UNIVERSITY OF CALIFORNIA
RIVERSIDE

Survivability Considerations in Wireless Networks

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

Doctor of Philosophy

in

Computer Science

by

Jianxia Ning

December 2012

Dissertation Committee:

Dr. Srikanth V. Krishnamurthy, Chairperson
Dr. Harsha V. Madhyastha
Dr. Vassilis Tsotras
Dr. Ertem Tuncel

The Dissertation of Jianxia Ning is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I would like to express my heartfelt gratitude to my Ph.D advisor, Dr. Srikanth V. Krishnamurthy for supporting my research ideas and continuing to inspire me along this long but fulfilling road. It is thanks to his endless patience, compassion, and guidance that I decided to stay in graduate school and finish my doctorate. I wish to thank Dr. Harsha V. Madhyastha, for his academic support during our collaboration. I would like to thank Dr. Vassilis Tsotras and Dr. Ertem Tuncel for participating in the committee for my defense. I would like to thank my family, for supporting me throughput my pursuit of Ph.D. overseas and for all the sacrifices they bear when I could not be with them for many moments in years. I would especially like to thank Dr. Konstantinos Pelechrinis, Dr. Ioannis Broustis, Dr. Ece Gelal for providing help in many aspects of my research. I would like to acknowledge the valuable input from Dr. Ramesh Govindan in the forensic analysis project. I would like to thank all the people in the UCR Networking Lab for making graduate school a wonderful experience for me. Finally, I would like to thank Chen Huang and many more for joining me in the life journey.

To my parents and friends.

ABSTRACT OF THE DISSERTATION

Survivability Considerations in Wireless Networks

by

Jianxia Ning

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, December 2012
Dr. Srikanth V. Krishnamurthy, Chairperson

Survivability is usually defined as the capability of a network to deliver data successfully in a timely manner, even in the presence of attacks. Security as well as reliability are included in survivability considerations. Due to the lossy and open nature of wireless links, design of solutions to make the network survive against attacks as well as wireless idiosyncrasies is difficult. In this dissertation, we approach several issues that have to be considered in order to ensure survivability in various contexts. Firstly, we need to understand the causes for why the network is failing or does not deliver data efficiently as supposed. Network forensics could help with this matter. However, provisioning forensic evidence could impact the network performance in terms of throughput and delay. Secondly, we need to make the network reliable when switching from commonly used bands to another, because conventional network protocols may be affected by the characteristics of that particular frequency. Finally, we need to make sure there is security for the communication relying on the online social networks against active seekers of information.

For the first issue, we examine the problem of monitoring packet level transmissions and provisioning forensic evidence. We provide an analytical framework that computes the likelihood that forensic evidence exists with respects to packet transmissions. We validate our analytical framework via simulations and real-world experiments on two different wireless testbeds. Then we use the analytical framework as a basis for a protocol within a forensic analyzer to determine the likelihood that nodes drop packets. Our assessments are shown to be close to the ground truth, with an average deviation of 2.3%. Furthermore, we quantify the trade-offs between provisioning forensic evidence and achieving high performance in wireless networks. We find that the performance remains unaffected up to a certain evidence availability requirement. Beyond this, the throughput could degrade and the delay could increase by as much as 30%.

Then, this dissertation presents our work in evaluating neighbor discovery in 60 GHz band. Neighbor discovery is a fundamental process in the self-configuration of ad-hoc wireless networks. However, the unique physical layer characteristics of 60 GHz make neighbor discovery different from that in traditional 2.4 and 5 GHz bands. In particular, neighbors can be discovered via both direct and reflected beams. We analyze two neighbor discovery approaches in 60 GHz band. We examine the impact of system parameters on the discovery efficiency.

Finally, this dissertation includes our efforts in doing covert communication on public photo-sharing sites. Steganography can be used to embed secret messages in shared images. However, the image processing performed by photo sharing sites may destroy the

embedded messages. We provide an in-depth measurement study of the feasibility of hiding data on four popular photo sharing sites and identify the challenges in communicating covertly on these sites. We propose an approach of embedding secret messages while ensuring their integrity is preserved. We also present an approach for bootstrapping the communication without an out-of-band channel.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Wireless ad-hoc and mesh networks have drawn a lot of attention in the last decade, and will continue to be an important research topic in the future. Applications for these types of networks are numerous and diverse ranging from tactical deployment, disaster recovery to public safety, health and environmental applications. The most attractive merit of ad-hoc wireless networks is their ease of deployment compared to wired networks that need an infrastructure to be installed first to operate. However, such flexibility compromises the robustness of these networks. For example, the wireless communication medium is prone to channel impairments and interference, which cause the wireless link quality to change according to the channel conditions. Besides the natural wireless induced effects in the physical and medium access layers, a variety of malicious attacks such as jamming and packet dropping would cause packet delivery failures and hurt the network performance. These problems emphasize the need for research to enhance the capability of the network

to survive against attacks as well as wireless idiosyncrasies .

Survivability is usually defined as the capability of a network to deliver data successfully in a timely manner, even in the presence of attacks. Security as well as reliability are included in survivability considerations. Several issues have to be considered in order to ensure survivability in various contexts. First of all, we need to understand the causes for why the network is failing or is not efficient as supposed. Network forensics could help towards figuring out what has been wrong. However, provisioning forensic evidence could involve some overhead that affects network performance, since security and performance are usually studied separately in wireless protocol design. Secondly, we need to make the network reliable when moving from commonly used bands to another, because the characteristics of the particular frequency in use may affect the conventional network protocols thus disrupt the normal function of the network. Finally, we need to make sure there is security for the successful covert communication relying on the online social networks against active seekers of information. In this dissertation, we present our research in network survivability with regards to these issues in diverse contexts. We provide a brief summary of our work in what follows.

1.1 Survivability Considerations in the Context of Wireless Network Forensics

Forensic systems collect evidentiary data towards detecting malicious attacks such as packet dropping attacks [2, 3]. However, they do not make an analysis to distinguish

between wireless induced losses due to channel impairments and interference, and malicious discarding.

In the first work, our primary objective is to perform a forensic analysis on the cause of packet losses based on some macroscopic network parameters (such as traffic intensity and network density) in multi-hop wireless networks. Specically, we seek to answer the questions: (a) Given a set of macroscopic network parameters, what is the likelihood that evidence exists relating to transmissions? and, (b) How can one perform a forensic assessment to determine if packet losses on links are due to natural eects in a wireless network or due to malicious discarding, based on these macroscopic network parameters?

Our contribution in the problem of monitoring packet level transmissions and provisioning forensic evidence is as follows:

(1) Computing the likelihood of evidence availability. We construct an analytical framework for computing the likelihood of forensic evidence availability. We capture the factors that affect such availability on both individual links and on an end-to-end path. We find that the availability depends on network parameters such as packet size, bit-rates, traffic load and node density. We make several interesting observations on the trends in availability when tuning these parameters.

(2) Validating the analytical models via simulations and real experiments. We perform extensive simulations to validate our analytical framework. We also perform experiments on (a) a 802.11 testbed and, (b) a testbed with five WARP boards [5] towards our validation. We find that our analytical framework can adequately capture the likelihood of

evidence availability in real networks.

(3) Forensic analysis of packet losses. Our analytical framework facilitates the estimation of the likelihood of either a transmitter and/or a receiver discarding packets, given the conditions in the network. The framework is used as the basis for a protocol within a forensic analyzer. It takes as input (a) the network parameters and (b) monitoring logs for the considered link; it then yields the likelihood that the transmitter or the receiver on the link has discarded packets. We perform extensive simulations and compare the assessment results with ground truth. We find that our analyzer facilitates assessments with high accuracy; in particular, they deviate from the ground truth by 2.3%, on average.

Our work reveals that a number of operational network parameters may have to be tuned appropriately to aid the collection of evidence. However, choosing the operational parameter space towards facilitating evidence collection may negatively impact the data delivery performance. Building on the above efforts, we further seek to answer the question: If one were to tune network parameters to achieve a high degree of evidence availability, how is the performance affected? Our study reveals interesting dependencies between provisioning forensic evidence and achieving high network performance. To our best knowledge, this is the first study examining this trade-off. Our main contribution is analyzing and understanding the aforementioned trade-off between evidence availability and network performance. Our results indicate the presence of a tipping point. Specifically, we find that the best performance remains fairly unchanged if we were to turn the parameters so as to achieve a certain availability requirement. Beyond this, the maximum throughput drops

and the minimum delay increases considerably.

1.2 Survivability Considerations in the Context of 60 GHz Indoor Wireless Networks

The unlicensed 60 GHz band has been the focus of recent attention as a candidate for multimedia applications such as high definition video streaming over the wireless medium [34, 35, 36]. Compared with the maximum data rate of 54 Mbps supported by current wireless networks operating in the 2.4 GHz band (802.11 b/g) or the 5 GHz band 802.11a), the 60 GHz band can potentially support multi-gigabit data rates using 7 GHz of bandwidth.

The 60 GHz band has distinct signal propagation characteristics as compared with the 2-5 GHz band. In particular, the propagation loss in the 60 GHz band is about 20-30 dB higher than in the 2-5 GHz bands. The penetration loss is also higher in the 60 GHz band. The diffraction effects are much smaller in the 60 GHz band in comparison to the 2-5 GHz bands. The electromagnetic (EM) field of the 2-5 GHz signals is composed of diffracted and reflected waves. On the other hand, the EM field of a 60 GHz signal has a structure consisting of a few rays coming from the direct path and from first-order reflections. The experiments performed in indoor scenarios in [37, 38, 39] suggest that due to these reasons, communications between a transmitter and a receiver in this band are established via both direct and reflected beams.

The use of high-gain directional antennas can be especially attractive in the 60 GHz band, given the above propagation effects. They can significantly increase communication

range and this can be especially useful given the high propagation loss in typical indoor 60 GHz environments. Furthermore, the use of directional antennas can limit the number of reflected beams and thereby increase space reuse [37]. In addition, the mitigated diffraction effects make it possible for highly directional antennas to focus most of the transmitted energy at the intended recipient; this drastically reduces the interference between links in the same geographic area.

Neighbor discovery is an essential process in the self-configuration of ad-hoc wireless networks. Our objective is to analyze neighbor discovery in the 60 GHz band. Neighbor discovery in the considered setting is much different from that in traditional 2-5 GHz bands. In this band, neighbors can be discovered not only via direct line-of-sight (LOS) beams, but also via reflected beams, which has not been considered before. We consider two possible approaches for neighbor discovery, direct discovery and gossip-based discovery. With direct discovery, a node discovers a neighbor only when it successfully receives a transmission from that neighbor. With gossip-based discovery, a node can discover a neighbor either via direct discovery or from some other node (possibly a different neighbor) that has information about that neighbor. Our work provides a basis for establishing links in the 60 GHz regime and insights on the choice of operational parameters to employ under different conditions.

The major contributions we made are as follows:

(1) We develop analytical models for both direct discovery and gossip-based discovery in the 60 GHz band. The proposed models take into account both direct and reflected beams. These models can effectively characterize the performance of the considered

approaches in different settings; in particular, variations in node density, and directional beamwidth can be characterized.

(2) We build a simulation framework that reflects operations in the 60 GHz band and validate our analysis through extensive simulations. In particular, our simulation framework accounts for the presence of obstacles (such as exterior and interior walls) in an indoor wireless network. We use a ray-tracing method to simulate the interactions between signal propagation and the obstacles. Our implementation accounts for penetration and reflection with proper loss. The behaviors of directional antennas with varying beamwidth are also captured in the simulations.

(3) We comprehensively examine the impact of various key parameters (such as node density and antenna beamwidth) on the performance of the neighbor discovery schemes. Our extensive studies in some typical indoor scenarios reveal interesting trade-offs in performance, resulting from tuning the above parameters.

1.3 Survivability Considerations in the Context of Covert Communication in Public Photo-sharing Sites

The explosion of photo-sharing on publicly available websites opens the door for secret communications on the Internet; users can use steganography [59] to embed information in the photos that they upload. However, while the idea of using steganography to embed secret information in shared content on photo-sharing sites is conceptually attractive, it is not straight forward. There are several challenges that exist in creating a viable

covert channel of this type.

First, it is known that photo sharing sites often process uploaded images. While some of the processing functions are clearly specified on the photo-sharing sites [62, 63], not all such functions are publicly known. These (possibly unknown) processing functions often interfere with the use of steganography.

Second, steganography does not offer perfect secrecy [64]. Censors can try to read the embedded message by applying a list of extraction algorithms on every image. Thus, one will have to encrypt the secret information embedded in the shared photographs, to prevent exposure in the rare cases of interception. Encryption requires the establishment of secret keys between the communicating entities. Often, prior work assumes the existence of an out-of-band channel via which such keys are established. However, in many cases (e.g., where people are trying to hide information from government-controlled censors), the creation of such an out-of-band channel may be difficult (e.g., e-mail or voice calls may be monitored).

Our goal in this work is to address the above challenges and provide a framework for covert communication on public photo-sharing sites. In summary, our contributions are as follows:

(1) We perform an extensive measurement study that provides an understanding of the feasibility of hiding data with image steganography on four popular photo-sharing sites, viz., Google+, Facebook, Twitter and Flickr. Our study reveals the impact of the image processing done on these photo sharing sites on secretly embedded content.

