UNIVERSITY OF CALIFORNIA
RIVERSIDE

Security, Reliability and Performance Issues in Wireless Networks

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Zi Feng

March 2013

Dissertation Committee:

Dr. Srikanth Krishnamurthy, Chairperson
Dr. Harsha V. Madhyastha
Dr. Eamonn Keogh

The Dissertation of Zi Feng is approved by:

_____

_____

_____

Committee Chairperson

University of California, Riverside

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to all the folks who made this dissertation possible. First, I would like to thank my Ph.D advisor, Dr. Srikanth V.Krishnamurthy, whose guidance has been invaluable over the years. Dr. Krishnamurthy's passion, dedication and knowledge in this field has never ceased to amaze me and continued to inspire me through this journey. I would also like to thank Dr. Michalis Faloutsos for the academic support. I would like to extend my gratitude to Dr. Harsha V. Madhyastha and Dr. Eamonn Keogh for giving me their valuable time and participating in the committee for my defense. In addition, I wish to thank Dr. Ioannis Broustis, Dr. Konstantinos Pelechrinis, and Dr. George Papageorgiou for collaborating with me on various research projects and being an integral part of my success. I would like to give a special thanks to my dear labmates, Indrajeet Singh, Jianxia Ning and Shailendra Singh, who provided help, support and advises along the way and helped make my experiences more valuable and memorable. Last but not least, I would like to thank my dear mother, and my loving husband, Paul, who sacrificed so much so I can focus on the research I love.

To my family

ABSTRACT OF THE DISSERTATION

Security, Reliability and Performance Issues in Wireless Networks

by

Zi Feng

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, March 2013
Dr. Srikanth Krishnamurthy, Chairperson

There are trade-offs between security, reliability and performance. In this dissertation, we consider these aspects of wireless operational networks in different settings. Specifically, with respect to security we look at replay attacks [1] and functional reliability [2], with respect to reliability and performance we look at video transmission [3] and visible light communications.

Packet replay attack is a type of denial-of-service(DOS) attack, wherein an attacker replays overheard packets in the wireless network. Our experiments indicate that even a single attacker can degrade the route throughput by up to 61%. We design a lightweight detection and prevention system, COPS (for Copycat Online Prevention System), that intelligently uses a combination of digital signatures and Bloom filters to cope with the attack. We implement COPS on real hardware and perform experiments on our 42 node wireless testbed. Our measurements indicate that COPS achieves its objective; it can efficiently contain the effects of replayed packets to a local neighborhood without incurring high resource consumption penalties.

Later, we look into the functional reliability (FR) of the nodes in wireless networks. The FR is typically assessed based on evidence collected by nodes with regards to other nodes in the network. However, such evidence is often affected by factors such as channel induced effects and interference. We design a framework for collaborative assessment of the FR of nodes, with

respect to different types of functions; our framework accounts for the above factors that influence evidence collection. We also design a module that drastically reduces the overhead at the expense of slightly increased uncertainty in the assessed FR values. We implement our framework on an indoor/outdoor wireless testbed. We show that with our framework, each node is able to determine the FR for every other node in the network with high accuracy.

Next, we research on the performance of video transmission in wireless networks. The end-user experience in viewing a video depends on the distortion; however, also of importance is the delay experienced by the packets of the video flow since it impacts the timeliness of the information contained and the playback rate at the receiver. Unfortunately, these performance metrics are in conflict with each other in a wireless network. We investigate this trade-off between distortion and delay for video. We validate our analysis via extensive simulations. Surprisingly, we find that the trade-off depends on the specific features of the video flow: it is better to trade-off high delay for low distortion with fast motion video, but not with slow motion video. Our simulation results further quantify the trade-offs in various scenarios.

Finally, we look into a visible light system in two rooms with a door open in between. Two emitters tx1 and tx2 are located in room 1 and room 2 respectively. We use BPPM, vary pulse width within the slot to provide different dimming levels. We propose a modified ray-tracing algorithm to calculate the channel impulse response. We also provide a BER analysis considering different combinations of system parameters. Our results show that if tx2 is just illuminating, it does not impact the performance of communications in room 1. However, if both tx1 and tx2 are transmitting, the performance in room 1 is degraded to different levels depending on the position of the receivers. To improve the performance of communications in room 1 in this case, we can increase the dimming level of tx1. Moreover, when dimming level is limited, our results show that reducing the bit rate of tx1 improves the performance in room 1 dramastically. For example, when the bit rate of tx2 is 8Mb/s, reducing the bit rate of tx1 from 8Mb/s to 4Mb/s makes the BER drop from 10^{-3} to 10^{-13}.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Wireless networks have gained a lot of popularity due to its ease of deployment. Various applications are supported by wireless networks, such as tactical deployment, disaster recovery, internet access, etc. However, wireless networks do not guarantee performance because of three main factors:

(i) The quality of the wireless links is not stable, which is induced by channel factors like noise, multi-path degradation, attenuation, and shadowing. This makes the performance of wireless networks unpredictable.

(ii) Wireless links are subject to interference because of the shared nature of wireless medium. Multiple nodes could be accessing the medium at the same time. The level of interference depends on the MAC protocols, the traffic pattern and also the density of the network.

(iii) In addition, wireless networks are vulnerable to a variety of malicious strategies. For example, attackers can eavesdrop and reveal the secret information. Attackers can also inject unwanted packets to jam the network. Malicious network nodes can just drop packets or refuse to reply to requests. Those malicious behaviors can degrade network performance and impair user experience.

Due to the factors mentioned above, there are tradeoffs between security, reliability and performance in wireless networks. In this dissertation, we consider these aspects of wireless

operational networks in different settings. With respect to security we first present a scheme to cope with replay attacks, and then discuss about FR assessment in wireless networks. As for reliability and performance, we look into the tradeoff between distortion and delay for video transmission performance in wireless networks and we also examine on the performance of visible light communication with dimming. Next in this chapter we provide a brief summary of related work and our contributions.

1.1 Security Issues

There are a variety of malicious attacks in wireless networks. Denial of attack (DoS) attack is simple to launch, wherein the attacker injects continuous traffic to the network attempting to jam the wireless medium. One of the best known DoS attacks is jamming attack [4]. Different countermeasures have been proposed in previous work [5, 6, 7]. Another type of DoS attack is replay attack, wherein the attackers generally replay the overheard information to the network. Replay attacks are used in different scenarios. For example, replay attacks can target at assessing secret information exchange among legitimate nodes in wireless networks. In [8, 9] the authors provide design principles or methods to mitigate attacks in this context. Routing message replay attacks have been researched in [10, 11, 12] .

In the first work of this dissertation, we focus on mitigating packet replay attacks. We seek to provide an efficient scheme to detect and minimize the impact of packet replay attacks in wireless networks.

1.1.1 Packet Replay Attack

The objective of the packet replay attacker is to make the packet to look like a legitimate unit avoiding at the same time detection. The intelligence of such an attack lies in convincing (i) the MAC level recipient(s) of a packet to accept and forward it and, (ii) the final destination into believing that this was a legitimately retransmitted packet and that no attack is being launched.

We consider two attacker models: **(a)** the attacker does not manipulate any packet contents. **(b)** the attacker edits the packet header. Attackers are motivated into editing the packet header because the **(a)** attack strategy can be easily detected. We refer to the attacks with these two models as copycat attacks. Our measurements on a wireless testbed, show that the impact of the copycat attack on performance can be devastating.

1.1.2 Technologies to Mitigate Replay Attack

As discussed earlier, with the copycat attack the attacker can either replay the overheard packets or manipulate the headers of overheard packets. Bloom filters can be utilized to determine whether a newly arrived packet is original or replayed. The capability and accuracy of a Bloom filter depends on its size, the hash functions and the number of its recorded packets. We discuss the details of Bloom filters in chapter 2. Digital signatures can be used for authenticating the identity of message senders so that packets with modified headers can be detected. In brief, digital signatures work in this way: each node has a private and a public key. The former is used to create a digital signature and the latter is used to verify the signature. The details of digital signatures can be found in [13].

1.1.3 Our Contributions in Mitigating Packet Replay Attack

The main contributions of our work in mitigating packet replay attacks can be summarized in the following:

(i) We experimentally assess the trade-off between the processing overhead and attack resilience. The results show that by signing just 40% of the packets the throughput on a 2-hop route is increased by up to 56% with one attacker and by up to 95% with two attackers.

(ii) Based on the above assessment, we design a lightweight scheme COPS [1]. COPS is an adaptive scheme that includes a detection and restrainer mechanism to counter copycat attacks. A very limited number of randomly generated packets are signed and verified in benign settings but upon the detection of copycat attack, a more aggressive signing/verification policy is adopted.

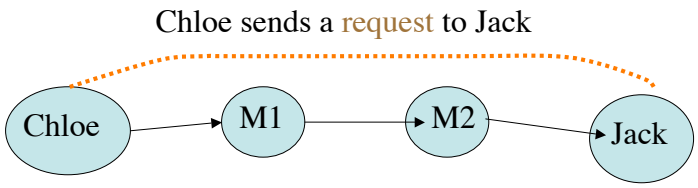


Figure 1.1: A Simple Example of Challenges of Functional Reliability Assessment.

(iii) We implement and experiment with COPS on our 802.11 testbed. Our measurements show that our scheme can effectively contain the copycat attack.

1.2 Reliability Issues

In mission-critical deployments, such as tactical missions or disaster recovery, of multihop wireless networks, nodes are expected to perform specific functions. The functions include forwarding packets and responding to queries. Assessing the reliability of nodes in performing these functions is critical for efficient operations and management of a network.

1.2.1 Challenges in FR Assessment in Wireless Networks

FR assessment in wireless networks is challenging. Fig. 1.1 shows a simple example. Chloe sends a request to Jack and expects for a reply. If Chloe does not get a response from Jack, there are a few possible reasons: **(a)** The links could be lossy. **(b)** Interference may cause unreliable operations or faulty observations. **(c)** M1 or M2 could be unreliable in forwarding packets. **(d)** Jack could be unreliable in responding to requests.

To assess the FR correctly we need to account for the above factors and have nodes collaboratively assess the FR of every other node.

1.2.2 The Basis for FR Assessment

To assess FR in wireless network, we collect evidences for each node. Using Fig. 1.1 as an example, Chloe builds evidence for Jack based on the direct interactions she has with him. This is referred to as direct evidence. Based on a series of such transactions, Chloe applies the Maximum Likelihood Estimation (MLE) framework, to estimate a direct FR value for Jack. She accounts both for the FR of each relay (M1 and M2) on the path to Jack in forwarding packets, as well as the qualities of the links en route to Jack. Besides direct evidences, Chloe also uses indirect evidences from other nodes' feedback with regards to Jack's FR. Indirect evidence is important since Chloe may not sufficiently interact with Jack so that in some extreme cases Chloe may rely on other nodes to assess Jack's FR.

1.2.3 Our Contributions in FR Assessment

In brief, our main contributions in FR Assessment [2] can be summarized as follows:

(i) We design a collaborative FR assessment framework that jointly considers the impact of the unique aspects of a wireless network.

(ii) We incorporate a lightweight evidence propagation scheme in our framework, which filters duplicated evidence and reduces the message complexity.

(iii) We implement and evaluate our scheme on our wireless testbed. Our experiments show that each node infers the FR values for every other node in the network with high accuracy. Our lightweight evidence propagation scheme reduces the propagation overhead by 37% compared to a simple flooding based evidence propagation.

(iv) We experimentally examine the impact of using different routing metrics on FR establishment.

1.3 Performance of Video Transmission

Wireless networks are now deployed in all kinds of environments and are expected to support different applications. Satisfying the network performance requirements of various applications is critical for the design of wireless technologies. Thus, a variety of wireless network standards have been employed. Those standards aim at fulfill different requirements for throughput and transmission range. However, the user experience with respect to some applications (such as video and voice) depends on more metrics than throughput. The quality of video communications is related to distortion and delay, which are two conflicting metrics. In this dissertation, we investigate trading off the distortion for delay for video transmissions in wireless networks.

1.3.1 Quality of Video Transmissions

Video Transmissions have become very popular and prevalent today. There are many video applications such as YouTube and streaming services that have become immensely popular. More importantly, video transmissions are critical in other contexts such as disaster recovery, tactical networks, and surveillance. As aforementioned, the user experience in viewing a video depends on both distortion and delay. The video distortion is affected by both the end coding process at the source and the wireless channel induced errors and interference. The packet delay experienced in transferring a video clip is based on the transmission rate controlled by the channel access protocol. Distortion and delay are in conflict with each other. Distortion level corresponds to packet loss, which is minimized by avoiding interferences. While packet delay is decreased by relaxing the requirement for interference avoidance.

1.3.2 Channel Access Control

The channel access scheme affects both the performance metrics we discuss here. Interference is minimized by dispersing transmissions in the frequency or time domain at the cost of higher packet delay. On the other hand, there are access schemes that allow the concurrent usage of the channel

by multiple transmitters, to decrease packet delay at the expense of potential higher interference. To capture the impact of interference management by an access mechanism on the trade-off between packet delay and distortion, we consider a simple channel access scheme. In brief, the scheme is characterized by an "access probability" that represents the likelihood with which a node transmits packets on the shared medium. In order to be able to represent the whole gamut of channel access mechanisms, we introduce a parameter which we call *aggressiveness* (α); this is used to tune the channel access probability to the medium. A low value of α (≤ 1) corresponds to a case with low interference wherein the nodes access the channel in such a way as to avoid collisions; the distortion in this case is low. However, the delay is high. If α is high, the delay is lowered. At the same time however, the probability of a collision and therefore of a packet loss increases, thereby resulting in higher video distortion. More details on the channel access model are discussed later in chapter 4.

1.3.3 Our Contributions in Video Transmissions in Wireless Networks

In brief, our contributions in video transmissions [3] are as follows:

(i) We develop an analytical framework to capture the trade-off between distortion and timeliness. The framework computes the expected values of the video distortion and the transfer delay of a video clip while accounting for system parameters both at the lower link level (interference, channel induced errors) and the application semantics (motion levels and structure of video content).

(ii) With simulations, we demonstrate the validity of our analytical model. We then quantify via both analysis and simulations the distortion versus delay trade-off for different types of video flows (fast versus slow motion) in a variety of scenarios.

(iii) Our key observation is that trading off high delay for low distortion is important for fast motion video, but not for slow motion video. We find that if the PSNR (Peak Signal to Noise ratio) requirement for a video clip is increased from 20dB to 25dB, the fast motion video clip suffers from a delay increase penalty that is 91 times higher than the penalty incurred with slow motion video. This shows that slow motion video is able to better tolerate packet losses than fast motion video and

thus, should be handled differently in a wireless network.

1.4 Performance of Visible Light Communications

Visible Light Communications (VLC) are gaining popularity ever since an indoor VLC system utilizing white LED light has been proposed. It is considered a promising replacement to radio frequency (RF) communications in indoor settings. In a VLC system, the LED lights not only illuminate the room, but also provide optical wireless communication. Due to its importance, IEEE has a standard [14] for VLC. White light LEDs have the advantages of reliability, security, lower power consumption, easy maintenance, harmlessness to the human eye and cost-effeciency. Furthermore, it is feasible to deploy a VLC network since in most indoor settings the lighting infrastructure already exists.

1.4.1 a VLC system with Dimming

In this dissertation we consider a VLC system deployed over two rooms separated by a door, with each room containing a set of emitters and receivers. This type of indoor setting is typical, especially in home or office establishments. Needless to say, if the door is closed, the VLC system can provide a separate channel for each room since the visible light signal cannot go through opaque surfaces. On the other hand, if the door is open, the visible light signal in one room interferes with the signal in the next room. In Chapter 5 , we study the communication performance of this scenario. Since the primary use of the LED emitters is illumination, dimming control is one of the desired functions of the system. Thus, we use the Variable Pulse Position Modulation (VPPM) scheme, which combines the Binary Pulse Position Modulation (BPPM) scheme for data transmission and the Pulse Width Modulation (PWM) scheme for dimming control. Note that VPPM is easy to implement and has been discussed in [14].

1.4.2 Our Contributions in Visible Light Communications

Our contributions in Visible Light Communications are as follows:

(i) we first present the system model where we characterize the channel based on simulation and propose an algorithm that uses a modified ray-tracing model to calculate the channel impulse response.

(ii) We utilize a simple symbol detection method and compute the Signal to Noise Ration (SNR) to characterize the quality of the connection. We show the simulation results for the SNR distribution in the room for two different cases: (a) both emitters are transmitting, and (b) Tx1 is transmitting and Tx2 is illuminating.

(iii) We also provide BER performance analysis for the system for varying data rates, dimming levels and door sizes. The results show that increasing the data rate or increasing the door size can degrade the BER performance. Increasing the dimming level of Tx1 improves the BER performance. On the other hand, increasing the dimming level of the interfering emitter impacts the BER performance in a negative way, especially for the receivers close to the door.

1.5 Outline of This Dissertation

The rest of this dissertation is organized as follows. In Chapter 2, we propose a lightweight detection and prevention system, COPS (for Copycat Online Prevention System), that intelligently uses a combination of digital signatures and Bloom filters to cope with the packet replay attack. We implement COPS on real wireless testbed and perform experiments. Our measurements show that the system can efficiently contain the effects of replayed packets without inducing high resource consumption. In Chapter 3, we design a framework for collaborative assessment of functional reliability in wireless networks. Each node in the network derives the FR of other nodes based on direct and indirect evidence. We also design a module that drastically reduces the overhead incurred in the propagation of indirect evidence. We implement the framework on an indoor/outdoor wireless testbed. We show that with our framework, each node is able to determine the FR for

every other node in the network with high accuracy. In Chapter 4, we investigate the trade-off between distortion and delay for video transmissions in wireless networks. We develop an analytical framework that accounts for characteristics of the network and the video content, assuming as a basis, a simple channel access policy that provides flexibility in managing the interference in the network. Our simulations validate our model. We also observe that the trade-off depends on the motion level of the video flow. In Chapter 5, we study the performance of a visible light communications system. The system consists of two emitters, Tx1 and Tx2, located in two neighboring rooms, Room 1 and Room 2, respectively. The two rooms are connected via a door. We propose a modified ray-tracing algorithm to calculate the channel impulse response between Tx1 and the receivers in Room 1. We also provide a Bit Error Rate (BER) analysis considering different combinations of system parameters. Our results show that if both Tx1 and Tx2 are transmitting, the performance in Room 1 is degraded to different levels depending on the position of the receivers.

Chapter 2

Coping with Packet Replay Attacks in Wireless Networks

Packet replay attack is a type of denial-of-service(DOS) attack, wherein an attacker replays overheard packets in the wireless network. In this chapter, we consider a variant of packet replay attacks wherein, an attacker simply replays overheard frames as they are, or with minor manipulations in the packet header; we refer to this as the copycat attack. When routers forward such replayed packets, the levels of congestion and interference increase in large portions of the network. Our experiments indicate that even a single attacker can degrade the route throughput by up to 61%. While simple to use techniques such as digitally signing every packet can stem the dissemination of such packets, they are resource intense. Thus, we design a lightweight detection and prevention system, COPS (for Copycat Online Prevention System), that intelligently uses a combination of digital signatures and Bloom filters to cope with the attack. With our system, the task of identifying and discarding replayed packets is distributed across a plurality of nodes on a route. We implement COPS on real hardware and perform experiments on our 42 node wireless testbed. Our measurements indicate that COPS achieves its objective; it can efficiently contain the effects of replayed packets to a local neighborhood without incurring high resource consumption penalties. Specifically, we show that COPS reduces the route throughput degradation by up to 66%.

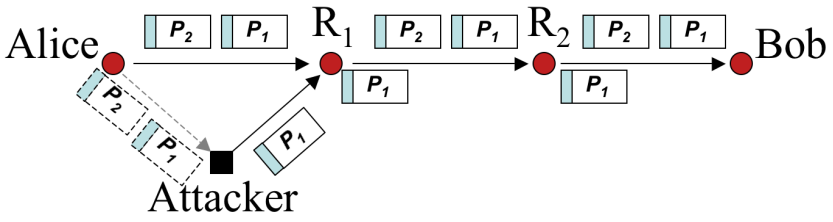


Figure 2.1: The data copycat attack: Alice's packets towards Bob are replayed by the attacker; routers R_1 and R_2 forward the replayed packets.

2.1 Introduction

A simple, yet effective strategy for wireless DoS is to replay locally overheard data packets. These packets are then carried by other forwarding nodes resulting in increased levels of congestion on a wider scale. There are variations of the attack, where either control or data packets are replayed. In this work, we are focus on adversaries that replay *data* packets either without modifying them or after manipulating their contents (typically the header); we refer to this attack as a copycat attack. The objective of the attacker is to make the packet to look like a legitimate unit avoiding at the same time detection. The intelligence of such an attack lies in convincing (i) the MAC level recipient(s) of a packet to accept and forward it and, (ii) the final destination into believing that this was a legitimately retransmitted packet and that no attack is being launched.

To illustrate this attack strategy, consider the simple topology depicted in Fig. 2.1. Here, Alice has established a 3-hop route to Bob via the relays R_1 and R_2. Without loss of generality, the attacker is in Alice's vicinity and overhears her packets.

a. The attacker does not manipulate any packet contents: Let us assume that in the topology of Fig. 2.1, the PDR (Packet delivery Ratio) on all of the links is equal to 1. Alice first transmits packet P_1, which is successfully received by R_1 and overheard by the attacker. Thereafter, when Alice observes the medium to be idle, she transmits packet P_2 to relay R_1, which is also acknowledged. Relay R_1 will place P_1 and P_2 in its MAC output queue and will forward them to R_2 whenever it gains access to the medium. When the attacker observes that relay R_1 has received P_2, the packet P_1 is replayed and received by R_1 again. This replayed packet has Alice's source address and appears

to be new from the perspective of the MAC layer of R_1 (there are no pending ACKs for the replayed P_1). Hence, R_1 is deceived into forwarding it to R_2, *again*. Note here that in order for the attack to be successful, the attacker has to make sure that R_1 will not discard the replayed P_1 packet. This is possible only if R_1 successfully receives subsequent packets (e.g. P_2) from Alice. Otherwise, R_1 will assume that Alice has not received the ACK for P_1, and simply discard the replayed P_1 [15] (until a new packet such as P_2 is received). Note here that the attacker can temporarily store and replay packets that are quite old, thereby effectively circumventing this constraint.

b. The attacker edits the packet header: The above attack strategy can be easily detected. In particular, the original transmitter of the packet (e.g., Alice) can easily detect the malicious node, when overhearing a replayed packet by inspecting the source MAC address. In addition, even if the above is not possible in some topologies, an "unmatched" MAC layer ACK (e.g., sent from R1 to Alice due to the replayed packet from the attacker) will trigger the detection. Upon detection, this attack strategy can be overcome by using a Bloom filter to ascertain that a newly arriving packet has not been received in the past. To bypass such safety countermeasures, the attacker may slightly modify a packet in a way that it still looks legitimate. A simple approach would be to spoof the MAC address of a legitimate node [16] that is not part of the Alice-Bob route, and replace Alice's address with the spoofed address in the header of the data packet. R_1 and R_2 are thus misled into believing that this is a new legitimate packet and Alice cannot detect the attack. Our measurements on a wireless testbed, show that the impact of the copycat attack on performance can be devastating. In particular, the existence of a single attacker can degrade the route throughput by up to 61%, while multiple attackers can further reduce the total network throughput.

The above effects can be completely overcome by using simple, previously proposed techniques. In particular, a **basic scheme** that utilizes digital signatures and Bloom filters can mitigate copycat attacks. Our prototype implementation, presented later, demonstrates the robustness of the basic scheme (up to 66% throughput improvement). However, performing signature and filtering operations on each data packet introduces considerable processing overheads, especially at high data rates. This in turn, can significantly hit performance, especially under benign

settings. Our measurements with different devices verify this fact, particularly when the available CPU and memory resources are limited. COPS balances the trade-off between the incurred processing overhead (due to the use of digital signatures) and the level of achieved protection against DoS, by only requiring signatures on a randomly chosen subset of packets. This subset is determined a priori by the source of the packets, and the processing load of the verification operations is distributed among all the nodes of a route. In brief, the main contributions of our work can be summarized in the following:

- We assess the trade-off between the processing overhead and attack resilience via extensive experimentation. Our experiments show that by signing just 40% of the packets the throughput on a 2-hop route is increased by up to 56% with one attacker and by up to 95% with two attackers.

- Based on the above assessments, we design COPS. COPS is an adaptive scheme that includes a detection and restrainer mechanism to counter copycat attacks. In brief, a very limited number of randomly generated packets are signed and verified in benign settings but upon the detection of a copycat attack, a more aggressive signing/verification policy is adopted.

- We implement and experiment with COPS on our indoor/outdoor 802.11 testbed. Our measurements show that our scheme can effectively contain the copycat attack.

The remainder of the chapter is structured as follows. In section 2.2 we provide brief background on digital signatures and Bloom filters; we also discuss previous related studies. In section 2.3 we describe the attacker model and we quantify the impact of an attack. In section 2.4 we present our measurement guidelines that lead to the design of COPS which is presented in section 2.5. In section 2.6 we evaluate the effectiveness of COPS. Our conclusions form section 2.7.

2.2 Background and Related Work

In this section, we first provide brief background on packet authentication towards fighting DoS attacks. Subsequently, we discuss relevant previous studies.

2.2.1 Using Bloom filters and digital signatures

As discussed earlier, with the copycat attack the attacker can manipulate the headers of overheard packets, by using spoofed addresses. The use of digital signatures can help prevent the propagation of such manipulated packets. Routers may utilize Bloom filters to determine whether a newly arrived packet is original or replayed; while Bloom filters can catch packets that are replayed as is, they cannot catch manipulated packets. Thus, COPS intelligently employs both Bloom filtering and digital signing functionalities. We briefly explain the default operations of Bloom filters and digital signatures, in what follows.

Bloom filters in a nutshell: A Bloom filter is an array of bits of size m for representing a set $B = j_1, ..., j_n$; initially all the bits are set to 0. A Bloom filter uses k independent hash functions $f_1, ..., f_k$ with range $1, ..., m$ [17]. For every $i \in B$, the bits $f_i(j_i)$ are set to 1 for $1 \leq i \leq n$ if j is to be indexed using the filter. Although a location can be set to 1 multiple times, the first change is the only one that has an effect. To check whether indeed an element $a \in B$ is indexed, one needs to examine whether $f_i(a)$ are set to 1, $\forall i$. If not, then $a \notin B$. Otherwise, $a \in B$ with some probability. The probability of error is controlled by choosing an appropriate size for the data structure relative to the size of the set of elements to be represented. A detailed overview of how Bloom filters have been previously used in a plurality of networking problems can be found in [17]. In essence, each packet that is seen, is indexed using the filter. In other words, for a packet p, $f_i(p)$ is set to 1, $\forall i$. When a new packet p' is received, the filter checks to see if $f_i(p') = 1$, $\forall i$. If yes, it is classified to be a replayed packet.

The use of digital signatures: Digital signatures are used for authenticating the identity of message senders. Each node in the network has a private and a public key. The former is needed in

order for a *digital signature* to be created, while the latter is used towards verifying this signature. The signature creation and verification operations typically use the Secure Hash Algorithm (SHA-1) [18, 13]. Since a private key is unique and held secret, attackers cannot reconstruct the same digital signature, unless they compromise the node. If a message is digitally signed, any change in the message will invalidate the signature. Any node that has the public key (typically made known) that corresponds to the signature of a received message can verify the message. In this work we assume that the identity of each legitimate node is a priori bound with a private and a public key by a trusted authority. Further details on signature procedures and algorithms can be found in [13].

2.2.2 Previous studies

In what follows, we discuss related studies on replay attacks.

Attacks on crypto-based key establishment: Such attacks target accessing secret information exchanged among legitimate nodes. The work in [19] describes an interesting classification of such attacks. Aura et al. [8] present a set of design principles to avoid replay attacks during crypto-based key establishment. Malladi et al. [9] propose a method that uses hashed values of random numbers in conjunction with the identities of all nodes to protect information.These approaches cannot mitigate copycat attacks, since data replaying takes place *after* secure communication establishment (perhaps achieved by the aforementioned schemes). In our work we assume that authenticated identity information exchange among legitimate nodes in the network has been established a priori.

Replay attacks related to wireless routing: Papadimitratos and Haas [10] present a secure route discovery protocol based on a message authentication code that can only be verified by the end nodes of a route. This, however, makes the protocol vulnerable to route and data frame replay attacks at intermediate nodes. Zhen and Srinivas [11] propose an approach to cope against a routing message replay attack that generates multiple, redundant RREQ packets. Winjum et al. [12] propose a scheme to address replay attacks in OLSR (Optimized Link State Routing protocol). However, none of the above approaches consider data replay or copycat attacks.

Replaying broadcast packets: Perrig et al. [20] propose a broadcast authentication protocol, TESLA, which uses one-way hash chains and delayed key releases to authenticate broadcast traffic. This idea has been utilized by various other studies, such as [21, 22]. However, as discussed in [23], TESLA induces a security vulnerability: if a secret hash key is released before forwarding nodes authenticate a packet, the newly arrived packets signed by an attacker with this hash key will be falsely deemed legitimate.

Securing data transmissions: Although there exist previous studies on secure data transmissions, they are inadequate in mitigating copycat attacks. Heer et al. [24] provide an adaptive and lightweight security protocol, ALPHA. Before every new (typically large) data packet is to be routed, a small path reservation packet is sent to the final destination in order for all nodes on a path to examine the integrity of the (larger) data packet that will follow. However, since copycat attacks might not modify the contents of the overheard packets, ALPHA cannot efficiently block the propagation of replayed packets.

Using time-stamping and counters to address replay attacks: The use of crypto-synchronization has been widely used in CDMA/EVDO networks for protection against replay attacks [25]. Similar time-stamping strategies for mitigating replay attacks have also been proposed in [26] and [27]. Such techniques, however, require that nodes are very strictly synchronized, and this can typically be only achieved with specialized hardware, especially in dynamic environments.

Anti-replay techniques for wireline networks: There has been some work on replay packet detection in wireline networks. However, these studies do not take into account the inherent properties of the wireless medium. The IP security protocol (IPSec) [28] includes an optional technique for the detection of duplicate IP datagrams. Gouda et al. [29] propose a variation of the IPSec anti-replay mechanism. However, since IPSec establishes a shared symmetric key between the source and the destination, replayed packets are only detected by the end destination, i.e., after they have already travelled along the route. For the same reason, any end-to-end symmetric key based approach is inadequate. Our proposed framework adopts an asymmetric key cryptographic technique, based on digital signatures, as we discuss in the following section.

To the best of our knowledge, our study is the first to provide a complete and effective software framework to mitigate copycat attacks while inducing low processing overhead.

2.3 The Copycat/Replay Attack

In this section, we begin with defining the attacker model that we consider in this study. Subsequently, we demonstrate how the attack can impact network performance.

2.3.1 The Attacker model

The goal of the copycat attack is to mislead routers into forwarding replicas of previously transmitted packets. However, the use of Bloom filters can effectively block the forwarding of packets. The attacker can bypass such blocking by slightly manipulating the packet header; with this, the Bloom filter decision engine will infer that the packet is new. However, if the manipulated packet is signed, the signature verification process at the next router will fail.

Note also that Bloom filters are not of infinite size and thus, they are flushed as soon as they cannot store more information. Hence, a copycat attack may still bypass a Bloom filter. In particular, we consider that the adversarial device has the following capabilities; these can be easily implemented in most commercial wireless cards nowadays :

- It has a wireless interface using which it can overhear packets . It stores the packets locally and retransmits them after an arbitrary time. For each overheard packet, the attacker decides randomly to either perform modifications in the packet header, or replay it without modifications.

- It has sufficient processing and memory capabilities, to store and process large volumes of packets.

- It is not an authenticated device, i.e., it does not have a private or a public key assigned by an authority. However, it can spoof the credentials of legitimate nodes.

