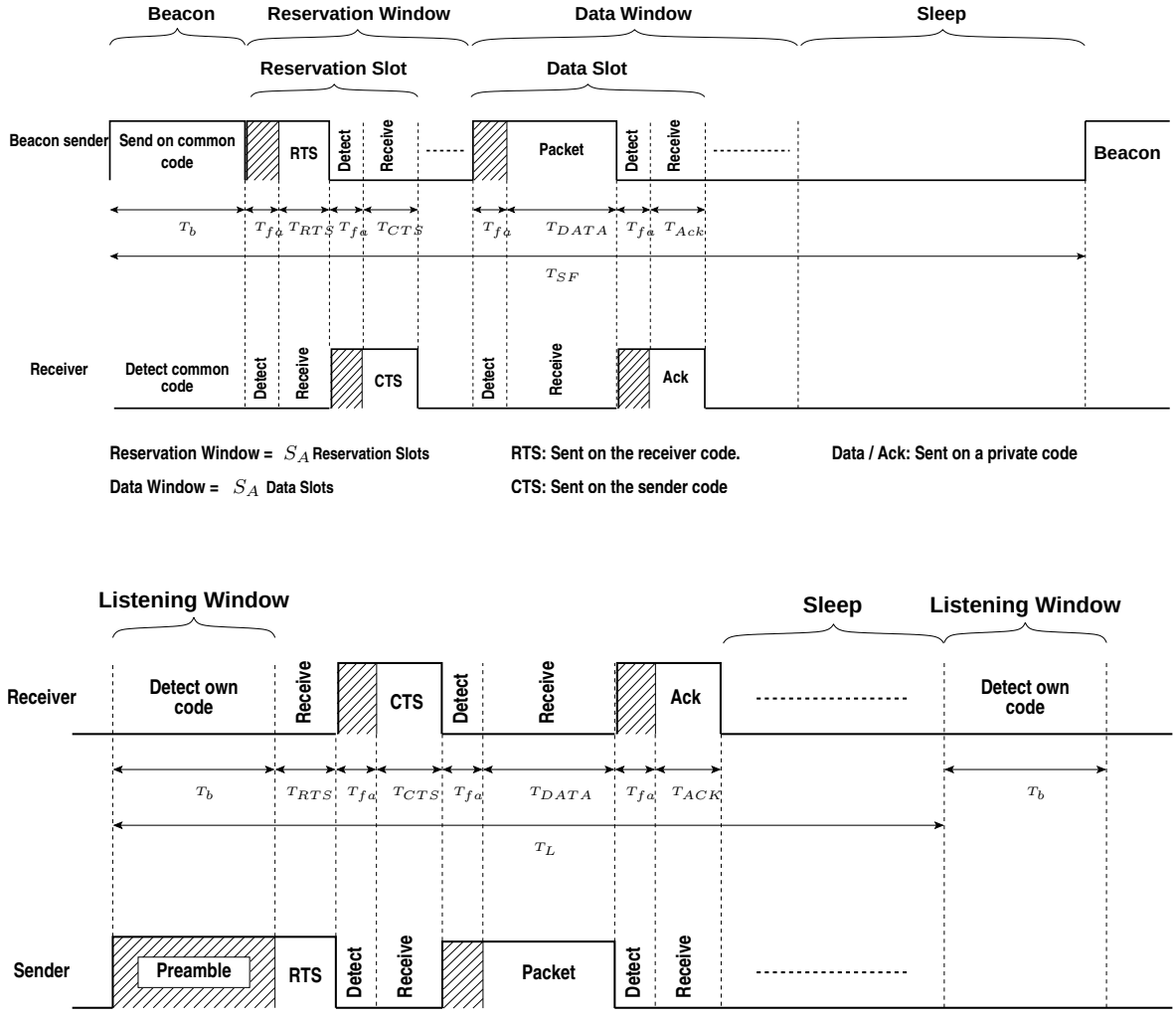


Figure 8.1: The *slotted sleeping protocol* is depicted on the top; T_b is the length of the superframe beacon necessary to achieve coarse acquisition. Afterward, there is only a short preamble of length T_{fa} before every packet. The *unslotted sleeping protocol* is depicted on the bottom; T_L is the time interval between two listening windows.

may send RTSs to the same receiver; if the receiver is waiting for a request, it is able to detect one of these RTS and answer the detected sender by a CTS.

Unlike the RTS/CTS exchange used with 802.11 MAC layer as a mutual exclusion mechanism, this exchange is now merely used for data request notification, as concurrent transmissions are allowed.

During the data window, nodes that are neither sending nor receiving switch to sleep mode. Thanks to the global synchronization ensured by the beacon, only a short synchronization preamble is needed at the beginning of each transmission in order to achieve fine synchronization between the sender and the receiver.

8.3.2 Unslotted

In this case, each receiver wakes up according to its own listening schedule (see Fig. 8.1. A transmitter that wants to communicate with a given receiver first needs to learn its listening schedule. Typically, if all nodes have the same sleeping scheme (but are delayed in time), a transmitter simply has to send a long preamble, as long as the maximum sleeping time is followed by an RTS. The destination, sure to wake up at some time in between, will receive the preamble and answer to the transmitter by a CTS. The data transmission takes place immediately afterwards. Similarly to the slotted case, the RTS is sent on the receiver TH code, the CTS on the sender TH code and the data on a private TH code.

8.4 Performance Evaluation

8.4.1 Energy Consumption Model

Our goal is to define an energy model that can be applied early in the design process, *before* an actual hardware is developed and can be instrumented. This is a serious challenge, but we can take advantage of the nature of IR-UWB to derive a generic model, which is flexible enough to account for a large set of options.