(2) We propose a new simple way of embedding secret information in photos to preserve the integrity of secret messages on popular photo-sharing sites. With the aid of steganalysis tools, we demonstrate that our approach does not result in higher detection likelihood as compared to other commonly used embedding techniques.

(3) We propose an in-band approach for bootstrapping secret conversations. Specifically, we use the same channel, secret embedding on uploaded images, as that used for covert communications to exchange keys and other essential metadata.

1.4 Outline of this Dissertation

The rest of this dissertation is organized as follows. In Chapter 2, we present our analytical framework for computing the likelihood of forensic evidence availability. We capture the factors that affect such availability on both individual links and on an end-to-end path. We find that the availability depends on network parameters such as packet size, bit-rates, traffic load and node density. In Chapter 3, we quantify the trade-offs between provisioning forensic evidence and achieving high performance in wireless networks. We find that the performance remains unaffected up to a certain evidence availability requirement. Beyond this, the throughput could degrade and the delay could increase by as much as 30%. In Chapter 4, we present our work in evaluating neighbor discovery in 60 GHz band. We give analytical models for both direct discovery and gossip-based discovery in the 60 GHz band. We examine the impact of system parameters on the discovery efficiency. Chapter 5 presents our efforts in covert communication on public photo-sharing sites with image

steganography. We provide an in-depth measurement study of the feasibility of hiding data on four popular photo sharing sites and identify the challenges in covertly communicating on these sites. We propose an approach of intelligently embedding secret messages while ensuring their integrity is preserved. We also present an approach for bootstrapping the covert communication without an out-of-band channel.

Chapter 2

Forensics Analysis of Packet Losses in Wireless Networks

Due to the lossy nature of wireless links, it is difficult to determine if packet losses are due to wireless-induced effects or from malicious discarding. Many prior efforts on detecting malicious packet drops rely on evidence collected via passive monitoring by neighbor nodes; however, they do not analyze the cause of packet losses. In this work, we ask: (a) Given certain macroscopic parameters of the network (like traffic intensity and node density) what is the likelihood that evidence exists with respect to a transmission? and, (b) How can these parameters be used to perform a forensic analysis of the reason for the losses? Towards answering the above questions, we first build an analytical framework that computes the likelihood that evidence (we call this transmission evidence or TE for short) exists with respect to transmissions, in terms of a set of network parameters. We

validate our analytical framework via both simulations as well as real-world experiments on two different wireless testbeds. The analytical framework is then used as a basis for a protocol within a forensic analyzer to assess the cause of packet losses and determine the likelihood of forwarding misbehaviors. Through simulations, we find that our assessments are close to the ground truth in all examined cases, with an average deviation of 2.3% from the ground truth and a worst case deviation of 15.0%.

2.1 Introduction

Wireless ad hoc and mesh networks find application in municipal networks, tactical deployments and disaster recovery missions. In such networks, packet forwarding along a path is an inherent functional requirement. There have been studies on packet dropping attacks, wherein malicious routers that are required to forward packets do not do so (e.g. [1]). Unfortunately, due to the lossy nature of wireless links it is not easy to determine whether packet losses are due to natural wireless induced effects (channel impairments or interference) or due to such malicious drops.

Forensic systems typically collect evidentiary data towards detecting such packet dropping attacks (e.g. [2, 3]); however, they do not make any analysis to distinguish between wireless induced losses and malicious drops. Nodes that are part of the network themselves may act as witnesses and monitor transmissions [4]; this is an attractive option when networks are rapidly deployed and dedicated monitoring nodes are unavailable. Depending on the deployment, witnesses may not have evidence (e.g., due to very few witnesses or because

of high levels of interference) relating to certain transmissions.

In this paper, our primary objective is to perform a forensic analysis on the cause of packet losses based on some macroscopic network parameters (such as traffic intensity and network density) in multi-hop wireless networks. Specifically, we seek to answer the questions: **(a)** Given a set of macroscopic network parameters, what is the likelihood that evidence exists relating to transmissions? and, **(b)** How can one perform a forensic assessment to determine if packet losses on links are due to natural effects in a wireless network or due to malicious discarding, based on these macroscopic network parameters?

Towards answering the above questions, we construct an analytical framework that takes as input, macroscopic measurements or configurations of network properties (as alluded to above) and provides as output the probability that evidence exists relating to transmissions. We call this evidence, "transmission evidence" or TE for short. The analytical framework forms the basis for a protocol used within a forensic analyzer to assess the most likely cause of packet losses on links.

In particular, our contributions in this paper are as follows:

(i) Computing the likelihood of TE availability: We construct an analytical framework for computing the likelihood of TE availability. We capture the factors that affect TE availability on both individual links and on an end-to-end path. We find that the availability depends on network parameters such as packet size, bit-rates, traffic load and node density. We make several interesting observations on the trends in TE availability when tuning these parameters.

(ii) Validating the analytical models via simulations and real experiments: We perform extensive simulations to validate our analytical framework. We also perform experiments on (a) a 802.11 testbed and, (b) a testbed with five WARP boards [5] towards our validation. We find that our analytical framework can adequately capture the likelihood of TE availability in real networks.

(iii) Forensic analysis of packet losses: Our analytical framework facilitates the estimation of the likelihood of either a transmitter and/or a receiver discarding packets, given the conditions in the network. The framework is used as the basis for a protocol within a forensic analyzer. It takes as input (a) the network parameters and (b) monitoring logs for the considered link; it then yields the likelihood that the transmitter or the receiver on the link has discarded packets. We perform extensive simulations and compare the assessment results with ground truth. We find that our analyzer facilitates assessments with high accuracy; in particular, they deviate from the ground truth by 2.3%, on average.

Scope of our work: Our analytical models yield a quick and effective way of capturing the TE availability in large sets of scenarios. The advantage of the approach is that only a coarse estimate of network parameters is used in order to make the assessments. In this paper we have considered relatively static, homogeneous (e.g. a single packet size is used) settings; even for this, the construction of the analytical framework that forms the basis of our forensic analyzer, is non-trivial. More importantly, the models capture the trends in TE availability in practice as validated by both our simulations and experiments on real systems. A consideration of more complex settings is left to future work.

The output of our forensic analyzer provides coarse-grained assessments on forwarding misbehaviors. Because of the generality of the analytical framework applied therein, the assessments of the cause of packet losses on specific links inevitably deviate from the ground truth. However, our evaluations show that in all cases we examine, the average deviation from the truth is about 2.3%, while the maximum deviation is 15.0%.

We wish to point out here that since malicious drops are always supplementary to wireless induced losses, it is impossible for an attacker to exactly mimic natural wireless effects. The likelihood of an attack being detected will directly depend on the aggressiveness of the attacker; the more the drops, the more the deviation from what is expected due to natural wireless effects and thus, the higher the chance of detection.

Organization: The paper is structured as follows. Section 3.2 discusses related work. In Section 3.3, we provide a description of our analytical framework. We consider specific network parameters and apply these in our framework in Section 2.4. The applicability of our framework in a forensic analyzer is discussed in Section 2.5. In Section 2.6, we present our performance evaluations. We conclude in Section 4.6.

2.2 Related Work

In this section we briefly discuss related literature on network forensics and analyzing packet losses in wireless networks.

2.2.1 Network Forensics

There is prior work on wireless monitoring at the *mechanism and system design* level [1, 6, 3, 7, 8, 9, 10, 11]. Marti *et al.* design a *watchdog* scheme to identify malicious nodes which do not forward packets along a multi-hop path. McGrath *et al.* [6] design and implement FLUX; FLUX automates the collection of forensic data and identifies abnormal traffic and network weaknesses. Ramach *et al.* [3] design and implement DAMON, a distributed monitoring system for MANETs. In summary, almost all of the above approaches propose techniques for solving specific network problems that require evidentiary data. None of them study the impact of various network parameters on the collection of evidence as we do here. Moreover, only a few use the evidentiary data to detect packet dropping attacks [3, 8]; however, unlike in these efforts we try to determine the likely cause of packet losses using a macroscopic view of network parameters.

2.2.2 Analyzing Packet Losses

ETX [12] and ETT [13] are metrics that have been designed to estimate the packet delivery ratio on links; however, they are empirical and more importantly these metrics reflect the packet loss rate but do not give insights into the root-cause of packet losses.

Some prior efforts attempt to distinguish packet losses due to interference from those due to channel fading [14, 15, 16]. Reis *et al.* present models for the physical layer behaviors of static wireless networks, focusing on the successful packet reception and carrier sense with interference. Qiu *et al.* propose a general model that is able to capture collision-

induced losses in multihop wireless networks, based a limited number of measurements. Wong *et al.* propose Robust Rate Adaptation Algorithm (RRAA), by which they try to differentiate between fading-related and collision losses by leveraging the RTS option. While these work look into the cause of packet losses due to wireless physical and medium access layer effects, none of these efforts however, consider the possibility of malicious discarding of packets.

The work that is closest to ours is in [4]. It proposes a specific witness-based detection scheme to identify forwarding misbehaviors. The authors analytically show that their scheme has low false positive and false negative rates. However, they do not evaluate how various network parameters would affect the evidence availability. To our best knowledge, we are the first to propose analytical models and experimental validation for this purpose.

2.3 Our Analytical Framework

In this section, we develop our analytical framework to compute the likelihood of TE availability. At this time, we assume that neither the transmitter nor the receiver discards packets maliciously. We defer a discussion of how our framework can be applied in a forensic analyzer to identify such possibilities, to Section 2.5.

2.3.1 Evidence Maintenance

In a multi-hop static wireless network, nodes maintain evidence relating to transmissions as follows: **(a)** A sender (or transmitter) keeps the signed ACK it receives for each

packet it sends. **(b)** A receiver creates an entry locally for each unique packet received and digitally verified. **(c)** A monitoring (witness) node creates an entry locally for each packet that it overhears and verifies. We assume that storage is not a limiting factor in evidence collection; one can envision nodes sending coarse-grained information relating to collected evidence periodically, to a central forensic controller. The overhead due to digital signatures has been previously studied (e.g., [17]) and hence is not considered here. The signed ACK helps in assuring non-repudiation. The sender can validate that it sent the packet in question and the receiver cannot deny receiving the packet. Without loss of generality, we assume that an ACK includes sender and receiver IDs and thus, an overheard ACK is of evidentiary value.

We expect that evidence is only sent infrequently to the controller since our objective is to investigate long-term effects. It can be easily piggybacked onto other control information (e.g. routing updates) and thus, we expect that the overhead is likely to be small. Our focus in this paper is more on the forensic analysis itself and not on the evidence collection process; thus, we do not perform an analysis of the overhead consumed due to evidence gathering.

2.3.2 Hop-level Transmission Evidence (HTE)

The availability of hop-level TE reflects the likelihood that evidence exists relating to transmissions on a link. In other words, the availability of HTE reflects the level of confidence provided to a network forensic system, about the occurrence of a transmission on a single link.

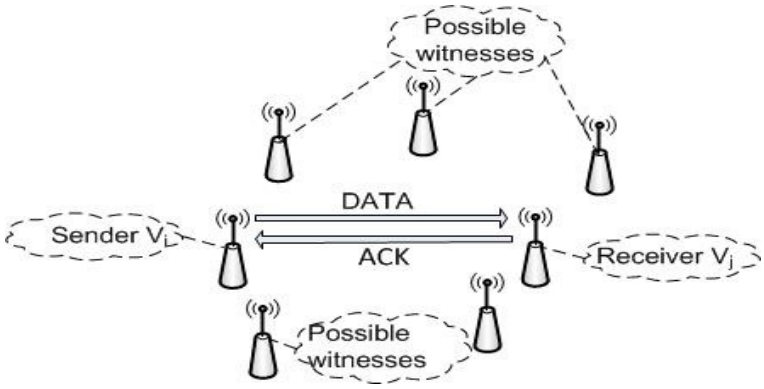


Figure 2.1: Three sources of evidence for a transmission from v_i to v_j: last hop v_i, next hop v_j, and other nodes serving as possible witnesses

As stated above, for a transmission between v_i and v_j, there are three sources of evidence. The locations of these sources are shown in Figure 2.1. Commonly used notations are enlisted in Table 3.1.

Source 1: v_i ***has the signed ACK from*** v_j ***for packet*** $(src, dest, pkt\ SQN)$. This requires that: **(a)** v_i's data packet is successfully received by v_j and, **(b)** v_j's ACK packet is successfully received by v_i.