- It can replay every packet an arbitrary number of times.
- It adheres to the 802.11 MAC protocol rules.

We assume that public and private keys needed for device authentication have been distributed before the application of our scheme. We also assume that two legitimate nodes can negotiate a shared secret key using their public/private keys, which allows them to communicate secret information.

2.3.2 Demonstrating the impact of the attack

Next, we present some of our testbed measurements that demonstrate the throughput degradation due to the data copycat attack.

Testbed description and experimental methodology: We conduct our experiments on our 42-node wireless testbed, which is deployed on the 3rd floor of the Engineering Building Unit II, at UC Riverside. The testbed configuration is such that the network consists of both indoor and outdoor links; we depict the layout in Fig. 2.2.

The nodes are based on the Soekris net5501 hardware configuration [30], and run a Debian Linux distribution with kernel v2.6.16.19 over NFS. Each node is equipped with 500 MHz CPU, 512 Mbytes of RAM, and a WN-CM9 wireless mini-PCI card, which carries the AR5213 Atheros main chip. Every card is connected to a 5 dBi gain external omnidirectional antenna.

Our measurements encompass an exhaustive set of links and routes of different lengths. We experiment with both 802.11a and g modes of operation (unless otherwise stated our observations are consistent for both modes of operation). The experiments are performed late at night in order to avoid interference from co-located WLANs. All devices (legitimate nodes and attackers) set their transmission powers to 20 dBm.

In order to demonstrate the effectiveness of the considered attack strategy, we perform the following set of experiments. We consider different 2, 3 and 4 hop (overlapping[1]) routes on our

[1]As an example we consider the 4 hop path 37-11-36-38-16, the 3 hop path 37-11-36-38 and the two hop path 37-11-36.

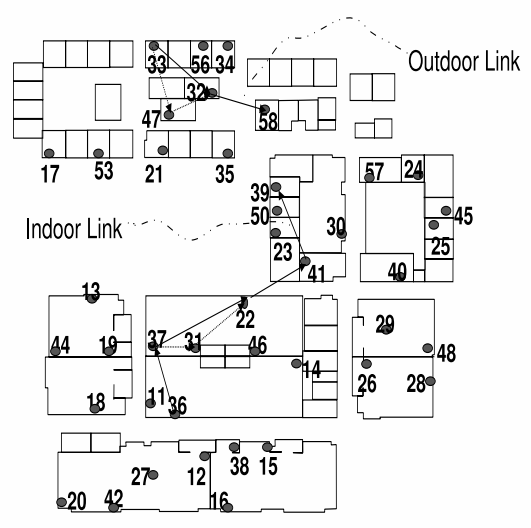


Figure 2.2: Our experimental testbed layout.

testbed and we measure the end-to-end performance degradation due to the presence of an attacker. We repeat each experiment 10 times and Fig. 2.3 depicts the average degradation observed, along with its standard deviation. The attacker is positioned such that it affects the first hop of the route (the effect of the attacker's position will be studied in the following sections). The attacker overhears packets for 120 seconds and replays them for the following 120 seconds. Our results show that the degradation can be as high as 54%. In addition, we observe that increasing the hop count on the route increases the degradation. This is an artifact of the attacker position; since the attack is at inception, the replayed packets traverse a larger portion of the network with longer routes, thus causing higher levels of degradation. Going forward, we examine the reasons behind the performance degradation, considering different settings, and we propose our countermeasure.

2.4 Deriving guidelines for system design

In this section, we first describe the implementation details of digital signatures and bloom filters; both are used in our system. Next, we describe experimental results with a baseline system where these features are employed to various extents. The results from these experiments provide insights

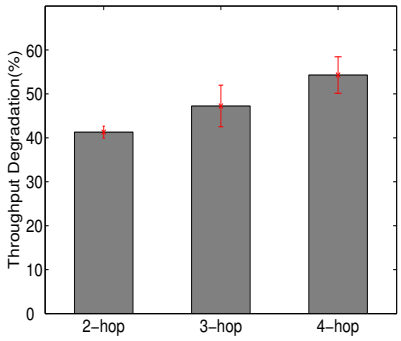


Figure 2.3: The effect of the data copycat attack routes of different lengths.

that guide the design of our system, COPS.

2.4.1 Implementation details

Our implementation is built on a combination of the Linux Click modular router platform v1.6 [31], the OpenSSL library [32] and a C++ Bloom filter library from Google [33]. We implement five new Click elements: *AddSQN, AddSID, AddSign, CheckSign, Bloomfilter* and *CheckRand*. For each packet, the element *AddSQN* adds a 32-bit monotonically increasing sequence number; *AddSID* adds a 32-bit nonce, and *AddSign* adds a 48-Byte digital signature. The element *CheckSign* verifies the signature of each incoming packet. We use the SHA1 function and DSA_sign/DSA_verification functions from the OpenSSL library to generate the message digest of the data as well as to sign/verify the digest. The element *Bloomfilter* uses 8 hash functions to produce hash values for the tuple <SQN, SID> (SQN is the sequence number and SID is the source identifier) in every packet. The Bloom filter size is set by default to 200000 entries. The source node uses the *CheckRand* element to determine whether to sign the packet or not; receivers use this element to determine whether the incoming packet is on the list for verifying the signature. This function can be used to sign and verify only a fraction of the packets generated.

2.4.2 The basic system

We first consider a basic scheme that employs a combination of digital signatures and Bloom filters towards addressing copycat attacks. The scheme does not account for the security-performance tradeoff and its objective is to simply constrain the replayed packets to their local neighborhood; the scheme provides insights on the design of our adaptive approach later.

Source Authentication : The basic scheme authenticates the packets as follows. First an additional header field is inserted into each packet by the source; we call this the BS header. The field includes a sequence number (SQN), a nonce (SID) and a digital signature (SIG). The 32-bit sequence number (SQN) is assigned for each data packet sent by the source. When the SQN space is used out, the SQN wraps around. In the case where two packets with the same SQN arrive at a receiver will not be able to determine if there is a copycat attack or not. Since we want the COPS header of each packet to be unique, we use a 32-bit randomly-generated nonce in conjunction with the SQN for every packet. With this, it is almost impossible for two or more packets to have the same pair of SQN and SID. The source node also uses its private key to produce a digital signature (SIG), which it appends to the header. In order to render the signing process efficient, a one-way hash function is first used on the packet contents to generate a fixed-size bit string. The private key is then used to sign this returned string. We use the popular SHA-1 as the cryptographic hash function [18], and we use DSA as the digital signature algorithm. SHA-1 takes the SQN, SID and the payload of the packet as input and returns a 20-byte output . Then, DSA generates a 48-byte string by signing the 20-byte hashed data.

Packet validation:

Each intermediate node R maintains a Bloom filter $Filter_{S,R}$ to keep track of data packets from every source S that uses R as a relay. Upon receiving a data packet from S, router R first checks the tuple <SQN, SID> of this packet by passing it to $Filter_{S,R}$. If the packet fails this check, R decides that the packet is replayed and discards it. Otherwise, R proceeds with authenticating the data packet. R uses the public key of S to verify the signature in the packet. If this packet fails the

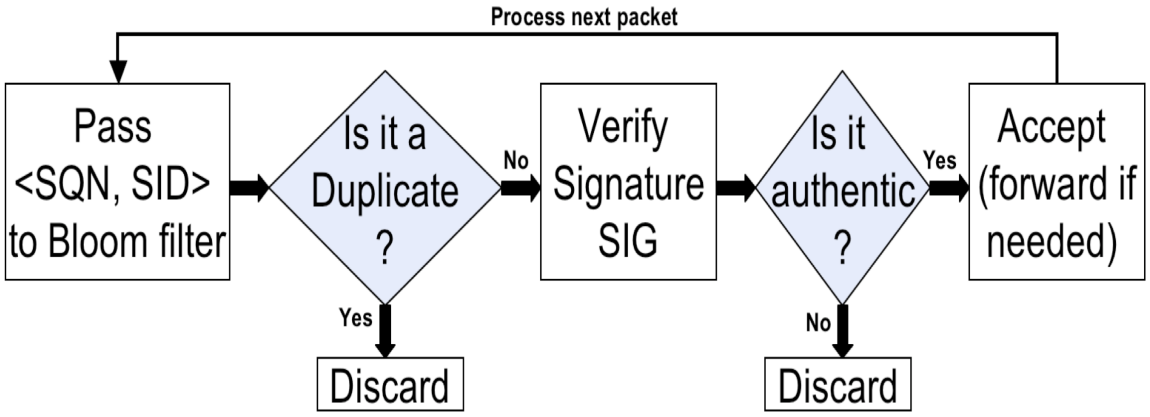


Figure 2.4: Data forwarding at each router: the <SQN, SID> tuple of every data packet is checked by the Bloom filter, which is locally maintained at the router. If it passes the check, the SIG is verified using the public key of the source node. If either of the two verifications fails, the packet is dropped.

signature verification, R considers it to be a spoofed packet and drops it. Otherwise, R forwards the packet to the next hop. The Bloom filter is flushed as soon as no more information can be stored. The final destination of a data packet performs the same operations as the intermediate nodes; it conducts the two verification steps for each incoming data packet and only accepts a packet if it passes the verifications. The steps of this procedure are shown as a block diagram in Fig. 2.4.

Note here that the design of BS, as well as COPS, adopts the use of small-size Bloom filters (order of a few KBytes) in conjunction with digital signatures. This allows its applicability in networks with devices of limited memory, such as sensor motes. One could employ large-size Bloom filters (order or several Mbytes), which can store packets for much longer times. However, the memory and processing requirements would render this inapplicable in networks that consist of CPU and/or memory limited devices.

Processing Overhead: The default operations of BS, described above, provides an almost 100% warranty that replayed packets are blocked from propagating in the network. We have verified the effectiveness of these joint operations against data copycat attacks through extensive experiments on a 42-node wireless network discussed later in this section. However, our

Soekris	500B/pkt	1000B/pkt	1500B/pkt
nosign	5793	3974	1817
sign10%	723	724	726
sign50%	155	155	154
sign80%	109	109	109
sign100%	69	69	68
Laptop	500B/pkt	1000B/pkt	1500B/pkt
nosign	31269	30460	16766
sign10%	7392	7242	7203
sign50%	2358	2469	2410
sign80%	1821	1802	1741
sign100%	1119	1115	1139

Table 2.1: Maximum data injection rates of a source node on a Soekris net5501 box or a DELL laptop with different packet sizes.

Click + DSA signature	Average Process Time(sec)	
	sign	verify
Soekris	0.014644	0.019361
Laptop	0.000955	0.001508

Table 2.2: Average process time of signing/verifying a DSA signature on a Soekris box and the a DELL laptop.

measurements also reveal that these operations induce considerable processing overheads; this is especially the case for packet sources digitally signing and verifying packets.

To quantify the additional overhead that is imposed, we measure how quickly devices with different hardware and with fully-saturated traffic queues, can sign packets (as per the afore-described procedure) and inject them into the network. For this, we observe the ability of two different types of devices to sign and inject packets into the network. Specifically, we test: (a) a Soekris net5501 box [34] with 500 MHz CPU and 512 MB of RAM, and (b) a Dell laptop with 2.4 GHz CPU and 2 GB of RAM. We conduct experiments on 2-hop routes with each of the above devices; in every experiment we sign only a percentage of packets. We also consider various data packet sizes. We measure (a) how many packets/sec each device can inject into the network, and (b) how

much time it takes to perform the signature and verification operations. Our measurements are tabulated in Tables 2.1 and 2.2. We observe that the processing unit capabilities play a significant role in the ability of the device to inject traffic into the network. In particular we observe that more than a 10 fold reduction in the injection rates is possible in same cases, if a 100% of the packets are signed. This exorbitant processing penalty necessitates the design of a *lighter* scheme, which balances security and performance.

2.4.3 Evaluating the performance with the basic system

In the following we will present the evaluations of the BS. The insights from the experimental results drive the design of COPS, presented later.

Towards reducing the processing load due to signing each data packet, we experiment with slight variations of BS. In particular, we utilize *random signing* (RS). BS-RS performs the following actions:

- It determines a subset of packets that correspond to a certain percentage of the packets in the output queue of a source node; only these packets are actually signed.
- The verification load is then distributed among all the nodes of a route. The set of packets that are verified at each node is decided a priori by the source node.

The source node sends the seeds to each node on the route separately.

Note here that the seed information is encrypted by the pairwise secret keys between the source and the intermediate nodes, so that *only the legitimate nodes on the route know which packets are signed*. When a node R gets the seed from source node S, it generates a sequence of random numbers. Packets from S whose SQNs are on the list of this sequence are authenticated by node R.

Packets whose SQNs do not correspond to one of the random numbers have "dummy" signatures (perhaps a random number) inserted in place of real signatures; this prevents the copycat from knowing which packets are really signed and which are not. With this approach, generating and verifying packets becomes faster. Taking the 3-hop route of Fig. 2.1 as an example, Fig.

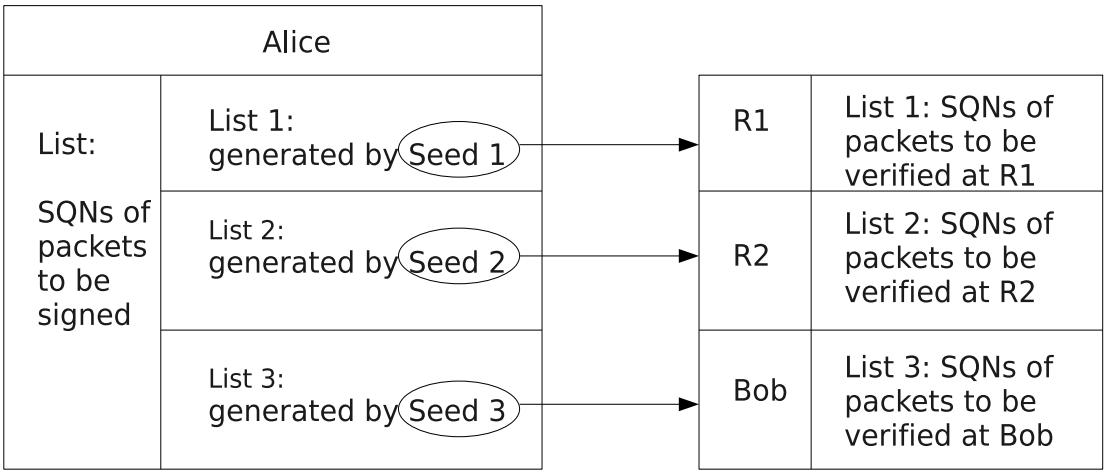


Figure 2.5: Random packet signature scheme for the 3-hop route of Fig. 2.1. Alice has a List consisting of List1, List2 and List3; these are generated by seeds 1, 2, 3 respectively. The List will be the SQNs of the set of packets that are signed by Alice. She sends the three seeds to R_1, R_2 and Bob, respectively. Each of them now knows which packets are to be verified.

2.5 shows how the random packet signature functionality is executed. Note that this introduces a performance vs. security trade-off. If only a fraction of the packets are signed the attacker can effectively have some of the replayed packets forwarded. However, once a replayed packet is recognized, the attack is detected. A higher percentage of signed packets decreases the detection time, but increases the processing overhead.

In what follows, we discuss our experiments towards understanding BS's performance on our testbed.

Assessing the efficiency of BS-RS in scenarios with short routes: To begin with, we consider 2-hop routes and one attacker. The attacking device is located close to the source (we examine other cases later) and overhears packets for the first 120 sec. Subsequently it launches a copycat attack for another 120 sec. For each new packet, prior to transmission the attacker decides randomly on whether to modify the packet header or not. First, we measure the throughput of 12 different 2-hop flows (a) in benign conditions and (b) with the copycat attack. Fig. 2.6 shows the average

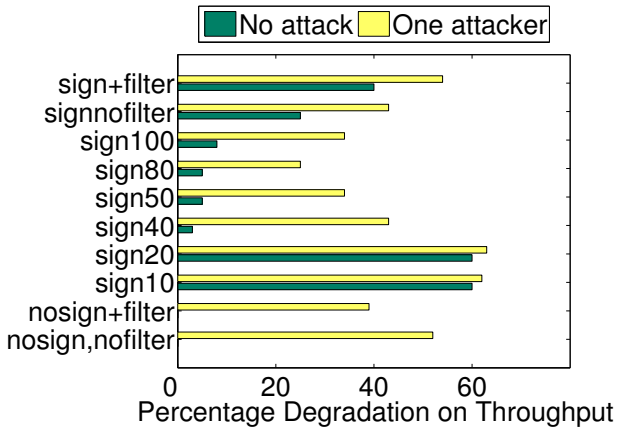


Figure 2.6: Percentage of throughput degradation in 2-hop routes under various conditions.

percentage degradation in the end-to-end throughput for different versions of BS; as an example, sign50 indicates a BS-RS version where 50% of the packets are signed and verified. To compute the degradation, we compare each throughput value with the throughput when (i) no attack occurs and (ii) the network is unprotected i.e., there is no overhead of any sort. We observe that when the attack takes place, if the network is unprotected the degradation in throughput can be as high as 54%. *Surprisingly, we find that signing and validating all the packets (a 100%) in this scenario degrades the throughput to a higher extent than if there was no protection, even under attack.* This is directly attributable to the processing overhead induced by the these operations. On two hop paths, the impact of the attack is constrained to a small portion of the network; the processing overhead incurred in order to prevent the replayed packets from traversing the second hop hurts the performance more than the attack itself. We observe that there is an inherent trade-off in terms of how many packets are signed and verified versus the reduction in degradation in the presence of the attacker. If fewer packets are signed, there is less overhead; however, more of the attacker's packets make it through; if more packets are signed fewer replayed packets are forwarded, but the processing increases. Fig. 2.6 suggests that BS-RS with 80% of the packets signed provides the best trade-off to yield the highest throughput when under attack; the throughput degradation is only about 5% in benign conditions and only about 20% in the presence of the attacker. Note that it is impossible to completely eliminate the impact of the attacker; the attacker's packets will always

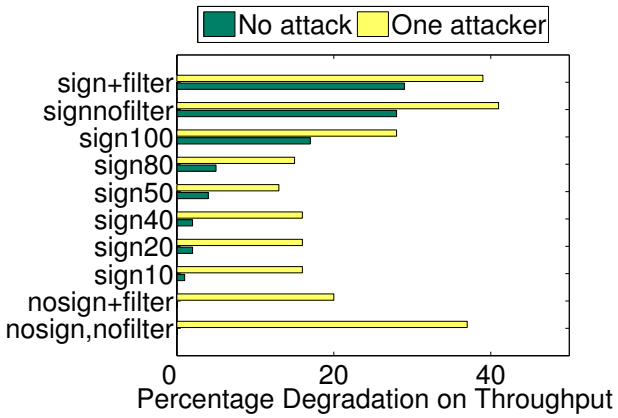


Figure 2.7: Percentage of throughput degradation in 3-hop routes under various conditions.

affect the local neighborhood. In essence, with protection from BS the attacker is reduced to a local jammer; anti-jamming is considered in [35, 36] and is not the focus of this work.

Experimenting with longer routes and more attackers:

Next, we consider routes with length 3 and 4 hops. As before, the attackers overhear and store packets during the first 120 sec of traffic. During the following 120 sec, the attackers launch the copycat attack. First, we consider 12 such 3-hop flows, with one attacker per flow, who overhears and replays packets transmitted from the first *relay* node (near the second hop). In Fig. 2.7 we observe that in the presence of the attack, due to the trade-off discussed earlier between the processing overhead and the level of achieved protection against DoS, BS-RS with 50% of the packets signed has the lowest throughput degradation. Even in this case, signing and verifying all of the packets is not viable since it can lead to significant overhead penalties.

Next, we show experimental results with 4-hop routes; we also consider 2 active attackers (one of which is placed close to the source while the other is placed by the last hop) at the same time. Fig. 2.8 demonstrates that with two attackers launching copycat attacks at the same time and without applying the protection scheme, the degradation in throughput can be as high as 61%. The difference in throughput degradation with BS(-RS) is less pronounced as in the previous cases with 2 and 3 hop routes, with a single attacker (Fig. 2.6 and Fig. 2.7). This is because the attack in this

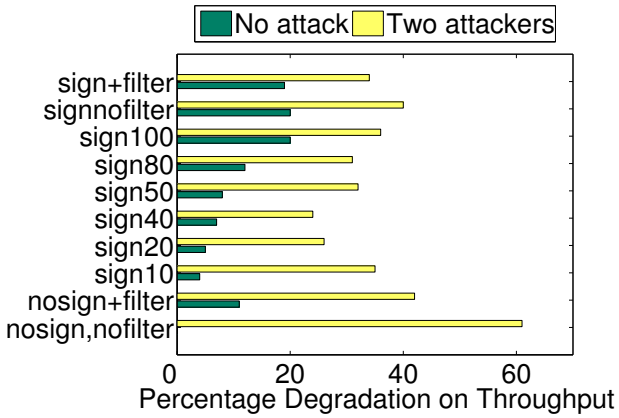
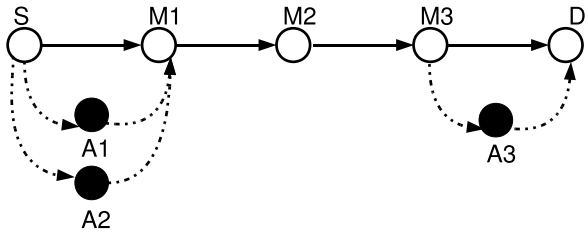


Figure 2.8: Percentage of throughput degradation in 4-hop routes under various conditions.



	RSSI (dBm)		RSSI (dBm)		RSSI (dBm)
S-A1	-50	S-A2	-77	M3-A3	-65
M1-A1	-47	M1-A2	-79	D-A3	-65

Figure 2.9: Experimental set-up for assessing the impact of the attack at various locations.

case is more severe than in the previous cases because: (a) with two attackers the imposed levels of interference are higher and (b) the replayed packets traverse a longer 4-hop route and thus, impact the performance to a larger extent. Signing 40% of the packets provides the best performance versus security trade-offs in this case. We would like to emphasize that similar trends were observed for paths of different hop count and different attacker location as seen in Figs. (2.6)-(2.8).

Cases with attackers with differing signal qualities to the relays: Next we observe how the distance between the replay attackers and victim routes affects the network performance; the larger

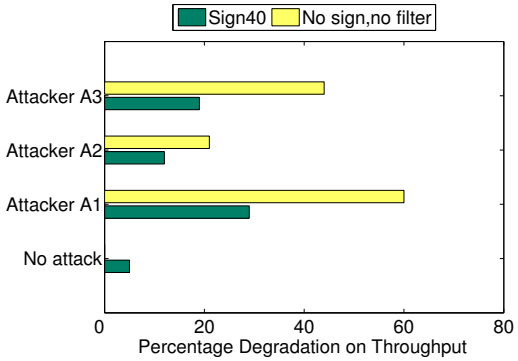


Figure 2.10: Percentage of throughput decrease for various attacker locations along a 4-hop path.

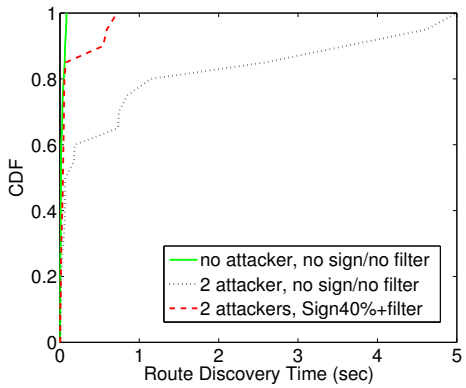


Figure 2.11: The CDF for the route request time based on our measurements on 20 pairs.

the distance the poorer the signal quality between the attacker and the victim route. While we have performed many experiments to validate our general findings, we present a specific sample scenario with a 4-hop route viz., 36→31→22→23→39. We enable fully-saturated UDP traffic from node 36 to node 39, and we activate an attacker at each of 3 different locations, namely A1, A2 and A3, as shown in Fig. 2.9; the measured RSSI values from the attacker, as measured at the victim relays are also shown. At each of these locations, the attacker overhears packets for 2 minutes and subsequently, replays them. Since signing 40% of the packets demonstrates good performance in all of our experiments, we choose this percentage. As we observe in Fig. 2.10, the percentage degradation in throughput is lower as the distance between the attacker and the victims increases. This is expected, since the PDR on the attacker's links decreases due to poorer link quality from increased distance. With this, there are two effects that act in conjunction. First, it becomes more difficult for the attacker to successfully overhear packets; second, replayed packets are not successfully received by the relay.

In the absence of any protection mechanism, when the attacker is close to the source (e.g. location A1 in Fig. 2.9), the copycat attack pushes a significant number of replayed packets along the route. However, when the attacker is closer to the destination (e.g. location A3), replayed packets travel only a few hops and, thus, the impact of the attack is not so prominent.

Assessing the impact of the copycat attack on the routing performance: The transmission

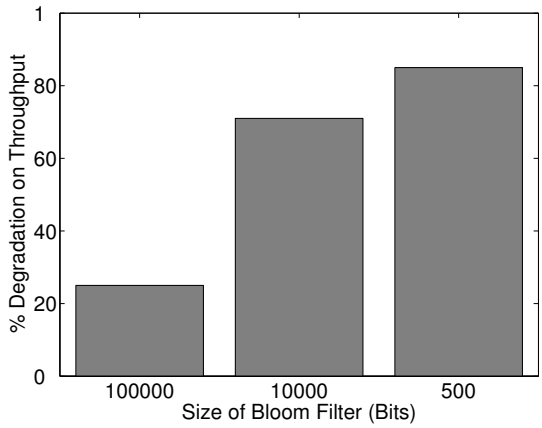


Figure 2.12: The performance of COPS-Lite with Bloom filters of different sizes and hash functions.

of replayed packets increases the medium occupancy and this leaves less time for legitimate nodes to send packets. This can have a significant impact on the performance of routing operations, which depend on the broadcasting of routing control messages. To determine the effect of replay attacks on the performance of routing, we perform experiments with two routes that are active in parallel and are affected by two attackers. We also randomly choose 20 pairs of legitimate nodes on the testbed that wish to establish routes, using the DSR protocol [37]; note that these flows are not directly targeted by the attacker. Fig. 2.3 shows the flows being attacked for this experiment. Nodes 31 and 47 are the attackers; node 36 generates traffic towards node 39, while node 33 generates traffic towards node 58. We measure the route discovery time for each of the *other* 20 randomly selected pairs. Fig. 2.11 shows the cumulative distribution function (CDF) of the average route discovery time: (a) under benign conditions, where BS is not deployed, (b) under the presence of one attacker where BS is not deployed, and (c) under the presence of one attacker where BS-RS (with 40 % of the packets signed[2]) is enabled. We observe that when the attacker is active, the route discovery time in the absence of BS is generally higher; this is because the copycat attack *indirectly* affects the routing performance of the other flows. Since the route discovery and route response packets, in many cases, have to traverse the areas congested due to the replayed packets, the latency incurred

[2]We present results for the Sign40 scheme only, since it provided the best performance among the different SignX schemes examined.

increases. With random signing, this problem is significantly alleviated. To illustrate this with an example, let us take a closer look at the route 44→57 (see Fig. 2.3). While node 31 is launching a copycat attack on the flow 36→39, the route query packets from 44 and the route response packets from 57 have to traverse regions that experience high levels of interference and congestion. In fact, when BS is disabled, the route request from node 44 has only a small chance to getting through to 57 before the route query times out. On the other hand, when BS is enabled, replayed packets are blocked to a large extent from traversing beyond node 22 and this alleviates the problem.

Experiments with various Bloom filter sizes: As one may expect, the size of the Bloom filter affects the potency of the attack. If the size is small, the filter has to be frequently flushed and with this, old packets that were overheard before the flushing will be considered as legitimate after the flushing. We perform experiments with different route lengths and Bloom filter sizes. We provide a representative example in Fig. 2.12, for a 3-hop route, wherein the attacker is located close to the source. Fig. 2.12 shows the throughput degradation with the Sign40 scheme, as compared to applying no protection. We observe that with the 100-Kbit filter, BS-RS provides the best performance; the attack degrades the throughput by 25% only. With filters of size 10-Kbit and 500-bit, many malicious packets go through, leading to higher levels of throughput degradation. This suggests that, depending on the available memory resources, long-size filters are preferable. Note however that if the attacker is able to store packets for very long periods of time, this may still be inadequate.

2.5 Designing COPS

The experimental results presented earlier show that different variations of BS achieve different security levels and network performance. Due to the processing overheads of the core functionalities of BS, blindly applying them can have a negative impact on the network performance. In order to achieve the best trade-off between the processing overhead and the protection level against the replay attack, we utilize the understanding obtained from our

experimental assessments and design COPS. In brief, COPS works as follows; in benign settings and COPS induces signature and verification of only a small fraction (e.g., 10%) of the packets. However, upon attack detection, the source is notified and the fraction of signed and verified packets is increased drastically. Note here that, COPS distributes the verification operations across the nodes along the route in order to reduce the verification load, in a way similar to BS. In the following we provide details on our design.

Detection mechanism: COPS incorporates a detection mechanism for both variations of the copycat attack. For the case where packets have not been manipulated by the attacker, the original transmitter can detect the attack relatively easily. Considering, without loss of generality, the topology in Fig. 2.1, once the attacker replays an unmodified packet, the attack can be detected with one of the two (or both) following ways:

- Alice overhears a packet that includes her credentials.
- Alice receives an unexpected/spurious MAC layer ACK (R1 upon reception of the replayed packet will transmit a MAC layer ACK). This helps if Alice cannot directly overhear the replayed packets.

When the attacker modifies the header of the packet (e.g., spoofs the source MAC address), the attack is detected by utilizing digital signatures; the attacker will not be able to sign the modified packets with a valid private key. If every packet is signed, then the attack is detected upon transmission of the first replayed-modified packet. However, as discussed earlier signing every packet induces significant processing overhead. Thus, COPS' detection scheme incorporates random signing i.e., only a randomly chosen fraction of the generated packets are digitally signed and a dummy signatures are inserted in the other packets (as discussed). In this case, there is a delay introduced in detecting an attack; the attack is detected when the first "signed" packet is modified and replayed. Our measurements indicate that with sign10, the detection time is on average *3 sec*. Our results also indicate, as one might expect, that more aggressive signing schemes (e.g., sign50),

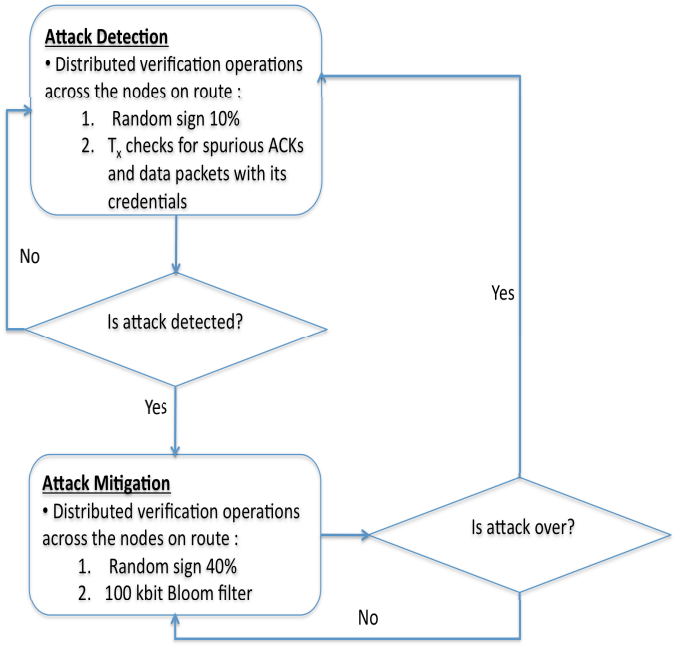


Figure 2.13: The operations of COPS.

reduce the detection time. However, overall, this reduction in detection time cannot compensate for the performance degradation induced by the processing overhead.