With IR-UWB, time is divided into frames of N_c short duration chips[1]. We use this to define a *chip-level* model of energy consumption. During a chip, the physical layer can either

[1]Only one pulse is transmitted per frame

transmit a pulse, receive a pulse, perform signal acquisition, be in an active-off state, or sleep. The active-off state occurs due to time-hopping. When a node is between two pulse transmissions or receptions, energy is consumed only to keep the circuit powered up, but no energy is used for transmitting or receiving pulses.

Hence, we model the energy consumption by considering the energy *per chip* for each state. An energy consumption model is defined by the vector

$$\vec{q} = [q_{tx}\ q_{rx}\ q_{ao}]$$

where q_{tx} is the cost for transmitting a pulse, q_{rx} receiving a pulse and q_{ao} for being in the active-off state. As the same transceiver elements are used for signal acquisition and reception, the acquisition energy consumption is also equal to q_{rx}. The cost while sleeping is negligible. It is currently impossible to give precise figures for $\vec{q}$, but only relative values are relevant to our performance evaluation. It is thus possible to limit our analysis to a small set of scenarios, as shown on the top of Table 8.1.

We now show on an example of how our energy model is used. The energy consumption E_{packet} to receive a packet of 127 bytes (including a synchronization preamble of 20 bytes) using binary modulation (one pulse carries one bit) is

$$E_{packet} = 8 \cdot \left(\underbrace{20 \cdot N_c \cdot q_{rx}}_{\text{Energy for the preamble acquisition}} + \underbrace{107 \cdot q_{rx}}_{\text{Energy when a pulse is present}} + \underbrace{107 \cdot (N_c - 1) \cdot q_{ao}}_{\text{Energy in the active-off state}} \right)$$

where the factor eight appears as we consider bytes. With this model, the energy consumed for each received or transmitted packet can be easily computed. The lifetime of a node is then the time necessary to consume all the energy contained in the battery of the node.

8.4.2 Performance Metric and Parameter Setting

Our evaluation metric is the node life time, which reflects the network life time in an homogeneous topology. The remaining assumptions about the physical layer parameters for our analysis are given in Table 8.1. Note that the physical layer supports several transmission rates (from 100kbit/s to 1Mbit/s). We assume that all nodes have an identical physical layer and the same initial battery power.

Energy consumption models $\vec{q} = [q_{tx}\ q_{rx}\ q_{ao}]$	1	$\vec{q} = [1\ 1\ 1]$	Baseline model
	2	$\vec{q} = [1\ 5\ 1]$	Higher cost for reception
	3	$\vec{q} = [1\ 1\ 0.5]$	Lower cost for active-off
	4	$\vec{q} = [1\ 5\ 0.5]$	Higher cost for reception, lower cost for active-off
Physical layer parameters	Frame length $N_c = 1000$ chips Chip duration $T_c = 1$ ns		
Sleeping protocols parameters[a]	$T_b = 50\mu$s, $T_{fa} = 10\mu$s		
	$T_{RTS} = T_{CTS} = T_{ACK} = 800\mu$s		Packet size is 20 bytes
	$T_{DATA} = 10200\mu$s		Packet size is 127 bytes

Table 8.1: Energy consumption model, traffic load model, physical layer parameters and assumptions for the performance analysis

[a]Packet lengths are computed assuming the smallest data rate

8.4.3 Performance Evaluation Results

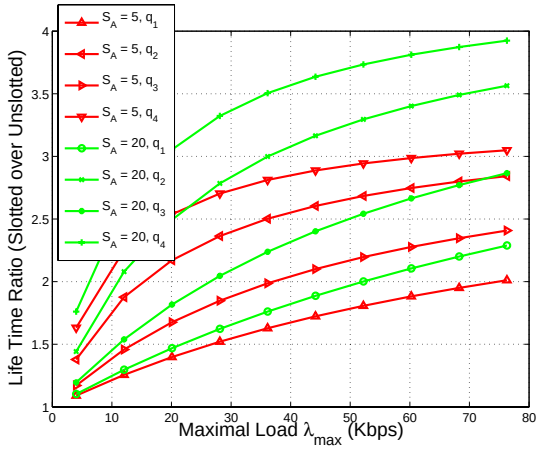
If occasional bursts must be supported, slotted sleeping is better than unslotted

We consider a slotted and an unslotted sleeping protocol (Sect. 8.3) as depicted in Figure 8.1. We analyze which protocol is more efficient with respect to average node lifetime.

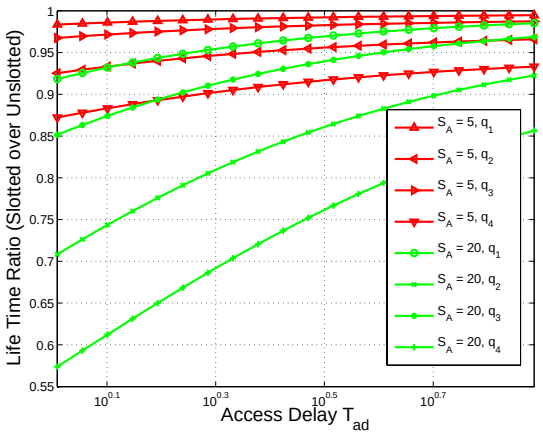
We compute the lifetime, assuming that most of the time the node is subject to a load λ_0. However, the network is designed to occasionally sustain a traffic load $\lambda_{max} > \lambda_0$ per receiver during burst intervals.