$Pr\left(succ \mid r, l_D\right)$ denotes the probability of a successful data transmission with rate r and packet length l_D. Similarly, the probability of a successful ACK transmission is $Pr\left(succ \mid r_0, l_A\right)$, assuming that an ACK is sent at the base rate r_0 and has a length l_A. The probability that the first source of evidence is available is

$$Pr_{src1} = Pr\left(succ \mid r, l_D\right) \cdot Pr\left(succ \mid r_0, l_A\right). \tag{2.1}$$

Source 2: v_j ***has a stored entry*** $|v_i|src|dest|pkt\ SQN|\ timestamp|$. This

N	Total number of nodes
v_i	Transmitter
v_j	Receiver
d_{v_i,v_j}	Distance between v_i and v_j
P_{v_i,v_j}	Received power at v_j from v_i
h_{v_i,v_j}	Channel attenuation between v_j and v_i
η	Expected value of $\|h_{v_i,v_j}\|^2$
P_t	Transmission power
P_n	Noise power
α	Path loss exponent
z	Number of interferers
Z	Set of interferers
λ	Expected traffic sent per node in unit time
Λ	Expected interference level perceived by a node projected from another node in unit time
r	Transmission bit-rate
γ	SINR threshold
l_D	data packet length
l_A	ACK packet length

Table 2.1: Notations

source of TE requires a successful transmission from v_i to v_j, the probability of which is

$$Pr_{src2} = Pr\left(succ \mid r, l_D\right). \tag{2.2}$$

Source 3: At least one witness has a stored entry $|v_i|src|\ dest|pkt\ SQN|timestamp|$. This requires at least a node other than v_i or v_j to overhear the data transmission from v_i or the ACK transmission from v_j.

Let Pr_{src3_D} and Pr_{src3_A} denote the probabilities that at least one witness overhears the data and ACK, respectively. Note that due to the half-duplex property of typical

radio devices, it is assumed that a node cannot be an interferer and a witness for the same transmission[1]. Therefore, when there are totally N nodes in the network and z interferers, the number of witnesses cannot exceed $N - z - 2$. Thus we have:

$$Pr_{src3\_D} = \sum_{z=0}^{N-2} Pr\left(\mathrm{z}\ \ int \mid r, l_D\right) \cdot \left(1 - \left(1 - Pr\left(succ \mid r, l_D, z\right)\right)^{N-z-2}\right), \tag{2.3}$$

in which $\left(1 - \left(1 - Pr\left(succ \mid r, l_D, z\right)\right)^{N-z-2}\right)$ is the probability that given z interferers, at least one witness overhears the data transmission. Considering all possible values that z can take, we get the marginal probability that at least one witness overhears a given transmission.

In a similar way, we compute Pr_{src3_A} as follows[2]

$$Pr_{src3\_A} = Pr\left(succ \mid r, l_D\right) \cdot \left(\sum_{z=0}^{N-2} Pr\left(\mathrm{z}\ \ int \mid r_0, l_A\right) \cdot \left(1 - \left(1 - Pr\left(succ \mid r_0, l_A, z\right)\right)^{N-z-2}\right)\right). \tag{2.4}$$

The successful overhearing of data and ACK transmissions by any given witness does not affect each other. In other words, these two events are assumed to be independent. In reality, there may be correlations due to interference effects at the overhearing node. However, the assumption (which we make for tractability) is shown to be reasonable by our simulations/experiments. With this:

$$Pr_{src3} = Pr_{src3\_D} + \left(1 - Pr_{src3\_D}\right) \cdot Pr_{src3\_A}. \tag{2.5}$$

[1]Without loss of generality we assume that the monitoring devices or witnesses are active nodes in the network. It is easy to modify the analysis if evidence is collected only by passive monitoring nodes.

[2]For v_j to transmit an ACK, it must have successfully received the corresponding data packet.

Hop-level TE availability: The probability that at least one source of TE is available for a transmission, under the assumption of evidence independence[3] is

$$\begin{aligned} Pr_{HTE} &= 1 - \prod_{i=1}^{3} Pr(source\ i\ is\ unavailable) \\ &= 1 - (1 - Pr_{src1}) \cdot (1 - Pr_{src2}) \cdot (1 - Pr_{src3}). \end{aligned} \tag{2.6}$$

Accounting for retransmissions: Next we consider a limit of n_r retransmissions for the same data packet. The success of each transmission is independent from that of another (assuming that these are staggered in time, this is a reasonable assumption since the temporal network conditions are likely to change). A successful exchange of data and ACK packets results in the termination of retransmission attempts. This probability of successful exchange, denoted by Pr_{succ_ex} is

$$Pr_{succ\_ex} = Pr\left(succ \mid r, l_D\right) \cdot Pr\left(succ \mid r_0, l_A\right), \tag{2.7}$$

The probability that there are i retransmissions ($i + 1$ transmission attempts) is denoted as $Pr(rtx)$ and is given by:

$$Pr(rtx = i) = \begin{cases} Pr_{succ\_ex} \cdot (1 - Pr_{succ\_ex})^i & 0 \leq i \leq n_r - 1 \\ 1 - \sum_{j=0}^{n_r - 1} Pr(rtx = j) & i = n_r \end{cases}.$$

Hence, the TE availability probability with a retransmission limit of n_r is:

[3]Dependencies between sources of evidence are discussed in Section 2.5.

$$Pr_{HTE}[\,n_r\,] = \sum_{i=0}^{n_r} Pr(rtx = i) \cdot \left(1 - (1 - Pr_{HTE})^{i+1}\right). \tag{2.8}$$

2.3.3 Path-level Transmission Evidence (PTE)

Next we look at the path-level TE, i.e., the evidence relating to all transmissions on an end-to-end path. The TE availability on each hop along the path is assumed to be independent of that on the other hops. Again, in reality the TE availability across hops may be correlated but we make this assumption for tractability; our simulations and experiments (where there is correlation) verify that this assumption is indeed acceptable. The PTE requires the HTE on all the hops of the path. The PTE for a H-hop path, denoted by $Pr_{PTE}[\,H\,]$, is given by:

$$Pr_{PTE}[\,H\,] = \prod_{h=1}^{H} Pr_{HTE}[\,at\ h^{th}\ hop\,]. \tag{2.9}$$

Bit-rate selection: Different bit-rates used on different hops will cause the TE availability on each hop to differ. The bit rate used on a link depends not only on the physical conditions (e.g., the distance between the communicating pair, the temporal fluctuations due to fading) but also on the rate adaptation algorithm in use. Given these, it is difficult to come up with a distribution for the bit rates used by nodes in a network. For simplicity, we assume that a bit rate is selected randomly from among the set of available rates. Note however, that our analysis can easily incorporate other distributions characterizing the usage

of different bit rates. The probability of PTE availability is computed by considering all possible combinations of rates, on each hop of the path.

With (3.10), we see that the same parameters that affect HTE affect PTE. In addition, the hop count H impacts PTE; generally, as one may expect the longer the path, the lower PTE.

Note that the PTE as defined here is strict in the sense that it requires HTE on all hops. The TE of the transmission on hop h, can imply the success of transmissions on the previous $h-1$ hops, even though the HTE may not be available for all such hops. We will consider this sort of implicit PTE in future work.

2.4 Explicitly Computing the Likelihood of Evidence Availability

Now that we have computed the high level formulation of the likelihood of TE availability in Section 3.3, we need compute the probabilities of success in (3.1), (3.2) and (3.5). However, in order to do so, we need to provide specific characteristics of the network. We proceed to do so in this section using commonly used models for representing the channel, the node density and the generated traffic; the models seem to characterize practical settings with good accuracy as seen in our real experiments later. Note here that, other models can be easily incorporated into our generic analytical framework.

2.4.1 Specific Characteristics of the Network

The channel model: The received signal strength from node v_i, at node v_j is:

$$P_{v_i,v_j} = \frac{P_t \cdot |h_{v_i,v_j}|^2}{d^{\alpha}_{v_i,v_j}}, \tag{2.10}$$

where, P_t is the transmission power. h_{v_i,v_j} is the attenuation due to fading between the communicating pair. As typical, we assume that h_{v_i,v_j} is a Rayleigh distributed random variable [18]; thus, $|h_{v_i,v_j}|^2$ is exponentially distributed. d_{v_i,v_j} is the distance between v_i and v_j. α is the path loss exponent.

The collision model: There are several models used to capture collisions in the literature [19]. We use the *SINR (Signal-to-Interference-and-Noise) physical model*, where node v_j successfully receives the transmission from node v_i *iff*:

$$\frac{P_{v_i,v_j}}{P_n + \sum\limits_{k\in\{1,\ldots,N\}\setminus\{i,j\}} P_{v_k,v_j}} > \gamma, \tag{2.11}$$

where, P_{v_i,v_j} is the received power from v_i to v_j and is computed using (3.17). P_n is noise power. v_k is one of the interfering nodes. $\sum_{k\in\{1,\ldots,N\}\setminus\{i,j\}} P_{v_k,v_j}$ is the accumulative interference power perceived by v_j. γ is the SINR threshold which varies with transmission bit-rate.

Use of multiple bit rates: The data packets are sent at a chosen transmission bit-rate from a set of available rates. For each rate there is a corresponding SINR threshold. The ACK packet is sent at the base rate.

Media access control (MAC): MAC provides nodes an access mechanism on a shared medium. The collision handling capability is different across different MAC protocols. To remove protocol dependencies, we do not assume a specific MAC scheme. Instead, we use a parameter to characterize the interference that nodes perceive, which in turn reflects the interference resolving ability of the MAC in use. This simplified representation avoids modeling the operations of specific MACs. As demonstrated later in Section 2.6, this model can be used to characterize multiple commonly used MAC protocols.

Node distribution: The network consists of N uniformly distributed static nodes (v_i, $i \in \{1, ..., N\}$).

Traffic pattern: Nodes send Poisson traffic, including their own packets and those to be simply forwarded.

2.4.2 Computing Transmission Evidence Availability

Towards computing (3.6), we start by considering a transmission of a data packet from v_i to v_j. Given the distance between them d_{v_i,v_j}, the transmission bit-rate in use r, packet length l_D, and the number of interferers z, the probability of the transmission succeeding (denoted as $Pr\left(succ \mid r, l_D, d_{v_i,v_j}, z\right)$) is:

$$Pr\left(succ \mid r, l_D, d_{v_i,v_j}, z\right) = Pr\left(\frac{P_{v_i,v_j}}{P_n + \sum_{k \in Z} P_{v_k,v_j}} > \gamma\right), \tag{2.12}$$

where Z is the set of interferers. $Z \subset \{1, ..., N\} \backslash \{i, j\}$ and $|Z| = z$. The value of γ here, corresponds to the rate r in use.

With respect to the right hand side (RHS) of (2.12) there are two cases: (i) in the absence of interference (when $z = 0$) and, (ii) with the presence of interference (when $1 \leq z \leq N-2$). Detailed derivations of (2.12) for these two cases, are presented in Section 2.4.3.

Next, we remove the conditioning on the number of interferers z from $Pr\left(succ \mid r, l_D, d_{v_i,v_j}, z\right)$. Λ is a parameter that captures the expected interference at a given node from a neighbor node, per unit time. In reality this is dependent on both the traffic intensity and the MAC protocol in use. However, we try to capture the interference experienced at a node, simply with this parameter. If interference is managed (e.g. with TDMA or CSMA/CA), Λ is likely to be low. If the interference is unmanaged (as with say Aloha) and is high, Λ will be high. If we assume asynchronous transmissions and fixed sized packets, it is easy to see that a packet is interfered with, if another node initiates a transmission within the packet transmission time (say τ), or for a duration of τ prior to the beginning of the intended transmission (similar to the analysis of the Aloha medium access scheme in [20]). Thus, if the traffic load of a node is λ, the projected interference load can be characterized by 2λ. Hence in this specific case, $\Lambda = 2\lambda$. When the traffic load is Poisson, the probability that a transmission does not *overlap* with the intended transmission between v_i and v_j is:

$$g(0, \Lambda\frac{l_D}{r}) = e^{-\Lambda\frac{l_D}{r}}\left(\Lambda\frac{l_D}{r}\right). \tag{2.13}$$

The probability that there are z $(0 \leq z \leq N-2)$ interference sources during the data packet transmission time $\frac{l_D}{r}$ is:

$$Pr\left(\text{z } int \mid r, l_D\right) = \left(1 - g(0, \Lambda\frac{l_D}{r})\right)^z \cdot \{g(0, \Lambda\frac{l_D}{r})\}^{(N-2-z)}. \tag{2.14}$$

The probability of a successful data transmission given the bit-rate and the packet length in use, and the distance between the communicating pair, is:

$$\begin{aligned} &Pr\left(succ \mid r, l_D, d_{v_i,v_j}\right) \\ &= \sum_{z=0}^{N-2} Pr\left(\text{z } int \mid r, l_D\right) \cdot Pr\left(succ \mid r, l_D, d_{v_i,v_j}, z\right). \end{aligned} \tag{2.15}$$

Next we remove the conditioning on d_{v_i,v_j} from $Pr\left(succ \mid r,\ \ l_D, d_{v_i,v_j}\right)$. As discussed, if one were to assume a uniform node deployment distribution, the PDF of d_{v_i,v_j} is $\frac{2d}{R^2}$. Thus:

$$Pr\left(succ \mid r, l_D\right) = \int_0^R Pr\left(succ \mid r, l_D, d_{v_i,v_j}\right) \frac{2d}{R^2} dd. \tag{2.16}$$

We emphasize that (2.14) and (2.16) can be easily modified to incorporate other distributions of interference levels and node deployments.