The attack mitigation mechanism: In a nutshell, by default (benign conditions are assumed) COPS uses random signing with 10% of packets being signed/verified. Every legitimate transmitter also keeps watching for spurious MAC layer ACKs and/or overheard replayed packets.

When an attack is detected, COPS performs the following steps: **(a)** increases the percentage of signed packets to 40% to contain the modified replayed packets, and **(b)** makes use of a 100 Kbit Bloom filter to contain replays of unmodified packets. Note here that, these values are derived based on our understanding from our previous experimental results. After applying the above countermeasures for restraining the effects of the attack, COPS continually monitors the network behavior for replayed packets. If no replayed packet seen for up to x seconds, COPS restores the default operations. Picking the value of x is not trivial; it heavily depends on the attack strategy.

If we were to pick x to be small, the overhead due to signing higher percentages of the packets under benign settings would be "minimized". However, with an attacker who waits for some time between replayed packets transmissions, a small value would yield undesirable effects. On the contrary, picking a high value for x can increase the hit due to the processing overhead introduced by the digital signatures. In the current implementation of COPS, we pick $x = 20sec$ which we find to be a good value for all considered scenarios. A pictorial representation of COPS is in Fig. 2.13. Our experimental evaluations of COPS are in the following section.

2.6 Evaluating COPS

In this section we present our testbed measurements for evaluating the efficacy of COPS in dealing with replay attacks.

2.6.1 COPS through a lens

We first take a close look at the operations of COPS. For this, we demonstrate the benefits of COPS, via a representative experiment in Fig. 2.14; flow 36→39 is considered with 31 as the attacker (see Fig 2.3). The abscissa indicates the progression in time during the experiment and the ordinate represents a moving average of the end-to-end throughput measurements. Initially there is no attack. The source signs only 10 % of the packets; the signature is verified by various relays en route, distributively. At around 57 sec into the experiment, node 31 launches the attack. The attack is detected within about 3 sec; after this, the source is notified and it sends out a new message, requiring the verification and authentication of 40% of the messages and the activation of Bloom filter. From this point on, 40% of the packets are signed. Note from the figure that this adaptive version provides the best of both worlds; it provides good performance in benign settings is able to effectively alleviate the impact of an attack. Finally, the COPS returns to signing 10% of the packets and deactivates Bloom filtering if no malicious packets are caught for at least 20 seconds (Fig. 2.14).

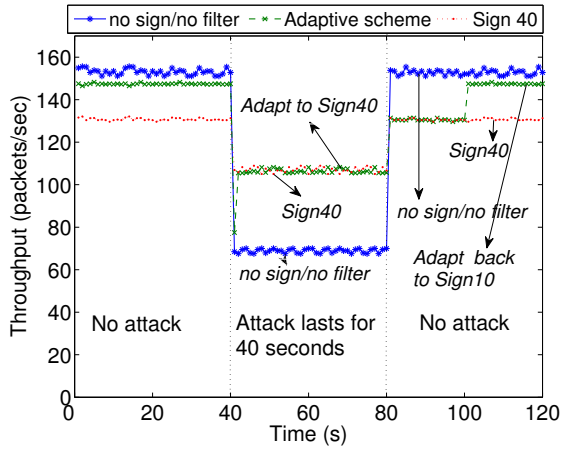


Figure 2.14: A real time trace of the throughput performance, in the presence of COPS.

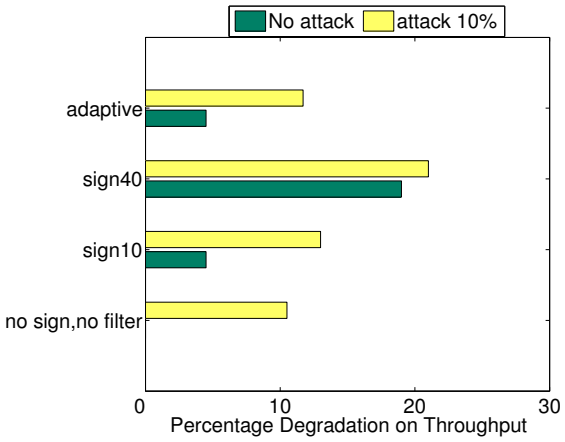
Attack model	Time slots (secs)
10%	[60,180]
30%	[60,180], [540,60]
50%	[60, 180], [300, 360], [420, 540]
70%	[60,360], [420,540]
90%	[60, 600]

Table 2.3: Attack load distribution

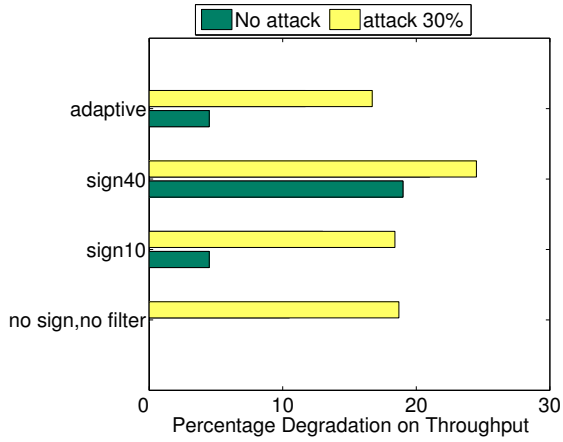
2.6.2 Macroscopic view of COPS

In this section we present our macroscopic results for the efficiency of COPS. In particular, we perform experiments with (i) a large set of routes on our testbed and (ii) a variety of attack models. Each experiment lasts for 600 seconds and we account for different attack loads. Our metric of interest is the average performance degradation when using COPS as compared with the corresponding hit when using static approaches based on BS. In particular we compare COPS with sign10, sign40 and no countermeasure at all.

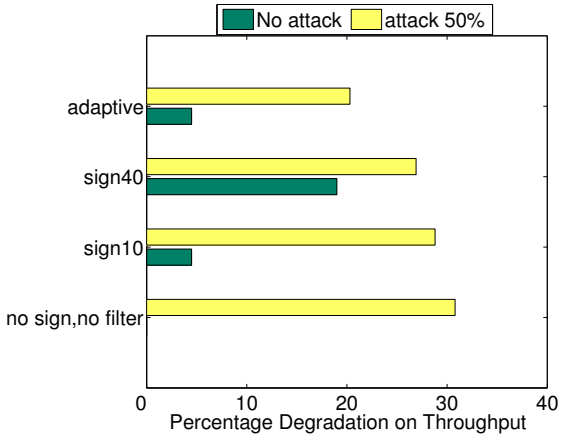
In a attack scenario corresponding to "attack X%", the attacker is active for X% of the total time. Table 2.3 shows the different time slots of the 600 seconds experiment duration, used from every attack model.



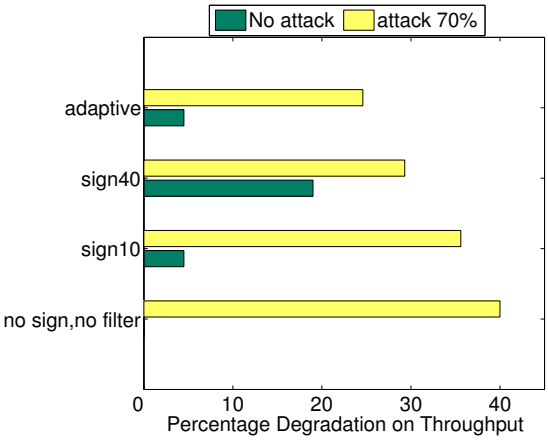
(a) 10%



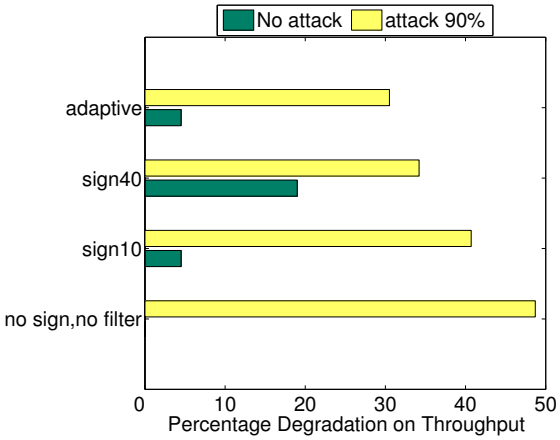
(b) 30%



(c) 50%



(d) 70%



(e) 90%

Figure 2.15: Performance degradation comparison when using COPS for different attack loads.

Benign settings: We observe from Figs. (2.15(a))-(2.15(e)) that in the absence of any attack, the degradation is the lowest when no protection scheme is being employed (as one might expect). In addition, COPS exhibits almost the same performance as with sign10 (recall that under benign conditions COPS randomly signs and verifies 10% of the packets, but does not employ Bloom filter). Employing sign40 and Bloom filters results in at least a 3x higher degradation as compared to COPS.

Attack settings: In all considered scenarios, COPS delivers the best performance (the least throughput degradation), since it adapts to attack changes. Irrespective of the attack intensity, COPS maintains low overhead under benign conditions. A higher processing overhead is incurred only when under attack; in these scenarios, 40% of the packets are signed as this yielded the best overhead versus detection/mitigation efficiency as demonstrated in our prior experiments.

Note here that, estimating the intensity of the attack on the fly and adapting the percentage of packets that are signed with a finer granularity is challenging; it is attacker specific and could increase the complexity of a detection/mitigation system. We will consider such possibilities in the future.

Figs. (2.15(a))-(2.15(e)) provide a gist of our experimental results under the copycat attack when COPS is employed. COPS provides a performance improvement of up to 66% (compared with the case with no security solution). As compared with the static schemes derived from BS, COPS provides an improvement of up to 45%.

2.7 Conclusions

In this chapter, we consider a variant of the data packet replay attack in wireless networks. Replayed packets increase the levels of contention and interference. An attacker can either replay packets as is, or modify the headers to create the illusion of new packets.

Packet signature and verification operations and Bloom filters can help tag packets as authentic or replayed. However, our measurements on a large-scale testbed suggest that a naive application

of these basic functionalities, wherein all packets are checked, project significant amounts of processing overhead. Towards reducing this processing overhead, while effectively mitigating the attack, we propose COPS, where (a) only a percentage of packets is signed, and (b) the load of verification operations is distributed along the nodes constituting the route. The fraction of packets that are signed and verified is adaptively varied based on the understanding gained from our measurements. Our experiments demonstrate that our system can efficiently address the attack in all the considered settings.

Chapter 3

Collaborative Assessment of Functional Reliability in Wireless Networks

Nodes that are part of a multihop wireless network, typically deployed in mission critical settings, are expected to perform specific functions. Establishing a notion of reliability of the nodes with respect to each function (referred to as functional reliability or FR) is essential for efficient operations and management of the network. This is typically assessed based on evidence collected by nodes with regards to other nodes in the network. However, such evidence is often affected by factors such as channel induced effects and interference. In multihop contexts, unreliable intermediary relays may also influence evidence. In this chapter, we design a framework for collaborative assessment of the FR of nodes, with respect to different types of functions; our framework accounts for the above factors that influence evidence collection. Each node (say Chloe) in the network derives the FR of other nodes (say Jack) based on two types of evidence: **(i)** *direct* evidence, based on her direct interactions with each such node and **(ii)** *indirect* evidence, based on feedback received regarding Jack from others. Our framework is generic and is applicable in a variety of contexts. We also design a module that drastically reduces the overhead incurred in the propagation of indirect evidence at the expense of slightly increased uncertainty in the assessed FR values. We implement our framework on an indoor/outdoor wireless testbed. We show that with

our framework, each node is able to determine the FR for every other node in the network with high accuracy. Our indirect evidence propagation module decreases the overhead by 37% compared to a simple flooding based evidence propagation, while the accuracy of the FR computations is decreased only by 8%. Finally, we examine the effect of different routing protocols on the accuracy of the assessed values.

3.1 Introduction

In mission-critical deployments (e.g., tactical missions, disaster recovery) of multihop wireless networks, nodes are expected to perform specific functions (such as forward packets or respond to queries). The reliability of nodes in performing these functions, referred to as functional reliability or FR, is critical for the efficient operations and management of a network. Other nodes may rely on those nodes that are deemed reliable in performing a desired function. We defer a formal definition of FR to Section 3.3. Roughly, the FR of a node with respect to a specific function is the reliability (or responsiveness) of that node in performing the function. A node may become functionally unreliable for various reasons; e.g., it may misbehave due to a low battery or being disconnected, misconfigured, or compromised. Assessing a node's FR in a wireless network is challenging. First, links are lossy; second interference may cause unreliable operations or faulty observations. Finally, information is relayed by other users (nodes), who themselves may not be completely reliable. To our best knowledge, the dependencies between these factors and a user's FR have not been previously investigated. We design a framework accounting for the above factors wherein nodes collaboratively assess the FR of every other node. Every node (say Chloe) maintains an *FR tuple* with respect to every other node (say Jack). Each element of the tuple corresponds to Jack's assessed FR with respect to a different functionality (e.g., routing/forwarding, responding to queries etc.). For instance, when Chloe wants to assess the end-to-end (e2e) FR of Jack (whether he is reliable in responding to a query), she accounts for the possibility that a transaction may fail due to wireless induced effects or due to an unreliable relay.

In particular, Chloe builds evidence for Jack based on the *direct* interactions she has with him. This is referred to as direct evidence. Note that direct evidence does not mean that there exists a physical one-hop distance between Chloe and Jack. "Direct" here pertains to the fact that Chloe gathers this evidence solely based on her transactions with Jack. Based on a series of such transactions, Chloe applies the Maximum Likelihood Estimation (MLE) framework, to estimate a direct FR value for Jack. Here, she accounts both for the FR of each relay on the path to Jack in forwarding packets (forwarding FR), as well as the qualities of the links en route to Jack.

Chloe then combines the above *direct FR* for Jack, with feedback relating to Jack from other users (say Tony) using a *gossiping* scheme; this is referred to as *indirect* evidence. The direct FR is combined with this *indirect FR* using the Dempster-Shafer theory of evidence (DSTE). Indirect evidence is vital since Chloe may not sufficiently interact with Jack; in some extreme cases she may have no interactions at all and may have to rely on other nodes to assess Jack's FR.

The interactions between the FR assessment process and the different network functionalities have complex interdependencies. On the one hand, the assessed values can influence various network functionalities (e.g., relay node selection). On the other hand, the FR inference engine can itself be affected by the operations of various network protocols. For instance, different routing metrics, will result in the use of different paths; the choice of the path influences the evidence collected for FR assessment. In our work, we also experimentally assess the impact of various routing policies on FR assessment.

In brief, our main contributions are summarized below:

(a) We design a collaborative FR assessment framework that jointly considers the impact of the unique aspects of a wireless network (i.e., channel related effects and unresponsive relays). To our best knowledge, this is the first framework to jointly consider these factors.

(b) We incorporate a lightweight evidence propagation scheme in our framework, which intelligently filters duplicated evidence and reduces the message complexity of propagation from $O(N^2)$ to $O(N)$ (N is the number of nodes in the network). The reduction in message complexity comes at a price—increased uncertainty in the reliability computations. **(c)** We implement and

evaluate our scheme on our wireless indoor/outdoor 802.11 testbed. Our experiments show that each node infers the FR values for every other node in the network with high accuracy. Our lightweight evidence propagation scheme reduces the propagation overhead by 37% compared to a simple flooding based evidence propagation.

(d) We experimentally examine the impact of using different routing metrics on FR establishment.

Scope of our work: As implicitly alluded to earlier, reliability is typically function dependent. Jack may forward packets destined to Chloe. However, he may not reply to e2e queries for a specific application because the corresponding application software (residing in his machine) is malfunctioning or he is restricted by policy (Chloe may be unaware of this). Our proposed framework is generic and can be used to assess the FR relating to various wireless network functional contexts. We showcase our framework by assessing e2e (response to queries) and forwarding FRs. However, the applicability of our framework is not limited to these contexts.

A limitation of our approach is that its assesses FR based on Boolean outcomes (e.g., *Did Jack respond to a query?*). It does not take into account possible subjectivity in assessing an outcome. Further, if for example, the question is *Did Jack provide the relevant information in response to a query?*, there may be a response that indicates that he only provided partial information. The extent of this partial information is not accounted for and the system just counts the observation to indicate a success or a failure. Taking into account subjectivity of observations and partially successful outcomes is beyond the scope of this work. We emphasize that our framework is designed to capture the average FR based on observations over sufficiently long periods. It does not address short-term trust variations. Likewise, we do not consider security aspects such as nodes that lie or collude.

Organization: This chapter is organized as follows. Section 3.2 discusses related work and provide background for our framework. Section 3.3 describes our FR establishment scheme. Section 3.4 presents our lightweight evidence propagation mechanism. Section 3.5 presents our implementation and the evaluations of our scheme and Section 3.6 our conclusions.

3.2 Background and Related Work

Related Work: Wireless multihop networks require users to perform specific functions for required network operations. There exists work in the literature to determine whether or not nodes are performing their functions in a non-cooperative setting. Specifically, reputation systems to evaluate, and incentive-based mechanisms to encourage cooperation and functional compliance, have been studied. While our work is similar in spirit to reputation systems, we believe we are the first to account for wireless effects and the impact of other unreliable nodes while estimating the functional reliability of nodes.

Reputation systems: Marti *et al.* [38] propose a scheme for identifying reputable nodes with respect to the routing functionality. They propose *watchdogs* that identify nodes that drop packets based on promiscuous observations and a *pathrater* that avoids paths with such misbehaving nodes. CONFIDANT [39] [40] seeks to identify the routing reliability of nodes. The architecture is similar to that of [38]; a monitoring system is used along with reputation and path selection mechanisms (no details are provided on how the reputation of a node is updated in time). The above schemes focus only on the routing/forwarding functionality. Moreover, they do not account for loss of information due to channel induced effects. Michiardi *et al.* [41] design CORE, which is the first work to define functional reputation. A node might have different reputation values for different network functionalities. Without getting into the details on every possible functionality, the authors present a general scheme that makes use of observations from the users of the network to estimate the functional reliability of a user. What is missing from the above scheme however, is that it does not account for the effect of wireless induced factors or interference while assessing reputation. *In summary, none of the above studies account for the impact of the unique factors that exist in a wireless network, on the estimation process.* To our best knowledge, we are the first to account for wireless induced factors and the network functional context while assessing the FR of a node.

Trust Assessment: *Trust* assessment is loosely connected to our work. Probst *et al.* [42] propose local *trust* computations based only on neighbors' past behaviors. They do not consider aggregation

of trust values and their scheme is specific to the topology and density of the network. Velloso *et al.* [43] present an approach which combines local measurements with aggregated trust values computing a weighed trust. However, they do not provide a method to efficiently propagate these values in the network. The interested reader can find a detailed study on trust management in [44]. In contrast, our work is focused on the assessment of the FR of a node (not trustworthiness) in a wireless network, taking into account wireless induced factors. ***Incentive-based mechanisms:*** Buttyan and Hubaux's [45] scheme provides incentives for users to cooperate and forward packets for other users; however, it does not provide a rating mechanism for the users. Users need to pay credits in order to get their packets forwarded. Relays can accumulate credits for future use; a node that does not have enough credits cannot use the network services itself. SPRITE [46] also uses credits to provide incentives to selfish users to cooperate; however, it does not require any tamper-proof hardware as is the case with [45]. Our work is on assessing the functional reliability of nodes and does not design methods toward ensuring compliance of non-cooperative nodes.

Dempster Shafer Theory of Evidence: DSTE is a generalization of the Bayesian inference theory. Based on evidence from one or more observations (possibly by different entities called *sensors*) of a system, DSTE estimates the system's state.

Let us assume that Θ, is the set of all possible states of the system and H (hypothesis) is a subset of Θ. Every sensor that reports evidence is described by a *Basic Probability Assignment (bpa)*, m, representing a *"measure of belief committed exactly at (each) H"* [47]:

$$m : 2^{\Theta} \rightarrow [0, 1] \tag{3.1}$$

$$m(\emptyset) = 0 \tag{3.2}$$

$$m(H) \geq 0, \forall H \subseteq \Theta \tag{3.3}$$

$$\sum_{H \subseteq \Theta} m(H) = 1 \tag{3.4}$$

Defining the belief (Bel) and plausibility (Pl) of H as:

$$Bel(H) = \sum_{B \subseteq H} m(B),\ Pl(H) = \sum_{B \cap H \neq \oslash} m(B), \tag{3.5}$$

the true belief on H lies within the interval $[Bel(H),\ Pl(H)]$.

In the case of multiple sensors reporting independent evidence for the system's state, the DSTE rule of combination (also known as **orthogonal product** $\oplus$) can be used. In particular, let's assume that we have two sources of independent evidence with assigned bpas m_1 and m_2, respectively. Then, these two sources of evidence can be combined to form a single source of evidence with bpa, $m_{12}(\Theta)$ for hypothesis Θ:

$$m_{12}(\Theta) = m_1 \oplus m_2 = \frac{\sum_{B \cap C = H} m_1(B) m_2(C)}{\sum_{B \cap C \neq \oslash} m_1(B) m_2(C)} \tag{3.6}$$

Intuitively, since the two sources of evidence are independent, the product of the corresponding bpas for the two hypotheses (e.g., B and C) gives the belief value on their intersection. As a result, Eq. 3.6 provides the portion of the total belief committed to hypothesis H from both sources of evidence. The numerator computes the belief on H, since B and C are constrained to the pair of sets whose intersection is H, while the denominator computes the total belief ($B \cap C \neq \oslash$). More details on DSTE can be found in [47] and [48].

3.3 Assessing Functional Reliability

We now describe our FR assessment framework.

Formally, the reliability or responsiveness of a node with respect to a function (or operation) is the likelihood that it will perform the function. For instance, if Chloe seeks some information from Jack, Jack's FR (from Chloe's perspective) reflects the likelihood that he will respond to that query. In a different context, if Chloe relies on Jack to forward her packets to Bob, Jack's FR captures the likelihood that he will relay her traffic toward the destination. As will be clear in the

following, Jack is associated with a tuple of FR values, whose elements embody the likelihood that Jack reliably performs a corresponding network function. ***Our approach in brief:*** FR (as defined above) is assessed based on a node's own interactions with a peer and responsiveness information (relating to the same peer) obtained from other nodes. We refer to the former as *direct evidence* and the latter as *indirect evidence*. The details of the interactions depend on the specific function considered. Irrespective of the specifics of a function, given a series of observations (of whether or not the peer performed the function), we use MLE to determine the probability that the peer is reliable with regards to the particular function. For instance, Chloe establishes direct evidence on e2e transactions with regards to Jack based on the success or failure of her transactions with him. We take into account the "forwarding reliability" of relay nodes[1] en route the peer (say a vector T), and the qualities of the links on the path used for the transaction (say a vector Q). These factors capture the possibilities that a transaction may fail not because the end peer did not respond, but because of link failures or an intermediate node being unreliable with respect to forwarding traffic. Note that the cardinality of T is the number of relays on the route and that of Q is the number of links on the route. We apply MLE to determine the probability that a peer is reliable in responding to the e2e queries, given a series of observations and the vectors T and Q, associated with each transaction attempt.

The FR established based on direct evidence is next updated based on indirect evidence, i.e., through **gossiping** with other nodes. We incorporate a degree of uncertainty in the computed values as explained later. For simplicity, the term reliability or responsiveness (FR) refers to e2e reliability unless explicitly specified. We discuss the applicability of our framework in other contexts in Section 3.3.4.

[1] As explained later, "forwarding FR" is assessed using a different set of observations but using the same statistical framework.

3.3.1 FR representation

If one were to have a strict notion of FR, it should be represented by a binary variable Z; Z is 0 if the node is unreliable and, 1 otherwise. However, in reality there is an uncertainty associated with FR and thus, we denote Z to be the likelihood or probability that the node is responsive (with respect to a function) and hence, $Z \in [0, 1]$.

However, this single *crisp* value does not capture the *degree* of uncertainty with regards to the peer entity under discussion. To account for this, the actual value is considered to lie within $I = [a, b] \subseteq [0, 1]$. The interval signifies the uncertainty associated with the determination of the probability; its width captures uncertainty that we have in our estimation. However, in some parts of the chapter, we will reduce this interval I to a single point value r, through a function h for clarity and tractability. Specifically, the function returns the mean value of the interval I i.e., $r = h(I) = \frac{a+b}{2}$. One can easily use other functions such as $min\{a, b\}$ or $max\{a, b\}$ instead.

We assume that nodes either have a priori perceptions of initial FR levels with respect to other nodes (as an example, a resource rich node may initially be deemed completely reliable), or that every node is reliable or unreliable with an equal likelihood (i.e., each node associates an FR value in the interval $[0.5 - \epsilon, 0.5 + \epsilon]$ for all other nodes). These (initial) values dynamically evolve as entities interact. If a node is responsive, it should eventually be deemed reliable with a low uncertainty.

3.3.2 Updating FR values based on direct evidence

The first source of evidence for a node's (say Chloe's) view of the FR of a peer (say Jack) originates from Chloe's direct interactions with Jack. The outcomes of these interactions/transactions via a wireless network, depend on 3 factors: **(i)** the *forwarding FR* of intermediate nodes that are responsible for relaying the transaction data, **(ii)** the wireless link qualities on the route R from Chloe to Jack, **(iii)** Jack's reliability (which Chloe wants to estimate).

In order to perform her estimation, Chloe monitors the outcome of k consecutive direct

transactions with Jack. These observations form a sample set, indexed by j. For each transaction i Chloe records the outcome, e_i, the probability that the communication path meets the requirements of the application, Q_i, and the forwarding FR of the path, T_i. For a successful transaction, we have $e_i = 1$; otherwise $e_i = 0$. Q_i depends on the specifics of the e2e transaction considered. When only the delivery of the transaction packets is required (e.g., no delay constraints), Q_i is the delivery probability on the route R_i, followed for the transaction i; this is estimated based on the link quality q_l of each of the intermediate links l of the route from Chloe to Jack. Here, q_l is essentially the Packet Delivery Ratio (PDR) on link l and it can be calculated by having neighbor nodes exchange probe packets[2] [49]. Section 3.3.4 examines transactions with different requirements and their mapping onto Q_i. T_i is calculated based on the forwarding reliability intervals I_j of the intermediate nodes j that comprise the route R_i. In particular:

$$Q_i = \prod_{l \in R_i} q_l, \; T_i = \prod_{j \in R_i} h(I_j) \tag{3.7}$$

The above equations assume the independence (i) of the quality of the links on a route and (ii) of the forwarding FR of the intermediate relay nodes. In practice, there may be correlations. First, the projected interference (which affects the quality of the links) on consecutive links may not be independent. Second, the forwarding reliability of the intermediate relays may depend on evidence from common sources causing the independence assumption to not hold. We make the independence assumption due to the complexity in modeling correlations;

however, our evaluations suggest that in spite of these assumptions, our models work well in practice.

Let us assume that Chloe associates with Jack a reliability value of p_i during her i^{th} transaction with him. Then, it is easy to see that the i^{th} transaction is a Bernoulli trial X, with a probability of

[2]We assume that this function(exchanging probe packets) is reliable; however, our framework could be used to assess the reliability of this function as well.

success $p_i \cdot Q_i \cdot T_i$. Thus, the pdf of X is:

$$f_i(X = e_i) = (p_i \cdot Q_i \cdot T_i)^{e_i} \cdot (1 - p_i \cdot Q_i \cdot T_i)^{1-e_i} \tag{3.8}$$

We use the MLE method [50] to update the estimate of the FR of Jack, p, based on the current trial and the previous $k-1$ trials. Then, Chloe's view of Jack's FR is the solution to the optimization problem:

$$\max_{p_j} \quad \frac{1}{k} \cdot \sum_{i=1}^{k} log(f_i(e_i|p_j)) \tag{3.9}$$

$$p_j \in [\widehat{p}, 1] \tag{3.10}$$

where p_j is the FR estimate based on sample set j.

Given $\overrightarrow{e_j}$, Jack's FR cannot be smaller than the percentage of successful transactions in $\overrightarrow{e_j}$. When $\overrightarrow{e_j} = \oslash$, $\widehat{p}$ captures the non-zero probability that all Chloe's transactions with Jack in the sample window fail due to wireless induced failures or unreliable intermediaries. Considering that Jack is 100% reliable, this probability is equal to $x = \prod_{i=1}^{k}(1 - Q_i \cdot T_i)$. If $x = 0$, then all the transactions failed due to Jack and hence, $p = 0$. As x increases, the minimum FR of Jack increases as well. Even if all transactions failed due to bad links or non-reliable relays (i.e., $x = 1$), Jack cannot be deemed 100% reliable. In fact here, Chloe does not know anything about Jack, which implies that $p = 0.5$. As we see, there is a dependence between p and x (i.e., $p = f(x)$ for some function $f()$). Assuming, for simplicity, a linear relation between x and p we can calculate the minimum FR of Jack in the average case to be:

$$\widehat{p} = \begin{cases} (\sum_{i=1}^{k} e_i)/k & \text{if } \overrightarrow{e_j} \neq \oslash \\ (\prod_{i=1}^{k}(1 - Q_i \cdot T_i))/2 & \text{if } \overrightarrow{e_j} = \oslash \end{cases} \tag{3.11}$$

p_j cannot be smaller than $\widehat{p}$, the min FR of Jack as per Chloe's view. Note here that, **the**

optimization problem Eqs. (3.9)–(3.10) always has a solution since the objective function is continuous, and is constrained on a closed and bounded set.

Considering one sample set j and solving the MLE problem provides Chloe with a single point estimate $\widetilde{p_j}$. In order to compute the uncertainty on the FR value, she uses m consecutive sample sets, i.e., a *sliding window* of samples. In particular, if the first sample set consists of the observations indexed by $\{1, 2, \ldots k\}$, the second sample set consists of the observations $\{2, 3, \ldots, k+1\}$, and so on. Using the estimates computed from MLE for each of the above sets, Chloe computes the average estimator $\widetilde{\overline{p}}$ and its standard deviation $\widetilde{p_{sd}}$. Then for the real FR value p^*, the following approximations hold:

$$p^* \in [p^*_{min}, p^*_{max}] \tag{3.12}$$

$$p^*_{min} = \max\{0, \widetilde{\overline{p}} - \frac{\widetilde{p_{sd}}}{2}\} \tag{3.13}$$

$$p^*_{max} = \min\{\widetilde{\overline{p}} + \frac{\widetilde{p_{sd}}}{2}, 1\} \tag{3.14}$$

One could have used a wider interval (e.g., equal to two or three standard deviations). However, we want to keep the uncertainty lower, by possibly trading some level of accuracy. Eqs. (3.12)-(3.14) define the FR interval I with respect to Jack from the perspective of Chloe, based on the direct evidence.

The use of a sliding window results in a subset of the samples being common across windows. Thus, the estimates $\widetilde{p_j}$ are *biased* by the samples that are common across the windows. To obtain unbiased estimates, one would need to use non-overlapping windows. However, in such a case, the updates are performed less frequently and one runs into the problem of the evidence becoming *stale*. Our evaluations show that the sliding window works well in practice.

3.3.3 Combining indirect evidence

Chloe can update her direct view of Jack's FR via feedback from other entities (say Tony) in the network. These entities are the *gossipers*. Using the DSTE, Chloe can combine the obtained feedback to derive an **aggregated** FR value for Jack. The use of indirect evidence is vital; Chloe may have conducted only a few or no transactions with Jack. In such cases, indirect evidence helps her assess Jack's FR. Our trust propagation technique helps address the challenge that indirect evidence may be unreliable.

As mentioned earlier, DSTE can be used to infer the likelihood of a *system* of being in a particular state based on a set of possibly contradicting pieces of evidence. Here, there are two states in our "virtual" system; θ_1, Jack is reliable, and θ_2, he is unreliable. Without loss of generality, we assume that we have two independent sources of evidence; the interval I_d derived from Chloe's direct observations on Jack, and the interval, I_g, that Chloe obtains from a gossiper, Tony. More than two sources of evidence can be aggregated sequentially in pairs.