Let us define by γ the network utilization. In the slotted case, a receiver can receive $\gamma\frac{S_A}{T_{SF}}$ packets per second where S_A is the number of reservation slots in the reservation window and T_{SF} is the superframe length. In the unslotted case it can receive $\gamma\frac{1}{T_L}$ where T_L is the time interval between two listening windows. One packet at most can be received during T_L. As a network with utilization close to 100% is unstable, we take $\gamma = 0.7$ to guarantee stability. Note that if two requests at the same destination overlap, one is very likely to be accepted due to time hopping and the signal acquisition procedure. Therefore, we assume that the total submitted traffic is close to λ_0 per receiver.

For two extreme values of S_A and the four energy models, we compare the lifetimes achieved with slotted and unslotted protocols. The parameters T_{SF} and T_L are chosen to sustain the bursty maximum load λ_{max}. The lifetime is then computed assuming a load $\lambda_0 = 10$kbit/s. The ratios of the lifetime in the slotted over the unslotted case are plotted on Figure 8.2(a). With slotted sleeping protocols, the lifetime is 15%-50% longer. If the lifetime is around one year, it can be significantly increased by 2 to 6 months. If the slotted structure comes at a low cost, or for free (as in a master-slave system like Bluetooth), its use is optimal. If this is not the case, we need to compare the implementation overheads to compare the two protocols. The main overhead of a slotted protocol is distributing the beacon and managing the cases when communicating nodes hear several different superframes. The main overhead of an unslotted protocol is the learning time when a node learns schedules of neighbors, either due to a topology change or due to a clock drift.



(a) Ratio of the average node lifetime in the slotted case over the unslotted case with respect to the maximal load λ_{max} (q_i stands for energy model i). In all cases, the slotted protocol outperforms the unslotted one by 15%-30%.



(b) Ratio of the average node lifetime in the slotted case over the unslotted case with respect to the access delay T_{ad} (q_i stands for energy model i). In this case, the unslotted protocol outperforms the slotted one.

Figure 8.2: Lifetime comparison for slotted and unslotted sleeping protocols under various traffic constraints. We compare the performance for for S_A equal to 5 and 20 and all energy models (Table 8.1), q_i stands for energy model i. In all cases $\lambda_0 = 10$Kbp/s.

If occasional maximum latency must be supported, unslotted sleeping is better than slotted

We consider a variant of the previous section. We still assume that most of the time, the network is subject to an average traffic load λ_0. However, it has to occasionally support a small number of unpredicted, but very urgent messages instead of a bursty high load.

When a node generates a packet, it cannot send it immediately. For the slotted protocol, a node has to wait at most T_{SF} to send a packet. For the unslotted one, the worst case delay is T_L. In both cases, we assume that the worst case is limited by application constraints to T_{ad}. We then compare the energy savings for the two approaches as a function of T_{ad} for the different energy models.

The ratios of the lifetime in the slotted case over the unslotted case are plotted on Figure 8.2(b). The conclusions are the opposite of the previous section: the unslotted protocol always performs better or equal to the slotted protocol. Indeed, the unslotted protocol has only one listening window per time T_{ad}, whereas the slotted one has S_A reservation slots and every node has to listen for an RTS during these S_A slots.

8.5 Conclusions

With emerging pervasive networks, the focus is on minimizing energy consumption in order to maximize the network lifetime. The most effective way to conserve energy is letting nodes sleep. We identify five key-design elements for UWB-IR sleeping protocols. The first design element is the ability to transmit concurrently without collisions. The second is the possibility of multi-access to the same destination, where one of the accessing signals is detected and others are ignored. The third is the hardware power consumption model that is different with UWB-IR than with narrow band systems. The forth is whether the system is slotted or not. And finally, the fifth is the sleeping cycle-traffic load trade-off. Then, we came up with two sleeping protocols for slotted and unslotted systems. We evaluated their performance analytically. We consider the node lifetime as our evaluation metric. We could show that slotted sleeping is better than unslotted if occasional bursts must be supported. In contrast, unslotted sleeping is better than slotted if occasional maximum latency must be supported.

Chapter 9

Conclusions

This dissertation is divided in two parts. The first addresses data dissemination over WIFI. The second concerns a cross-layer optimization for UWB-IR systems. With both parts, we consider fully self-organized ad-hoc networks, where the main challenge is the absence of a centralized coordination.

As to data dissemination over WIFI, we propose SLEF, a complete practical middleware for multi-hop broadcast in ad hoc networks. It adapts itself to the variability of the ad hoc network environments. This includes the implementation of an adaptive TTL (through the spread control), an adaptive forwarding factor (inhibition) and congestion control. In addition, SLEF achieves buffer management, an efficient use of the MAC broadcast and source-based fairness. All these functions are achieved using only local information to the node and do not need any knowledge about the network topology. We derive simple system equations in order to tune the SLEF parameters, and we deliver default values for them. We validate our design through simulations applied on different vehicular network scenarios ranging from very sparse (DTN like) to very dense (traffic jam). SLEF shows good adaptation and succeeds in avoiding congestion collapse, even in the extreme scenarios where other multi-hop broadcast schemes fail.

Then, we identify vulnerabilities that are specific to epidemic forwarding over wireless ad-hoc networks. We classify these vulnerabilities into two categories: malicious and rational. We evaluate their impact according to the number of attackers and the different network settings. We find that the effect of malicious attacks depends on the position of the attacker relative to

the victim, the network density, the traffic load and mobility. In static scenarios, we identify the attacks that reduce dramatically the victim spread, whereas the harm of other attacks is reduced due to the adaptive forwarding factor control and the injection rate control. In highly mobile vehicular networks, the effects of malicious attacks are minimized due to the spread control. We have studied the rational case in presence of only one attacker in the network. The attacker could achieve considerable profit in all scenarios.