$Pr\left(succ \mid r, l_D, z\right)$ in (3.3), is obtained by removing the conditioning on d_{v_i,v_j} from $Pr\left(succ \mid r,\ \ l_D, d_{v_i,v_j}, z\right)$ (similar to that in (2.16)). Together with $Pr\left(\text{z } int \mid r, l_D\right)$ and $Pr\left(succ \mid r,\ \ l_D\right)$, we get Pr_{src1}, Pr_{src2} and Pr_{src3}, and finally the probability that hop-level TE is available (Pr_{HTE}) in (3.6).

Summary: TE analytical models take as inputs a set of network parameters (node distribution, data traffic volume, interference level and etc.) and output the likelihood that

the evidence for transmissions exist in the network. The values of these network parameters can be pre-configured by the network administrator, otherwise the approximate values can be measured in real time.

2.4.3 The Probability of a Successful Data Transmission

The notation used here is carried over from Table 3.1.

There are two cases when considering (2.12). In the absence of interference,

$$\begin{aligned} &Pr\left(succ \mid r, l_D, d_{v_i,v_j}, z=0\right) = Pr\left(\frac{P_{v_i,v_j}}{P_n} > \gamma\right) \\ &= Pr\left(|h_{v_i,v_j}|^2 > \frac{d^{\alpha}_{v_i,v_j} \cdot P_n \cdot \gamma}{P_t}\right). \end{aligned} \tag{2.17}$$

We denote $\frac{d^{\alpha}_{v_i,v_j} \cdot P_n \cdot \gamma}{P_t}$ by c. Recall that $|h_{v_i,v_j}|^2$ is an exponentially distributed r.v. with parameter η. Thus,

$$Pr\left(|h_{v_i,v_j}|^2 > c\right) = \int_c^{\infty} \eta e^{-\eta x} dx. \tag{2.18}$$

With interference, the success probability is computed as:

$$\begin{aligned} &Pr\left(succ \mid r, l_D, d_{v_i,v_j}, 1 \le z \le N-2\right) \\ &= Pr\left(\frac{P_{v_i,v_j}}{P_n + \sum_{k \in Z} P_{v_k,v_j}} > \gamma\right) \\ &= Pr\left(\frac{P_t \cdot |h_{v_i,v_j}|^2}{d^{\alpha}_{v_i,v_j}} > \gamma \cdot \sum_{k \in Z} \frac{P_t \cdot |h_{v_k,v_j}|^2}{d^{\alpha}_{v_k,v_j}} + \gamma \cdot P_n\right) \end{aligned}$$

It is difficult to compute the above since the distances from v_j, to different interferers will be different. For tractability, we make a conservative approximation that all the interferers are at the same distance as that to the closest interferer to v_j[4] (denoted as $min\{d_{v_k,v_j}\}$). With this, we find a lower bound on the success probability (upper bound on failure probability) as follows:

$$\begin{aligned}
&Pr\left(succ \mid r, l_D, d_{v_i,v_j}, 1 \leq z \leq N-2\right) \\
&> Pr\left(\frac{P_t \cdot |h_{v_i,v_j}|^2}{d^{\alpha}_{v_i,v_j}} > \gamma \cdot \sum_{k \in Z} \frac{P_t \cdot |h_{v_k,v_j}|^2}{min\{d_{v_k,v_j}\}^{\alpha}} + \gamma \cdot P_n\right) \\
&= Pr\left(P_t \cdot |h_{v_i,v_j}|^2 \cdot min\{d_{v_k,v_j}\}^{\alpha} > \right. \\
&\left. \gamma \cdot P_t \cdot \sum_{k \in Z} |h_{v_k,v_j}|^2 \cdot d^{\alpha}_{v_i,v_j} + \gamma \cdot min\{d_{v_k,v_j}\}^{\alpha} \cdot P_n \cdot d^{\alpha}_{v_i,v_j}\right).
\end{aligned} \tag{2.19}$$

On the RHS of (2.19), there are three r.v.s $min\{d_{v_k,v_j}\}^{\alpha}$, $\sum_{k \in Z} |h_{v_k,v_j}|^2$ and $|h_{v_k,v_j}|^2$. Since $|h_{v_k,v_j}|^2$ is exponentially distributed with parameter η, the sum $\sum_{k \in Z} |h_{v_k,v_j}|^2$ follows an Erlang distribution with parameters η and z [21]. Next we derive the distribution of $min\{d_{v_k,v_j}\}^{\alpha}$ (d^{α}_{min} for short):

$$F_{d_{min}}(d) = Pr(d_{min} \leq d) = 1 - Pr(d_{min} > d). \tag{2.20}$$

Recall that d_{min} is the minimum of the distances from the z interferers to v_j (denoted by $d_1, d_2, \cdots, d_z$). Thus,

[4] Our simulations validate that this is reasonable.

$$\begin{aligned} 1 - Pr(d_{min} > d) &= 1 - Pr(min\{d_1, d_2, ..., d_z\} > d) \\ &= 1 - Pr(d_1 > d, d_2 > d, ..., d_z > d). \end{aligned} \tag{2.21}$$

Since $d_1, d_2, \cdots d_z$ are independent r.v.s with the same distribution, (2.21) can be written as:

$$\begin{aligned} &1 - Pr(d_1 > d) \cdot Pr(d_2 > d) \cdots Pr(d_z > d) = 1 - Pr(\hat{d} > d)^z \\ &= 1 - \Big(1 - Pr(\hat{d} \leq d)\Big)^z = 1 - \big(1 - F_{\hat{d}}(d)\big)^z \end{aligned} \tag{2.22}$$

where $\hat{d}$ is the distance from an interferer to v_j. The probability distribution of $\hat{d}$ is simply the distribution of distance between node pairs in the network (since the interferer could be anywhere within the range of v_j). If one assumes a uniform deployment of nodes, the probability density function (PDF) that a node is d units away from another node is $\frac{2d}{R^2}$ where R is the maximum possible distance units between a pair.

The PDF of d_{min}, $f_{d_{min}}(d)$ is then given by:

$$\begin{aligned} f_{d_{min}}(d) &= \frac{d}{dd} F_{d_{min}}(d) = z \cdot (1 - F_{\hat{d}}(d))^{(z-1)} \cdot f_{\hat{d}}(d) \\ &= z \cdot \left(1 - \frac{d^2}{R^2}\right)^{z-1} \cdot \frac{2d}{R^2}. \end{aligned} \tag{2.23}$$

The PDF of d_{min}^{α} (denoted as $\tilde{d}$), a function of d_{min}, is expressed as [21]:

$$f_{\tilde{d}}(d) = \frac{1}{\alpha} \cdot d^{(\frac{1}{\alpha} - 1)} \cdot f_{d_{min}}(d^{\frac{1}{\alpha}}). \tag{2.24}$$

We now have the PDFs for the three r.v.s $min\{d_{v_k,v_j}\}^\alpha$, $\sum_{k\in Z}|h_{v_k,v_j}|^2$ and $|h_{v_i,v_j}|^2$. To compute (2.19), we need to further get the PDFs for $\{P_t\cdot|h_{v_i,v_j}|^2\cdot min\{d_{v_k,v_j}\}^\alpha\}$ (denoted as a new r.v. V) and $\{\gamma\cdot P_t\cdot\sum_{k\in Z}|h_{v_k,v_j}|^2\cdot d^\alpha_{v_i,v_j}+\gamma\cdot min\{d_{v_k,v_j}\}^\alpha\cdot P_n\cdot d^\alpha_{v_i,v_j}\}$ (denoted as r.v. U).

We start by computing the PDF of $\{|h_{v_i,v_j}|^2\cdot min\{d_{v_k,v_j}\}^\alpha\}$ (denoted as r.v. W). Note that $|h_{v_i,v_j}|^2$ (Y for short) and $min\{d_{v_k,v_j}\}^\alpha$ ($\tilde{d}$) are independent. Further, $\tilde{d}$ varies from 0 to $min(R^\alpha, w/y)$ where, w and y are variables representing the value assumed by r.v.s W and Y, respectively. Thus,

$$F_W(w)=\int_{\frac{w}{R^\alpha}}^{\infty}\int_0^{\frac{w}{y}} f_Y(y)f_{\tilde{d}}(d)dd\,dy+\int_0^{\frac{w}{R\alpha}}\int_0^{R^\alpha} f_Y(y)f_{\tilde{d}}(d)dd\,dy. \tag{2.25}$$

Differentiating (2.25) yields $f_W(w)$. Since $V=P_t\cdot W$:

$$f_V(v)=\frac{1}{P_t}\cdot f_W(\frac{v}{P_t}). \tag{2.26}$$

The derivation of the PDF of U is similar to that of V and is omitted due to space constraints.

With the new notation, the RHS of (2.19) is

$$Pr(V>U)=\int_0^{\infty}\int_0^{v} f_V(v)f_U(u)du\,dv. \tag{2.27}$$

Having derived (2.18) and (2.19), we have the expression for $Pr\left(succ \mid r,\ l_D, d_{v_i,v_j}, z\right)$.

2.5 Our Forensic Analyzer

Our analytical framework is used as the basis for a protocol within a forensic analyzer. Using the framework the analyzer computes offline, the probabilities of packet losses and TE availability under different conditions, in a benign setting on a link, based on a set of network parameters. It then compares these computed values with what is observed during network operations to estimate the likelihood of a transmitter or receiver discarding packets and lying about the same. As discussed earlier, the packet losses due to malicious dropping will always be in addition to what is experienced in benign settings due to wireless effects. The more aggressive an attacker, the more will be the deviation between what is observed and the expected number of packet losses in benign settings.

In this section, we describe our forensic analyzer in detail.

2.5.1 Performing the Forensics Analysis

As illustrated in Fig. 2.2, the forensic analyzer takes as inputs 1) the estimated probabilities from our analytical framework and 2) the evidence collected by nodes at run-time and the packet delivery ratios (PDRs) reported by the receivers. It outputs the assessment results on possible forwarding misbehaviors as discussed below.

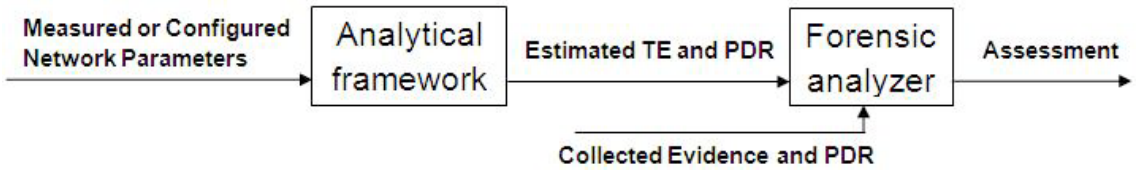


Figure 2.2: Forensic analyzer

Forwarding misbehaviors: Nodes on an end-to-end path in a multi-hop wireless network may indulge in forwarding misbehaviors. A ***lying transmitter*** may claim to have attempted to forward packets, but may not have done so. Evidence for the transmissions that did not occur will not exist. A ***lying receiver*** may claim to have not received packets that were in fact received. If a receiver denies receiving packets, the only source of TE comes from any witness overhearing the data transmissions (available with probability Pr_{src3_D}). We wish to point out here that we do not differentiate between misbehaviors due to malicious activity and that from misconfigurations.

Threat model: In this work, we only consider forwarding misbehaviors as above. We assume that the network parameters are accurately gathered and nodes do not lie with regards to these parameters. We assume that keys cannot be compromised to create fake signatures. We also assume that there is no evidence manipulation i.e., none of the nodes create fake evidence or delete the genuine evidence. While a receiver discards packets as above, we assume it still follows the protocol in sending ACKs (only) for packets that it does not discard. Given these assumptions, the first source of evidence is conditional on the second source i.e., an ACK is possible only if the receiver says that it received the data packet successfully. Overheard ACKs, as part of the third source of evidence, are also dependent on the event that the receiver successfully receives the data packets. In other words, the first source of evidence and evidence with overheard ACKs will be available *iff* the second source of evidence is available (receiver has successfully recorded the data packet). Therefore, it is sufficient for TE to consist of the entries stored in the receiver and

the entries stored in any witnesses about overheard data transmissions. With this, it is easy to see that (3.6) can be refined to $\{1-(1-Pr_{src2})\cdot(1-Pr_{src3\_D})\}$ since the first source of evidence subsumes the second source.

2.5.2 Analysis of Misbehaviors

Suppose that $Pr[transmitter\ lying]$ is the likelihood of a transmitter lying about sending packets (which it does not send). Let $Pr\ [receiver\ lying]$ be the likelihood of a receiver lying of not receiving packets (when it discarded such received packets). With these forwarding misbehaviors, the likelihood of TE availability is:

$$\begin{aligned}
&\tilde{Pr}_{HTE} = 0\cdot Pr[transmitter\ lying]+\\
&1\cdot(1-Pr[transmitter\ lying])\cdot Pr_{succ}\cdot(1-Pr[receiver\ lying])+\\
&Pr_{src3_D}\cdot(1-Pr[transmitter\ lying])\cdot Pr_{succ}\cdot Pr[receiver\ lying]+\\
&Pr_{src3_D}\cdot(1-Pr[transmitter\ lying])\cdot(1-Pr_{succ}),
\end{aligned} \tag{2.28}$$

where, Pr_{succ} is simply a shortened notation for $Pr\,(succ \mid r,\ l_D)$. The terms in the summation on the RHS of (2.28) correspond to the TE availability over all possible combinations of the transmitter and the receiver lying as detailed in Table 2.2. As discussed, if the transmitter is lying, no TE is available. If the receiver is lying, witnesses may or may not have evidence to the transmission.