Directly performing the aggregation on the intervals I_* is hard. Thus, we perform two separate aggregations; one on the lower bounds of the intervals, and one on the upper bounds. Each aggregation will yield an interval in which the real value lies. Thus, Chloe will end up with an interval for the lower bound for Jack's FR, and another interval for the upper bound. However, as we show later, for our system these intervals are reduced to a single value.

A sketch of the aggregation process: Assume that $I_d = [a_1, b_1]$, $I_g = [a_2, b_2]$ and consider the aggregation on the lower bound of the FR interval. First, we define the bpa functions (recall section 3.2), m, associated with each source of evidence. The powerset $2^{\Theta} = \{\emptyset, \{\theta_1\}, \{\theta_2\}, \{\theta_1, \theta_2\}\}$. Note that the elements of the powerset are the different hypotheses H, as introduced in Section 3.2. For the bpa of the direct observations we have:

$$m_d^{min}(\emptyset) = 0 \tag{3.15}$$

$$m_d^{min}(\{\theta_1\}) = a_1 \tag{3.16}$$

$$m_d^{min}(\{\theta_2\}) = 1 - a_1 \tag{3.17}$$

$$m_d^{min}(\{\theta_1, \theta_2\}) = 0 \tag{3.18}$$

For the bpa of the gossip-based/indirect evidence we have:

$$m_g^{min}(\emptyset) = 0 \tag{3.19}$$

$$m_g^{min}(\{\theta_1\}) = a_2 \tag{3.20}$$

$$m_g^{min}(\{\theta_2\}) = 1 - a_2 \tag{3.21}$$

$$m_g^{min}(\{\theta_1, \theta_2\}) = 0 \tag{3.22}$$

We assume that the available pieces of evidence lead to a ***probabilistic binary decision*** (i.e., a node is functionally reliable or not). In other words, there is no uncertainty or ambiguity with regard to the state, i.e., the probability that $\theta_1 \cap \theta_2$ is 0. This results in Eqs. (3.18) and (3.22) and these are key for proving Lemma 1. As discussed in Section 3.2, the bpa m_d^{min} expresses the measure of belief committed on each hypothesis from the direct evidence with regards to the minimum FR value of a node. Since the direct FR of Jack as per Chloe is given by the interval I_d, the belief committed on the hypothesis that a node is responsive (hypothesis θ_1), with respect to its minimum FR, is $m_d^{min} = \inf\{I_d\} = a_1$ (Eq. (3.16)). Given, that θ_1 and θ_2 are complementary we get Eq.

(3.17). The above steps apply to m_g^{min} as well. (Below, we write $m(*)$ instead of $m(\{*\})$.)

Lemma 1. *With the bpas defined in Eqs. (3.15)–(3.22), the aggregated intervals are reduced to a single value.*

Proof. Using the rule of combination (see Eq. 3.6), we first compute the aggregated bpa, m_{agg}^{min}. It is easy to see that: $m_{agg}^{min}(\emptyset) = m_{agg}^{min}(\theta_1, \theta_2) = 0$. In addition we have:

$$m_{agg}^{min}(\theta_i) = \frac{2 \cdot m_d^{min}(\theta_i) \cdot m_g^{min}(\theta_i)}{K},\ for\ i = 1, 2 \tag{3.23}$$

where $K = 2 \cdot m_d^{min}(\theta_1) \cdot m_g^{min}(\theta_1) + 2 \cdot m_d^{min}(\theta_2) \cdot m_g^{min}(\theta_2)$.

Using the above aggregated bpa and the definitions of belief and plausibility (Eq. (3.5)), we have:

$$Bel^{min}(\theta_1) = Pl^{min}(\theta_1) = m_{agg}^{min}(\theta_1) =$$

$$= \frac{a_1 \cdot a_2}{a_1 \cdot a_2 + (1 - a_1) \cdot (1 - a_2)} \tag{3.24}$$

This concludes the proof for the lower bound of the FR. Similar steps can be followed for the upper bound.

□

Thus, after the aggregation process, Chloe's updated estimate of Jack's FR is the interval:

$$T_{\{Chloe,Jack\}} = \left[\frac{a_1 \cdot a_2}{a_1 \cdot a_2 + (1 - a_1) \cdot (1 - a_2)},\right.$$

$$\left.\frac{b_1 \cdot b_2}{b_1 \cdot b_2 + (1 - b_1) \cdot (1 - b_2)}\right] \tag{3.25}$$

3.3.4 Our framework in different contexts

For ease of discussion, we have so far assumed a scenario where Chloe estimates the e2e FR of Jack with a simple transaction type without any QoS requirements. Our framework, however, is independent of the context as long as the observations are Boolean outcomes.

To illustrate this, we consider three contexts next.

e2e FR: The first scenario is a case where Jack is expected to perform a function to satisfy an end-to-end requirement. In the simplest case (as considered in our narrative), the desired function is to just respond to a query. The only metric of interest is the delivery of transaction packets e2e; in this case, the effect of the wireless medium that is of interest is simply the PDR. However, one can easily envision applications that have other requirements. For example, there may be a requirement that the response is received within a prespecified delay. In such a case, one will have to hypothesize about the *timeliness* of Jack's response. The constraints will be the delays imposed by retransmissions and queuing on the wireless medium and the likelihood of the packets being delayed by unresponsive relays. As a second example, if a query requests a video clip, one can impose a requirement on the quality of the clip. Then one needs to compute the likelihood that the degradation was caused by channel induced failures or packet drops by relays as opposed to Jack sending a poor quality clip. The examples here are not exhaustive; however, if one can compute the likelihood of a transaction not meeting the requirements due to wireless effects or packet drops/delays by relays, one can apply our framework to provide an assessment of Jack's FR.

Forwarding FR: One may envision the forwarding FR (alluded to in our earlier discussion) to be independent of the e2e FR. Due to intermittent link qualities, interference, poor battery state, or because of compromise Jack may not forward traffic as expected. Jack's neighbors can monitor (perhaps promiscuously) his activities [38] with respect to forwarding packets and obtain direct evidence $\overrightarrow{e}$, which will lead to their assessments of his forwarding FR. An important difference with the e2e FR is that Chloe may not have a direct link to Jack and thus, no opportunity to observe Jack's forwarding behavior. Thus, she has no direct evidence on Jack. Our framework can still

be applied by combining indirect evidence from Jack's neighbors. We evaluate our framework in terms of its effectiveness in assessing the forwarding FR of nodes in Section 3.5.

Gossiping FR: In our framework, Chloe updates the FR of Jack using indirect evidence from Tony. We have thus far implicitly assumed that all nodes do provide such indirect evidence. For a variety of reasons discussed earlier (e.e., poor battery, loss of connectivity)

Tony may not however, provide timely or accurate information with regards to Jack. Furthermore, the accuracy of the evidence from Tony may depend on factors such as his distance from Jack and Chloe (the greater the distance the less accurate the evidence). The probability of Tony providing timely/accurate evidence refers to the gossiping FR of Tony as per Chloe.

Modeling this leads to additional complexities (finding the likelihood that timely and accurate evidence is received from Tony). In our evaluations, we assume that nodes are reliable with regards to the gossiping function and we evaluate our framework on e2e and forwarding FR. Determining the gossiping FR is cumbersome in terms of obtaining the required evidence. Nevertheless, once the evidence is in place, it is straightforward to infer the gossiping FR. We will consider this in the future.

3.4 Lightweight evidence propagation

Implicit in our FR assessment scheme was the use of a propagation protocol for distributing indirect evidence. The indirect evidence that Chloe obtained from Tony with respect to Jack was simply Tony's *direct* FR assessment of Jack. A simple approach for evidence propagation is a flooding scheme; Tony propagates his assessed direct FR values with respect to all other nodes, to everyone. With this, every node (say, Chloe again) will have a global view of the direct relationships, and thus, she can use DSTE's orthogonal product to compute the aggregated FR on each of her peer network users. While the scheme provides simplicity and accuracy the associated overhead is large; the number of messages that need to be transmitted is $O(N^2)$, where N is the number of nodes in the network.

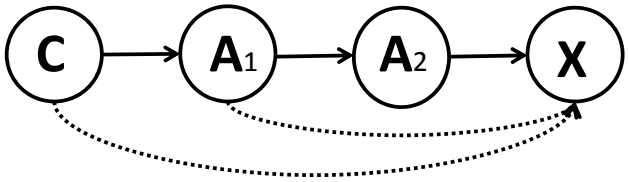


Figure 3.1: Fusion on path P: $P(X) = C(X) \odot A_1(X) \odot A_2(X)$.

We therefore design a new mechanism for propagating the assessed FR values (evidence) with the objective of reducing the communication overhead incurred. If each node only communicates with its direct physical neighbors, the overhead can be reduced drastically. As will be evident, the number of messages that need to be transmitted with such an approach is linear with respect to the number of users in the network, i.e., $O(N)$.

Double Counting of Evidence: The use of local broadcasts results in a challenge that we have to address. Each node propagates evidence only to its physical neighbors. These neighbors then *fuse* or *combine* this evidence and propagate it to other nodes. Let us assume that Chloe gets observations with regards to Jack from two of her neighbors, say, Jill and Jane. However, this evidence may have originated at a single node, say, Tony. Thus, Chloe will have to ensure that she does not "double" count this evidence when computing an FR value for Jack (since the originator is Tony for both pieces of evidence). Below we present three operators that are essential for our scheme for filtering such duplicate evidence. Indirect relaying of evidence (as will be the case here) could also result in a decrease in the accuracy of the computation of FR at each node. Our experimental evaluations presented in Section 3.5 demonstrate, however, that this impact is low.

3.4.1 Path FR operators

Fusion $\odot$

For simplicity, let us consider the 3-hop physical topology of Fig. 3.1. The extension to an n-hop case is trivial. Node C, wants to update X's assessed FR, using information gathered along the

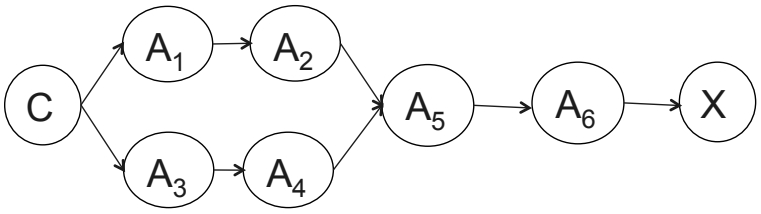


Figure 3.2: Select operator on two dependent paths.

shown path. The steps followed are:

Step 1: A_1 updates X's FR value through DSTE's rule of combination. The two sources of evidence combined are: (i) X's direct FR as per A_1 (direct evidence) and, (ii) X's direct FR as per A_2.

Step 2: C updates X's FR in a similar manner using: (i) its own direct FR for X and, (ii) A_1's updated FR for X (as computed in Step 1).

Note that there are two crucial features of the fusion operator. At every step, (i) the sources of the pieces of evidence that are being combined are independent. Thus, the only requirement for using the DSTE's orthogonal product is fulfilled. (ii) Only new information is being added, which guarantees that there is *no double counting of evidence*.

Path Aggregation $\otimes$

In the majority of the cases, there are multiple physical paths from source node C to node X. In general, each of these paths will result in a different assessed FR interval for X. C should be able to aggregate the different FR intervals with respect to X derived on the basis of the different paths. For this we will use the path aggregation operator $\otimes$.

Let us assume that (a) we have k **independent** paths (i.e., they do not share any intermediate common nodes) from node C to node X and (b) using path P_i, C derives the interval $[a_i, b_i]$ to reflect the FR of node X. Then, the path aggregated FR interval for X, from C's perspective, is computed as

$$P(X) = \otimes_{i=1}^{k} P_i(X) = [\min_{j \in \{1,2,\ldots,k\}} \{a_j\}, \max_{j \in \{1,2,\ldots,k\}} \{b_j\}] \tag{3.26}$$

The path aggregation operator computes the global min (max) of the individually estimated lower (upper) bounds from each considered independent path. Thus, the computed interval is likely to be large and hence the uncertainty on the computed FR value will be larger than what is computed with the basic approach where all information is strictly accounted for.

One could more carefully try to combine evidence from the multiple paths but the processing complexity will be higher. We choose lower complexity in lieu of lower uncertainty with the objective of keeping the process lightweight. Our experimental evaluations show that this results in a small increase in inaccuracy.

In topologies similar to the one in Fig. 3.2, if one were to apply the fusion operation on the two paths leading to node X and then aggregate the FR intervals, double counting of evidence will occur. This is because the paths are not independent (they share common intermediate nodes); the evidence from nodes A_5 and A_6 will be counted twice. We adopt a variant of the select operator $\langle S \rangle$ [51] that chooses the stronger of two paths (trivially extended to multiple paths).

Select makes use of the fusion operator, to compute the FR interval on the furthest common node Y (A_6 in our example) along the two different paths ($P_i = C - A_1 - A_2 - A_5 - A_6$ and $P_j = C - A_3 - A_4 - A_5 - A_6$ as in Fig. 3.2). Let us assume that these intervals are: $P_i(Y) = [a_i, b_i]$ and $P_j(Y) = [a_j, b_j]$. Then, the select operator picks path P as follows:

$$P = P_i(C,Y)\langle S \rangle P_j(C,Y) = \begin{cases} P_i, & \text{if } a_i > a_j, \\ \\ P_j, & a_i < a_j. \end{cases} \tag{3.27}$$

The select operator is *optimistic*, in the sense that it chooses the path that leads to the *maxmin* FR value on the common intermediate node. Note that, if $a_i = a_j$, P is randomly selected.

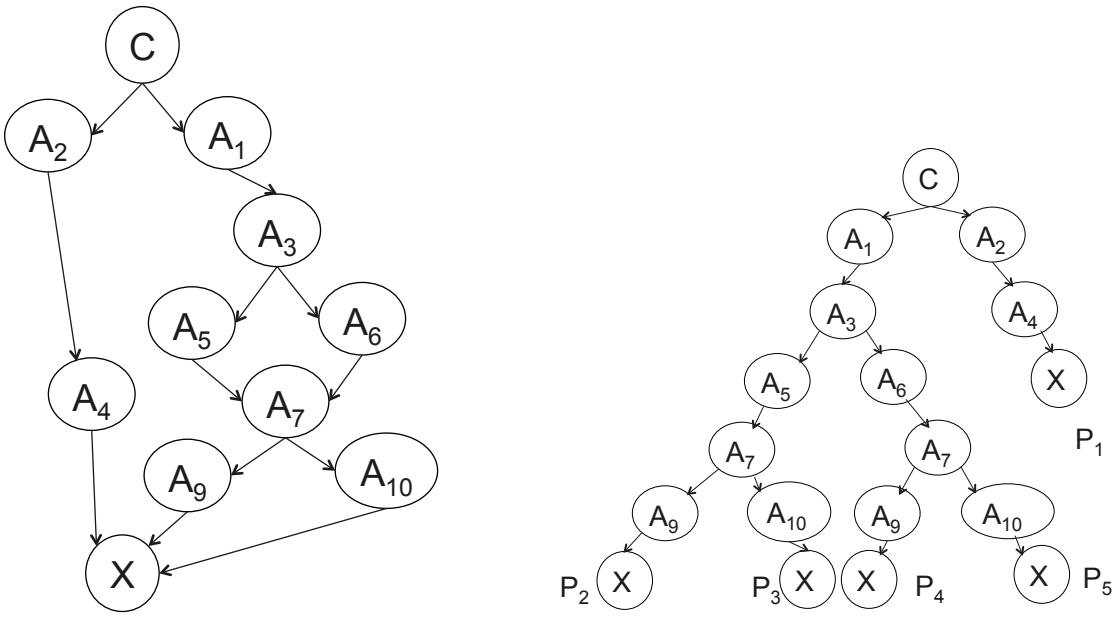
3.4.2 Tree construction and evidence propagation

Next, we present our tree based, lightweight evidence propagation protocol. The goal is to identify a set of intermediate nodes that will provide the indirect evidence in order to update the assessed FR value on a specific network entity. By only considering a subset of nodes in the network to provide indirect evidence, we may reduce the accuracy of the assessments; however, it helps overcome problems arising from the duplication or double counting of evidence while reducing the overhead incurred in FR propagation. Our scheme is based on the physical network topology. In brief, all the independent paths toward the target node are identified and the FR for a node is updated only via these paths, utilizing the operators presented above.

Toy example: Let us consider the physical topology presented in Fig. 3.3(a). Node C wants to update X's FR value. To achieve this, C needs to assimilate the knowledge obtained from the nodes along the path toward X. However, blindly aggregating the FR values reported by the intermediate nodes can lead to double counting of evidence.

In order to construct the tree, we first identify all the different paths that lead to node X. In the scenario under consideration we have five paths. Among these, P_1 is the only path that does not share a common node with any other path i.e., P_1 is *independent* of all the other paths. Using the fusion operator we can estimate X's FR through P_1 to be $P_1(X)$, where $P_1(X) = C(X) \odot A_2(X) \odot A_4(X)$.

The remaining four paths are not independent and therefore we need to eliminate the dependencies. Starting bottom up (i.e., from the target node X to the source C), P_2 and P_4 have A_9 as a common node, while P_3 and P_5 both include A_{10}. For each of the above pairs of paths we apply the select operator. In particular, we have $P_{2,4} = P_2(C, A_9)\langle S\rangle P_4(C, A_9)$ and $P_{3,5} = P_3(C, A_{10})\langle S\rangle P_5(C, A_{10})$. Now we have reduced the number of paths from four to two. These two paths however, are still *dependent* (node A_7 is common to both of them; A_1 and A_3 are also common on both paths, but A_7 is the deepest match). Thus, we apply the select operator again on the two resulting paths and we have: $P_{2,3,4,5} = P_{2,4}(C, A_7)\langle S\rangle P_{3,5}(C, A_7)$.



(a) Physical topo

(b) FR propagation tree.

Figure 3.3: Lightweight Propagation.

The final step is to combine the evidence obtained from the two independent paths, P_1 and $P_{2,3,4,5}$ to form an FR value for X, using the path aggregation operator. In other words, C computes $P(X) = P_1(X) \otimes P_{2,3,4,5}(X)$.

Generalizing our algorithm: The first step toward updating the assessed FR on a network entity is to construct a logical tree that gathers all the possible physical paths from the source to the target node X. The root of the tree is the source node, C, while a leaf of the tree corresponds to the target X. The 1^{st} level of the tree (*children* of root C) includes the physical neighbors of root C. Recursively, the children of the nodes of the i^{th} level (forming the $(i+1)^{st}$ level) include the neighbors of the nodes residing at this level. We continue until we cannot further update the tree paths and we keep only the paths that end at node X. The procedure requires nodes to indicate the *chain of evidence* in their local broadcasts; in other words, they announce the path via which the evidence was propagated. This allows a node (say node C) to determine the topology. Clearly, in the worst case, a piece of evidence has $O(N)$ associated node identities in the chain. For moderate

sized networks, we expect that this will not result in much overhead (assuming no more than 32 bits if IP addresses are used as identifiers). Using hashes of addresses could further decrease this overhead.

Next, we parse the tree and perform the following 3 steps:

1) Step 1 - Identify independent paths: If all nodes A_i, $i \in \{1, 2, ..., n\}$, belonging to a path P_i do not belong to any other path P_j, then P_i is independent of any other path. Thus, the FR of the leaf X along P_i is estimated using the fusion operator, $P_i(X) = C(X) \odot_{i=1}^{n} A_i$.

2) Step 2 - Prune path dependencies: If two paths P_m and P_n, share common nodes, we use the select operator to eliminate the dependencies. As discussed, we first identify the *deepest* matching node (e.g., node f). Note that the common nodes can appear at different tree levels across the different paths;

however, they will appear in the same order. This is easy to verify since the tree is based on the physical network topology.

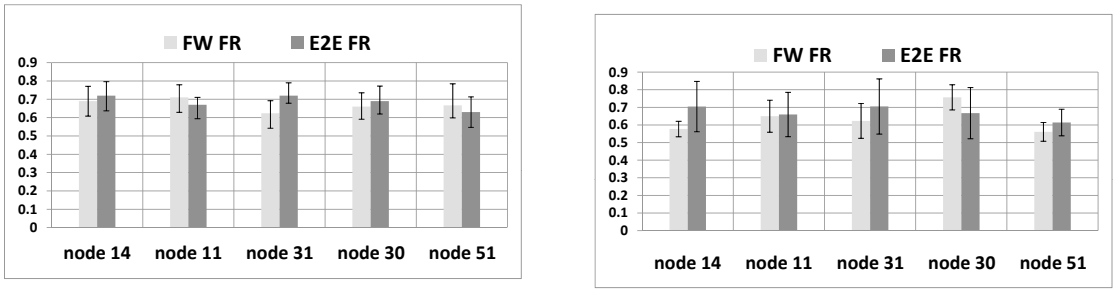
After identifying f we retain path P, where $P = P_m(C, f)\langle S \rangle P_n(C, f)$. This process continues until we remove all dependencies.

At the end of this step all paths that are still under consideration, are independent.

3) Step 3 - Aggregate path FR: Using the path aggregation operator, we aggregate the FR values along all the z (independent) paths identified at the end of Step 2. In particular, the FR of node X is updated to be: $P(X) = \otimes_{i=1}^{z} P_i(X)$.

Message complexity: The tree-based algorithm reduces the communication overhead compared to the flooding approach. It can be shown formally that the message complexity of our scheme is $O(N)$, where N is the number of nodes in the network. Similarly, the time-complexity of tree-construction is also O(N). We omit the proofs due to space limitations.

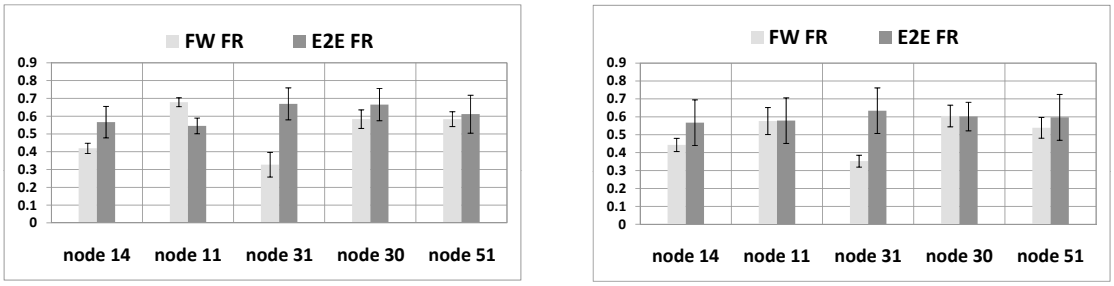
Discussion: With lightweight propagation, due to either link failures or poor forwarding FR, indirect evidence from some of the paths may be lost. However, we find in our experiments that this does not significantly affect the accuracy in FR assessment since in most cases, evidence is collected along the most reliable paths. Finally note here that if a node on a path does not have



(a) Flood-based propagation

(b) Lightweight scheme

Figure 3.4: FR assessment under benign settings (Preconfigured FR is 1 for all nodes).



(a) Flood-based propagation

(b) Lightweight scheme

Figure 3.5: Non-responsive relays can affect the e2e FR assessment (Nodes 14 and 31 have a preconfigured forwarding FR of 0. All other FR values are set to 1).

any evidence relative to a node (say, A_7 does not have any evidence relating to X), it may simply forward the evidence from its predecessor or use a value of 0.5 relating to the FR of X. The latter simply indicates that from A_7's perspective, the events that X is functionally reliable *or* unreliable are equally likely. We use the latter approach in our experimental studies.

Finally, we point out that the mechanics of the lightweight evidence propagation is not new. While the mechanics of the flood based approach is similar to link state routing update propagation, that of the lightweight approach is similar to the propagation of routing updates in distance vector routing [52]. The novelty of the approach is in filtering duplicate evidence.

3.5 Implementation and evaluation of our system

We now present the implementation and experimental evaluations of our framework. We implement our scheme with both (a) flooding based direct evidence propagation and (b) our lightweight evidence propagation.

Protocol implementation and experimental setup: Our implementation is on our 42-node wireless testbed, which consists of both indoor and outdoor links as detailed elsewhere [30].

Our measurements span many wireless links and routes of different lengths, and packet delivery ratios (PDR).

We experiment with the 802.11g mode. Our framework is implemented using the Click toolkit [31]. By default, we use ETX routing [49] and the ETX metric is used to estimate link qualities.

Ground truth: We preconfigure each node's forwarding FR and the likelihood of its responding to e2e queries (e2e FR); this defines the *ground truth* in terms of the actual long term behavior of the node. Each node also has an *initial FR value* for both forwarding and e2e queries with regards to every other node. We set this to be 0.5 with an uncertainty of 0 for both operations. With time, we expect that with our framework, FR values evolve from this initial state, based on both direct and indirect evidence; the FR values at the end of an observation period is the *assessed FR* at the end of the period. Our objective is to see how the assessed FR compares with the ground truth.

Functions examined: Each node (Chloe) randomly picks a target (Jack) and sends *ICMP* queries; these queries form the basis for the direct e2e observations. To decide on the success/failure of an e2e transaction, we send 10 ICMP_ECHO_REQUEST messages and we expect $x\%$ these to successfully result in ICMP_ECHO_REPLY messages. We disable link layer retransmissions and we pick $x\%$ to be the minimum delivery probability among all the links of the route. To determine the success/failure of *forwarding operations*, we configure the sender to be in the promiscuous mode to overhear forwarded packets. If the PDR of the link between the forwarder and the sender is $y\%$, we expect the sender to overhear at least $y\%$ of the ICMP_ECHO_REQUEST messages ***delivered*** to the forwarder. Each experiment runs for 3000 seconds in which each node makes on

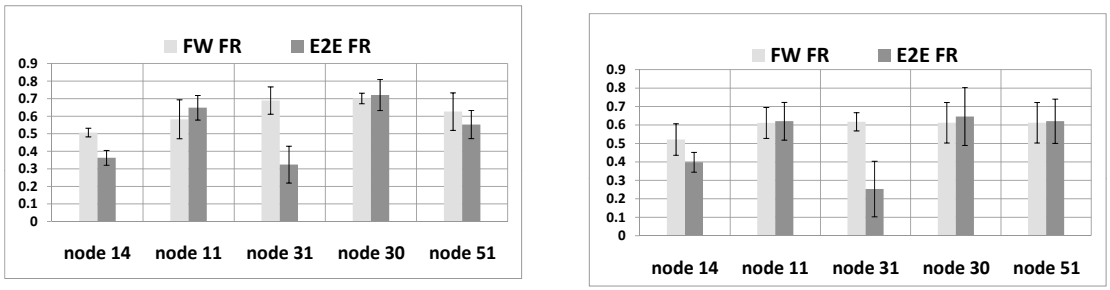
average 10–15 observations for every other node.

Protocol Details: With flood based propagation, the direct evidence of a node is broadcast every 10 seconds (this forms indirect evidence for other nodes). Each node appends any new information and re-broadcasts a received broadcast.

With the lightweight propagation scheme, each node *locally* broadcasts its *aggregated* FR estimates for other nodes, every 10 seconds. These broadcasts include the chain of evidence, which allows each node receiving them to locally recreate the network topology. When a node receives such local information, she updates its aggregated FR (using DSTE) for each of its peers (indirect evidence aggregation) and re-broadcasts the new information. For both schemes, direct evidence is computed using MLE with a sliding window of eight observations.

Accuracy in reliable and unreliable settings: First, we examine the accuracy of the estimation process when using (i) the flood-based evidence propagation and (ii) our lightweight protocol. Initially, we preconfigure all nodes to be responsive (forwarding and e2e FR values are 1 and their uncertainty is 0); this represents the ground truth in terms of FR.

Fig. 3.4 shows representative assessed FR values and their uncertainty for 5 nodes at the end of our experiment (later we present statistics from a large set of trials). To annotate, the bars corresponding to node 14 indicate that, the mean FR (computed over all nodes in the network) on this node is 0.7, the maximum of the mean FR values from among all these FR values is 0.8 and the minimum is about 0.6. The uncertainty on the estimated values for each individual assessment is typically < 10 % of the mean FR value and these are not plotted to ensure clarity. These results suggest that with both schemes the average assessed FR values are sufficiently *close* to the ground truth. Since there are uncertainties that influence the computed FR (wireless effects, varying FR reports from gossipers), the average value almost never converges to the ground truth within the experiment duration. We see that the accuracy is typically lower with the lightweight protocol since, with the latter fewer observations are combined to form indirect evidence in the gossiping phase. Although it is possible that with an increased number of samples, the accuracy can sometimes decrease (rather than increase), we do not observe this to be the case here.



(a) Flood-based propagation

(b) Lightweight scheme

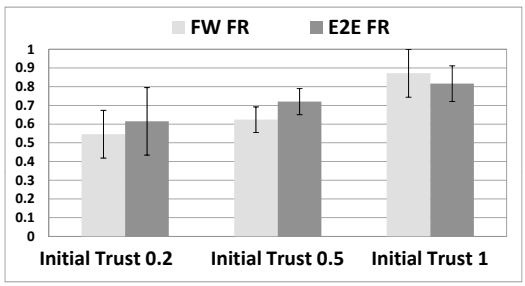
Figure 3.6: The assessed e2e FR for unreliable nodes (Nodes 14 and 31 have a preconfigured e2e FR of '0'. Other nodes are responsive).

Our results with the preconfigured forwarding FR values of nodes 14 and 31 set to 0 with an uncertainty of 0 (ground truth) are in Fig. 3.5.

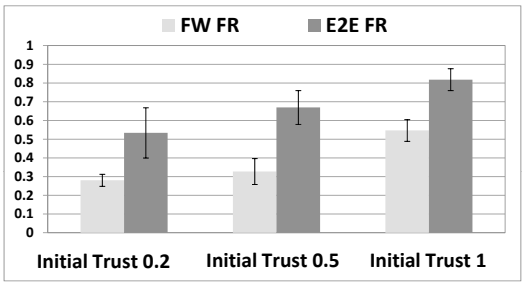
We observe that the assessed "average" e2e FR is lower as compared with that in the "reliable" settings scenario. For example, the inferred average e2e FR of node 11 is approximately 20% lower for both schemes. Many transactions fail due to the forwarding unreliability. Unfortunately, the estimation engines (slightly) penalize the end node as well, due to the uncertainty in ascertaining the reason that caused the failure.

Finally, we preconfigure the e2e FR of nodes 14 and 31 to 0 (i.e., they do not respond to ECHO_REQUESTS) and restore their forwarding responsiveness to '1' (ground truth). The results with our framework are presented in Fig. 3.6. We see that the average e2e FR of these nodes is significantly lower as compared to the other nodes (e.g., node 31 exhibits an approximately 60% lower e2e FR as compared with node 30 for both schemes). This value still is about 0.2, due to the small number of transactions. It is also influenced by the initial FR value of 0.5. These factors result in increased uncertainty in the assessment process, which reduces accuracy.