Then, we want to stress test SLEF in the real world, far from the simulation simplifications. Therefore, we build a solid and widely adaptable experimental testbed for wireless networks. It is composed of 57 wireless routers. We show the feasibility of running SLEF on such devices, which are very resource-constrained. Then, we show measurements results that aim at understanding the behavior of the testbed and ensuring its readiness. Further, we compared SLEF to fixed TTL and showed that SLEF performs significantly better.

As to the UWB-IR part, the main problem we address is the presence of uncontrolled multi-user interference. We deal with this problem through a cross-layer optimization. First, we identify the signal acquisition problem that arises using the conventional detection method, which correlates the received UWB Impulse Radio (IR) signal with a Template Pulse Train (TPT) and performs a threshold check on the output of the correlation; we show that the synchronization is either unfeasible or entails an extremely large overhead due to the Inter-User Interference (IUI) in these scenarios. In order to solve the extreme IUI case (near-far problem), we propose a new detection method, which we call Power Independent Detection (PID) method; it splits the correlation into elementary correlations. Each one corresponds to one pulse in the TPT. Then, two threshold checks are performed. The first is to detect pulses whereas the second is to detect the signal based on the number of detected pulses. Our PID method solves the problem without any additional complexity overhead, e.g. for a digital receiver, it employs the same sampling frequency and number of operations as the conventional detection method. We evaluated the performance of the PID detection method based on analysis and simulations. The adopted metrics are (1) the probability of Missed Detection (P_{MD}) when the received IR signal contains the synchronization sequence to be detected (2) the probability of false alarm (P_{FA0}) when the received IR signal does not contain the synchronization sequence to be detected and (3) the total error defined as $E_t = P_{MD} + P_{FA0}$. The results presented in Chap. 7 show a significant improvement compared to the conventional detection

method. Some of the results are that, for $E_0/N_0 = 15dB$ and in the presence of 10 users transmitting simultaneously in the LOS case, $E_t = 10^{-8}$ with the PID method at the optimal working point whereas E_t is very close to 2 with the conventional detection method.

We also investigate the particular case where all concurrent transmissions have the same time-hopping code. Our results show that, with high probability, no collision occurs and we are still able to detect one of the transmitted signals independently of its power.

At the MAC level, we focus only on one component of a MAC layer which is the sleeping mode that could be added to any MAC layer proposal adequate to UWB IR. First, we identify five key design elements for UWB-IR sleeping protocols. The first design element is the ability to transmit concurrently without collisions. The second is the possibility of multi-access to the same destination, where one of the accessing signals is detected and others are ignored. The third is that the power consumption model of the hardware with UWB-IR is different than the one with narrow band systems. The fourth is the possibility for the system to be slotted or not. And finally, the fifth is the sleeping cycle-traffic load trade-off. Then, we came up with two sleeping protocols for slotted and unslotted systems. We evaluated their performance analytically. We consider the node lifetime as our evaluation metric. We could show that slotted sleeping is better than unslotted if occasional bursts must be supported. In contrast, unslotted sleeping is better than slotted if occasional maximum latency must be supported.

9.1 Future Work

Within the framework of a collaboration with Laboratory for computer Communications and their Applications 3 (LCA3) [5], we want to organize extensive series of measurements of wireless network protocols. On one hand, we want to evaluate the performance of SLEF through factorial analysis applied on the measurement results. On the other hand, we are interested in testing and evaluating mesh network protocols. Also, we want to make our testbed publicly accessible to researchers in order to carry out their own measurements.

Appendix A

Appendix

A.1 Analysis

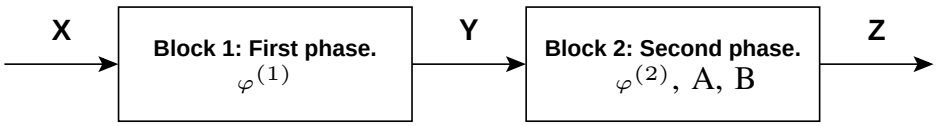


Figure A.1: The synchronization method is modeled as 2 blocks with 3 random variables. Blocks 1 and 2 represent the first and the second phases, respectively. X (input of block 1) indicates the presence of the true sequence in the received IR signal. Y (output of block 1 and input of block 2) indicates whether a detection is declared or not and, if a detection is declared, whether it is a good detection or an FA. Z (output of block 2) gives the result of the verification of the second phase.

Our analysis treats 3 random variables, X, Y and Z, as it is indicated in Fig. A.1. Block 1 represents the first phase in our synchronization method and block 2 illustrates the second phase. X is the input of block 1. Y forms the output of block 1 and the input of block 2. Z is the output of block 2. The values that X, Y and Z can take are as follows:

$$X = \begin{cases} 1 & \text{The IR signal contains the true sequence} \\ 0 & \text{Else} \end{cases}$$

$$Y = \begin{cases} 1 & \text{Good detection and X = 1} \\ 2 & \text{Bad detection either X = 0 or X = 1} \\ 3 & \text{No detection and X = 0} \\ 4 & \text{No detection and X = 1} \end{cases}$$

$$Z = \begin{cases} 1 & \text{Good detection: Y = 1 and the detection is confirmed by the second phase.} \\ 2 & \text{False Alarm: Y = 2 and the detection is confirmed by the second phase.} \\ 3 & \text{Y = 1 but the detection is canceled by the second phase.} \\ 4 & \text{Y = 2 but the detection is canceled by the second phase.} \end{cases}$$