If $Pr[transmitter\ lying]$ and $Pr[receiver\ lying]$ are set to 0, (2.28) reduces to $\{1-(1-Pr_{src2})\cdot(1-Pr_{src3\_D})\}$, which is exactly the TE availability in benign settings.

Case	**TE availability probability**
Transmitter lying	0
Transmitter not lying, receiver receiving the packet and not lying	1
Transmitter not lying, receiver receiving the packet and lying	Pr_{src3_D}
Transmitter not lying, receiver not receiving the packet	Pr_{src3_D}

Table 2.2: TE availability under all possible cases

If a transmitter or/and receiver indulges in forwarding misbehaviors, the PDR reported by the receiver is affected. Only those packets that are sent by the transmitter, successfully received and truthfully reported by the receiver are counted towards successful delivery. This PDR is expressed as:

$$PDR = (1 - Pr[transmitter\ lying]) \cdot Pr_{succ} \cdot (1 - Pr[receiver\ lying]). \tag{2.29}$$

Solving (2.28) and (2.29) yields $Pr[transmitter\ lying]$ and $Pr\ [receiver\ lying]$ as follows:

$$Pr[transmitter\ lying] = 1 - \frac{\tilde{Pr}_{HTE} - PDR + PDR * Pr_{src3\_D}}{Pr_{src3\_D}}. \tag{2.30}$$

$$Pr[receiver\ lying] = 1 - \frac{PDR \cdot Pr_{src3\_D}}{Pr_{succ} \cdot (\tilde{Pr}_{HTE} - PDR + PDR \cdot Pr_{src3\_D})}. \tag{2.31}$$

From (2.30) and (2.31), we see that there are four values essential towards computing the desired probabilities. First, one would need the measured actual TE availability

and reported PDR from the network during operations. The probability $\tilde{Pr}_{HTE}$ is simply the ratio of the number of packets for which evidence is available to the total number of packets the transmitter claims to have sent. Pr_{src3_D} and Pr_{succ} are obtained from the analytical models. Using these, the desired probabilities for the setting are computed. Finally, the probability of packet losses due to either the transmitter lying or the receiver lying is $\{Pr[transmitter\ lying] + (1 - Pr[transmitter\ lying]) \cdot Pr[receiver\ lying]\}$; The complementary probability to this yields the likelihood of the losses being because of natural effects (channel induced or interference) in the wireless network.

Discussion: We wish to acknowledge here that the actual TE availability on specific hops, even without any forwarding misbehaviors, may vary from that predicted by our analytical framework. Our assessment may inevitably deviate from the ground truth. However, the approach provides a quick and coarse-grained estimation on the likelihood of forwarding misbehaviors. In Section 2.6, we find via simulations that our assessments do not deviate much from the ground truth. For further fidelity, the local traffic and topology in the proximity of a link of interest can be considered and the analysis modified for that setting; however, note that this would increase the volume of information collected towards performing the forensic analysis (since microscopic information from local neighborhoods are needed).

The threshold values for flagging attacks can be left to be determined by the network administrator according to if the policy is stringent or relaxed, which we do not discuss in this work.

Note that the our assessments are only based on transmissions on a single link. More intelligent assessments can be made, for example, by looking at the assessments of a large number of links each node involves in to build suspicion profile for the node.

2.6 Evaluations

In this section, we first validate our analytical framework (in benign settings) with both simulations, and experiments on two different testbeds. We also examine the trends in TE availability by varying different network parameters. These provide an understanding of the likelihood of the existence of TE in various settings. Finally, we conduct a forensic analysis of packet losses to assess the likelihood of forwarding misbehaviors via simulations.

The default parameter settings (unless specified otherwise) are listed in Table 3.2. Without loss of generality, the values for the rates and SINR thresholds are adopted from 802.11a.

N	10
P_t	3.16E-2 *watts*
P_n	3.16E-10 *watts*
λ	20 pkt/sec
R	100 m
η	1.0
α	2.0
Data packet length	50/100/200/400/800/1500 bytes
ACK length	20 bytes
Rates and SINR thresholds	See Table 3.3 for the SINR thresholds and rates.

Table 2.3: Default parameter settings

Rate	6	9	12	18	24	36	48	54
SINR	6.02	7.78	9.03	10.79	17.04	18.8	24.05	24.5

Table 2.4: 802.11a Rates (Mbps) and SINR thresholds (dB).

2.6.1 Model Validation via Simulations

Simulation setup: The simulations are performed using OPNET modeler version 14.5 [22]. In our simulations, we first consider a single hop wireless network. Here, N nodes are uniformly distributed in a circle with diameter $2R$. The considered receiver is positioned at the center, while the transmitter is randomly picked from among the other $N-1$ nodes. The transmissions experience both path loss and Rayleigh fading. Next, we consider a multi-hop network by spreading $5N$ nodes uniformly in a circle with diameter $2\sqrt{5}R$. The paths whose PTE is considered, are selected such that the nodes on the paths are near the center of the circle, instead of being at the network edge. We choose these configurations to eliminate edge effects while keeping the node density of the network fixed. Traffic are sent between randomly chosen source destination pairs. The traffic generated at a node is 20 pkts/sec. Shortest path routes are used. Nodes transmit packets at random instances in time to contend for channel access. At the end of a run, we combine the traces from all nodes and calculate the number of unique transmissions recorded. The fraction of this number over the total number of transmissions occurring during this run, is the TE availability probability. The data collected for each specific scenario is an average over 20 runs.

Trends in hop-level TE availability: We first examine the trends in HTE

availability when various parameters are tuned. Benign settings are considered.

Bit-rate and packet length: We first vary the transmission bit-rate and packet length. The other parameters are at default settings. Figs. 2.3 and 2.4 show the trend in HTE availability from the analysis and simulations[5]. We see that for a fixed bit-rate, a smaller packet length leads to higher hop-level TE availability. This is because, with a smaller packet length, the air time is small; thus, the chances of a packet being corrupted due to interference either at the receiver or at the witnesses is lower. Furthermore, using lower bit rates results in almost perfect TE availability when the packet length is small. When the rate increases the TE availability decreases. With higher rates, packets are more susceptible to channel induced losses. This decreases the probabilities that the receiver and the overhearing witnesses successfully receive the transmissions. The above effects are more pronounced with larger packet lengths. In the case we consider, when the packet length grows beyond 800 bytes, even the lower rates do not guarantee high TE availability.

The results from simulations are shown in Fig. 2.4. We observe that the trends hold in terms of TE availability , thus validating the applicability of our assumptions.However, the numerical values of counterparts from simulations are generally slightly higher than those generated with our analysis. This could be due to the conservative approximation we use in our analysis (all the interferers are assumed at the same distance as the closest interferer to the receiver). However, we notice that the difference between the analytical and simulated results is not very much and they are almost in conformance.

[5]The simulation results for packet lengths of 50 and 100 bytes are similar to that of length 200 bytes; they are not shown for clarity.

Node density: We increase the number of nodes deployed from 10 to 20 and 30. As a consequence the node density increases. The analytical results are in Figs. 2.5 and 2.6 (simulation results are similar and not shown for purposes of clarity). The total interference levels imposed on a node is higher due to the higher node density. This hurts both packet reception and overhearing and eventually hurts TE. However, a high node density means that there are more nodes serving as potential witnesses. This helps TE collection. Figs. 2.5 and 2.6 indicate that (a) when low rates are used, the *first* factor seems dominant and, thus TE availability tends to decrease as N increases; (b) when high rates are used, the *second* factor seems dominant and TE availability increases with N in the case we consider. Comparing across packet lengths and rates, we conclude that when node density is high, moderately large packet lengths and high rates facilitate high TE. In this case, the benefit of large witness amount and short air time exposed to interference overrides other factors.

Traffic volume: Now we adjust the traffic generated per node to be 5 times less (4 pkts/sec) and 5 times more (100 pkts/sec). As we see from Fig. 2.7, TE availability is fairly high and alike across all packet lengths with low traffic volume. It is because the main reason for packet failure is the effect of the channel and not the interference. In addition, with less traffic to send, nodes are more likely to be witnesses and collect TE. When traffic volume is high (Fig. 2.8), TE availability drops drastically; high interference hurts TE collection and nodes have less time for overhearing.

Retransmission limit: We vary the retransmission limit from 0 (default) to

7. We notice that at low and moderate loads allowing more retransmissions increases TE availability as one might expect (Fig. 2.9). The effect is more pronounced when the retransmission limit increases from 0 to 1 and 2, and less pronounced with further increases. The TE availability drops when the retransmission load increases beyond a certain point. Note here that node density is a factor in determining when such a switch over would occur. Due to space constraints, we do not discuss more details here.

Trends in path-Level TE availability: We vary the packet length and hop count; other parameters remain at default settings. We look at paths with hop counts from 1 to 7. We believe that this reflects a reasonable network size. Especially, the path level TE with hop count 1 is the same to the hop level TE. Recall that we assume a uniform rate selection at each hop. The results in terms of PTE availability probability generated with our analytical models and from simulations, are in Figs. 2.10 and 2.11 respectively. We observe that: (1) A shorter packet length yields a higher PTE availability. For example, the path level TE with a packet length of 200 bytes can be 2-3 times of that with a packet length of 1500 bytes, when the hop count is larger than 1. This trend is consistent with that in HTE. (2) The PTE decreases quite fast with increasing hop count. With the hop count being increased by 1, the PTE drops by 10%-20%. When the hop count increases from 1 to 7, the path level TE can decrease to about less than half of the hop level TE. (3) The simulation results are similar to the analytical results.

Fig. 2.12 presents the CDFs of the PTE for all the cases that we examine. We observe in about 80% of the cases, the PTE is above 0.5; in about 30% of the cases, it

exceeds or approaches 0.8. This implies that, there is a good likelihood that the PTE is available in typical settings.

2.6.2 Model Validation via Experiments

We examine the TE availability in real networks: (a) in a 802.11 testbed with CSMA/CA and (b) in a testbed of WARP nodes with Aloha. Our objective is to show that our framework accurately characterizes the TE availability that one can expect in real networks.

Experiment setup: We conduct two sets of experiments, to examine the TE availability in *interference-managed* (802.11 CSMA/CA) and -*unmanaged* (Aloha) scenarios, respectively.

The first set of experiments is performed on a 32-node wireless testbed deployed on an entire floor of our campus building as shown in Fig. 2.13. The testbed configuration is such that the network consists of both indoor and outdoor links. The nodes are based on the Soekris net5501 hardware configuration, and run a Debian Linux distribution. 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. More details about our wireless testbed can be found in [23]. We experiment with the 802.11a mode in order to avoid interference from co-located 802.11b campus WLANs. RTS/CTS is disabled. Each scenario involves different sets of about ten nodes. In each scenario, every node generates traffic to a randomly chosen neighbor, while running TCPdump to record all packets it receives or overhears. The TE

availability probability is computed as described earlier with our simulations.

The second set experiments is on a 5-node WARP (Wireless open-Access Research Platform) [5] Radio testbed in a lab setting. Nodes access the channel using Aloha, which runs on top of the WARP OFDM implementation. Traffic is generated on the board itself. When a node is idle (not transmitting) it logs the data and ACK transmissions going on in its neighborhood and these logs are continuously sent to the central server which is connected to all the nodes using an Ethernet interface provided on each node. The central server is a PC which is basically used to send control messages to the nodes and to collect logs sent by the nodes.

Note here that in a lab setting (distance between nodes no more than ten meters), with default transmission power the channel induced errors are fairly minor across all modulation schemes. To see the channel induced effects on links we have to reduce the transmission power. In our experiments all nodes operate at half of their maximum allowed power.

We evaluate the trends in TE using this testbed by varying packet lengths and modulation schemes. There is no option to change the forward error correction or FEC, code rate on our boards. The packet generation rate is 200 pkts/sec. We use 3 transmitters and 2 receivers for a period of 10 seconds.

Empirical hop-level TE: First, we report our experiments on the 802.11a testbed. We plot the TE availability probability for various packet lengths and with different rates in Fig. 2.14. We see that low rates offers high TE availability. At higher rates, the TE avail-

ability drops slightly ($\approx 10\%$), especially for larger packet sizes. A quick look at Fig. 2.7 shows that the results with our model with low traffic volume is similar to what is seen here. This is because, CSMA/CA manages the interference well to avoid interference in the vicinity of an active transmission. There is still interference due to hidden terminals, but the levels are low. Thus, by calibrating our model with a low Λ one can obtain trends that are likely to exist with MAC protocols that manage interference (as with 802.11 or TDMA).

Next, we consider unmanaged interference with Aloha. We use our WARP testbed here. Since, our testbed consists of only five nodes, we use a high packet generation rate in order to have a desired interference level.