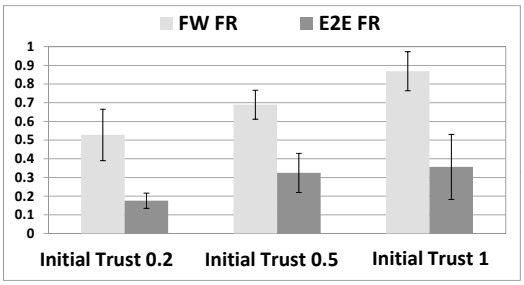
FR evolution with different initial values: As alluded to above, the initial FR value that bootstraps the assessment process can affect the estimated FR value since this is used in the aggregation. To examine the impact of this parameter, we experiment with different initial FR values. In particular, we examine the average FR for node 31 (over the observation period) with 3



(a) Benign node



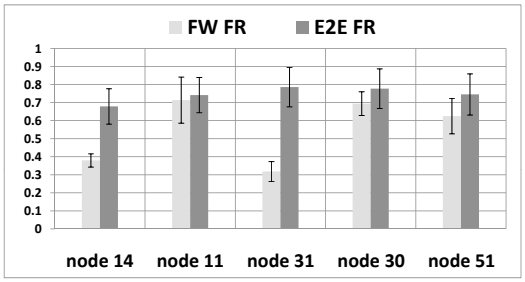
(b) Non-responsive relay



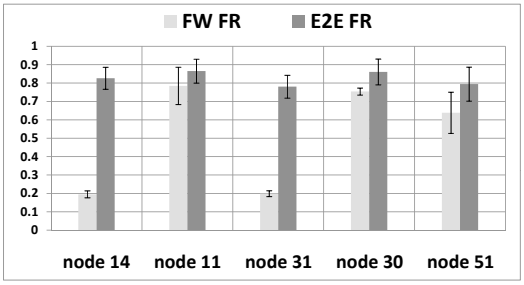
(c) e2e non-responsiveness

Figure 3.7: Higher accuracy is achieved when the initial FR is *closer* to the preconfigured FR.

different initial FR values, 0.2, 0.5 and 1 (all with uncertainty 0). Fig. 3.7 presents our results, for 3 different scenarios; node 31 is (i) a responsive node, (ii) an unreliable relay and (iii) an unreliable node with respect to e2e queries. The values depicted are the estimated average FR values (the average computed on the perception of the mean responsiveness of 31 by all other nodes) after 3000 seconds. It is evident, that when the initial value is close to the actual value, the estimation within the considered time is much more accurate. For instance, when node 31 does not respond to e2e queries (e2e FR is 0), when the initial FR is 0.2, the assessed value is approximately 0.18, while with an initial FR of 0.5 (respectively, 1) the estimated values are larger, 0.31 (respectively, 0.34). With an increase in the number of observations the effect of the initial FR values decreases and the assessed values come closer to the actual preconfigured FR values (shown next). However, *strict* convergence is not achieved since there is always some degree of uncertainty with regards to whether or not other factors (e.g., wireless effects) contributed to transaction/operation failures.



(a) 20 transactions



(b) 40 transactions

Figure 3.8: More observations lead to higher accuracy (Nodes 14 and 31 are non-responsive relays. Other nodes are responsive).

Accuracy vs number of observations: The number of e2e transactions between users affects the accuracy of estimation of both the forwarding FR as well as the e2e FR.

Considering the same set up as above, and preconfiguring nodes 14 and 31 to be non-responsive relays we run our experiments for a larger period ($\approx$ 10000 seconds), enough to perform up to 40 transactions pairwise. Fig. 3.8 presents the estimated FR values with 20 and 40 pairwise transactions. As one might expect, with more transactions (and thus, more observations), the assessed FR values are closer to the *actual* preconfigured ones for both responsive and unreliable nodes.

Overhead comparison of the flood based and lightweight propagation protocols: We compare the two propagation protocols in terms of the induced overhead; we also look at the accuracy achieved (in terms of the *distance* between the assessed and the preconfigured FR values i.e., the ground truth). We see from Table 3.1 that as expected flooding results in smaller uncertainty. However, the mean distances from the ground truth are very similar with both schemes.

It is also evident that the lightweight propagation results in about a 37% decrease in the induced overhead. We believe that this is a significant reduction, at the expense of a slightly higher inaccuracy and uncertainty.

Interactions between routing and FR establishment/propagation protocols: Next we want to study the impact of different routing protocols on the evolution of the FR values using our

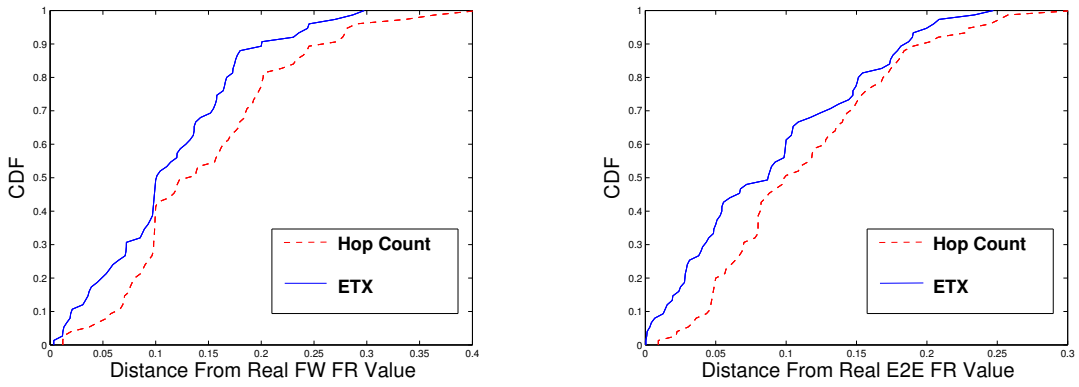
	Traffic Load(Bytes)	Distance	Uncertainty
Flooding	12387191	0.0945	0.121
Lightweight	7751196	0.1168	0.222

Table 3.1: Comparing flood based and lightweight propagation.

lightweight protocol. As observed in our first set of experiments, the presence of non-responsive relays can affect the establishment of the e2e FR values when ETX routing is used. The routes do not account for the forwarding FR of nodes and hence, the presence of *bad* relays on a path can cause transactions to fail. This consequently results in a reduction in the accuracy of the assessed e2e FR.

We perform a large set of experiments where each node is preconfigured with randomly chosen FR tuples (for forwarding and e2e FR). We ensure that these FR values are evenly spread across $[0, 1]$. We use 2 different routing metrics to find routes, minimum hop count and ETX. Running 10 repetitions of our experiments for 5000 seconds each, we obtain the results in Fig. 3.9. These figures depict the CDF of the distance between the preconfigured (*real*) and the assessed average FR values for the nodes. We observe that in all scenarios, minimum hop distance performs the worst in terms of accuracy due to long unreliable links that contribute to high uncertainty. A routing metric that not only accounts for the link qualities (e.g., ETX), but also for the forwarding responsiveness of the relays can further improve the assessment accuracy. Designing such a metric is beyond the scope of our study and is left for future work.

On the hardness of convergence: We observe from Fig. 3.9 that in the best case, almost 40% of our assessments differ by at least 0.1 from the ground truth in terms of both the forwarding and e2e FR. It is really hard, if not impossible, to achieve *strict convergence* to the real FR values. There are several reasons that contribute to this hardness. As seen earlier, the forwarding FR affects the e2e FR (Fig. 3.5). Failures due to forwarding attackers, will influence the assessed e2e FR. As our experiments indicate (omitted due to space constraints) the same happens when the transactions fail due to wireless induced effects. In addition, as discussed earlier, the initial value affects the



(a) Forwarding FR

(b) e2e FR

Figure 3.9: CDF of the distance between the assessed and real FR.

assessment as well. Finally, gossiping adds uncertainty and decreases the accuracy of the assessed FR. So even for a completely (e2e) responsive node the assessment engine cannot converge to the value of 1. We believe that this level of accuracy however, is sufficient in most cases when nodes make coarse grained assessments to hypothesize about the reliability of peers.

3.6 Conclusions

We design a framework for collaborative FR assessment in wireless networks. Unlike in prior work, we account for wireless induced factors and the reliability of intermediary relays. The framework accounts for both direct interactions between nodes and indirect feedback obtained from gossipers about other nodes in the network. It consists of a lightweight evidence propagation scheme that carefully filters out duplicate evidence. Our evaluations on an indoor/outdoor wireless testbed show that each node is able to estimate the FR values for other nodes with a sufficiently high accuracy.

Chapter 4

Trading Off Distortion for Delay for Video Transmissions in Wireless Networks

The end-user experience in viewing a video depends on the distortion; however, also of importance is the delay experienced by the packets of the video flow since it impacts the timeliness of the information contained and the playback rate at the receiver. Unfortunately, these performance metrics are in conflict with each other in a wireless network. Packet losses can be minimized by perfectly avoiding interference by separating transmissions in time or frequency; however, this decreases the rate at which transmissions occur, and this increases delay. Relaxing the requirement for interference avoidance can lead to packet losses and thus increase distortion, but can decrease the delay for those packets that are delivered. In this chapter, we investigate this trade-off between distortion and delay for video. To understand the trade-off between video quality and packet delay, we develop an analytical framework that accounts for characteristics of the network (e.g. interference, channel variations) and the video content (motion level), assuming as a basis, a simple channel access policy that provides flexibility in managing the interference in the network. We validate our model via extensive simulations. Surprisingly, we find that the trade-off depends on the specific features of the video flow: it is better to trade-off high delay for low distortion with fast motion video, but not with slow motion video. Specifically, for an increase in PSNR (a metric that

quantifies distortion) from 20 to 25 dB, the penalty in terms of the increase in mean delay with fast motion video is 91 times that with slow motion video. Our simulation results further quantify the trade-offs in various scenarios.

4.1 Introduction

Video communications have become much more popular and prevalent today due to two factors. First, wireless devices have become much more sophisticated (e.g., tablets); second, there are many video applications such as YouTube and streaming services (e.g., Amazon Instant Video) that have become immensely popular. Not only is video of interest in the commercial world, it is also of importance in other contexts such as disaster recovery, tactical networks, and surveillance.

The user experience with respect to a video flow depends on the experienced distortion. The end-to-end video distortion is affected by both the encoding process at the source and the wireless channel induced errors and interference. Also of importance is the packet delay experienced in transferring a video clip; for example, a rescue operation in disaster recovery may depend on the timely delivery of the information. The play back at a receiver could get affected due to large delays experienced by packets (and thus, video frames) [53].

Unfortunately, the two performance metrics viz., distortion and packet delay are in conflict with each other. The channel access mechanism at the link layer affects both of these performance metrics. Among the plethora of such mechanisms, there are those that minimize interference by dispersing transmissions in the frequency or time domain at the cost of higher packet delay. On the other hand, there are access mechanisms that allow the concurrent usage of the channel by multiple transmitters, to decrease packet delay at the expense of potential higher interference. Our goal is to capture the impact of interference management by an access mechanism on the trade-off between packet delay and distortion in wireless multi-hop networks.

Towards this end, we consider a simple channel access scheme (described later) that allows us to do so. Specifically, it allows us to develop a tractable analytical framework that computes the

expected values of the video distortion and packet delay experienced in a video flow, taking into account the characteristics of the channel (e.g. interference) and the video content (motion level). In brief, the scheme is characterized by an "access probability" that represents the likelihood with which a node transmits packets on the shared medium. In order to be able to represent the whole gamut of channel access mechanisms, we introduce a scalar parameter which we call *aggressiveness* (α); this is used to tune the channel access probability to the medium. A low value of α (≤ 1) corresponds to a case with low interference wherein the nodes access the channel in such a way as to avoid collisions; the distortion in this case is low. However, the delay is high. If α is high, the delay is lowered. At the same time however, the probability of a collision and therefore of a packet loss increases, thereby resulting in higher video distortion. We provide more details on the channel access model later in the chapter.

Our key finding in this work is that the characteristics of the video traffic and in particular, the motion level (described below) affects the distortion versus delay trade-off. The characteristics of the video traffic depend on the video content and the encoder that is used. The motion level of a video clip can be computed through appropriate detection algorithms; typically these algorithms classify a video clip as a fast motion or a slow motion video. In order to capture the effect of the motion level on the distortion vs delay trade-off, we introduce a second parameter in our model, which we call *sensitivity* (s). The sensitivity represents the robustness of the decoder to packet losses. When a video clip is characterized by high motion levels, the output at the encoder exhibits high variability. Therefore, transmission induced errors have a significant impact on the quality of the decoded signal, since it is more difficult to compensate for the lost information. In such cases, the robustness of the decoder to packet losses is small and consequently the sensitivity to such losses is high. The contrary holds for slow motion level video clips.

In summary, our contributions and key findings are:

- We develop an analytical framework to capture the trade-off between distortion and timeliness. The framework computes the expected values of the video distortion and the transfer delay of a video clip while accounting for system parameters both at the lower link

level (interference, channel induced errors) and the application semantics (motion levels and structure of video content).

- We demonstrate, through extensive simulations, the validity of our analytical model. We then quantify via both analysis and simulations the distortion versus delay trade-off for different types of video flows (fast versus slow motion) in a variety of scenarios.

- Our key observation is that trading off high delay for low distortion is important for fast motion video, but not for slow motion video. We find that if the PSNR (Peak Signal to Noise ratio) requirement for a video clip is increased from 20dB to 25dB, the fast motion video clip suffers from a delay increase penalty that is 91 times higher than the penalty incurred with slow motion video. This shows that slow motion video is able to better tolerate packet losses than fast motion video and thus, should be handled differently in a wireless network.

Our work in perspective: Our work demonstrates that application semantics determine the appropriateness of specific protocols in a wireless network. Specifically, we show that interference has a much more significant effect on fast motion video as compared to slow motion video. Channel access schemes that account for this can drastically improve the performance of video flows. One could also envision transmission of fast motion video on congestion free paths to ensure reliability.

Our analytical framework not only provides an understanding of the distortion versus delay trade-off for video flows, but also provides a quick way of obtaining performance results. Simulations on the other hand, take much longer time to provide the same results. To the best of our knowledge, this is the first analysis that characterizes the distortion versus delay trade-offs for video flows in wireless networks.

Organization: In Section 4.2 we present related work. The channel access scheme considered for our analysis is described in Section 4.3. Our analysis is detailed in Section 4.4. In Section 4.5 we validate our analysis via simulations and discuss the main implications of our results. We conclude in Section 4.6.

4.2 Related Work

Standards like the MPEG-4 [54] and the H.264/AVC [55] provide guidelines on how a video clip should be encoded for transmission over a communication system. Predictive source encoding where motion estimation and motion compensation play important roles in the process, is very popular for video compression. Typically, each video clip is separated into a repetitive structure called a Group of Pictures (GOP). Each GOP consists of I, P and B frames (B frames are optional). An I-frame can be decoded independently of any other information within the GOP and each of the P-frame or B-frame use the I-frame as a reference to encode information [53]. In the following, we assume an $IPP \ldots P$ encoding structure for each GOP.

Handling missing frames is critical for the decoder since frame losses affect the video quality perceived by the end user. Typically, the last decoded frame is substituted for a frame that is lost at the receiver [53, 56]. With this method, the difference between the substitute frame and the original (lost) frame determines the error and hence, the video quality perceived by the user. We discuss this further when we present our analysis and articulate the relationship between the Peak-Signal-to-Noise-Ratio or PSNR (a common metric to assess video quality) and the distance between the missing and substitute frames.

While protocols have been designed towards trying to achieve an optimal trade-off between video quality and delay in wireless networks, there has not been a formal analytical assessment of this trade-off; these protocols do not account for video semantics. In [57], a priority-based video stream scheduling algorithm which considers channel conditions, frame types and traffic burstiness is proposed. In [58] the authors propose a cross layer error control scheme for Fine Granularity Scalable (FGS) video transmission in an IEEE 802.11a network; the scheme computes the optimal combination of modulation and the FEC rate for a video flow. In [59], a cross layer architecture is designed for multicast streaming in wireless LANs. These efforts have only considered single hop wireless networks and to reiterate, have not assessed the impact of video semantics (fast versus slow motion video) on this trade-off.

The impact of packet losses and delay on the video quality depends on the motion level in the video clip. Video motion detection algorithms can be used during the video encoding process (e.g., [60]).Tools such as PhysMo [61] and AForge [62] can also be used to determine the motion level in a video clip.

4.3 Channel Access System Model

There is a trade-off between the video quality and the timely delivery of a video clip from a source to a destination node. Spreading parallel transmissions in the frequency or time domain eliminates collisions and can provide the best video quality (since the packet losses are then minimized), but may increase the delay in transferring the video content. Many approaches have been designed towards avoiding collisions; these include scheduled access schemes, random access schemes with carrier sensing and exponential backoffs etc.

On the other hand, allowing concurrent transmissions, reduces the delay but increases packet losses due to collisions and interference, and therefore introduce higher video distortion. Under high load, some random access schemes exhibit this behavior. In order to explore the space between mechanisms that manage interference in different ways, we consider a simple channel access scheme that is described below. With this scheme, by tuning a parameter (*aggressiveness*), we are able to roughly characterize the whole gamut of channel access mechanisms. Importantly, the scheme allows us to characterize the trade-off between distortion and packet delay analytically.

We consider a slotted time system, where all the nodes are assumed to be synchronized. We set the slot duration to be equal to the transmission time of the largest allowable packet in the system. At the beginning of each time-slot, each node decides on whether or not it will access the channel in that slot, based on an access probability p_a, given by:

$$p_\alpha = \alpha \cdot p_r, \tag{4.1}$$

where, p_r is a reference access probability and α is the *aggressiveness*. The reference probability p_r is equal to the frequency with which a node accesses the channel when a perfect schedule is used. In such a case, based on the topology of the network, each node accesses the channel periodically. If this period is n (typically equal to the maximum of the sizes of the cliques [63] which the node belongs to, as discussed later), then the reference probability is:

$$p_r = \frac{1}{n}. \tag{4.2}$$

In the simple case where the network topology is a single clique (all nodes can directly communicate with each other), n is equal to the clique size. In general however, a node may belong to a plurality of cliques of different sizes. Computing a transmission schedule for nodes that belong to cliques of different sizes is NP-hard. A perfect schedule guarantees collision free transmissions for each node at the expense of long packet delays (e.g., see [64]).

When $\alpha = 1$, $p_a = p_r$ and access is as per the reference scheme. However, note that since the access is probabilistic, collisions could still occur. When $\alpha > 1$, a node accesses the channel more aggressively compared to the reference scheme. This results in smaller packet delays but also in increased interference and a higher probability of packet collisions. For $0 < \alpha < 1$, the node accesses the channel less frequently; here, the packet delay increases compared to the reference case but packet collisions are avoided with high probability.

Consider a simple case where two nodes that are one hop away belong to different cliques. Let the size of the clique that the first node belongs to, be n_1 and the size of the clique that the second node belongs to, be $n_2 < n_1$. Here, if a perfect schedule is implemented the first node will access channel much less frequently as compared to the second node. Similarly with the reference approach, the exchange of packets between these two nodes is slowed down by the first clique, since

$$p_{r,1} = \frac{1}{n_1} < \frac{1}{n_2} = p_{r,2}. \tag{4.3}$$

Increasing the aggressiveness of the first node by setting $\alpha > 1$ causes the first node to access the

channel with probability

$$p_{\alpha,1} = \alpha \cdot p_{r,1} > p_{r,1} \tag{4.4}$$

This reduces the packet delay between the nodes, but possibly at the expense of increased collisions in the first clique (depending on the access probabilities of nodes in that clique).

Note here that p_r is particular to each node i ($p_{r,i}$); it is essentially the reciprocal of the maximum of the sizes of the cliques to which the node belongs. To compute the reference access probabilities $p_{r,i}$, maximal clique enumeration is essential. A maximal clique is a clique in which all the composers are connected to each other and there is no other clique that contains this clique. There are various algorithms that compute the maximal cliques in an undirected graph. Any approach could be used here. In our performance evaluations in Section 4.5 we use the well-known Bron-Kerbosch algorithm [65]. Computationally less complex algorithms can be used in practice at the cost of optimality (e.g., [66]).

Other assumptions: We do not consider retransmissions or higher layer protocol effects to keep the analysis and the study simple. We will consider these issues in the future work. Our work however, does not require a node to have something to send all the time. Nodes could also be simultaneously forwarding multiple video streams.

4.4 Our Analytical Framework

We develop an analytical framework to compute the expected distortion of a video flow over a multi-hop static wireless network in the presence of interference from other video connections. We first compute the expected value of the Signal-to-Interference-and-Noise-Ratio (SINR) which we use to find the packet success rate. We then translate packet losses to video frame losses and hence to video transmission distortion.

4.4.1 SINR Computation

Consider the communication between a pair of nodes i and j that are one hop away. The SINR for this communication is:

$$\text{SINR} = \frac{P_{ij}}{N + \sum_{k \in \mathcal{I}} P_{kj}}, \tag{4.5}$$

where, N is the noise power, $\mathcal{I}$ is the set of interfering nodes and P_{sr} is the received power at node r of a signal transmitted by node s, for any pair of nodes s, r. Using the Friis propagation loss model [67], the received power P_{sr} can be written as:

$$P_{sr} = \frac{P_s G_s G_r}{L} \cdot \left(\frac{\lambda}{4\pi}\right)^2 \cdot \frac{1}{d_{sr}^2}, \tag{4.6}$$

where P_s is the transmission power, G_s and G_r are the transmission and reception gains respectively, λ is the wavelength of the signal, d_{sr} is the distance between the sender s and the receiver r and L is the system loss. We assume that all nodes use the same transmission power P, i.e. $P_s = P$ for all s.

To compute the expected value of the SINR we assume that R is the maximum communication range. If a uniform node distribution is assumed, the probability density function of the distance is given by:

$$f_d(t) = \frac{2t}{R^2}, \quad 0 < t < R. \tag{4.7}$$

The expected value of the SINR can then be computed as:

$$\text{E[SINR]} = \sum_{l=0}^{m} \text{E[SINR} \mid l \text{ interferers]} \cdot P\{l \text{ interferers}\}, \tag{4.8}$$

where we assume that the number of neighbors is at most m.

The number of interferers follows a binomial distribution with parameters m and p_α:

$$P\{l \text{ interferers}\} = \binom{m}{l} p_\alpha^l \, (1-p_\alpha)^{m-l}, \; l = 0, 1, \ldots m, \tag{4.9}$$

where p_α is given by (4.1). In general, the network topology may be such that the neighbors of the receiving node belong to cliques of different sizes, and therefore access the channel with different probabilities. In that case, we can compute the probability that l interferers exist, in (4.9), considering $p_\alpha^{\min}$ and $p_\alpha^{\max}$ as the minimum and maximum values, respectively, of the access probabilities from among the neighbors of the receiver. Using these values for the access probability, we can determine lower and upper bounds for the E[SINR].

Since SINR takes positive values, its conditional expected value given the number of interferers can be computed as:

$$\begin{aligned} &\text{E[SINR} \mid l \text{ interferers]} \\ &\qquad = \int_0^\infty P\{\text{SINR} > x \mid l \text{ interferers}\} \, dx \end{aligned} \tag{4.10}$$

Using (4.5) and (4.6), the conditional tail distribution of SINR can be written as:

$$\begin{aligned} &P\{ \text{ SINR} > x \mid l \text{ interferers}\} \\ &= P\left\{ \frac{\left(\frac{1}{d_{ij}}\right)^2}{\frac{N \cdot L \cdot (4\pi)^2}{P \cdot G_t \cdot G_r \cdot \lambda^2} + \sum_{k=1}^{l} \left(\frac{1}{d_{kj}}\right)^2} > x \mid l \text{ interferers} \right\} \\ &= \underbrace{\int_0^R \cdots \int_0^R}_{l \ times} P\left\{ d_{ij} < \left(\frac{xNL(4\pi)^2}{PG_tG_r \cdot \lambda^2} + \sum_{k=1}^{l} t_k^{-2} \right)^{-\frac{1}{2}} \right. \\ &\qquad \left. \mid l \text{ interferers} \right\} \cdot f(t_1, \ldots, t_l) \, dt_1 \ldots dt_l, \end{aligned} \tag{4.11}$$

where $f(\cdot,\cdot,\ldots,\cdot)$ is the joint probability density function of the distances from the l interferers to the destination node j. Using (4.7) and assuming statistical independence between these distances we have that:

$$f(t_1,\ldots,t_l)=\prod_{k=1}^{l}\frac{2t_k}{R^2}=t_1\cdots t_l\cdot\left(\frac{2}{R^2}\right)^l. \tag{4.12}$$

Using (4.7) and (4.12), (4.11) becomes

$$P\{\text{ SINR}>x\mid l\text{ interferers}\}$$

$$=\left(\frac{2}{R^2}\right)^{l+1}\underbrace{\int_0^R\cdots\int_0^R}_{l\ times}\int_0^{U(x)} t\cdot t_1\cdots t_l\,dt\,dt_1\ldots dt_l, \tag{4.13}$$

where $U(x)=\left(\frac{xNL(4\pi)^2}{PG_tG_r\cdot\lambda^2}+x\sum_{k=1}^{l}t_k^{-2}\right)^{-\frac{1}{2}}$. From (4.10) and (4.13) we have:

$$\text{E}\left[\text{SINR}\mid l\text{ interferers}\right]$$

$$=\left(\frac{2}{R^2}\right)^{l+1}\int_0^{\infty}\underbrace{\int_0^R\cdots\int_0^R}_{l\ times}\int_0^{U(x)} t\cdot t_1\cdots t_l\,dt\,dt_1\ldots\,dt_l\,dx. \tag{4.14}$$

From (4.8), (4.9), and (4.14), we get

$$\text{E}\left[\text{SINR}\right]=\sum_{l=0}^{m}\Bigg\{\binom{m}{l}p_\alpha^l\,(1-p_\alpha)^{m-l}$$

$$\times\left(\frac{2}{R^2}\right)^{l+1}\int_0^{\infty}\underbrace{\int_0^R\cdots\int_0^R}_{l\ times}\int_0^{U(x)} t\cdot t_1\cdots t_l\,dt\,dt_1\ldots\,dt_l\,dx\Bigg\}, \tag{4.15}$$

which we solve numerically. We point out here that we have the tail distribution for SINR, the outermost integral from 0 to ∞ converges fairly quickly i.e., as x increases $P\{SINR>x\}\to 0$.

4.4.2 Packet Success Rate

To compute the packet success rate we consider the IEEE 802.11b physical layer for high-rate (11Mbps), where complementary code keying (CCK) is adopted. We consider this version of the protocol since the analysis in [68, 69], directly provides the packet success rate; note that it is easy to incorporate other versions of 802.11 (a, g), by considering the appropriate modulation/encoding schemes. The packet success rate, p_s, as a function of the expected value S of the SINR is:

$$p_s(S) = [1 - P_{e,1}(S)]^B \tag{4.16}$$

where

$$P_{e,1}(S) = 1 - \left(\int_{-\sqrt{2S}}^{\infty} \left[2\Phi\left(x + \sqrt{2S}\right) - 1 \right]^{(N-1)} \times \frac{\exp\left(-x^2/2\right)}{\sqrt{2\pi}}\, dx \right)^2 . \tag{4.17}$$

In the above, B is the length of the packet in bits, $N = K/2$, K is the number of biorthogonal signals used and $S = \mathrm{E}[\mathrm{SINR}]$. Equations (4.16) and (4.17) provide a mapping from the expected value of the SINR to the packet success rate p_s.

4.4.3 Video Frame Success Rate

We map the packet success rate p_s to the video frame (referred to as simply 'frame') success rate P_f, which denotes the probability a frame is successfully transmitted over one hop. As was mentioned in Section 4.2, we assume that each GOP has an *IPP*. . . *P*-structure.

If n is the number of packets in each frame, then to successfully decode a frame, (a) the first packet of that frame needs to be successfully received, and (b) $0 \leq s \leq n-1$ of the remaining

$n-1$ packets need also be received successfully. The success probability of a frame is given by:

$$P_f = p_s \sum_{i=s}^{n-1} \binom{n-1}{i} p_s^i (1-p_s)^{n-1-i}. \tag{4.18}$$

We call the parameter s the *sensitivity* of the decoder to packet losses. It is the minimum number of packets that the decoder needs to receive without errors in order to decode the corresponding frame correctly. As discussed in Section 4.5, the sensitivity is associated with the video content itself and specifically with the motion level. When a video clip is characterized by high (or fast) motion, the sensitivity s has a higher value compared to a low (or slow) motion video. This is because in a high motion video clip, the difference between successive frames in the GOP structure is large and the loss of a frame has a higher impact on the overall video quality.

In general, the *I*-frame is much larger than a *P*-frame. Thus, the number of packets in the *I* and *P* frames differ. As a result, the frame success probabilities for an *I* and a *P*-frame also differ. We denote by P_I the success probability of an *I*-frame and by P_P the corresponding success probability of a *P*-frame.

We have validated this model via extensive experiments using the EvalVid tool.

4.4.4 Distortion

Let the GOP structure contain $G-1$ *P*-frames that follow the *I*-frame. We consider predictive source coding where, if the i^{th} frame is the first lost frame in a GOP, then the i^{th} frame and all its successors in the GOP are replaced by the $(i-1)^{st}$ frame at the decoder. If the *I*-frame of the GOP cannot be decoded correctly, then the whole GOP is considered unrecoverable and is ignored. In this case, these lost video frames are replaced by the most recent frame from a previous GOP that is correctly received. In all cases, the similarity between the missing frames and the reference frame (substitute frame) affects the distortion [70].

We compute the video distortion as the *mean square error* of the difference between the missing frame and the substitute frame. Initially, we focus on a one-hop transmission and then extend our

analysis for multi-hop connections. Specifically, we have the following cases:

Case 1 – Intra-GOP distortion: The *I*-frame of the current GOP is successfully received. The distortion for the current GOP depends on which, if any, of the *P*-frames of the GOP cannot be decoded without errors. If the first unrecoverable *P*-frame is the i^{th} frame in the GOP, the corresponding distortion is given by [56]:

$$d_i = (G-i)\frac{i \cdot G \cdot d_{min} + (G-i-1) \cdot d_{max}}{(G-1) \cdot G}, \tag{4.19}$$

for $i = 1, 2, \ldots, (G-1)$, where $d_{\max}$ is the maximum distortion when the first frame is lost and d_{min} is the minimum distortion when the last frame is lost. The values of $d_{\max}$ and $d_{\min}$ depend on the actual video content and have to be evaluated experimentally. The probability P_i that the i^{th} frame is lost is

$$P_i = P_I P_P^{i-1}(1-P_P), \quad i = 1, 2, \ldots, (G-1). \tag{4.20}$$

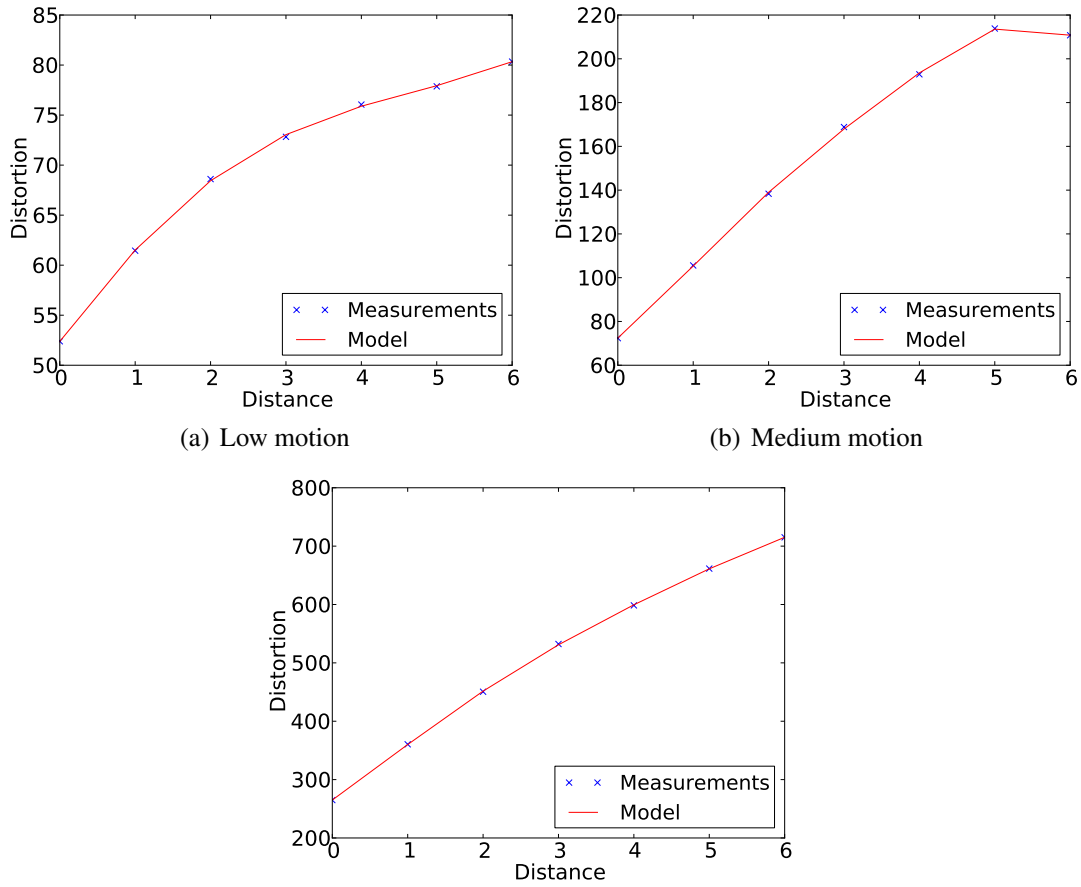
Using (4.19) and (4.20), the expected value of the distortion can be computed to be:

$$D^{(1)} = \sum_{i=1}^{G} d_i \cdot P_i \tag{4.21}$$

Case 2 – Inter-GOP distortion: The *I*-frame of the current GOP is lost and a frame from a previous GOP is used as the reference frame. In this case, the difference between the reference frame and the missing frames determine the distortion.