Let P_{GD} be the probability of Good Detection, we have by definition:

$$P_{GD} = P(Z = 1|X = 1) \quad \text{(A.1)}$$

$$P_{MD} = 1 - P_{GD} \quad \text{(A.2)}$$

$$P_{FA0} = P(Z = 2|X = 0) \quad \text{(A.3)}$$

Introducing the variable Y in (A.1) we can write:

$$\begin{aligned} &P(Z = 1|X = 1) = \\ &\quad P(Z = 1|Y = 1, X = 1)P(Y = 1|X = 1) + \\ &\quad P(Z = 1|Y \neq 1, X = 1)P(Y \neq 1|X = 1) \\ &= P(Z = 1|Y = 1, X = 1)P(Y = 1|X = 1) \end{aligned} \quad \text{(A.4)}$$

Similarly we have for (A.3):

$$P(Z=2|X=0) = \\ P(Z=2|Y=2,X=0)P(Y=2|X=0) \tag{A.5}$$

Let us express the terms on the right hand side in (A.4) and (A.5) in terms of P_i, $i = 1,\ldots,4$; By definition we have:

$$P(Y=1|X=1) \;=\; P_1 \tag{A.6}$$

$$P(Y=2|X=0) \;=\; P_2 \tag{A.7}$$

$$P(Z=1|Y=1,X=1) = \\ \sum_{i=B}^{A} \binom{A}{i} P_3^i \left(1-P_3\right)^{(A-i)} \tag{A.8}$$

$$P(Z=2|Y=2,X=0) = \\ \sum_{i=B}^{A} \binom{A}{i} P_4^i \left(1-P_4\right)^{(A-i)} \tag{A.9}$$

Plugging (A.6)-(A.9) in (A.4) and (A.5) we obtain (7.7) and (7.8).

A.2 Communication Range

The mean channel pathloss excluding antenna effects is defined as [14, 39]

$$PL(d) = \frac{P_{TX}}{E\left\{P_{RX}(d)\right\}} \tag{A.10}$$

where P_{TX} and P_{RX} are the transmission and reception powers, respectively, d is the distance between the transmitter and the receiver, and the expectation is taken over an area that is large enough to allow averaging out of the shadowing, as well as the small-scale fading. Due to the frequency dependence of propagation effects in a UWB channel, the wideband pathloss

is a function of frequency, as well as distance. The pathloss as a function of distance and frequency can be written as a product of the terms

$$PL(f,d) = PL(f)PL(d). \tag{A.11}$$

The distance dependence of the pathloss in dB is described by

$$PL(d) = PL_0 + 10nlog_{10}\left(\frac{d}{d_0}\right) \tag{A.12}$$

where the reference distance d_0 is set to 1 m, and PL_0 is the pathloss at the reference distance. n is the pathloss exponent that is equal to 1.63 in the LOS case and 3.07 in the NLOS case. Then a difference of 20 dB in the pathloss between the source and an interferer that is one meter far from the receiver can be written as:

$$\begin{aligned} 20 \;=\; & \left(PL_0 + 10nlog_{10}\left(\frac{d_s}{d_0}\right)\right)_s - \\ & \left(PL_0 + 10nlog_{10}\left(\frac{d_i}{d_0}\right)\right)_i \end{aligned} \tag{A.13}$$

the indexes s and i are to indicate the source and the interferer respectively. Resolving (A.13) we get with the LOS case:

$$d_s = 16.86\,m \tag{A.14}$$

and with the NLOS case:

$$d_s = 4.48\,m \tag{A.15}$$

Bibliography

[1] Haggle. A full Future and Emerging Technologies (FET) Integrated Project funded under the Situated and Autonomic Communication program of the Information Society Technologies priority area of the European Union's Framework Programme 6 (FP6). http://www.haggleproject.org/.

[2] J9 MIDP 2.0. https://www6.software.ibm.com/developerworks/education/wipim/section5.html.

[3] Java in simulation time / scalable wireless ad hoc network simulator , jist/swans, http://jist.ece.cornell.edu/.

[4] jpcap. http://jpcap.sourceforge.net/.

[5] Laboratory of computer Communications and their Applications. Ecole Polytechnique Fédérale de Lausanne. http://lca.epfl.ch/.

[6] Linux / unix command: packet. http://linux.about.com/library/cmd/blcmdl7_packet.htm.

[7] Madwifi. http://madwifi.org/.

[8] MICS. National Center of Competence in Research - Mobile Information & Communication Systems. http://www.mics.org/micsCluster.php?groupName=CL2&action=projects.

[9] OpenWrt. http://openwrt.org/.

[10] Senao :: Nmp-8602 (etsi). http://www.interprojekt.com.pl/senao-nmp8602-etsi-100mw-245ghz-minipci-p-215.html.

[11] Street random waypoint / vehicular mobility model for network simulations , straw, http://www.aqualab.cs.northwestern.edu/projects/straw/.

[12] Winpcap. http://www.winpcap.org/.

[13] Channel modeling sub-committee report final. IEEE P802.15-02/490r1-SG3a, February 2003.

[14] 802.15.4a channel model subgroup final report. IEEE P802.15-04-0535-00-004a, September 2004.

[15] A.-F.Molisch, K. Balakrishnan, C.-C. Chong, S. Emami, A. Fort, J. Karedal, J. Kunisch, H. Schantz, U. Schuster, and K. Siwiak. Ieee 802.15.4a channel model - final report, document 04/662r1, November 2004.