Fig. 2.15 shows the TE trends with the four modulation schemes with Aloha. With our WARP boards, the FEC code rate is fixed and hence, there are not as many bit-rates to select from. The fairly limited setting of the testbed makes it difficult to directly compare the results with the heavy load scenario in Fig. 2.8, although the trends are similar. At low rates the TE availability is low. As we increase the rate it increases and finally drops due to channel induced effects with QAM 64 modulation. Note that there is no one to one mapping between the x-axes in Figs. 2.15 and 2.8. Thus, it is difficult to get an exact match. However, the model does indeed predict the trend of what could be expected in practice.

Empirical path-level TE: We use our 802.11 wireless testbed to validate the model for PTE. We create a number of 3 and 4-hop paths to measure the probability of end-to-end TE availability. Note that due to the small scale, such an experiment was not possible with our WARP hardware. We used static routing to create multi-hop paths to

ensure that route flapping did not happen. Default rate adaptation is used by nodes. Each node along the path logs the traffic which is being transmitted in its vicinity, using which we calculate the PTE for each multi-hop path. Two packet sizes are considered. It is seen in Fig. 2.16 that the PTE availability decreases as the hop count and packet length increase. We see a good match with our analytical results generated with a low traffic volume.

Summary: To summarize, both our simulation and experimental results demonstrate that by appropriately calibrating our analytical framework with network parameters (packet size, bit rate in use, node density, interference level), one can get a good indication of the likelihood of TE availability in practice. This can not only aid forensic analysis (as discussed next), but also allow a network administrator to determine the efficacy of a monitoring system given specific network conditions.

2.6.3 Forensics Analysis using Transmission Evidence

In our last set of evaluations, we aim to provide the assessments of forwarding misbehaviors. We randomly select up to one hundred links from our simulated network and on each of these links, ten thousand packets are scheduled to be sent. We emulate forwarding misbehaviors at the transmitters and receivers, individually and jointly. A transmitter lying by $x\%$, implies that it does not transmit $x\%$ of the packets that it is supposed to send. A receiver lying by $x\%$ means that it claims to have received only $1 - x\%$ of the packets that it in fact receives. We vary the fraction of lying (0%, 10%, 20%, 40% and 60%) at the transmitter and receiver, respectively. These preset values correspond to the ground truth.

We collect the actual TE and PDR for each transmission period. Having the

Ground Truth (%)	**Assessment Results**		
	avg dev (%)	**min dev (%)**	**max dev (%)**
transmitter 0	2.65	0.39	5.98
receiver 0	4.84	0.44	13.83
transmitter 10	2.05	0.00	1.05
receiver 0	4.20	0.00	11.77
transmitter 0	5.28	0.02	10.39
receiver 10	2.71	0.00	9.62
transmitter 10	1.80	0.18	6.08
receiver 10	1.84	0.38	5.88
transmitter 20	5.36	0.32	15.00
receiver 0	2.92	0.00	11.21
transmitter 0	1.36	0.00	8.49
receiver 20	2.32	0.35	9.24
transmitter 20	1.56	0.38	5.11
receiver 20	1.66	0.55	5.09
transmitter 40	3.87	0.29	10.89
receiver 0	2.59	0.00	9.09
transmitter 0	0.76	0.00	4.90
receiver 40	1.38	0.09	7.53
transmitter 40	0.99	0.26	3.11
receiver 40	1.01	0.16	3.05
transmitter 60	2.34	0.24	5.52
receiver 0	1.84	0.00	8.85
transmitter 0	0.23	0.00	1.42
receiver 60	1.89	0.50	5.20
transmitter 60	0.39	0.12	1.28
receiver 60	0.43	0.09	1.34

Table 2.5: Assessments on transmitter and receiver lying

measured values from the simulations and their estimated counterparts from the analytical models, we use our forensic analyzer described in Section 2.5 to assess the likelihoods of the transmitter and/or the receiver lying.

The assessment results are presented in Table 2.5. Column one contains the ground truth, while columns two to four contain the average/minimum/maximum deviation of the assessments from the truth (expressed as percentages) across all considered links. The deviation is calculated as $|assessed\,value - truth|$ and is computed for both the transmitter and receiver. We see that, overall, our assessments are able to reflect the ground truth with good accuracy. However, due to the variance between the generic analytical models and the unique circumstances of each link, inevitably there is a deviation in the assessment on each specific link. The average deviation is 2.3% for all the cases that we examine, while the maximum value is 15.0%. These results demonstrate that our analytical framework can facilitate the assessment of the considered misbehaviors with good accuracy.

Note here that the deviation, as we define here, computes the "overestimate" or the "underestimate" of the misbehavior probability by the forensic analyzer in absolute terms. As aforementioned, the deviation is due to the variance between the coarse-grain estimation made and the actual TE availability on each specific link. If this deviation is small, the analyzer has a reasonable estimate of the likelihood of misbehaviors. False alarms or misses are inevitable due to the variance of the analytical and actual TE. If needed, fine grained information can be further gathered from the vicinity of the link in question to refine this probability estimate.

Finally, one interesting observation we made is that, when the misbehavior at either the transmitters or the receivers is at a mild level (the fraction of lying is 5% or 10%), the measured TE is not much distinguishable compared to the analytical TE, especially considering the inherent variance of actual TE. As the misbehavior at the transmitters goes more severe, the measured TE drops drastically. However when the discarding at the receiver goes aggressive, the measured TE only decreases slightly. The reason is that the transmissions are overheard by witnesses even the receivers deny receiving them.

2.7 Conclusions

In this work, we seek to differentiate between wireless induced packet losses and malicious discarding in wireless networks. Towards facilitating such a forensic analysis, we develop an analytical framework that takes as input various macroscopic network parameters and yields as output, the likelihood of evidence availability. We validate our analytical framework via both extensive simulations and experiments on two different wireless testbeds that employ different MAC protocols. We then discuss the applicability of our analytical framework in a forensic analyzer to determine the likelihood of a transmitter or receiver discarding packets maliciously. We show via simulations that the analyzer is able to determine these likelihoods with high accuracy.

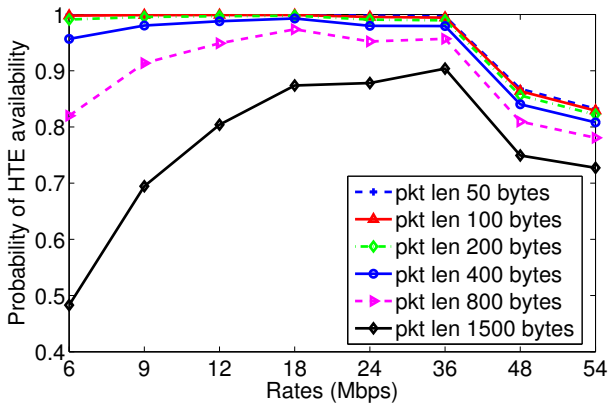


Figure 2.3: Hop-level TE availability (analytical)

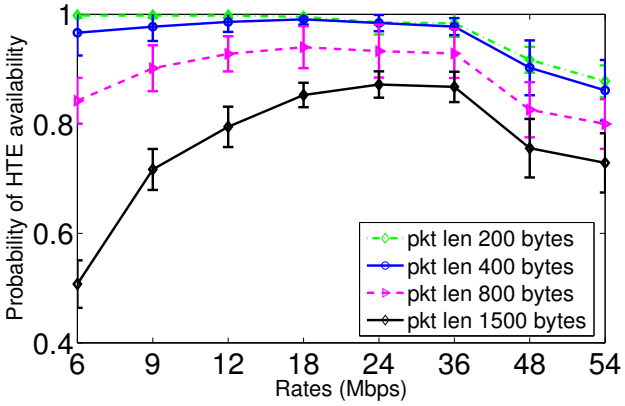


Figure 2.4: Hop-level TE availability (simulation)

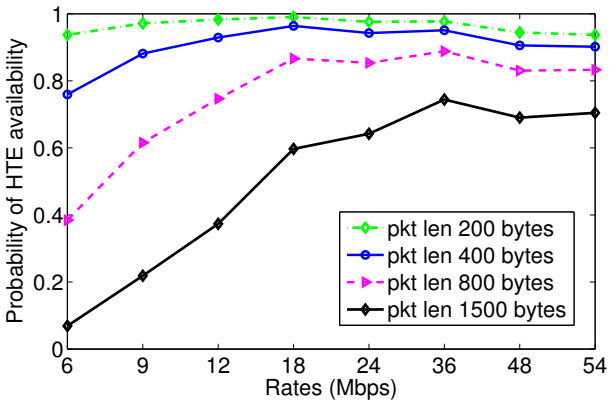


Figure 2.5: Node number 20

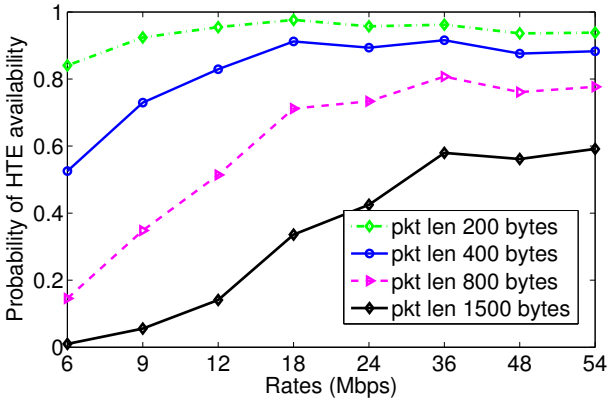


Figure 2.6: Node number 30

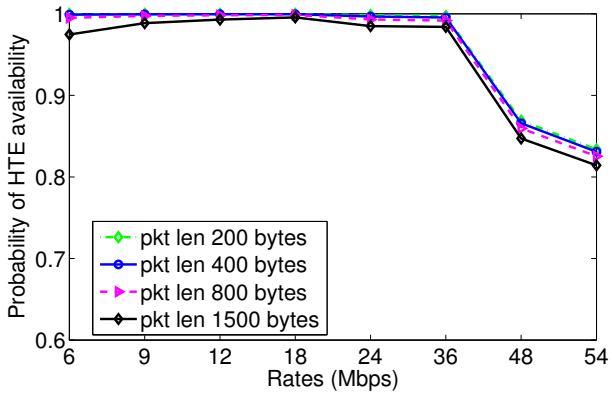


Figure 2.7: Low Traffic Volume

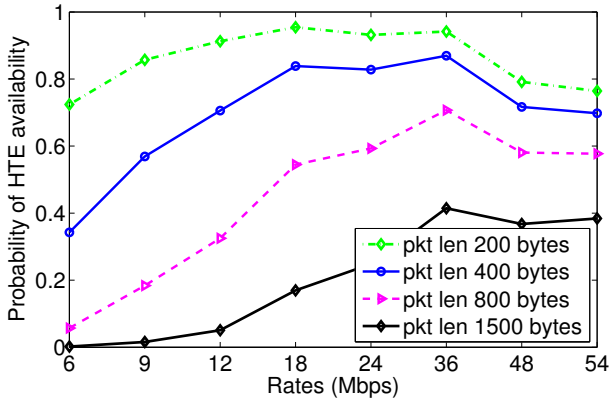


Figure 2.8: High Traffic Volume

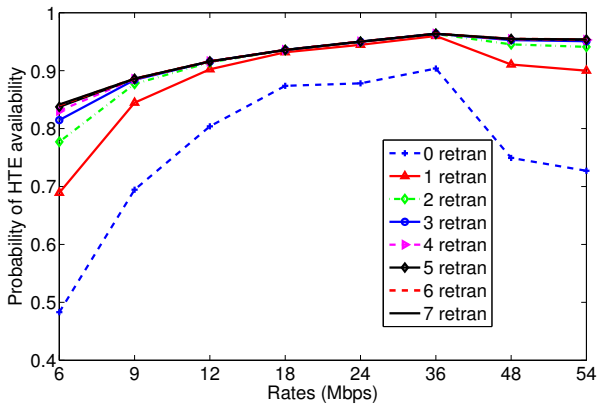


Figure 2.9: Impact of retransmission limit (packet length 1500 bytes)

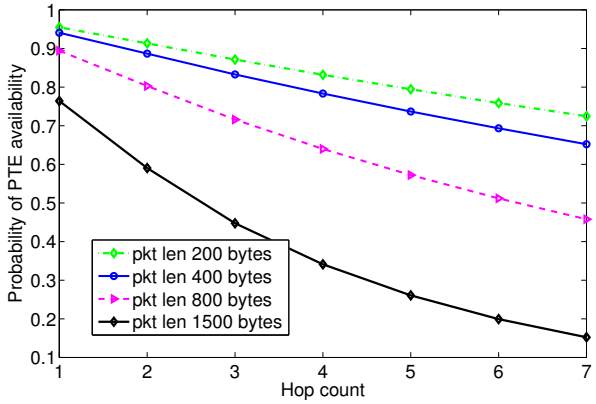


Figure 2.10: PTE availability probability (analytical)

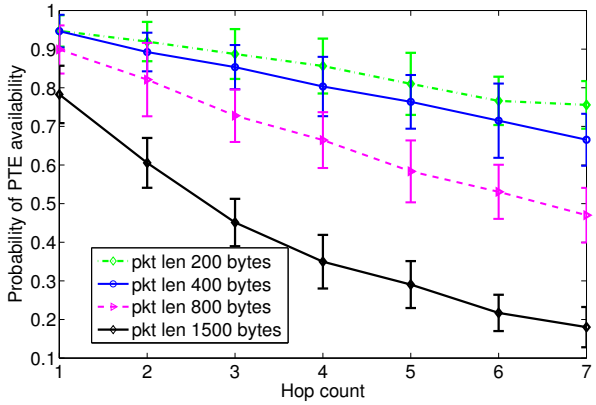


Figure 2.11: PTE availability probability (simulation)