Similar to the work in [70], we expect to see the motion characteristics of the video affecting the distortion. To capture the dependence of the inter-GOP distortion on the motion level of the video we perform a set of experiments and we use the collected results to statistically describe this association.

Specifically, we select a set of video clips from [71] and categorize them into three groups



(a) Low motion

(b) Medium motion

(c) High motion

Figure 4.1: Average distortion with distance.

according to their motion level: low, medium and high. All video clips have 300 frames each, with a frame rate of 30 frames per second. We use FFmpeg [72] to convert the video clips from the initial, uncompressed YUV format to the MP4 format. Then, we artificially create video frame losses in order to achieve reference frame substitutions from various distances. Finally, we use the Evalvid toolset [73] to measure the corresponding video distortion.

In Fig. 4.1 the dependence of the average distortion on the distance between the missing frame and the substitute is shown for the three categories. In order to use these empirical results in other experiments, we approximate the observed curves with polynomials of degree 5 using a multinomial regression. In particular, we define the approximate distortion $D^{(2)}$ as a function of the distance d:

$$D^{(2)}(d) = a_5 d^5 + a_4 d^4 + a_3 d^3 + a_2 d^2 + a_1 d^1 + a_0 d^0 \tag{4.22}$$

and compute the coefficients $a_0, \ldots, a_5$, through the regression.

Case 3 – Initial GOP: The *I*-frame of the current and all previous GOPs (including the first GOP) are lost. In this case the distortion D is maximized. If $\{D_{\max}^{(1)}, D_{\max}^{(2)}, \ldots, D_{\max}^{(\|\mathcal{G}\|)}\}$, where $\mathcal{G}$ is the set of all GOPs in the video clip, is the set of the maximum distortion values in all GOPs, then

$$D^{(3)} = \max_{k \in \mathcal{G}} D_{\max}^{(k)}. \tag{4.23}$$

4.4.5 Single-hop Transmission

Suppose the video clip has N GOPs and each GOP consists of an *I*-frame followed by $G-1$ *P*-frames. For each GOP of the video clip define the state $S_i, i = 1, 2, \ldots, N$ such that

$$S_i \in \mathcal{S} = \{0, 1, 2, \ldots, (G-1), G\}. \tag{4.24}$$

The state S_i for the i^{th} GOP indicates which is the first unrecoverable frame in that GOP. Specifically,

$$S_i = \begin{cases} 0, & \textit{I}\text{-frame is lost,} \\ k, & k^{th}\ \textit{P}\text{-frame is lost, } 1 \le k \le (G-1), \\ G, & \text{none of the frames is lost,} \end{cases} \tag{4.25}$$

for $i = 1, 2, \ldots, N$. The initial state for each GOP is G. The transition probability $p_i(G, q)$ of the state S_i from G to $q \in \mathcal{S}$ is

$$p_i(G,q) = \begin{cases} 1 - P_I, & q = 0, \\ P_I P_P^{k-1}(1 - P_P), & q = k, 1 \leq k \leq (G-1), \\ P_I P_P^{G-1}, & q = G, \end{cases} \tag{4.26}$$

for $i = 1, 2, \ldots, N$.

To compute the expected value of the distortion for the transmission of the video clip over the wireless channel we need to consider the states of all the GOPs. We define the vector $\boldsymbol{S}$ as

$$\boldsymbol{S} = (S_1, S_2, \ldots, S_N) \in \mathcal{S} \times \mathcal{S} \times \cdots \times \mathcal{S}. \tag{4.27}$$

The initial state of $\boldsymbol{S}$ is $\boldsymbol{G} = (G, G, \ldots, G)$ and its transition probability $p(\boldsymbol{G}, \boldsymbol{q})$ to a new state $\boldsymbol{q} = (q_1, q_2, \ldots, q_N)$ is

$$p(\boldsymbol{G}, \boldsymbol{q}) = \prod_{i=1}^{N} p_i(G, q_i). \tag{4.28}$$

The overall distortion for the video clip transmission depends on the final state $\boldsymbol{q}$. As was discussed earlier in Case 2, the distortion of a GOP may depend not only on the frame losses in that GOP but on losses in previous GOPs as well. Therefore, if D_i is the distortion of the i^{th} GOP, it is a function of the vector $\boldsymbol{q}$ and not only of the i^{th} component of $\boldsymbol{q}$. We define the random variable $\boldsymbol{D}(\boldsymbol{q})$ as:

$$\boldsymbol{D}(\boldsymbol{q}) = (D_1(\boldsymbol{q}), D_2(\boldsymbol{q}), \ldots, D_N(\boldsymbol{q})) \tag{4.29}$$

consisting of the distortions of each of the GOPs of the video clip. Using (4.28) we have:

$$\mathrm{E}[\boldsymbol{D}] = (\mathrm{E}[D_1], \mathrm{E}[D_2], \ldots, \mathrm{E}[D_N]) = \sum_{\boldsymbol{q}} p(\boldsymbol{G}, \boldsymbol{q}) \boldsymbol{D}(\boldsymbol{q}) \tag{4.30}$$

The average distortion that corresponds to the video file is

$$\bar{D} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{E}[D_i]. \tag{4.31}$$

4.4.6 Multi-hop Transmission

For the case of a multi-hop transmission we need to compute the transition probability from the initial state $\boldsymbol{G} = (G, G, \ldots, G)$ at the source node to a final state $\boldsymbol{q} = (q_1, q_2, \ldots, q_N)$ at the destination node. To do this we first compute the transition probability $p_i^{(j)}(n, m)$ of the i^{th} GOP (the likelihood that the first unrecoverable frame is m after this hop, given that it was n upon reaching this hop) at the j^{th} hop along the path from the source to the destination.

$$p_i^{(j)}(n,m) = \begin{cases} 0, & \text{if } m > n, \\ 1, & \text{if } m = n = 0, \\ P_I P_P^m, & \text{if } m = n \text{ and } n > 0, \\ 1 - P_I, & \text{if } m = 0 \text{ and } n > 0, \\ P_I P_P^{m-1}(1 - P_P), & \text{if } 0 < m < n, \end{cases} \tag{4.32}$$

for $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, k$, where N is the total number of GOPs and k is the number of hops along the path.

Then, the transition probability for the j^{th} hop is :

$$p^{(j)}(\boldsymbol{n}, \boldsymbol{m}) = \prod_{i=1}^{N} p_i^{(j)}(n_i, m_i). \tag{4.33}$$

for $\boldsymbol{n} = (n_1, n_2, \ldots, n_N)$ and $\boldsymbol{m} = (m_1, m_2, \ldots, m_N)$, where n_i is the current state of the i^{th} GOP and m_i is the next state of the i^{th} GOP.

For the multi-hop transmission, the transition probability $p_{\text{mu}}(\boldsymbol{G}, \boldsymbol{q})$ from the initial state $\boldsymbol{G} = (G, G, \ldots, G)$ at the source node to a final state $\boldsymbol{q} = (q_1, q_2, \ldots, q_N)$ at the destination node, is given by

$$p_{\text{mu}}(\boldsymbol{G}, \boldsymbol{q}) = \sum_{\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots \boldsymbol{q}_{k-1}} p^{(1)}(\boldsymbol{G}, \boldsymbol{q}_1) p^{(2)}(\boldsymbol{q}_1, \boldsymbol{q}_2) \cdots p^{(k)}(\boldsymbol{q}_{k-1}, \boldsymbol{q}) \tag{4.34}$$

As is the case for the single hop transmission, the overall distortion depends on the final state $\boldsymbol{q}$. Therefore, the average distortion $\bar{D}$ can be computed using (4.29), (4.30) and (4.31), where instead of using the one-hop transition probability $p(\boldsymbol{G}, \boldsymbol{q})$ we use the multi-hop transition probability $p_{\text{mu}}(\boldsymbol{G}, \boldsymbol{q})$.

4.4.7 Mapping Distortion to PSNR

In all the results we present in the sequel, we use the Peak Signal-to-Noise Ratio (PSNR) which is an objective video quality measure [53]. The relationship between distortion and PSNR (in dB) is given by [53]:

$$PSNR = 10 \log_{10} \frac{255}{\sqrt{\text{Distortion}}} \tag{4.35}$$

4.4.8 Delay

For the single hop communication, where a node accesses the channel with probability p_α give by (4.1), the average delay of a packet is given as the expected value of a geometric distribution

GOP Size	15
Frame Size	CIF (352×288)
Frames per second	30
MTU	1000

Table 4.1: Video Encoding Parameters

with parameter p_α:

$$\text{E[Delay]} = \text{time\_slot\_duration} \cdot \frac{1}{p_\alpha}. \tag{4.36}$$

In the multi-hop case, a packet successfully received by the destination traverses each intermediate hop. At each hop, the delay is given by (4.36). Hence, the overall delay is:

$$\text{E[Delay]} = \text{time\_slot\_duration} \cdot \sum_i \frac{1}{p_\alpha^{(i)}}, \tag{4.37}$$

where $p_\alpha^{(i)}$ is the access probability at the i^{th} hop. This can be directly used to compute the delay incurred in transferring the entire video clip.

Key Observations: It is evident from (4.37) that as p_α increases, the delay decreases. As a consequence however, the likelihood of an increased interference (see (4.9)) and thus, frame loss increases. However, for a given frame loss rate, the distortion depends on the sensitivity (see Section 4.4.3). For slow motion video, a given value of the frame loss rate results in lower distortion than for fast motion video. Thus, for a given distortion requirement, it is possible to achieve a lower delay through more aggressive scheduling for slow motion video as compared to fast motion video.

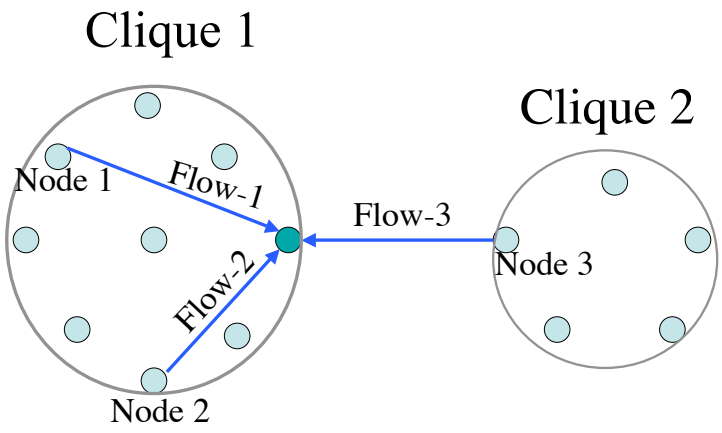


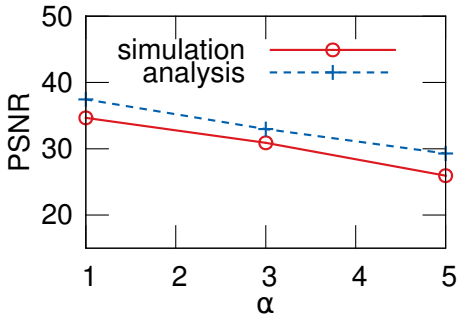
Figure 4.2: Example network topology.

4.5 Evaluations

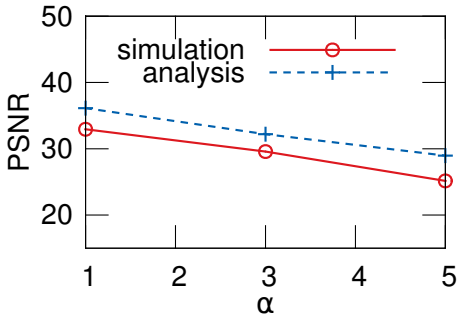
In this section, we seek to quantify the trade-off between distortion and delay. We also validate our analytical model via extensive simulations. The analytical results take an order of magnitude time (minutes) less than the simulation counterparts (hours); the simulation is complicated by the presence of both the network functions (channel access, PHY), and the application semantics (video).

Simulation set up and metrics: We use the network simulator ns-3 [74]. We modify the IEEE 802.11 module therein, to implement our scheme. We implement a slotted-time system to control the shared medium where each node is granted access to the channel with probability $p_\alpha = \alpha \cdot p_r$. As was described in Section 4.3, the reference probability $p_r = \frac{1}{n}$ where n is the maximum clique this node belongs to. In our implementation, the network topology information is collected when the network is set up and the Bron-Kerbosch clique enumeration algorithm [65] is used to compute the maximal cliques across the network.

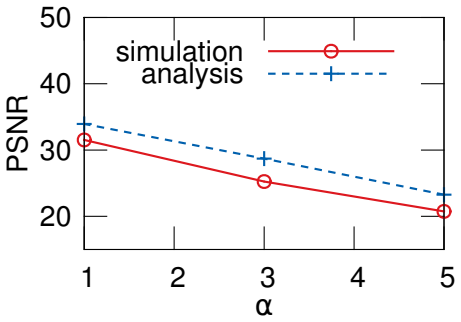
We also use EvalVid [73], which is a popular tool-set for evaluating the quality of video transmitted over a real or simulated network. The tool allows us to gather performance statistics with metrics such as the Peak-Signal-to-Noise-Ratio (PSNR) and the Mean Opinion Score (MOS) [75]. The PSNR metric compares the maximum possible signal energy to the error energy.



(a) Average PSNR for flow-1



(b) Average PSNR for flow-2

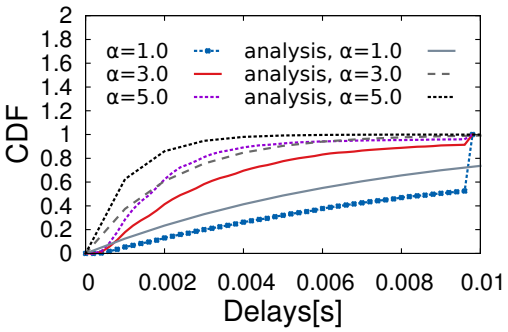


(c) Average PSNR for flow-3

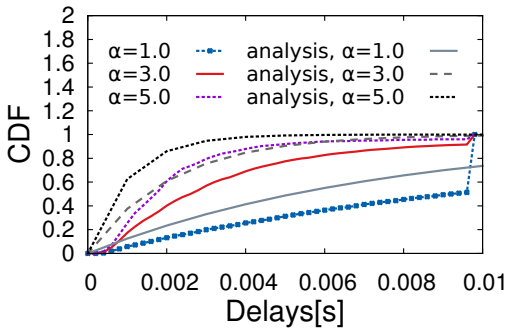
Figure 4.3: Average PSNR for a simple network topology.

The PSNR of each frame can be mapped to the MOS, which is quantified on a scale of five grades (from "bad" to "excellent"). Note that the lower the PSNR and MOS, the higher the distortion; we use these metrics since the tool we use provides these measures directly.

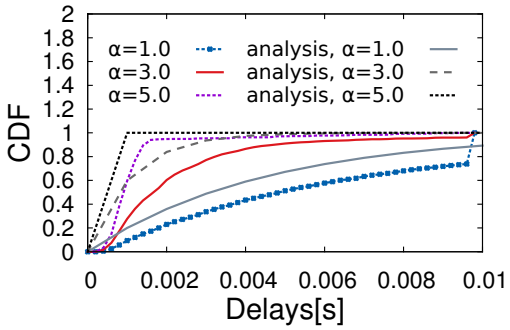
For each video flow in a simulation, we produce a sequence of files. We start with an initial video file (taken from [71]), composed of YUV frames. Using EvalVid we transform the YUV video to the MP4 format and then to the MPEG4 format. To simulate how the video would be transmitted over the real network, EvalVid provides a tool called `mp4trace` which creates a log from an attempted MPEG4 video transmission over a real network. We use this log file as an input to our ns-3 implementation. In the end, based on the log file, the EvalVid tool-set and the packet losses produced by the simulation, we reconstruct the video file as it is 'supposed to be' received at



(a) Delay distribution for flow-1



(b) Delay distribution for flow-2



(c) Delay distribution for flow-3

Figure 4.4: Average PSNR and distribution of the end-to-end delay for a simple network topology.

the destination node in a real network. Comparing the reconstructed video with the reference video file, we can measure the video quality degradation caused by the transmission over the network. Finally, note that we use the AForge tool to classify a video flow as fast or slow motion video [62].

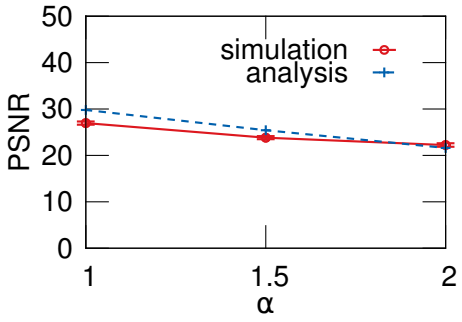
The video encoding parameters we use are in Table 4.1. We focus on three metrics: (i) the PSNR, which is an objective quality measure, (ii) the MOS, which is a subjective quality metric, and (iii) the delay distribution of each video flow.

An Example: First, we illustrate our approach by considering the simple network topology in Fig. 4.2. Since the topology is known, we refine our analysis to this specific topology (use topology specific parameters rather than estimated averages) to compute the average values of PSNR and

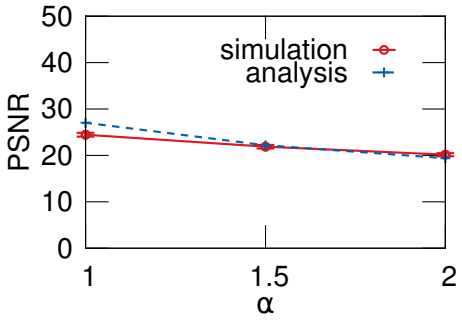
delay. We use a video clip with slow motion for this experiment. There are two maximum cliques in this network: "clique-1" of size 8 and "clique-2" of size 5. In this scenario, there are two video traffic flows originating in "clique-1" and one video traffic flow from "clique-2" both destined to the same receiver. The reference access rates of these flows are $p_{r,1} = p_{r,2} = \frac{1}{8}$, and $p_{r,3} = \frac{1}{5}$.

In Fig. 4.3 and Fig. 4.4 the average PSNR and the cumulative distribution of the end-to-end packet delay are shown for each of the three video flows for different values of the aggressiveness α. The effect of the aggressiveness α on PSNR is shown in Figs. 4.3(a), 4.3(b) and 4.3(c). As α increases, each transmitting node accesses the channel more frequently thus increasing the number of packet collisions and lowering the video quality. On the other hand, this increase in α has the opposite effect on the packet delay. As shown in Figs. 4.4(a), 4.4(b) and 4.4(c) the larger the value of α, the less the delay that each packet experiences in the network. For example, if we focus on flow-1, we notice from Fig. 4.3(a) that the average PSNR is about 35dB when $\alpha = 1.0$. The average PSNR drops to 26dB when $\alpha = 5.0$, i.e. the PSNR is decreased by 26%. On the other hand, Fig. 4.4(a) shows that when $\alpha = 1.0$, 30% of the video packets are delayed by 0.004 seconds, while 90% of the packets are delayed by 0.004 seconds when α increases to 5.0. If a PSNR value of 26dB is acceptable for a specific application, the aggressiveness α can be tuned to be equal to 5.0 to minimize the delay. However, if the video application has strict constraints regarding the video quality, a lower value of α can be used at the expense of larger delays. Finally, note that the analytical results match very well with simulation results, both for the PSNR and the delay distribution.

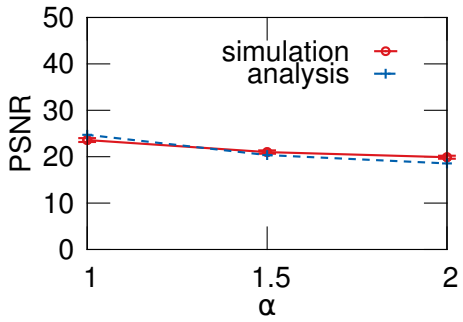
More general cases: Next, we consider general, randomly generated network topologies. In all our experiments, we focus on wireless multi-hop networks that cover a geographical area of 800×400 m^2. This area is separated into eight sub-areas, each of which is of size 200×200 m^2. In each sub-area, the nodes are distributed according to a Poisson random field. On the average there are 40 nodes in the network. Each node uses our MAC protocol stack with the ns-3 implementation of the IEEE 802.11b physical layer and the Friss propagation model [67]. The maximum transmission range is about 150 m. We run the experiments for 40 different



(a) Avg PSNR (2-hops).



(b) Avg PSNR (3-hops).

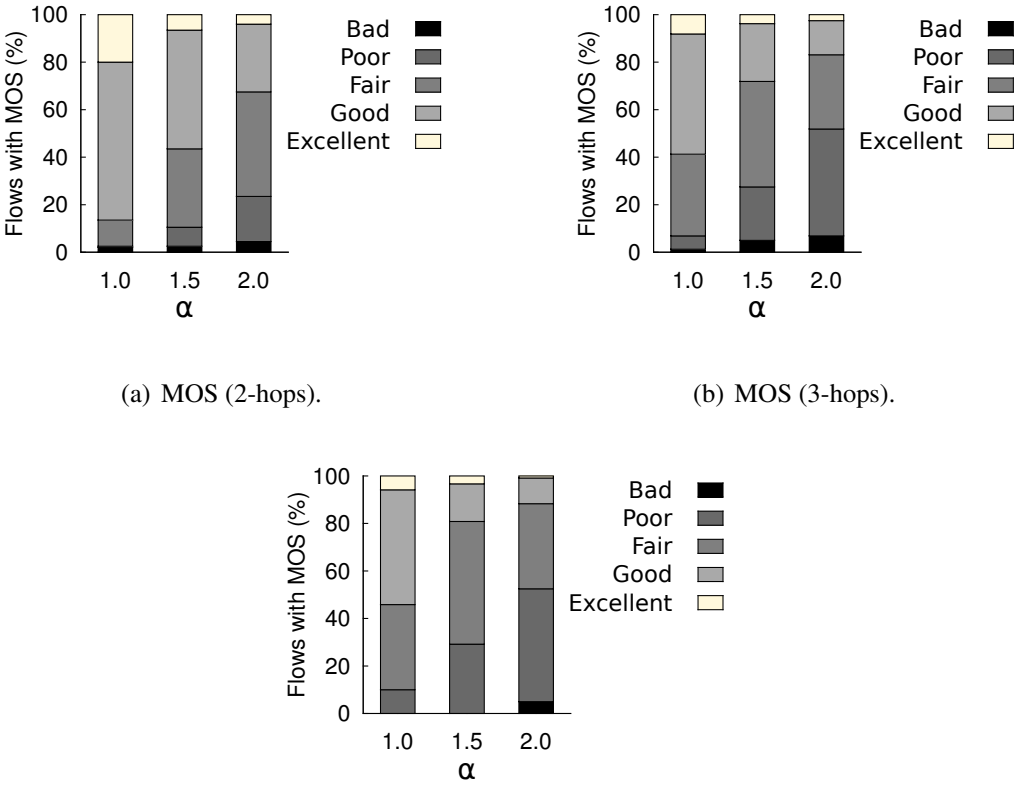


(c) Avg PSNR (4-hops).

Figure 4.5: Average PSNR for slow motion videos.

random topologies and for each video flow, we compute the average PSNR, the MOS of each flow, and the end-to-end delay. The PSNR that is computed is the average PSNR from among all the transmissions. The MOS results show the percentage of flows that fall in the five scales of MOS ratings (“Bad”, “Poor”, “Fair”, “Good”, “Excellent”). For the delay we find the cumulative distribution of the end-to-end delay for each flow.

Slow Motion Videos: The first set of experiments are for transmissions of video files with slow motion levels. We consider three different cases, where the flows consist of (i) 2 hops, (ii) 3 hops and (iii) 4 hops. We omit the results from the 1 hop case due to space constraints; the results are similar to that with the example described earlier. In each case, we create flows by choosing the source-destination pairs at random from among the nodes that satisfy in each case, our constraint



(a) MOS (2-hops).

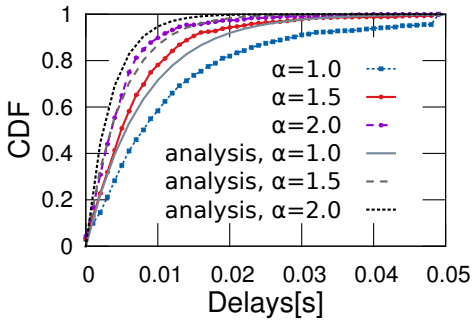
(b) MOS (3-hops).

(c) MOS (4-hops).

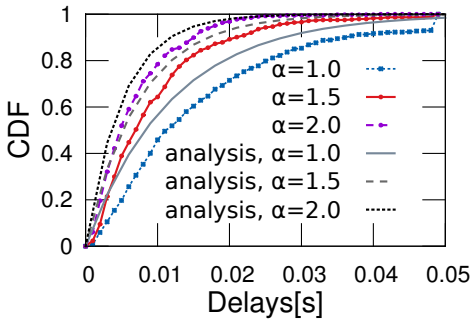
Figure 4.6: Average MOS for slow motion videos.

regarding the hop-count. To determine saturation conditions, we increase the number of flows gradually considering an one-hop scenario, with $\alpha = 1.0$. We find here that 10-12 concurrent transmissions are where the network achieves its capacity with acceptable quality for each video flow. If we consider each hop in a multi-path connection as leading to a single transmission, the total number of concurrent transmissions in the multi-hop case can be roughly estimated to be `hop_count` $\times$ `number_of_flows` (is likely to be true for the bottleneck links). Thus, to approximately reach the capacity, we use 5 flows for the 2-hop experiments, 4 flows for the 3-hop experiments and 3 flows for the 4-hop experiments.

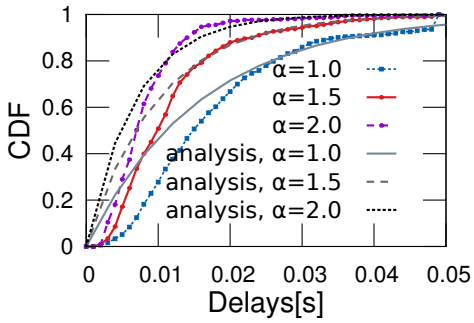
First, we notice from Fig. 4.5, Fig. 4.6 and Fig. 4.7 that the analytical model provides a good estimation of the video quality and the delay in all three cases and for the entire range of



(a) Delay distribution (2-hops).



(b) Delay distribution (3-hops).

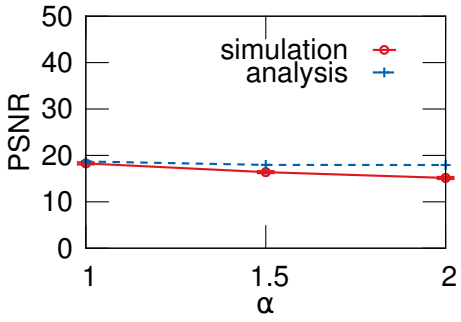


(c) Delay distribution (4-hops).

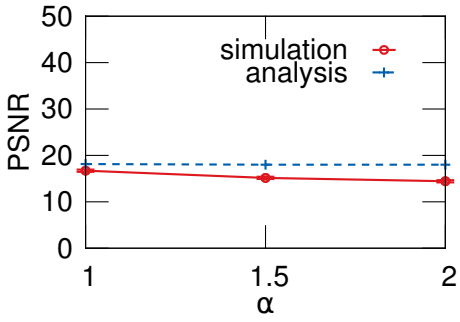
Figure 4.7: Delay distribution for slow motion videos.

the considered values of the aggressiveness α. Another observation is that for multi-hop video transmissions, increasing α carefully can yield significant delay gains if one could bear a slight decrease in PSNR. For example, in Fig. 4.5(c) if we allow the average PSNR value to drop from 24dB to 20dB (by increasing α), a large improvement in delay is possible as shown in Fig. 4.7(c). When $\alpha = 1.0$, only 27.7% packets are delayed by 0.01 seconds; when $\alpha = 2.0$, 73.6% packets are delayed by 0.01 seconds.

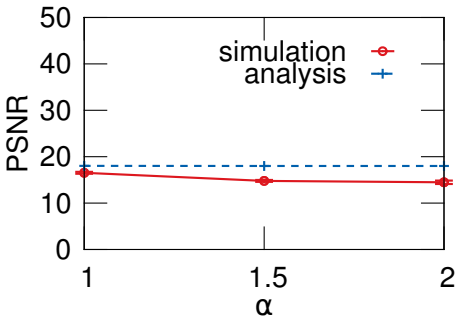
In Fig. 4.6(a)-4.6(c) the percentage of flows within the five MOS ratings for the three cases are shown. An increase in α decreases the percentage of flows with 'excellent' quality. Simultaneously, the percentage of flows with 'poor' and 'bad' quality increases. Furthermore, as the path length increases, the percentage of flows with higher quality decreases as one might expect.



(a) Avg PSNR (2-hops).



(b) Avg PSNR (3-hops).

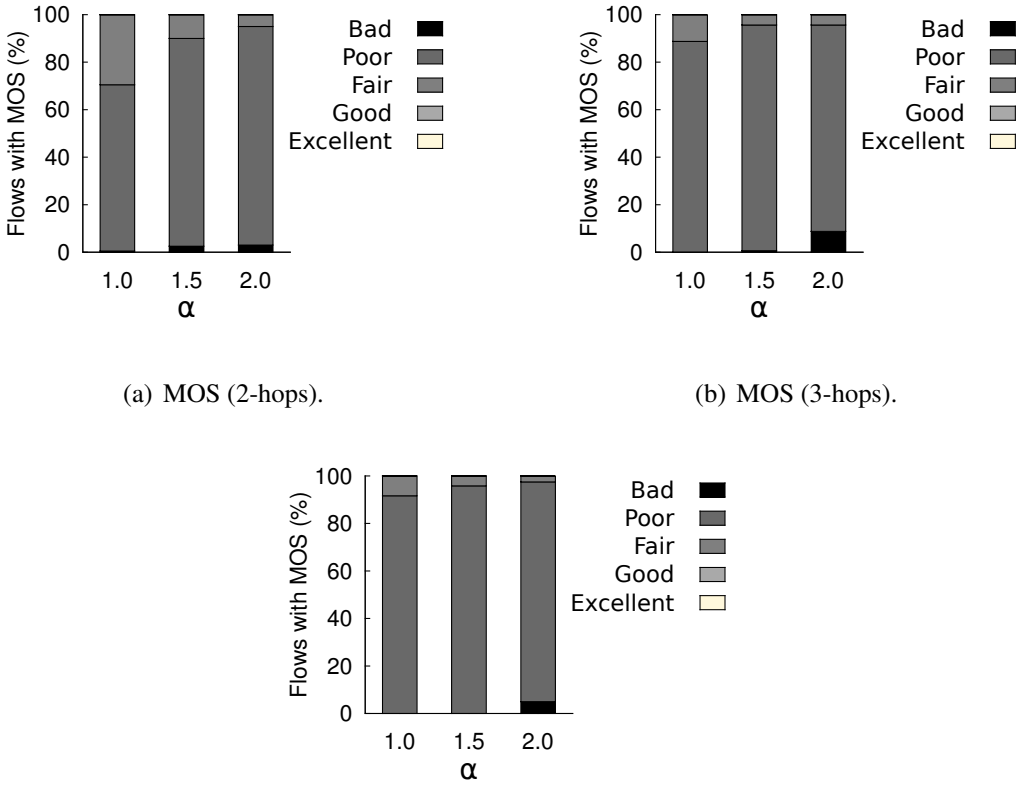


(c) Avg PSNR (4-hops).