[16] Sara Alouf, Iacopo Carreras, Alvaro Fialho, Daniele Miorandi, and Giovanni Neglia. Autonomic information diffusion in intermittently connected networks. In Autonomic Computing and Networking, August 2009.

[17] Hamada Alshaer and Eric Horlait. An optimized adaptive broadcast scheme for inter-vehicle communication. In *VTC 2005-Spring. IEEE 61st Vehicular Technology Conference*, volume 5, pages 2840– 2844, June 2005.

[18] Adel Aziz, David Starobinski, and Patrick Thiran. Stability of Random Access Wireless Mesh Networks. Technical report, 2008.

[19] M.-G. Di Benedetto, L. De Nardis, M. Junk, and G. Giancola. $(UWB)^2$: Uncoordinated, wireless, baseborn, medium access control for uwb communication networks. *Mobile Networks and Applications special issue on WLAN Optimization at the MAC and Network Levels*, 2005.

[20] Chiara Boldrini, Marco Conti, and Andrea Passarella. Modelling data dissemination in opportunistic networks. In *CHANTS '08: Proceedings of the third ACM workshop on Challenged networks*, pages 89–96, New York, NY, USA, 2008. ACM.

[21] Jean-Yves Le Boudec. Performance evaluation. http://ica1www.epfl.ch/perfeval/printMe/perf.pdf.

[22] E. Callaway, P. Gorday, L. Hester, J. Gutierrez, M. Naeve, B. Heile, and V. Bahl. Home networking with ieee 802.15.4: a developing standard for low-rate wireless personal area networks. *IEEE Communication Magazine*, 40(8), August 2002.

[23] D. Cassioli, M. Win, and A. Molisch. The ultra-wide bandwidth indoor channel: from statistical model to simulations. *IEEE Journal on Selected Areas in Communications*, 20(6), aug 2002.

[24] Augustin Chaintreau, Abderrahmen Mtibaa, Laurent Massoulie, and Christophe Diot. The diameter of opportunistic mobile networks. In *CoNEXT '07: Proceedings of the 2007 ACM CoNEXT conference*, pages 1–12, New York, NY, USA, 2007. ACM.

[25] C.-F. Chiasserini, E. Fasolo, R. Furiato, R. Gaeta, M. Garetto, M. Gribaudo, M. Sereno, and A. Zanella. Smart broadcast of warning messages in vehicular ad hoc networks. In *Workshop Interno Progetto NEWCOM (NoE)*, nov 2005.

[26] C.-F. Chiasserini, R. Gaeta, M. Garetto, M. Gribaudo, and M. Sereno. Efficient broadcasting of safety messages in multihop vehicular network. In *5th International Workshop on Performance Modeling, Evaluation, and Optimization of Parallel and Distributed Systems (PMEO-PDS 2006))*, april 2006.

[27] R. J. M. Cramer, R. A. Scholtz, and M. Z. Win. Evaluation of an ultra-wide-band propagation channel. *IEEE Trans. Antennas Propag.*, 50(5), 2002.

[28] John R. Douceur. The sybil attack. In *The 1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*, March 2002.

[29] Alaeddine El Fawal and Jean-Yves Le Boudec. Synchronizing Method for Impulse Radio Network. Pending patent: US-11/260,390. 2005.

[30] Alaeddine El Fawal and Jean-Yves Le Boudec. A Robust Signal Detection Method for Ultra Wide Band (UWB) Networks with Uncontrolled Interference. *IEEE Transactions on Microwave Theory and Techniques (MTT)*, 54(4, part 2):1769–1781, 2006.

[31] Alaeddine El Fawal, Jean-Yves Le Boudec, Ruben Merz, Bozidar Radunovic, Jrg Widmer, and Gian Mario Maggio. Tradeoff Analysis of PHY-aware MAC in Low-Rate, Low-Power UWB networks. *IEEE Communications Magazine*, 43(12):147, 2005.

[32] Alaeddine El Fawal, Jean-Yves Le Boudec, and Kave Salamatian. Multi-hop Broadcast from Theory to Reality: Practical Design for Ad Hoc Networks. In *Autonomics*, Rome - Italy, 2007.

[33] Alaeddine El Fawal, Jean-Yves Le Boudec, and Kave Salamatian. Vulnerabilities in Epidemic Forwarding. In *The First IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC2007)*, 2007.

[34] E. Fasolo, R. Furiato, and A. Zanella. Smart broadcast algorithm for inter-vehicular communications. In *IWS 2005/WPMC 2005*, September 2005.

[35] A. El Fawal and J. Y. Le Boudec. Fast synchronization in uwb self-organized networks and in the absence of power control. Doctoral School Project, July 2004.

[36] Alaeddine El Fawal and Jean-Yves Le Boudec. A power independent detection method for ultrawide band (uwb) impulse radio networks. In *Proceedings of IEEE International Conference on Ultra-Wideband (ICU 2005)*, Zurich, Switzerland, September 2005.

[37] J. Foerster and Q. Li. Uwb channel modeling contribultion from intel. Call for Contributions on Ultra-wideband Channel Models (IEEE P802/208r1-SG3a), June24 2002.

[38] A Garyfalos and K Almeroth. Coupons: Wide scale information distribution for wireless ad hoc networks. In *IEEE Global Telecommunications Conference (Globecom) Global Internet and Next Generation Networks Symposium Dallas, Texas, USA*, pages 1655–1659, December 2004.