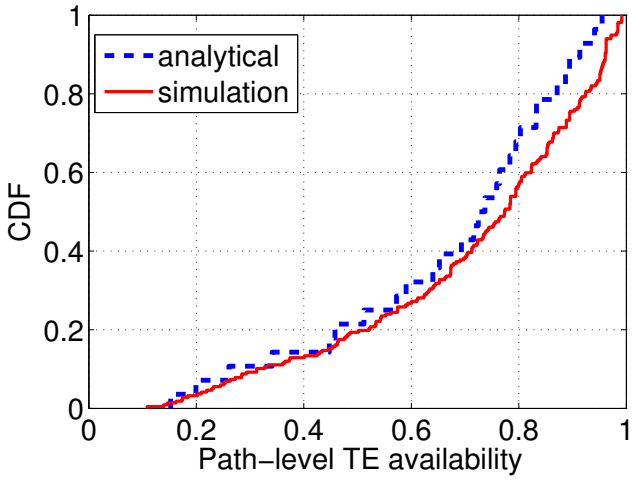


Figure 2.12: CDF of PTE availability

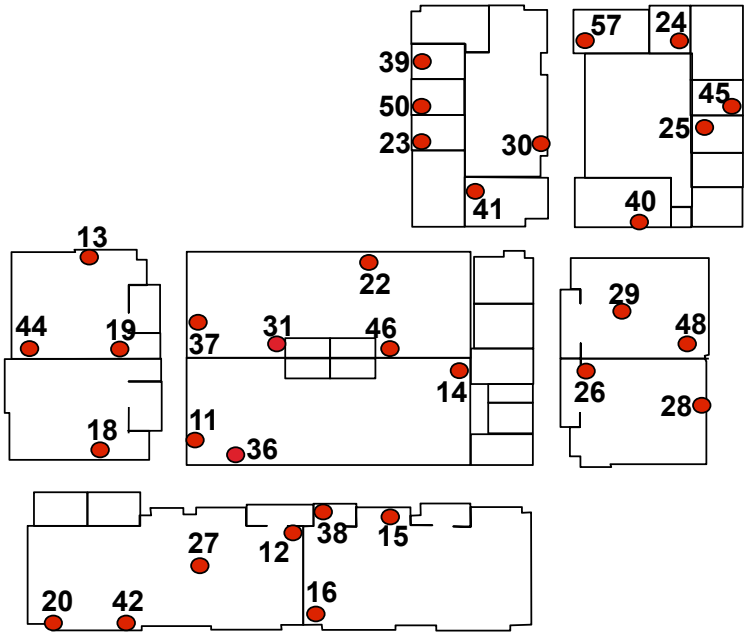


Figure 2.13: UCR wireless testbed

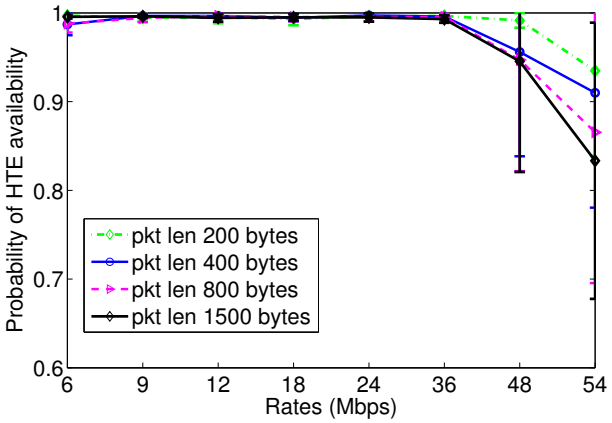


Figure 2.14: Empirical HTE in 802.11

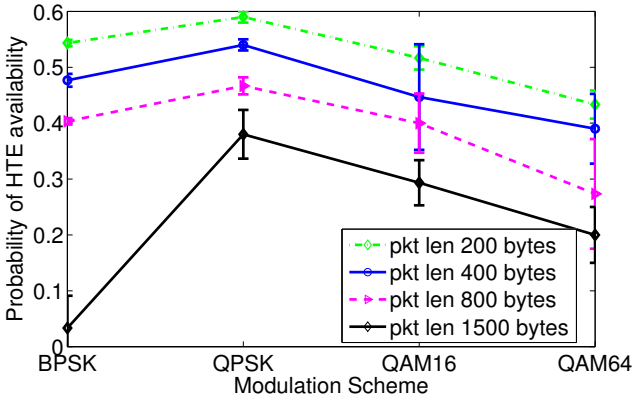


Figure 2.15: Empirical HTE in Aloha

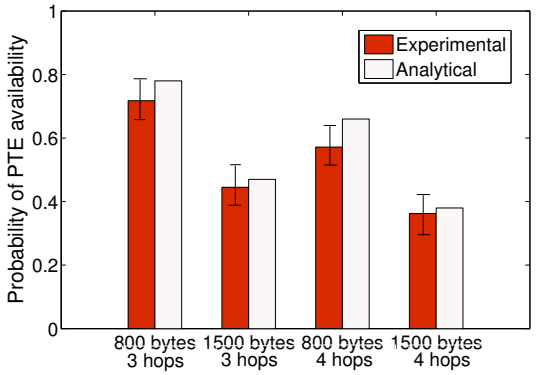


Figure 2.16: Empirical PTE in 802.11

Chapter 3

On the Trade-offs between Collecting Packet Level Forensic Evidence and Data Delivery Performance in Wireless Networks

Transmission Evidence (TE for short) refers to a historic trail of the packet transmissions in the network. TE is collected and maintained in a distributed manner by the nodes in the network and can be queried on demand by a network forensics system to trace past events. The latter can facilitate crucial applications such as identifying malicious or malfunctioning nodes. Recently, we developed an analytical framework towards computing the likelihood of TE availability in wireless networks. Our prior efforts [24] brought to light the impact of the network's operational parameters (such as transmission rate and packet length) on the availability of TE. However, provisioning for

TE could impact the network performance in terms of throughput and/or delay. Our objective in this work is to capture and quantify the trade-offs between provisioning transmission evidence and achieving high performance in wireless networks. In particular, we investigate the network performance hit, under the constraint of TE availability guarantees. Our results indicate that the performance remains unaffected up to a certain TE requirement. Beyond this, the throughput could degrade and the delay could increase by as much as 30%. To the best of our knowledge, this is the first study of its kind.

3.1 Introduction

The open nature of the wireless medium makes it easy for malicious users to adopt a variety of misbehaviors, varying from simple injection of electromagnetic energy that causes interference (i.e., jamming) to more sophisticated higher layer attacks such as replay attacks. With regards to many of these attacks, network forensics can help identify the cause of abnormal network behaviors [25] [26]. For instance, in a multi-hop wireless network, routing is facilitated by the nodes. Selfish routers might not want to spend their resources (e.g., power) to forward packets for others. In such cases, network forensics can be useful in tracing back to the misbehaving node(s) on paths.

A network forensics system should be able to (a) collect evidence, (b) detect long-term trends and (c) provide post-event understanding of what actually happened, and (d) playback events to the highest fidelity possible. As has been identified in the literature [2] [3] [27], collecting evidentiary data is a key component of such a system. Ideally, a complete and accurate information trail about network events should be recorded. However, this is not feasible in practice in wireless networks due to the lossy nature of wireless links. A number of operational network parameters may have to be tuned appropriately to aid the collection of evidence. However, traditionally, security and performance considerations have followed disjoint paths in wireless protocol design. Consequently,

choosing the operational parameter space towards facilitating evidence collection may negatively impact the data delivery performance. This is the target of our investigation in this paper.

We define *transmission evidence* (TE) to be the evidence collected by nodes with regards to link level packet transmissions. This collection takes place in a fully distributed manner. The nodes can then be queried by a central network forensics system when needed. To illustrate the importance of TE consider again our prior example of packet forwarding misbehaviors in a multi-hop wireless network. Such blackhole (or grayhole) attacks have been extensively studied in the literature (e.g., [2] [3] [1]). A typical detection scheme incorporates the collection of evidentiary data with respect to packet transmissions. This evidence is collected from the network users. In other words, nodes who utilize the infrastructure to communicate with each other, act at the same time as transmission witnesses and monitors [4]. This is a very attractive feature, since there is no need for dedicated monitoring nodes, hence reducing the deployment cost. In addition, the distribution of this task to more than a few dedicated monitoring nodes, increases the resilience of the system to node failures. However, depending on the deployment, witnesses may not have evidence with regards to specific packet transmissions, leading to low TE availability.

TE availability refers to the probability that transmission evidence exists in the network. In our previous work [24] we constructed an analytical framework for computing this likelihood. We captured the factors that affect TE availability and found that it depends on network parameters, and in particular packet size and bit-rates. We further made several interesting observations on the TE availability when tuning these parameters. We also demonstrated that the analytical framework could be used as the basis of a forensic analyzer to assess the cause of packet losses and determine the likelihood of packet dropping attacks.

In this work, we further seek to answer the question: ***If one were to tune network parameters to achieve a high degree of TE availability, how is the performance affected?***

Our study reveals interesting dependencies between provisioning TE and achieving high network performance. To our best knowledge, this is the first study examining this trade-off. Our main contribution in this paper is analyzing and understanding the aforementioned trade-off between TE availability and network performance. Our results indicate the presence of a tipping point. Specifically, we find that the best performance remains fairly unchanged if we were to turn the parameters so as to achieve a certain TE requirement. Beyond this, the maximum throughput drops and the minimum delay increases considerably. In what follows, we examine the reasons for this.

The rest of the paper is structured as follows. Section 3.2 discusses related work. In Section 3.3, we describe the TE collection process and provide a brief description of the TE availability models. Models that characterize the trade-offs between TE and network performance are formulated in Section 3.4. Section 3.5 presents our results. Finally, Section 4.6 contains our conclusions.

3.2 Related Work

In this section, we briefly discuss representative literature on network forensics and transmission evidence. We also survey existing work that examines the trade-offs between achieving certain level of security in various contexts and data delivery performance in networks.

3.2.1 Network Forensics and Transmission Evidence

As mentioned earlier, network forensics is the process of gathering and analyzing information that traverses a wired or wireless network and is an important part of network management. The goals include but are not limited to fault diagnostics, identifying the source of misbehaviors, interpreting traffic characteristics, and evaluating network performance [28].

There exist prior work on wireless network forensics at the *mechanism and system design* level (e.g., [1] [6] [3] [7] [8] [9] [10] [11]). Marti *et al.* [1] design a *watchdog* scheme to identify

malicious nodes who employ blackhole attacks, that is, they do not forward packets along a multi-hop path. McGrath *et al.* [6] design FLUX to automate the collection of forensic data for identifying abnormal traffic and network weaknesses. Ramach *et al.* [3] design and implement DAMON, a distributed monitoring system for MANETs while, Yang *et al.* [4] propose a specific witness-based detection scheme to identify forwarding misbehaviors. The vast majority of the above studies propose techniques for solving specific network problems that require evidentiary data. However, none of them study the impact of facilitating the collection of evidence on network performance.

3.2.2 Security and Performance Trade-offs in Networks

Security solutions almost always involve some overhead that affects network performance. The metrics for performance and security are different in various contexts. Soini *et al.* [29] examine how energy consumption and computation time are increased in sensor nodes due to utilization of different security features of encryption and authentication in wireless sensor networks. Yau *et al.* [17] develop a model for quantifying the trade-offs between service performance and security levels in a service-oriented architecture. System performance is measured with respect to timeliness and throughput, while the service security is realized through authentication, authorization and encryption mechanisms. Zeng *et al.* [30] present a trade-off model to adapt security configuration to provide sufficient protection while satisfying real-time dynamic performance requirements of the networked control systems. Unlike the aforementioned studies, we quantify security in terms of forensic evidence availability. We examine the trade-offs between the latter and network performance.

3.3 TE Availability Models

In this section, we describe the process of TE collection and provide a brief introduction to our TE availability models.

Evidence collection: In a multi-hop wireless network, nodes maintain evidence relating to transmissions as follows: **(a)** a sender (or transmitter) keeps the signed ACK it receives for each packet it sends. **(b)** A receiver creates a local entry for each unique packet received and digitally verified. **(c)** A witness node creates an entry locally for each packet that it overhears and verifies. Legitimate nodes adhere to the above rules and do not manipulate evidence. We would like to emphasize that packets (DATA and ACK) contain the senders' digital signature. Without loss of generality, we assume that an ACK includes sender and receiver IDs and therefore, an overheard ACK is of evidentiary value. While having digital signatures increases the computation overhead in nodes and may negatively impact the network performance, there are studies examining such trade-offs [29] [17], and hence this issue is not reconsidered in our work.