Figure 4.8: Average PSNR for fast motion videos.

As discussed in Section 4.4, the parameter s in (4.18) indicates the *sensitivity* of the decoder to packet losses. When computing the analytical results of the first set of experiments for slow motion video transmissions, we set s to 0. Since the motion level of the video content is low, the difference from frame to frame inside a GOP and across the GOPs is expected to be small. In this case, the substitution of a missing video frame by a previously, correctly received frame incurs minimum distortion. Therefore, the decoder is less sensitive to packet losses as compared to fast motion video as we see next.

Fast motion videos: We repeat the same set of experiments with fast motion video clips. The PSNR, delay and MOS results are shown in Fig. 4.8, Fig. 4.9 and Fig. 4.10. For this set of experiments we keep the same number of concurrent flows as we had for slow motion video



(a) MOS (2-hops).

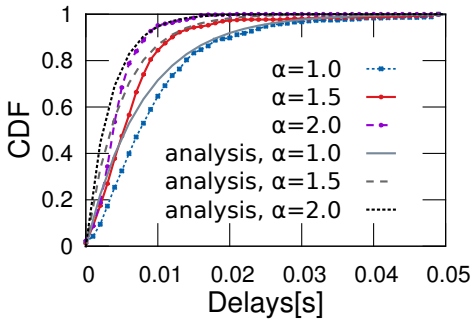
(b) MOS (3-hops).

(c) MOS (4-hops).

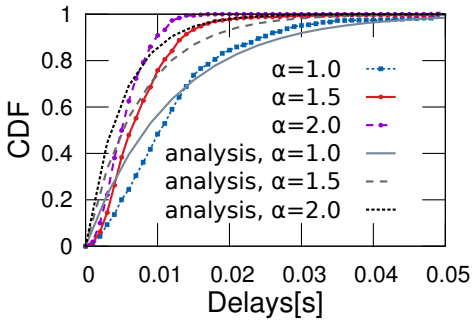
Figure 4.9: Average MOS for fast motion videos.

transmissions, i.e. 5 flows for 2-hop, 4 flows for 3-hop, and 3 flows for 4-hop connections.

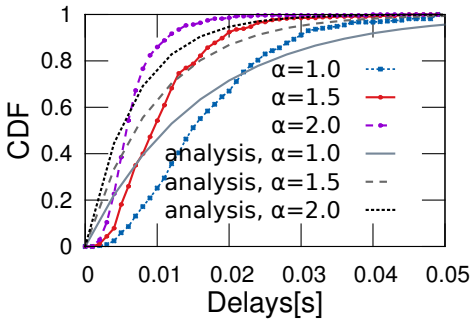
The results in Fig. 4.10 again show that our analytical model provides a good estimation of the video quality and delay for fast motion transmissions as well. In Figs. 4.8(a)-4.8(c) we observe that even when α is 1.0, the average PSNR value is much lower than the average PSNR value in Figs. 4.5(a)-4.5(c). This is because the fast motion video reconstruction is much more sensitive to packet losses as compared to slow motion video. Also as shown in Fig. 4.8(a), when $\alpha = 2.0$, the average PSNR value is as low as 15dB. If we want to keep the average PSNR value above 20dB, we need to tune α towards smaller values for fast motion video transmissions. With regards to the delay, the general effect of α is similar to that with slow motion video. Comparing the results in Figs. 4.8(a)-4.8(c), the decrease in video quality is not as prominent with an increase of path



(a) Delay distribution (2-hops).



(b) Delay distribution (3-hops).



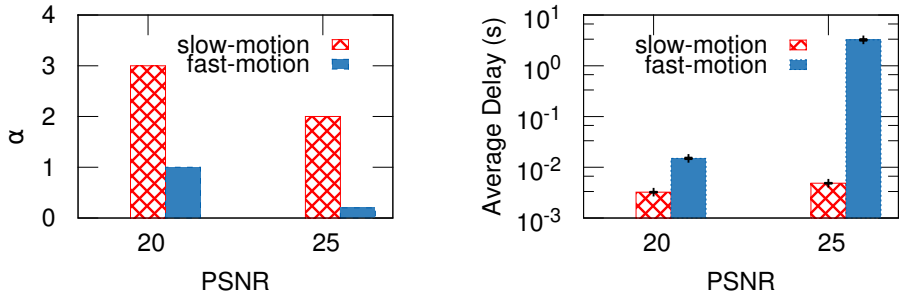
(c) Delay distribution (4-hops).

Figure 4.10: Delay distribution for fast motion videos.

length as with slow motion. This is because the video distortion is already high (approaching the maximum) and thus, the averaged PSNR value does drop much further when the path length grows.

Figs. 4.9(a)-4.9(c) show the percentage of flows within each category of MOS for fast motion video transmissions. We observe that very few flows are of quality higher than "Good". This is again due to the sensitivity of fast motion video decoding to packet losses. As shown in Fig. 4.9(b) even with $\alpha = 1.0$, half of the flows are of "fair" quality. This means that we should tune α more conservatively.

We wish to point out here that in the case of fast motion video content, the sensitivity of the decoder to packet losses is higher. A substitution of a missing frame by another frame, even from the same GOP, typically results in a significant increase in distortion. Therefore, we set s to 8 for



(a) α vs PSNR.

(b) Delay vs PSNR.

Figure 4.11: Value of α and mean delays for target PSNR.

fast motion video.

Delay for a target PSNR :

Next, we seek to estimate the delay incurred for a target PSNR value with both slow and fast motion video. Specifically, we seek to determine the delay increase penalties with fast and slow motion video, for increasing the PSNR value (decreasing distortion). We keep 10 flows in the network and tune the α to different levels, *targeting* average PSNR values of 20dB and 25dB (to reflect a desired video quality).

Fig. 4.11(a) shows the values of α needed to reach the targeted PSNR values for both slow and fast motion video clips. To reach a PSNR of 20dB, we need to tune α to 1.0 for fast motion video but can be more aggressive and set α to 3.0 for slow motion video. To achieve a PSNR value of 25dB, for fast motion video, α should be 0.2, which is a value smaller than 1.0. (In general, α should be tuned towards smaller values if the video is of fast motion, since it is more sensitive to packet losses.)

Fig. 4.11(b) shows the average packet delays associated with the targeted PSNR values of 20dB and 25dB for both slow and fast motion video clips. We see that the packet delays for slow motion video is much lower compared to that for fast motion video. To achieve a target PSNR of 20dB, average delays of 0.015 seconds and 0.003 seconds are incurred for fast and slow motion video, respectively. If the target PSNR is now increased to 25dB, the average delay increases to 0.005

(a) $\alpha = 1.0$ (b) $\alpha = 2.0$ (c) $\alpha = 3.0$

Figure 4.12: Slow motion video snapshots for different α.

(a) $\alpha = 1.0$ (b) $\alpha = 2.0$ (c) $\alpha = 3.0$

Figure 4.13: Fast motion video snapshots for different α.

seconds for slow motion video, and a staggering 3.2 seconds for fast motion video (increases by 99.8 %). *The delay increase penalty is 91 times higher with fast motion video as compared to slow motion video, for achieving this PSNR increase.* Thus, if one were to tolerate a slightly lower PSNR requirement with fast motion video, the delay could be drastically reduced. However, this reduction is not that prominent for slow motion video.

Visual quality: Finally, from the reconstructed videos at the receiver for both slow and a fast motion we have visually verified that the video quality of the slow motion video is clearly better than the fast motion video, for each considered value of α. This is consistent with the previously presented PSNR and MOS measurements results. Fig. 4.12 and Fig. 4.12 present snapshots of these videos.

4.6 Conclusions

Given the increasing popularity of video communications over wireless networks, we examine two key performance metrics associated with video flows, viz., distortion and delay. Unfortunately, these conflict with each other; one can increase distortion at the expense of delay and vice versa. Towards understanding the trade-off between distortion and delay we develop an analytical framework. We validate our framework with extensive simulation experiments. Since the sensitivities of the decoder to packet losses with respect to fast and slow motion video differ, we find that the motion level of a video clip affects this trade-off. Specifically, we find that for a target video quality, much lower delays are viable for slow-motion video clips. Our findings can be used as a design guideline for new protocols for networks that carry a considerable amount of video traffic.

Chapter 5

Performance of Visible Light Communications with Dimming

In this chapter, we study the performance of a visible light communications system. The system consists of two emitters, Tx1 and Tx2, located in two neighboring rooms, Room 1 and Room 2, respectively. The two rooms are connected via a door. We focus on the performance of the system in Room 1 (where Tx1 is located) and treat Tx2 as an interferer. We use Binary Pulse Position Modulation (BPPM) where we vary the pulse width within the slot to provide different dimming levels. We propose a modified ray-tracing algorithm to calculate the channel impulse response between Tx1 and the receivers in Room 1. Based on that, we generate the SNR distribution inside Room 1. We also provide a Bit Error Rate (BER) analysis considering different combinations of system parameters. Our results show that if Tx2 is only illuminating, it does not impact the performance of the communication in Room 1. However, if both Tx1 and Tx2 are transmitting, the performance in Room 1 is degraded to different levels depending on the position of the receivers. To improve in this case, the performance of communications in Room 1, we increase the dimming level of Tx1. Moreover, when the dimming level is limited, our results show that reducing the data rate of Tx1 improves the performance in Room 1 drastically. For example, when the data rate of Tx2 is 8Mbps, reducing the data rate of Tx1 from 8Mbps to 4Mbps results in a BER drop from

10^{-3} to 10^{-13}.

5.1 Introduction

Visible Light Communications (VLC) are gaining popularity ever since an indoor VLC system utilizing white LED light has been proposed in [76]. It is considered a promising replacement to radio frequency (RF) communications in indoor settings. In a VLC system, the LED lights not only illuminate the room, but also provide optical wireless communication. Due to its importance, IEEE has a standard [14] for VLC. White light LEDs have the advantages of reliability, security, lower power consumption, easy maintenance, harmlessness to the human eye and cost-effeciency. Furthermore, it is feasible to deploy a VLC network since in most indoor settings the lighting infrastructure already exists.

A lot of research has been conducted in the areas of VLC system design and channel characterization. In [77], the authors provide an indoor VLC system design with theoretical analysis and experimental proof of the feasibility of VLC. In [78], a typical basic configuration is provided and the performance that can be achieved with different modulation schemes is discussed. Regarding the VLC channel, a simulation based method in characterizing the infrared (IR) channel that has been proposed in [79] has been broadly adopted. Based on this work, [80] presents the VLC channel characteristics considering wavelength and spectral reflectance.

However, none of the above mentioned research efforts consider a VLC system deployed over two rooms separated by a door, with each room containing a set of emitters and receivers, as shown in Fig. 5.1. This type of indoor setting is typical, especially in home or office establishments. Needless to say, if the door is closed, the VLC system can provide a separate channel for each room since the visible light signal cannot go through opaque surfaces. On the other hand, if the door is open, the visible light signal in one room interferes with the signal in the next room. In this work, we study the communication performance of this scenario. Since the primary use of the LED emitters is illumination, dimming control is one of the desired functions of the system. Thus, throughout this

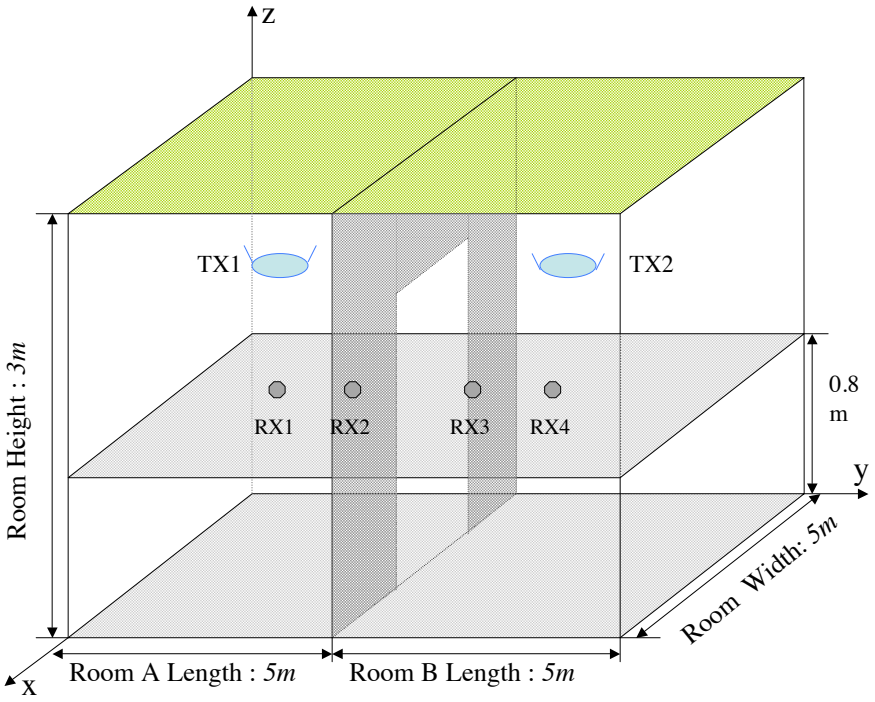


Figure 5.1: The visible light system in two rooms with an open door in between.

work, we use the Variable Pulse Position Modulation (VPPM) scheme, which combines the Binary Pulse Position Modulation (BPPM) scheme for data transmission and the Pulse Width Modulation (PWM) scheme for dimming control. Note that VPPM is easy to implement and has been discussed in [14].

In the following, we first present the system model. The details for the setting are discussed in Section 5.2. We characterize the channel based on simulation and propose an algorithm that uses a modified ray-tracing model to calculate the channel impulse response. We study the communication channel between Tx1 and a receiver in the same room (Room 1), treating the transmission from Tx2 as interference. We utilize a simple symbol detection method and compute the Signal to Noise Ration (SNR) to characterize the quality of the connection. We show the simulation results for the SNR distribution in the room for two different cases: (a) both emitters are transmitting, and (b) Tx1 is transmitting and Tx2 is illuminating.

We also provide BER performance analysis for the system for varying data rates, dimming levels and door sizes. The results show that increasing the data rate or increasing the door size can degrade the BER performance. Increasing the dimming level of Tx1 improves the BER performance. On the other hand, increasing the dimming level of the interfering emitter impacts the BER performance in a negative way, especially for the receivers close to the door.

Finally, we look at the performance when Tx1 reduces its data rate below that of Tx2. Our results show that this strategy can improve the BER performance significantly, which is useful when the dimming level has reached its limit.

5.2 A Visible Light Communication System Model

We consider a visible light indoor optical wireless system in two rooms with an open door between them, as shown in Fig. 5.1. There are two LED emitters, Tx1 and Tx2, in Room 1 and Room 2, respectively. We assume that the two emitters can transmit data and illuminate at the same time. The receivers are located on the plane that is 0.8m above the floor. We consider the receivers Rx1 and

Table 5.1: System parameters.

Room Length x(m)		5
Room Width y(m)		5
Room Height z(m)		3
Roof reflectivity		0.38
Floor reflectivity		0.6
Walls reflectivity		0.68
Door Width x(m)		3
Door Height z(m)		2
Tx1 Position		$(2.5, 2.5, 2)$
Tx2 Position		$(2.5, 7.5, 2)$
Rx1 Position		$(2.5, 2.5, 0.8)$
Rx2 Position		$(2.5, 4.0, 0.8)$
Rx3 Position		$(2.5, 6.0, 0.8)$
Rx4 Position		$(2.5, 7.5, 0.8)$
Receiver Area (cm^2)		1
Receiver FOV (deg)		85
Emitter Orientation	φ	0
	θ	90
Receiver Orientation	φ	0
	θ	90

Rx2 that are located in Room 1 and are associated with emitter Tx1. Due to geometrical symmetry, we expect that the performance of the receivers Rx3 and Rx4, which are associated with emitter Tx2, to be similar to that of Rx1 and Rx2, respectively. The parameters of the system are shown in Table. 5.1.

5.2.1 VPPM Transmitter

The transmitters Tx1 and Tx2 of Fig. 5.1 are at a distance of 0.5m from the ceiling and point straight up. We assume that each LED emitter uses VPPM modulation [14] and adopts the emission profile in [81]. The VPPM scheme is identical to the 2-PPM scheme when the duty cycle is 50%. The duty cycle δ, i.e. the pulse width within the slot T, corresponds to the dimming level. In this work, we consider the dimming level to be no more than 50% to simplify the analysis. Fig. 5.2 provides a simple example of the VPPM signal. The n^{th} transmitted bit $s_n \in \{0, 1\}$ corresponds to the symbol

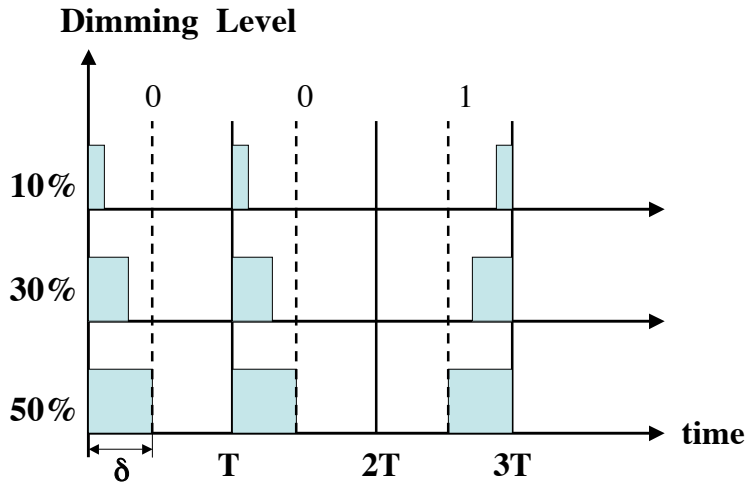


Figure 5.2: An example of VPPM signals.

S_n^δ, given by:

$$S_0^\delta(t) = \begin{cases} 2P_t, & \text{if } 0 \leq t < \delta T \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

$$S_1^\delta(t) = \begin{cases} 2P_t, & \text{if } (1-\delta)T \leq t < T \\ 0, & \text{otherwise} \end{cases} \tag{5.2}$$

where T is the duration of a VPPM symbol and P_t is the average transmitted power when the duty cycle is 50%.

5.2.2 Channel Impulse Response

The channel impulse response for the case where the emitters and the receivers reside in the same room has been presented in [82, 83, 84]. We extend this previous work by considering a situation where two emitters are located in two neighboring rooms with an open door in between. It is hard to calculate analytically the channel path loss in this case due to interference. Instead and motivated

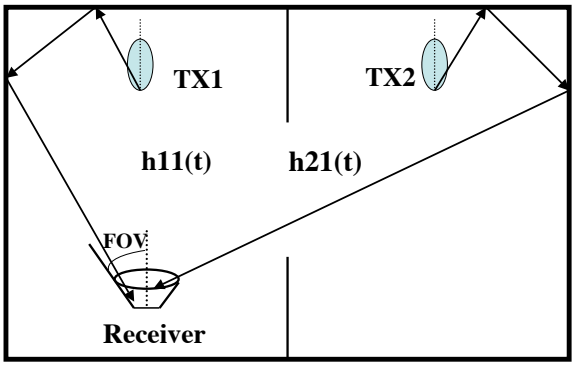


Figure 5.3: Reflection of rays.

by [84], we propose the use of a modified ray-tracing algorithm to generate the channel impulse response $h_i^{(k)}(t)$ for the channel between emitter i and receiver k. For a receiver k in Room 1, we consider two separate channels: (i) the channel between emitter Tx1 and the receiver $h_1^{(k)}(t)$, (ii) the channel between emitter Tx2 and the receiver $h_2^{(k)}(t)$.

We calculate the impulse response $h_i^{(k)}(t)$ through simulation that lasts t_max sec. The simulator determines the maximum number ray_max of rays to generate. The selection of the number of rays is discussed in [85]. The distribution of the generated rays is according to the emission profile. The propagation path of each ray may contain obstacles. These obstacles include the roof, the ceiling and the walls. Note that we assume the door is open, so the door is not considered an obstacle.

When a ray reaches an obstacle, the simulator checks if the point of impact (*PI*) is on the door. If the *PI* is not on the door, it reflects the ray and the power is reduced by the reflection coefficient of the obstacle. During simulation, a new ray is generated at *PI* with the new, reduced, power. If the *PI* is on the door, the ray propagates to the other room. The simulator computes the new point of impact *PI'* in the other room and generates a new ray at that point. In Fig. 5.3 the reflections of rays is shown. Only diffused reflection is considered in this algorithm. The direct power contribution of each ray is calculated each time it is reflected [84]. The calculated power is added to $h_i^{(k)}(t)$ if the ray can be intercepted by the receiver, i.e. it is within the FOV of the receiver.

Fig. 5.4 presents the impulse response computed through Algorithm 1 for the channel between

while $ray\_num < ray\_max$ **do**
  step 1 : Generate a new ray starting at the emitter ;
    $t = 0$, $P = 1$;
  step 2 : **while** $t < t\_max$ **do**
    Propagate the ray until it reaches any obstacle plane;
    Find the point of impact *PI* where the ray intersects with the obstacle;
    if PI *is on the door* **then**
      Propagate the ray to the neighboring room;
      Find the impact point *PI'* where the ray intersects with any obstacle planes in the neighbor room;
      Calculate the contribution from PI' to the receiver;
      Generate a new ray starting at PI', with reduced power $P = \rho P$;
      Back to Step 2;
    else
      Calculate the contribution from *PI* to the receiver;
      Generate a new ray starting at *PI*;
    end
  end
  $Increase\ ray\_num\ by\ 1.$
end

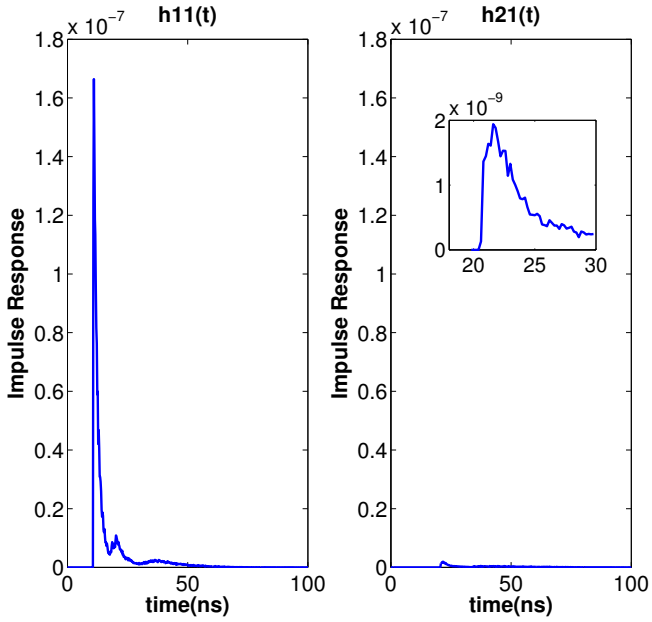
Algorithm 1: Ray-tracing Algorithm for $h_i^{(k)}(t)$

Tx1 and receivers Rx1 and Rx2. The number of rays is 10,000, the resolution time is 0.2ns and the simulation time is 120ns. The shapes of $h_1^{(1)}(t)$ and $h_1^{(2)}(t)$ look similar. Only the peak power of $h_1^{(1)}(t)$ is higher than $h_1^{(2)}(t)$. This is due to the positions of Rx1 and Rx2, i.e. Rx1 is closer to Tx1 and further from Tx2 while Rx2 is further to Tx1 and closer to Tx2. For the same reason, we see that the power of $h_2^{(1)}(t)$ is much smaller than the power of $h_2^{(2)}(t)$.

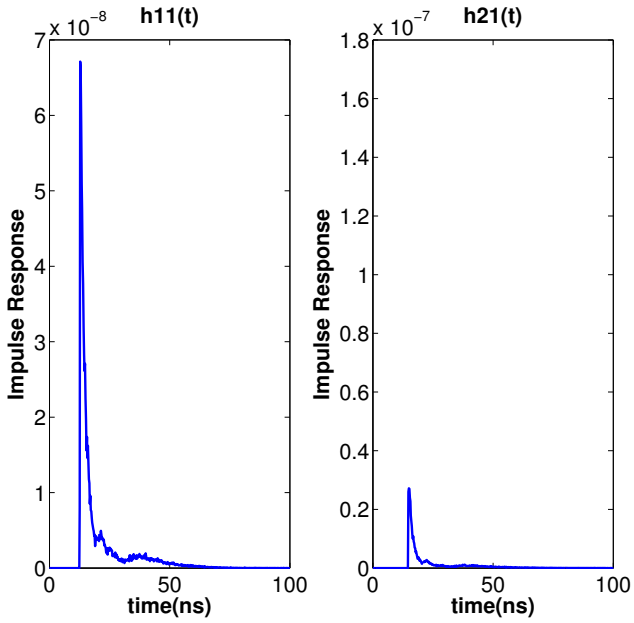
5.2.3 Received Signal

Receivers in Room 1 are associated with Tx1 and receivers in Room 2 are associated with Tx2. If not explicitly stated otherwise, we assume that the emitters Tx1 and Tx2 use the same data rate and the system is synchronized. Looking at the performance in Room 1, we consider two scenarios: (i) both Tx1 and Tx2 are transmitting data, and (ii) Tx1 is transmitting data and Tx2 is just illuminating.

Receivers Rx1 and Rx2 are connected to Tx1, therefore the signal from Tx2 acts as an interference to them. Following [79], the received signal $r_1^k(t)$ at receiver k in Room 1 is given



(a) Impulse Response between transmitters and Rx1.



(b) Impulse Response between transmitters and Rx2.

Figure 5.4: Impulse Responses.

as:

$$r^{(k)}(t) = R^{(k)} X_1(t) \otimes h_1^{(k)}(t) \; + \; I^{(k)}(t) \; + \; n(t) \tag{5.3}$$

where R^k is the responsivity of receiver k in Room 1, $X_1(t)$ is the transmitted signal of Tx1 and $n(t)$ is the noise. $I^{(k)}(t)$ is the interference from Tx2 to receiver k, given by:

$$I^k(t) = R^{(k)} X_2(t) \otimes h_2^{(k)}(t) \tag{5.4}$$

where $X_2(t)$ is the transmitted signal of Tx2. Note that for scenario (ii) above $X_2(t)$ is a signal with constant power. We further discuss this case in Section 5.2.4. The variance σ^2_{total} of the Gaussian $n(t)$ [77, 76, 79] is given by:

$$\sigma^2_{total} = \sigma^2_{thermal} \ + \ \sigma^2_{shot} \tag{5.5}$$

The shot noise variance is given by:

$$\sigma^2_{shot} = 2qRP_nI_2R_b \tag{5.6}$$

where q is the electric charge, R is the photodiode responsivity, P_n is the noise power, I_2 is the noise bandwidth factor and R_b is the data rate. The thermal noise variance is given by:

$$\begin{aligned}\sigma^2_{thermal} = \frac{4kT_f}{R_F} I_2 R_b \ + \ \frac{16\phi^2 kT_f}{g_m}(\Gamma \ + \ \frac{1}{g_m R_D}) C_T^2 I_3 R_B^3 \\ + \frac{4\phi^2 K I_D^a C_T^2}{g_m^2} I_f R_b^2\end{aligned} \tag{5.7}$$

We adopt the parameters defined in [79] except for the data rate R_b. The received waveform can be calculated using (5.3)-(5.7).

5.2.4 Symbol Detection and SNR Distribution

Ther are various methods designed for symbol detection [86, 87, 88]. We need a symbol detection mechanism that is simple and effective, and does not complicate the analysis. As discussed earlier, the system that we consider in this work has two channel impulse responses $h_1(t)$ and $h_2(t)$ for each receiver. Note that these two channel impulse responses cannot be combined because $X_1(t)$

and $X_2(t)$ can be different, i.e. Tx1 and Tx2 are transmitting different data. Thus, equalization [89] cannot be employed in this system. Considering dimming levels no more than 0.5 we can neglect ISI (Intersymbol Interference) when the symbol duration is sufficiently longer than the delay spread.

The uneuqalized receiver perform symbol-by-symbol ML detection. We assume the receiver is synchronized with the transmitter and the receiver has the information of dimming level and the data rate of the transmitter it is associated with, which is done before the data transmission [14]. Thus the received signal $r(t)$ can be sampled into two blocks y_1, y_2 for each symbol, where y_1 and y_2 correspond to the samples in the fist half slot $slot_1$ and second half slot $slot_2$ respectively. The receiver makes symbol decisions based on the relative magnitude of y_1, y_2.

Using this symbol detection method, the SNR can be defined as:

$$\text{SINR}_{s_n} = \begin{cases} \frac{P_{slot_1} - P_{slot_2}}{P_{noise}}, & \text{if } s_n \text{ is 0} \\ \\ \frac{P_{slot_2} - P_{slot_1}}{P_{noise}}, & \text{if } s_n \text{ is 1} \end{cases} \tag{5.8}$$

where $P_{noise} = \sigma_{total}$ (see (5.5)), s_n is the desired symbol and P_{slot_1} and P_{slot_2} are the average received power levels in the first half slot $slot_1$ and the second half slot $slot_2$, respectively. The average received power is computed based on the received signal $r^{(k)}(t)$,which can be computed by (5.3) We are interested in the expected value of the SINR across Room 1, which provides a measure of the communication performance in Room 1. As discussed earlier, we consider two scenarios: (i) both Tx1 and Tx2 are transmitting, and (ii) Tx1 is transmitting and Tx2 is just illuminating. Assuming the input bit stream is an independently and identically distributed (i.i.d.) Bernoulli$(1/2)$ process, the emitter in transmission mode has $1/2$ probability to transmit symbol [1] 0 or 1.

Considering the first scenario, the possible combination of symbols from Tx1 and Tx2 is one in

[1]To evaluate the performance of the system, we consider the system is uncoded throughout this work.

the set $S_{set}^{(1)} = \{(0,0),(0,1),(1,0),(1,1)\}$. The expected value of the SNR for the first scenario is:

$$\mathrm{E}[\mathrm{SINR}] = \frac{1}{4}\sum \mathrm{SINR}_{\{s_1,s_2\}} \tag{5.9}$$

where $\{s_1, s_2\} \in S_{set}^{(1)}$ and s_1 s_2 are the bits transmitted from Tx1 and Tx2 respectively. To compute P_{slot_1} and P_{slot_2} for a specific receiver k in Room 1, we first simulate its $h_1^{(k)}(t)$ and $h_2^{(k)}(t)$ with Algorithm 1. SINR_{s_1,s_2} is computed via (5.8) and s_1 as the desired symbol since we look at receivers in Room 1.

Regarding the second scenario, Tx2 is in illumination mode so that the signal $S_{illumin}$ it generates is of constant power:

$$S_{illumin}^{\delta}(t) = 2\delta P_t \tag{5.10}$$

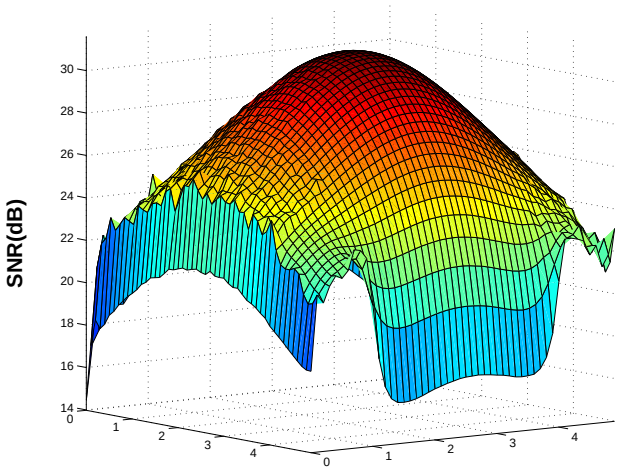
where δ is the dimming level of Tx2. Fig. 5.5 shows the expected value of the SNR across Room 1 for scenario (i) and (ii). The dimming level of both emitters is 0.5 and the data rate is 1 Mbps. The receiver plane is at a distance of $0.8m$ above the floor. Comparing the results shown in Fig. 5.5(a) and Fig. 5.5(b), we observe that if Tx2 is just illuminating it does not impact the transmission performance in Room 1, while if both Tx1 and Tx2 are transmitting data, the performance in Room 1 is affected. Especially for the receivers in Room 1 closer to the door the SNR is degraded largely.