[39] S. S. Ghassemzadeh, R. Jana, C. Rice, W. Turin, and V. Tarokh. Measurement and modeling of an ultra-wide bandwidth indoor channel. *IEEE Trans. Commun.*, 52(10), October 2004.

[40] Zygmunt J. Haas, Joseph Y. Halpern, and Li Li. Gossip-based ad hoc routing. In *Infocom*, pages 1707–1716, 2002.

[41] Zygmunt J. Haas, Joseph Y. Halpern, and Li Li. Gossip-based ad hoc routing. *IEEE/ACM Transactions on Networking*, 14(3):479–491, 2006.

[42] H. Hashemi. The indoor radio propagation channel. In *Proc. of the IEEE*, volume 81, page 943968, 1993.

[43] LAN/MAC Standard Committee IEEE Computer Society. IEEE standard for information technology- telecommunications and information exchange between systems- local and metropolitan area networks- specific requirements part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs). IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003).

[44] LAN/MAC Standard Committee IEEE Computer Society. Ieee p802.15.4a/d7 (amendment of ieee std 802.15.4), part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks, January 2007.

[45] B. Kannan et al. Uwb channel caracterization in indoor office environments. IEEE P802.15-15-04-0439-00-004a, August 2004.

[46] Sachin Katti, Dina Katabi, Wenjun Hu, Hariharan Rahul, and Muriel Medard. The importance of being opportunistic: Practical network coding for wireless environments. In *Allerton conference*, 2005.

[47] Lorenzo Keller and Alaeddine El Fawal. MAC layer functions for SLEF. http://infoscience.epfl.ch/record/98430, 2006. This work was supported by Haggle (a European project) and supervised by Alaeddine El Fawal.

[48] Andrzej Kochut, Arunchandar Vasan, A. Udaya Shankar, and Ashok Agrawala. Sniffing out the correct physical layer capture model in 802.11b, berlin, germany. In *IEEE International Conference on Network Protocols (ICNP 04)*, pages 252– 261, October 2004.

[49] Gökhan Korkmaz, Eylem Ekici, and Füsun Özgüner. An efficient fully ad-hoc multi-hop broadcast protocol for inter-vehicular communication systems. In *ICC '06. IEEE International Conference on Communications*, volume 1, pages 423–428, June 2006.

[50] Gökhan Korkmaz, Eylem Ekici, and Füsun Özgüner. Black-burst-based multihop broadcast protocols for vehicular networks. *IEEE Transactions on Vehicular Technology*, 56:3159 – 3167, September 2007.

[51] Gökhan Korkmaz, Eylem Ekici, Füsun Özgüner, and Ümit Özgüner. Urban multi-hop broadcast protocol for inter-vehicle communication systems. In *VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 76–85, New York, NY, USA, 2004. ACM.

[52] Xiang-Yang Li, Kousha Moaveninejad, and Ophir Frieder. Regional gossip routing for wireless ad hoc networks. *Mobile Networks and Applications*, (10):61–77.

[53] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. In *Proceedings of The First International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR 2004)*, August 2004.

[54] R. Merz, J. Widmer, J. Y. Le Boudec, and B. Radunovic. A joint phy/mac architecture for low-radiated power th-uwb wireless ad-hoc networks. *Wireless Communications and Mobile Computing Journal, Special Issue on Ultrawideband (UWB) Communications*, August 2005.

[55] Ruben Merz, Alaeddine El Fawal, Jean-Yves Le Boudec, Bozidar Radunovic, and Joerg Widmer. The Optimal MAC Layer for Low-Power UWB is Non-Coordinated. In *IEEE International Symposium on Circuits and Systems (ISCAS 2006)*, 2006. Invited paper.

[56] Nathalie Mitton and Eric Fleury. Efficient broadcasting in self-organizing multi-hop wireless networks. *Ad-Hoc, Mobile, and Wireless Networks*, (3738):192–206.

[57] Shah-D. Modiano, E. and G. Zussman. Maximizing throughput in wireless networks via gossiping. In *ACM SIGMETRICS / IFIP Performance'06*, June 2006.

[58] A. F. Molisch, J. R. Foerster, and M. Pendergrass. Channel models for ultrawideband personal area networks. *IEEE Wireless Communications*, 10(6), dec 2003.

[59] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *Mobicom, Seattle, Washington, United States, August 15 - 19, 1999*, pages 151–162.

[60] Tatsuaki Osafune, Lan Lin, and Massimiliano Lenardi. Multi-hop vehicular broadcast (MHVB). In *ITST*, 2006.

[61] Adrian Perrig, Ran Canetti, Dawn Song, and J. D. Tygar. Efficient and secure source authentication for multicast symposium, NDSS'01. In *Network and Distributed System Security*, February 2001.

[62] B. Radunovic and J. Y. Le Boudec. Optimal power control, scheduling and routing in uwb networks. *IEEE Journal on Selected Areas in Communications*, 22(7), 2004.

[63] Maxim Raya, Imad Aad, Jean-Pierre Hubaux, and Alaeddine El Fawal. DOMINO: Detecting MAC layer greedy behavior in IEEE 802.11 hotspots. *IEEE Transactions on Mobile Computing*, 2006.

[64] Maxim Raya, Panos Papadimitratos, and Jean-Pierre Hubaux. Securing Vehicular Communications. *IEEE Wireless Communications Magazine, Special Issue on Inter-Vehicular Communications*, 13(5):8–15, 2006.

[65] A. Saleh and R. Valenzuela. A statistical model for indoor multipath propagation. *IEEE Journal on Selected Areas in Communications*, 5(2), 1987.

[66] Xuemin (Sherman) Shen, Weihua Zhuang, Hai Jiang, and Jun Cai. Channel models for ultrawideband personal area networks. *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, 54(5), sep 2005.

[67] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi Raghavendra. Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In *Proceedings of IEEE PERCOM, on the International Workshop on Intermittently Connected Mobile Ad hoc Networks (ICMAN)*, March 2007.

[68] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Efficient routing in intermittently connected mobile networks: The multiple-copy case. *IEEE/ACM Transactions on Networking*, 16:77–90, 2008.

[69] Min-Te Sun, Lifei Huang, Anish Arora, and Ten-Hwang Lai. Reliable mac layer multicast in ieee 802.11 wireless networks. In *ICPP '02: Proceedings of the 2002 International Conference on Parallel Processing (ICPP'02)*, page 527, Washington, DC, USA, 2002. IEEE Computer Society.

[70] Ken Tang and Mario Gerla. Mac reliable broadcast in ad hoc networks. In *Military Communications Conference, MILCOM 2001*, volume 2, pages 1008– 1013, Washington, DC, USA, 2001.

[71] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report Technical Report CS-200006, Duke University, 2000.

[72] M. Z. Win, R. A. Scholtz, and M. A. Barnes. Ultra-wide bandwidth signal propagation for indoor wireless communications. In *Proc. IEEE Int. Conf. Communications*, volume 1, pages 56–60, Montréal, Canada, June 1997.

[73] Moe Z. Win and Robert A. Scholtz. Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communications. *IEEE Trans. Commun.*, 48(4), April 2000.

Appendix B

Curriculum Vitæ

Alaeddine El Fawal was born in Tripoli, Lebanon, in 1977. He earned his Telecommunication and Computer engineering degree in 2001 and a master's degree in Networking in 2002, both from The Lebanese University. He then joined the LCST (Laboratoire des Composants et des Systèmes de Télécommunications), INSA Rennes France, and the Planète team, INRIA Sophia Antipolis - France, where he carried out his engineering and master's thesis respectively.

In July 2004, he joined the Laboratory of computer Communications and their Applications (LCA), School of Computer and Communication Sciences at EPFL, and begin his PhD work under the sipervision of Prof. Jean-Yves Le Boudec. From 2004 to 2006, he participated in the National Center of Competence in Research on Mobile Information and Communication Systems (NCCR-MICS). From 2006 to 2009, he participated in Haggle, a European project. He was a teaching assistant for TCP/IP Networking, Performance Evaluation and C++ Programming. His main research interests include autonomic opportunistic communication, fully self-organized wireless networks, Ultra-Wide-Band communications and Cheating and security in wireless LANs.

B.1 Publications

Journal Papers

- Alaeddine El Fawal, Jean-Yves Le Boudec "A Robust Signal Detection Method for Ultra Wide Band (UWB) Networks with Uncontrolled Interference". IEEE Transactions on Microwave Theory and Techniques (MTT) (2006).
- Maxim Raya, Imad Aad, Jean-Pierre Hubaux, Alaeddine El Fawal. "DOMINO: Detecting MAC layer greedy behavior in IEEE 802.11 hotspots". IEEE Transactions on Mobile Computing, vol. 5, num. 12, 2006.
- Alaeddine El Fawal, Jean-Yves Le Boudec, Ruben Merz, Bozidar Radunovic, Joerg Widmer, Gian Mario Maggio. "Tradeoff Analysis of PHY-aware MAC in Low-Rate, Low-Power UWB networks". IEEE Communications Magazine, vol. 43, num. 12 (2005).
- Chadi Barakat, Alaeddine El Fawal. "Analysis of link-level hybrid FEC/ARQ-SR for wireless links and long-lived TCP traffic". Performance Evaluation Journal, vol. 57, no. 4, pp. 423-500, August 2004.

Conference Papers

- Alaeddine El Fawal, Jean-Yves Le Boudec, Kave Salamatian. "Multi-hop Broadcast from Theory to Reality: Practical Design for Ad Hoc Networks". First International Conference on Autonomic Computing and Communication Systems (Autonomics), Rome - Italy, 2007.
- Alaeddine El Fawal, Jean-Yves Le Boudec, Kave Salamatian. "Vulnerabilities in Epidemic Forwarding". The First IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC2007), Helsinki, Finland, 18 June 2007.
- Ruben Merz, Alaeddine El Fawal, Jean-Yves Le Boudec, Bozidar Radunovic, Joerg Widmer. "The Optimal MAC Layer for Low-Power UWB is Non-Coordinated". Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS 2006), Island of Kos, Greece, May 21-24, 2006.

- Alaeddine El Fawal, Jean-Yves Le Boudec. "A Power Independent Detection Method for UltraWide Band (UWB) Impulse Radio Networks". Proceedings of IEEE International Conference on Ultra-Wideband (ICU 2005), Zrich, Switzerland, September 2005.

- Alaeddine Al Fawal, Chadi Barakat. " Simulation-based study of link-level hybrid FEC/ARQ-SR for wireless links and long-lived TCP traffic". In proceedings of WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, Sophia Antipolis, France, March 2003.

Patents

- Alaeddine El Fawal, Jean Yves Le Boudec. "Synchronizing Method for Impulse Radio Network". Patent US-11/260,390, 2005.

Demos

- Alaeddine El Fawal, Kave Salamatian, David Cavin, Yoav Sasson, Jean Yves Le Boudec. "A framework for network coding in challenged wireless network". MobiSys 2006, Uppsala - Sweden, June 19-22, 2006.