5.3 BER Performance

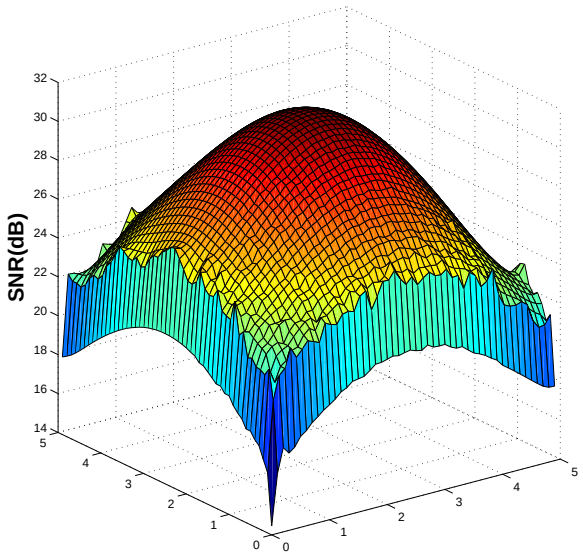
In this section we first provide basic BER performance analysis and then look into the performance with different system parameters. Our goal is to achieve an optimal BER performance by tuning the dimming level and the data rate for the system we consider in this work.

5.3.1 Bit Error Rate Analysis

First we seek to approximate the BER performance when Tx1 and Tx2 use the same data rate. As described earlier, we consider an unequalized VPPM system. Assuming the symbol detection



(a) Both Tx1 and Tx2 are transmitting.



(b) Only Tx1 is transmitting.

Figure 5.5: SNR distribution of Room 1 (datarate 1 Mb/s)

method in section 5.2.4, the probability of bit error for scenario **(a)** at high SNR can be estimated

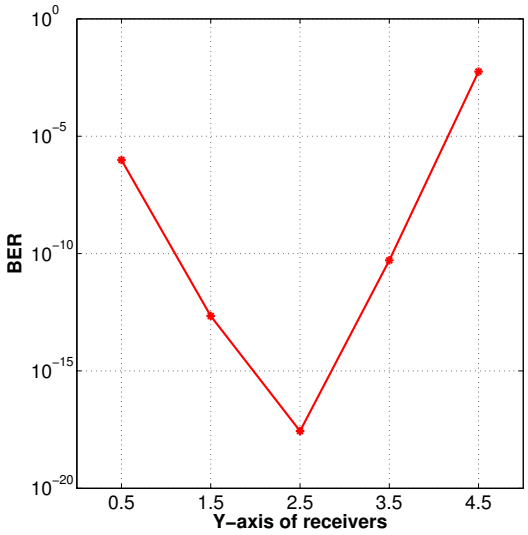


Figure 5.6: BER vs Distance

as :

$$P\{\ \text{bit error} \mid \text{Tx1 transmits symbol } s_1\ \} \tag{5.11}$$

$$\approx Q\Big(\sqrt{SNR_{\{s_1,s_2\}}}\Big) \tag{5.12}$$

where s_1 and s_2 are the symbols sent by Tx1 and Tx2 respectively, and $Q(x)$ is given by :

$$Q(x) = \frac{1}{\sqrt{2\phi}} \int_x^\infty \mathrm{e}^{-u^2/2} du \tag{5.13}$$

For a random input data, BER can be obtained by averaging over all possible symbol s_1 and over all possible interfering symbol s_2 from Tx2:

$$BER = \sum P\{\ \text{bit error} \mid \{s_1, s_2\}\ \} \cdot P\{\ \{s_1, s_2\}\ \} \tag{5.14}$$

where $\{s_1, s_2\} \in S_{set}^{(1)}$ and $P\{\ \{s_1, s_2\}\ \} = 1/4$. Note that $P\{\ \text{bit error} \mid \{s_1, s_2\}\ \}$ is the same with $P\{\ \text{bit error} \mid \text{Tx1 transmits symbol } s_1\ \}$ since s_1 is the desired symbol.

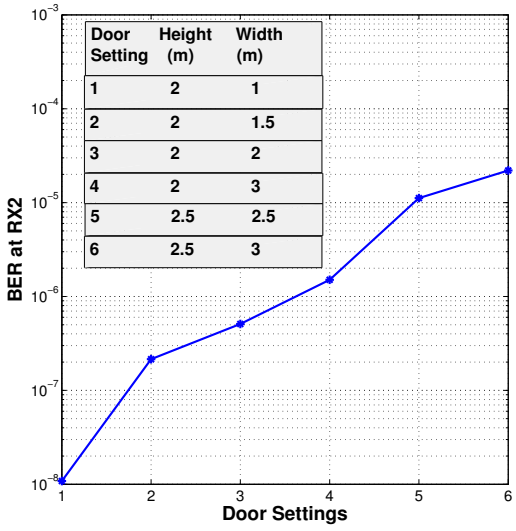


Figure 5.7: BER at Rx2 vs Doorsizes

5.3.2 BER Performance

We evaluate the BER performance with various combinations of the system parameters such as receiver positions, door sizes, data rates and also dimming levels, If not stated otherwise, the two emitters are using the same dimming level and the same data rate.

BER vs Distance: Fig. 5.5(a) shows that when two emitters are transmitting the receivers closer to the door are affected much more. We calculate the BER for five receivers in room 1, which are at different distances from the emitters. The two emitters use the same dimming level of 0.5 and data rate $10Mb/s$. The BER performance results shown in Fig.5.6 is consistent with the SNR results in Fig.5.5(a).

BER vs Door Size : Our previous analysis and results have shown that the interfering signal from Tx2 impacts the performance in room 1. It is interesting to look into how the different door sizes affects the performance. In Fig. 5.7 we show the BER at receiver RX2 in various door sizes. As expected, the larger the area of the door is, the higher the BER is. This is because more interfering signal goes through the door when the door is larger.

BER vs Data Rate Till now, we assume data rate of Tx1 and Tx2 are the same. In Fig. 5.8,

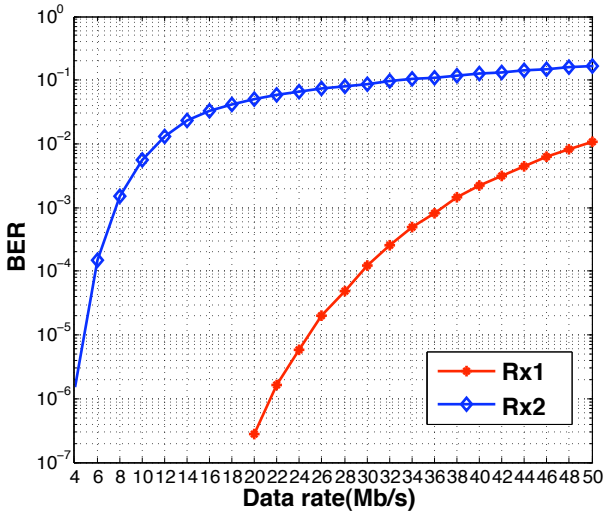


Figure 5.8: BER vs Data rate for Rx1 and Rx2

.

we show the BER for Rx2 and Rx1 with different data rates. Due to the higher interfering signal at Rx2, the maximum data rate is $4Mb/s$ to reach the 10^{-6} minimum requirement of BER. For Rx1, the maximum data rate is $20Mb/s$. Generally, increasing data rate degrades the BER at both receivers.

BER vs Dimming: In previous results, we consider Tx1 and Tx2 both use dimming level 0.5. Since dimming is a special feature of VPPM modulation, we look into the impact of dimming on the performance of the system. δ_1 and δ_2 are dimming levels of Tx1 and Tx2 respectively. Receivers in room 1 at different locations are affected by interfering signal from Tx2 so we look at the BER performance at Rx1 and Rx2 separately. In results of BER at Rx1, we use data rate of $20Mb/s$. As for results at Rx2, we use data rate of $4Mb/s$.

Results shown in Fig.5.9 present the BER performance at Rx2 with different combinations of δ_1 and δ_2. There are five sets of data, wherein each of them the dimming level δ_2 of Tx2 is fixed and the dimming level δ_1 of Tx1 is increased from 0.1 to 0.5. When δ_2 is fixed, increasing δ_1 improves the BER performance. For example, when δ_2 is 0.2, increasing δ_1 from 0.3 to 0.5 makes the BER drops from 10^{-2} to 10^{-6}. This trend is observable for each set of data where δ_2 is fixed to different levels. Also, if we look at Fig.5.9 in a different angle, i.e. considering each column as a set of

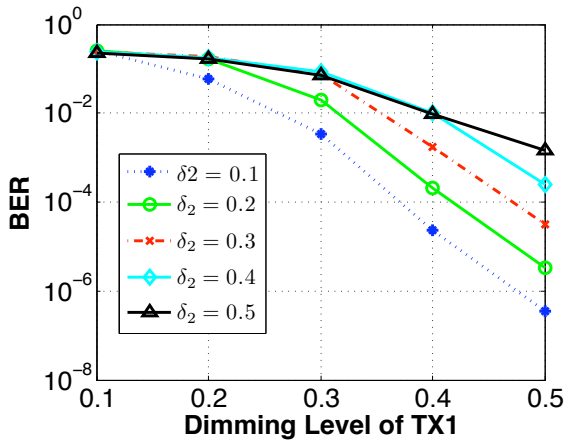


Figure 5.9: BER vs Dimming at Rx2.

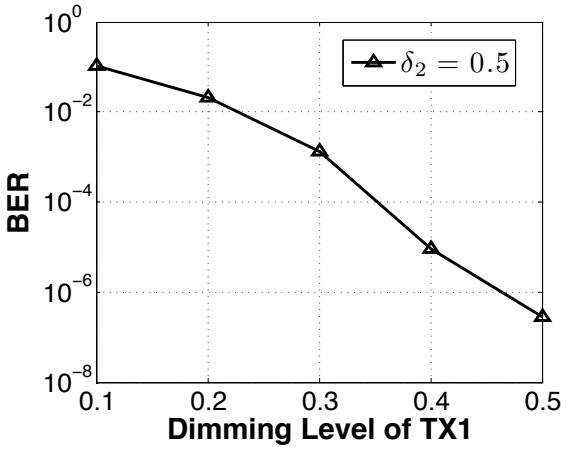


Figure 5.10: BER vs Dimming at Rx1.

data, we can conclude that if δ_1 is fixed, decreasing δ_2 can improve the BER performance. Another observation is that when δ_2 is of a high value, for example, if $\delta_2 = 0.5$ then tuning δ_1 is limited in helping improve BER. This is because the higher the δ_2 is, the stronger the interfering signal is, which impacts the BER performance. When $\delta_1 = \delta_2 = 0.5$ the interfering signal for Rx2 is considerable.

For Rx1, we observe that the dimming level δ_2 of Tx2 does not impact its performance and increasing δ_1 has the similar impact without relation to Tx2's dimming level. In Fig. 5.10 we show the results for when $\delta_2 = 0.5$ increasing δ_1, wherein the BER performance is improved.

How to improve BER when dimming is limited : The observation from previous results implies that increasing the dimming level of the desired emitter can improve BER performance. However, it also shows that when dimming reaches its limitation the BER is cannot be improved much. For example, in Fig. 5.9 we see that if δ_1 δ_2 are 0.5 the BER at Rx2 is 10^{-3}, which is not acceptable in data transmission. Thus, when dimming level is limited, we can reduce the data rate of Tx1 to improve the performance in room 1.

The BER analysis is a bit different than the analysis in 5.3.1. Because the possible combination of symbols from Tx1 and Tx2 is not the same as $S_{set}^{(1)}$ when Tx1 and Tx2 use the same data rate. We assume the data flows from Tx1 and Tx2 start at the same time to avoid complicating the analysis. If the data rate R_{b_1} is half of R_{b_2}, the symbol combination of Tx1 and Tx2 $\{s'_{1,i},s'_{2,j}\}$ is one in the set $S_{set}^{(2)}$: $\{\ \{0,00\},\{0,01\},\{0,11\},\{0,10\},\{1,00\},\{1,01\},\{1,11\},\{1,10\}\ \}$ Because $R_{b_1} =^1 /2R_{b_2}$ means $T_1 = 2T_2$ (T_1, T_2 are the symbol duration time for Tx1 and Tx2). Fig. 5.11 illustrates the symbol set for Tx1 transmitting 0. The BER can be calculated using Eq. (5.14) and $P\{\ \{s_{1,i}, s_{2,j}\}\ \}$ is $1/8$ here.

Similarly, we can compute the BER for $T_1 = 4T_2$. Fig. 5.12 shows the BER at RX2 when Tx1 and Tx2 using different data rates. It shows that increasing T_1 to be four times of T_2, i.e., reducing the data rate (R_{b_1}) of Tx1 to be $1/4$ of the data rate (R_{b_2}) of Tx2 can improve the BER performance at Rx2 significantly.

Specifically, when the data rate (R_{b_2}) of Tx2 is $16Mb/s$ reducing R_{b_1} from $16Mb/s$ to $8Mb/s$ makes the BER drops from 10^{-3} to 10^{-6}. Mo rover, if we reduce R_{b_1} to be $4Mb/s$, the BER is as low as 10^{-16}. Since the minimum required BER is 10^{-6} we set the maximum data rate of Tx1 accordingly, considering performance at Rx2: for $R_{b_2} = 32Mb/s$ the maximum R_{b_1} is $8Mb/s$ and for $R_{b_2} = 16Mb/s$ the maximum R_{b_2} is $8Mb/s$.

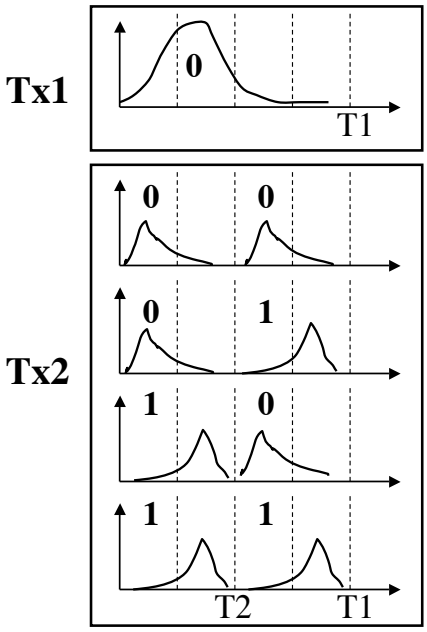


Figure 5.11: Symbol set for Tx1 transmitting 0, $T_1 = 2T_2$

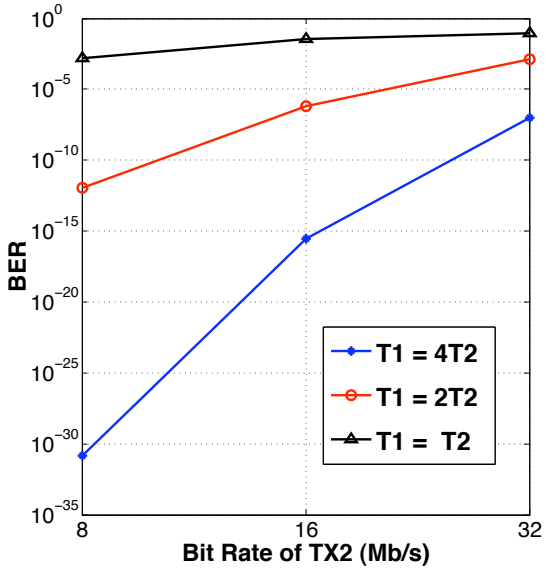


Figure 5.12: BER at Rx2 for Tx1 and Tx2 using different data rates.

5.4 Conclusions

In this chapter, we discuss the performance of a visible light system within two neighboring rooms, where two emitters, Tx1 and Tx2, are located in separate rooms and VPPM with dimming is used. We propose an algorithm to characterize the channel impulse response and the BER of the system. Our results show that if Tx2, which is the interferer, is just illuminating, it does not impact the performance of the communication between Tx1 and the receivers in the same room. However, if both Tx1 and Tx2 are transmitting, the performance is degraded, especially for the receivers closer to the door. We show that increasing the dimming level of the desired signal can improve the BER performance. Moreover, we find that when the interfering signal is strong and the dimming level reaches its limit, reducing the data rate of Tx1 improves significantly the performance of the communication between Tx1 and the receivers in the same room.

Bibliography

[1] Z. Feng, J. Ning, I. Broustis, K. Pelechrinis, S.V. Krishnamurthy, and Michalis Faloutsos. Coping with packet replay attacks in wireless networks. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on*, pages 368–376, June.

[2] Z. Feng, K. Pelechrinis, S.V. Krishnamurthy, Ananthram Swami, Felix Wu, and Singh M.P. Collaborative assessment of functional reliability in wireless networks. In *Mobile Ad hoc and Sensor Systems (MASS), 2012 9th IEEE International Conference on*, Oct.

[3] Z. Feng, G. Papageorgiou, S.V. Krishnamurthy, R. Govindan, and T.L. Porta. Trading off distortion for delay for video transmissions in wireless networks. In *the 2013 32nd IEEE International Conference on Computer Communications (INFOCOM)*, April.

[4] Wenyuan Xu, Wade Trappe, Yanyong Zhang, and Timothy Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '05, pages 46–57, New York, NY, USA, 2005. ACM.

[5] Wenyuan Xu, Ke Ma, W. Trappe, and Y. Zhang. Jamming sensor networks: attack and defense strategies. *Network, IEEE*, 20(3):41–47, May-June.

[6] K. Pelechrinis, Guanhua Yan, S. Eidenbenz, and S.V. Krishnamurthy. Detecting selfish exploitation of carrier sensing in 802.11 networks. In *INFOCOM 2009, IEEE*, pages 657–665, April.

[7] K. Pelechrinis, I. Broustis, S.V. Krishnamurthy, and C. Gkantsidis. A measurement-driven anti-jamming system for 802.11 networks. *Networking, IEEE/ACM Transactions on*, 19(4):1208–1222, Aug.

[8] T. Aura. Strategies against replay attacks. In *IEEE Computer Security Foundations Workshop*, pages 59–68. IEEE Computer Society Press, 1997.

[9] S. Malladi, J. Alves-Foss, and R. B. Heckendorn. On preventing replay attacks on security protocols. In *In Proc. International Conference on Security and Management*, pages 77–83. CSREA Press, 2002.

[10] P. Papadimitratos and Z. J. Haas. Secure routing for mobile ad hoc networks. In *CNDS*, pages 193–204, 2002.

[11] J. Zhen and S. Srinivas. Preventing Replay Attacks for Secure Routing in Ad Hoc Networks. In *Ad-Hoc, Mobile, and Wireless Networks*, pages 140–150. Springer Berlin/Heidelberg, February,2004.

[12] E. Winjum, A. M. Hegland, O. Kure, and P. Spilling. Replay Attacks in Mobile Wireless Ad Hoc Networks: Protecting the OLSR Protocol. In *Lecture Notes in Computer Science, ISSN 0302-9743*, 2005.

[13] FIPS-186-2. In *Digital Signature Standard (DSS), FIPS PUB 186-2*. U.S. Department of Commerce/National Institute of Standards and Technology, January 2000.

[14] Ieee standard for local and metropolitan area networks–part 15.7: Short-range wireless optical communication using visible light. *IEEE Std 802.15.7-2011*, pages 1–309, 6.

[15] ANSI/IEEE 802.11-Standard. 1999 edition.

[16] J. Wright. Detecting wireless LAN MAC address spoofing. Technical report, 2003.

[17] Andrei Broder and Michael Mitzenmacher. Network applications of bloom filters: A survey. In *Internet Mathematics*, pages 636–646, 2002.

[18] FIPS-180-1. In *NIST, FIPS PUB 180-1: Secure Hash Standard*, April 1995.

[19] P. Syverson. A Taxonomy of Replay Attacks. In *Naval Research Lab, Washington DC*, January 1994.

[20] A. Perrig, R. Canetti, J. D. Tygar, and D. Song. The tesla broadcast authentication protocol. *RSA CryptoBytes*, 5:2002, 2002.

[21] Y. Yang, X. Wang, S. Zhu, and G. Cao. Sdap: A secure hop-by-hop data aggregation protocol for sensor networks. In *ACM MOBIHOC*, pages 356–367. ACM Press, 2006.

[22] S. Zhu et al. Lhap: a lightweight network access control protocol for ad hoc networks. In *Journ. of Ad Hoc Networks, 46 DRDC Ottawa TM*, 2006.

[23] Y. Huang, W. He, K. Nahrstedt, and W. C. Lee. Dos-resistant broadcast authentication protocol with low end-to-end delay. In *INFOCOM Workshops 2008, IEEE*, pages 1–6, April 2008.

[24] T. Heer et al. Alpha: an adaptive and lightweight protocol for hop-by-hop authentication. In *ACM CoNEXT*, 2008.

[25] S. Mizikovsky, Z. Wang, and H. Zhu. CDMA 1xEV-DO Security. In *Wiley Interscience, Bell Labs Technical Journal, 11(4), 291-305*, 2007.

[26] S. Pallicara et al. A Framework for Secure End-to-End Delivery of Messages in Publish/Subscribe Systems. In *IEEE/ACM International Conference on Grid Computing*, 2006.

[27] C. Adjih et al. Securing the OLSR Protocol. In *Med-Hoc-Net*, 2003.

[28] Security Architecture for the Internet Protocol. http://www.rfc-editor.org/rfc/rfc4301.txt.

[29] M. G. Gouda, C.-T. Huang, and E. Li. Anti-Replay Window Protocols for Secure IP. In *IEEE ICCCN*, 2000.

[30] UCR Wireless Testbed. http://networks.cs.ucr.edu/testbed.

[31] Click Modular Router. http://read.cs.ucla.edu/click/.

[32] The OpenSSL Project. http://www.openssl.org.

[33] C++ Bloom Filter Library. http://code.google.com/p/bloom.

[34] Soekris-net5501. http://www.soekris.com/net5501.htm.

[35] V. Navda et al. Using Channel Hopping to Increase 802.11 Resilience to Jamming Attacks. In *IEEE INFOCOM Miniconference*, 2007.

[36] R. Gummadi et al. Understanding and mitigating the impact of rf interference on 802.11 networks.

[37] D. B. Johnson et al. Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. In *Ad Hoc Networking, Ch.5, pp. 139-172*, 2001.

[38] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *MobiCOM*, 2000.

[39] S. Buchegger and J.-Y. Le Boudec. Nodes Bearing Grudges: Towards Routing Security, Fairness and Robustness in Mobile Ad Hoc Networks. In *Proc. Euromicro Wkshp. Parallel, Distributed and Network-based Processing*, 2002.

[40] S. Buchegger and J.-Y. Le Boudec. Performance Analysis of CONFIDANT protocol: Cooperation of nodes - fairness and dynamic ad-hoc networks. In *ACM MobiHOC*, 2002.

[41] P. Michiardi and R. Molva. CORE: A Collaborative Reputation Mechanism to enforce node cooperation in Mobile Ad Hoc Networks. In *European Wireless Conference*, 2002.

[42] Matthew J. Probst and Sneha Kumar Kasera. Statistical Trust Establishment in Wireless Sensor Networks. In *Proc. 13th International Conference on Parallel and Distributed Systems*, 2007.

[43] Pedro B. Velloso, Rafael P. Laufer, Daniel de O. Cunha, Otto Carlos M. B. Duarte, and Guy Pujolle. Trust management in mobile ad hoc networks using a scalable maturity-based model. In *IEEE Trans. Netw. Service Manage., Vol. 7, No.3*, 2010.

[44] Kannan Govindan and Prasant Mohapatra. Trust Computations and Trust Dynamics in Mobile Adhoc Networks: A Survey. In *Communications Surveys Tutorials, IEEE, Vol. PP, Issue 99*, 2011.

[45] L. Buttyan and J.P. Hubaux. Enforcing Service Availability in Mobile Ad Hoc WANs. In *ACM MobiHOC*, 2000.

[46] S. Zhong, J. Chen, and Y. R. Yang. Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks. In *IEEE INFOCOM*, 2003.

[47] G. Shafer. *A Mathematical Theory of Evidence*. Princeton Univ. Press, Princeton, NJ, 1976.

[48] A. P. Dempster. A generalization of Bayesian inference. In *Journal of the Royal Statistical Society, Series B, Vol. 30, pp. 205247*, 1968.

[49] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A High Throughput Path Metric for MultiHop Wireless Routing. In *ACM MOBICOM*, 2003.

[50] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, ISBN 0-13-345711-7.

[51] C.-W. Hang, Y. Wang, and M. P. Singh. Operators for propagating trust and their evaluation in social networks. In *Proc. AAMAS*, 2009.

[52] C. Perkins. *Ad hoc Networking*. Addison Wesley Professional Series, 2001.

[53] Alan C. Bovik. *The Essential Guide to Video Processing*. Academic Press, 2009.

[54] ISO/IEC JTC1/SC29/WG11. ISO/IEC 14496 – Coding of audio-visual objects. `http://mpeg.chiariglione.org/standards/mpeg-4/mpeg-4.htm`.

[55] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the H.264/AVC video coding standard. 13(7):560–576, July 2003.

[56] Michel T. Ivrlac, Ruly Lai-U Choi, Eckehard G. Steinbach, and Josef A. Nossek. Models and analysis of streaming video transmission over wireless fading channels. *Signal Proessing: Image Communication*, 24(8):651–665, September 2009.

[57] Raja S. Tupelly and Junshan Zhang. Opportunistic scheduling for streaming video in wireless networks. In *Hopkins University*, 2003.

[58] Xiaofeng Xu. Fine-granular-scalability video streaming over wireless LANs using cross layer error control. In *ICASSP*, Montreal, Canada, May 2004.

[59] J. Villalon, P. Cuenca, L. Orozco-Barbosa, Yongho Seok Yongho Seok, and T. Turletti. Cross-layer architecture for adaptive video multicast streaming over multirate wireless LANs. *IEEE JSAC*, 25(4), May 2007.

[60] J. R. Renno, N. Lazarevic-McManus, Dimitrios Makris, and Graeme A. Jones. Evaluating motion detection algorithms: Issues and results. In *Proceedings of the* 6^{th} *IEEE International Workshop on Visual Surveillance*, 2006.

[61] PhysMo: Video motion analysis. `http://physmo.sourceforge.net`.

[62] AForge.NET. http://www.aforgenet.com/framework/features/motion_detection/_2.0.html.

[63] E. A. Akkoyunlu. The enumeration of maximal cliques of large graphs. *SIAM Journal on Computing*, 2(1):1–6, 1973.

[64] Haiyun Luo, Songwu Lu, and Vaduvur Bharghavan. A new model for packet scheduling in multihop wireless networks. In *ACM MobiCom*, 2000.

[65] Coen Bron and Joep Kerbosch. Finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, September 1973.

[66] Rajarshi Gupta, Jean Walrand, and Olivier Goldschmidt. Maximal cliques in unit disk graphs: Polynomial approximation. In *Proceedings of the* 2^{nd} *International Network Optimization Conference (INOC)*, 2005.

[67] H.T. Friis. A note on a simple transmission formula. *Proceedings of the IRE*, 34(5):254–256, May 1946.

[68] M. Pursley and T. Royster Iv. Properties and performance of the IEEE 802.11b complementary-code-key signal sets. 57(2):440–449, February 2009.

[69] G. Pei and T. Henderson. Validation of ns-3 802.11b PHY model, May 2009.

[70] Yubing Wang, Mark Claypool, and Robert Kinicki. Impact of reference distance for motion compensation prediction on video quality. In *Proceedings of ACM/SPIE Multimedia Computing and Networking (MMCN)*, 2007.

[71] YUV CIF reference videos (lossless H.264 encoded). http://www2.tkn.tu-berlin.de/research/evalvid/cif.html.

[72] FFmpeg. http://ffmpeg.org/.

[73] EvalVid with GPAC. http://www2.tkn.tu-berlin.de/research/evalvid/EvalVid/docevalvid.html.

[74] The network simulator, ns-3. http://www.nsnam.org/.

[75] Lajos Hanzo, Peter J. Cherriman, and Jürgen Streit. *Wireless Video Communications - Second to Third Generation Systems and Beyond*. Wiley-IEEE, 2001.

[76] T. Komine and M. Nakagawa. Fundamental analysis for visible-light communication system using led lights. *Consumer Electronics, IEEE Transactions on*, 50(1):100–107, Feb.

[77] Kaiyun Cui, Gang Chen, Zhengyuan Xu, and R.D. Roberts. Line-of-sight visible light communication system design and demonstration. In *Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on*, pages 621–625, July.

[78] D. O'Brien, L. Zeng, Hoa Le-Minh, G. Faulkner, J.W. Walewski, and S. Randel. Visible light communications: Challenges and possibilities. In *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, pages 1–5, Sept.

[79] J.M. Kahn and J.R. Barry. Wireless infrared communications. *Proceedings of the IEEE*, 85(2):265–298, Feb.

[80] Kwonhyung Lee, Hyuncheol Park, and J.R. Barry. Indoor channel characteristics for visible light communications. *Communications Letters, IEEE*, 15(2):217–219, February.

[81] Francisco J. Lopez-Hernandez, Rafael Perez-Jimenez, and Asuncion Santamara. Ray-tracing algorithms for fast calculation of the channel impulse response on diffuse ir wireless indoor channels. *Optical Engineering*, 39(10):2775–2780, 2000.

[82] J.R. Barry, J.M. Kahn, W.J. Krause, E.A. Lee, and D.G. Messerschmitt. Simulation of multipath impulse response for indoor wireless optical channels. *Selected Areas in Communications, IEEE Journal on*, 11(3):367–379, Apr.

[83] F.J. Lopez-Hernandez, R. Perez-Jimeniz, and A. Santamaria. Monte carlo calculation of impulse response on diffuse ir wireless indoor channels. *Electronics Letters*, 34(12):1260–1262, Jun.

[84] F.J. Lopez-Hernandez, R. Perez-Jimenez, and A. Santamara. Ray-tracing algorithms for fast calculation of the channel impulse response on diffuse ir wireless indoor channels. *Optical Engineering*, 39(10):2775–2780, 2000.

[85] O. Gonzalez, S. Rodriguez, R. Perez-Jimenez, B.R. Mendoza, and A. Ayala. Error analysis of the simulated impulse response on indoor wireless optical channels using a monte carlo-based ray-tracing algorithm. *Communications, IEEE Transactions on*, 53(1):124–130, Jan.

[86] M.D. Audeh, J.M. Kahn, and J.R. Barry. Performance of pulse-position modulation on measured non-directed indoor infrared channels. *Communications, IEEE Transactions on*, 44(6):654–659, Jun.

[87] J.R. Barry. Sequence detection and equalization for pulse-position modulation. In *Communications, 1994. ICC '94, SUPERCOMM/ICC '94, Conference Record, 'Serving Humanity Through Communications.' IEEE International Conference on*, pages 1561–1565 vol.3, May.

[88] G.D. Forney. Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference. *Information Theory, IEEE Transactions on*, 18(3):363–378, May.

[89] Lubin Zeng, D. O"brien, Hoa Le-Minh, Kyungwoo Lee, Daekwang Jung, and Yunje Oh. Improvement of date rate by using equalization in an indoor visible light communication system. In *Circuits and Systems for Communications, 2008. ICCSC 2008. 4th IEEE International Conference on*, pages 678–682, May.