With storage space becoming cheaper and more compact, we expect and thus assume that storage is not a limiting factor in evidence collection. Furthermore, one can envision nodes sending coarse-grained information relating to collected evidence periodically to a central forensics analyzer. Evidence is only sent infrequently since our objective is to investigate long-term effects. It can be easily piggybacked onto other control information (e.g. routing updates) and thus, we expect that the overhead is likely to be small, leading us to not consider it during the quantification of the trade-off under examination.

Hop-level TE (HTE): The availability of hop-level TE reflects the likelihood that evidence exists relating to transmissions on a link. Commonly used notations are listed in Table 3.1. As alluded to above, for a transmission between v_i and v_j, there are three sources of evidence, which we formally present in what follows.

Source 1: v_i ***has the signed ACK from*** v_j ***for packet*** $(src, dest, pkt\ SQN)$. This requires that: **(a)** v_i's data packet is successfully received by v_j and, **(b)** v_j's ACK packet is successfully received by v_i.

N	Total number of nodes
v_i	Transmitter
v_j	Receiver
z	Number of interferers
Z	Set of interferers
r	Transmission bit-rate
l_D	data packet length
l_A	ACK packet length

Table 3.1: Summary of notations

$Pr\left(succ \mid r, l_D\right)$ denotes the probability of a successful data transmission with rate r and packet length l_D. Similarly, the probability of a successful ACK transmission is $Pr\left(succ \mid r_0, l_A\right)$, assuming that an ACK is sent at the base rate r_0 and has a length l_A. The probability that the first source of evidence is available is:

$$Pr_{src1} = Pr\left(succ \mid r, l_D\right) \cdot Pr\left(succ \mid r_0, l_A\right). \tag{3.1}$$

Source 2: v_j ***has a stored entry*** $|v_i|src|dest|pkt\ SQN|\ timestamp|$. This source of TE requires a successful transmission from v_i to v_j, the probability of which is:

$$Pr_{src2} = Pr\left(succ \mid r, l_D\right). \tag{3.2}$$

Source 3: At least one witness has a stored entry $|v_i|src|\ dest|pkt\ SQN|timestamp|$. This requires at least a node other than v_i or v_j to overhear the data transmission from v_i or the ACK transmission from v_j.

Let Pr_{src3_D} and Pr_{src3_A} denote the probabilities that at least one witness overhears the data and ACK, respectively. Note that due to the half-duplex property of typical radio devices, it is assumed that a node cannot be an interferer and a witness for the same transmission[1]. Therefore,

[1]To reiterate, we assume that the monitoring devices or witnesses are active nodes in the network. It is easy to modify the analysis if evidence is collected only by passive monitoring nodes.

when there are totally N nodes in the network and z interferers, the number of witnesses cannot exceed $N-z-2$. Thus we have:

$$\begin{aligned} Pr_{src3\_D} = \sum_{z=0}^{N-2} Pr\left(\mathrm{z}\ \ int \mid r, l_D\right) \cdot \\ \left(1-\left(1-Pr\left(succ \mid r, l_D, z\right)\right)^{N-z-2}\right), \end{aligned} \tag{3.3}$$

in which $\left(1-\left(1-Pr\left(succ \mid r, l_D, z\right)\right)^{N-z-2}\right)$ is the probability that given z interferers, at least one witness overhears the data transmission. Considering all possible values that z can take, we get the marginal probability that at least one witness overhears a given transmission.

In a similar way, we compute Pr_{src3_A} as follows[2]:

$$\begin{aligned} Pr_{src3\_A} = Pr\left(succ \mid r, l_D\right) \cdot \left(\sum_{z=0}^{N-2} Pr\left(\mathrm{z}\ \ int \mid r_0, l_A\right) \cdot \right. \\ \left.\left(1-\left(1-Pr\left(succ \mid r_0, l_A, z\right)\right)^{N-z-2}\right)\right). \end{aligned} \tag{3.4}$$

The successful overhearing of data and ACK transmissions by any given witness are assumed to be independent. In reality, there may be correlations due to interference effects at each overhearing node. However, the independence assumption (which we make to keep the analysis tractable) is shown to be reasonable by both our simulations and experiments reported in [24], and hence:

$$Pr_{src3} = Pr_{src3\_D} + (1 - Pr_{src3\_D}) \cdot Pr_{src3\_A}. \tag{3.5}$$

Hop-level TE availability: The probability that at least one source of TE is available for a transmission is:

$$\begin{aligned} Pr_{HTE} &= 1 - \prod_{i=1}^{3} Pr(source\ \ i\ \ is\ \ unavailable) \\ &= 1 - (1 - Pr_{src1}) \cdot (1 - Pr_{src2}) \cdot (1 - Pr_{src3}). \end{aligned} \tag{3.6}$$

[2]For v_j to transmit an ACK, it must have successfully received the corresponding data packet.

In the above model, we have considered every packet being transmitted only once. Wireless MAC layer protocols usually introduce a maximum number n_r of retransmission attempts for every packet, in order to increase the reliability over an inherent unreliable medium. The success of each transmission is independent from the previous trails (assuming that these are staggered in time, this is a reasonable assumption since the temporal network conditions are likely to change). The probability of successful packet exchange, denoted by Pr_{succ_ex} is given by:

$$Pr_{succ\_ex} = Pr\left(succ \mid r, l_D\right) \cdot Pr\left(succ \mid r_0, l_A\right), \tag{3.7}$$

Since the transmission trials for a packet, stop either because the exchange was successful or due to reaching the retransmission limit, the probability that there are i retransmissions ($i+1$ transmission attempts) is denoted as $Pr(rtx)$ and is given by:

$$Pr(rtx = i) = \begin{cases} Pr_{succ\_ex} \cdot (1 - Pr_{succ\_ex})^i & 0 \leq i \leq n_r - 1 \\ 1 - \sum_{j=0}^{n_r - 1} Pr(rtx = j) & i = n_r \end{cases}. \tag{3.8}$$

Hence, the TE availability with a retransmission limit n_r is:

$$Pr_{HTE}[\, n_r \,] = \sum_{i=0}^{n_r} Pr(rtx = i) \cdot \left(1 - (1 - Pr_{HTE})^{i+1}\right). \tag{3.9}$$

Path-level TE (PTE): Having the TE availability on a single hop, we next extend it to the path-level TE, that is, availability of evidence related with all transmissions on an end-to-end path. The TE availability on each hop along the path is assumed to be independent of that on any other hops. Again, in reality the TE availability across hops may be correlated but we make this assumption for tractability; our simulations and experiments verify that this assumption is also acceptable as shown in [24]. The PTE requires the HTE on all the hops of the path. Hence, the PTE for a H-hop path, denoted by $Pr_{PTE}[\, H \,]$, is given by:

$$Pr_{PTE}[\, H \,] = \prod_{h=1}^{H} Pr_{HTE}[\, at\ h^{th}\ hop \,]. \tag{3.10}$$

In addition to the parameters that affect HTE, PTE is also impacted from the hop count. Generally, as one may expect the longer the path, the lower PTE.

Note also here that we choose to give a strict definition of PTE, in the sense that it requires HTE on all hops. The TE of the transmission on hop h, can imply the success of transmissions on the previous $h-1$ hops, even though the HTE may not be available for all such hops. However, in our model we do not consider such a cumulative definition of PTE.

3.4 Trade-offs between TE and Performance

Network administrators might require a specific level of evidence availability in order to ensure lower bounds on the ability to track misbehaviors in the network. However, this might hurt the network performance since the collection of evidence hinges on reliable transmissions which in turn either requires transmissions at lower rates or the use of smaller packet sizes.

3.4.1 The Trade-off between TE and Delay

The total delay on a given link (i.e., nodal delay) consists of the processing delay at the node, the queueing delay, the transmission delay and the propagation delay. The end-to-end delay is simply the sum of all these nodal delays over the path followed. Since the collection of TE is related only to the transmission of packets (and not for instance with their queueing at nodes' buffers), the TE availability is only related to the transmission delay. Therefore, we only consider the transmission time over hops, which in turn depends on the packet length, the transmission bit rate, and the link-level retransmission limit. The delay over a hop can be expressed as [31]:

$$D = \sum_{i=0}^{n_r} \frac{succ_i}{succ\_total} \cdot \left(T_o \cdot i + \sum_{j=0}^{i} \frac{L_D}{r_j} \right), \tag{3.11}$$

where n_r is the retransmission limit, $succ_i$ is the probability that the transmission is successful at the $(i+1)^{\text{th}}$ attempt and *succ_total* is the probability that the packet delivery is successful within the retransmission limit. Note here that we only consider the delays of successfully delivered packets; the ratio $\frac{succ_i}{succ\_total}$ is the fraction of packets that are successfully delivered with exactly i retransmission attempts. T_o is the time spent between two consecutive transmission attempts. This time may depend on the specific MAC protocol in use (e.g., exponential back-off in CSMA/CA). In our evaluations we simply set this to be a fixed constant time to remove MAC dependencies for simplicity. r_i is the rate used for the $(i+1)^{\text{th}}$ transmission attempt.

The probability of a transmission success on exactly the $(i+1)^{\text{th}}$ transmission attempt, $succ_i$, is:

$$succ_i = \begin{cases} Pr(succ \mid r_i, l_D) & i = 0 \\ \prod_{j=0}^{i-1} (1 - Pr(succ \mid r_j, l_D)) \cdot Pr(succ \mid r_j, l_D) & 0 < i \leq n_r \end{cases} . \tag{3.12}$$

It is easy to see that the probability of transmission success within the retransmission limit *succ_total* is:

$$succ\_total = \sum_{i=0}^{n_r} succ_i, \tag{3.13}$$

since the events that the packet succeeds in the i^{th} attempt and the j^{th} attempt are mutually exclusive for $i \neq j$.

Here we do not consider end to end retransmissions which may be used by the transport layer protocol to achieve high reliability (e.g., TCP). The procedure of computing end to end delay with a retransmission limit is similar to that with link level retransmission limit and can be extended from (3.11), (3.12) and (3.13).

The question that we ask is "given an TE requirement β, what is minimum end-to-end delay that we can achieve?". Formalizing our question, we seek to solve the following optimization problem:

$$\begin{aligned} \text{Min} \quad & \sum_{h=1}^{H} D_h \\ \text{s.t.} \quad & Pr_{PTE}[\,H\,] \geq \beta, \\ & 0 < \beta \leq 1. \end{aligned} \tag{3.14}$$

where, H is the number of total hops, and is a positive integer, D_h is the delay on hop h, $Pr_{PTE}[\,H\,]$ is the path level TE on a H-hop path, and is computed with (3.10), and β is the desired threshold on TE and is a real number between $[0, 1]$. The objective is to minimize the end-to-end delay under the constraint which imposes that the path level TE should be no less than β.

3.4.2 The Trade-off between TE and Throughput

Next we will examine the trade-offs between achieving a given level of TE and high end-to-end throughput.

The end-to-end throughput is defined for every communicating pair to be *the number of bits arriving at the destination per unit time* and can be expressed as:

$$Th[H] = \frac{L_P}{\sum_{h=1}^{H} D_h} \cdot \prod_{h=1}^{H} succ\_total_h, \tag{3.15}$$

where L_P is equal to the data payload (header and digital signature overheads are not considered). $\sum_{h=1}^{H} D_h$ is the end-to-end delay and $\prod_{h=1}^{H} succ\_total_h$ is the probability of successful end-to-end packet delivery, where $succ\_total_h$ is the probability of transmission success at hop h.

Similar to the case of the end-to-end delay, we want to find the maximum possible throughput on an H-hop path that can be achieved under the constraint of a given level of TE. Formally put we want to solve the following optimization problem:

$$\begin{aligned} \text{Max} \quad & Th[H] \\ \text{s.t.} \quad & Pr_{PTE}[\,H\,] \geq \beta. \\ & 0 < \beta \leq 1. \end{aligned} \tag{3.16}$$

Both optimizations problems are integer optimization problems, which in the general case are NP-hard [32]. However, in our scenario the variables bit-rate and packet length take only a limited number of values and thus, we solve these problems with an exhaustive search over the feasible set.

In order to find the capacity of the entire network, we need the distribution of the "hop counts" of the connections established in the network. The hop count distribution of multi-hop paths is jointly determined by factors such as the routing policy, the distance between the source and destination pairs, the node density, etc. Stated otherwise, the hop count distribution is case-dependent and not easy to generalize. We will consider looking at models for the hop count distribution and estimating the throughput capacity in our future work.

3.5 Numerical Results

In this section, we provide our results on quantifying the studied tradeoffs. First, we begin by briefly describing our evaluation environment.

The channel model: The received signal strength from node v_i, at node v_j is:

$$P_{v_i,v_j} = \frac{P_t \cdot |h_{v_i,v_j}|^2}{d^{\alpha}_{v_i,v_j}}, \tag{3.17}$$

where, P_t is the transmission power. h_{v_i,v_j} is the attenuation due to fading between the communicating pair. As typical, we assume that h_{v_i,v_j} is a Rayleigh distributed random variable [18]; thus, $|h_{v_i,v_j}|^2$ is exponentially distributed. d_{v_i,v_j} is the distance between v_i and v_j while α is the path loss exponent.

The collision model: There are several models used to capture collisions in the literature [19]. We use the *SINR (Signal-to-Interference-and-Noise) physical model*, where node v_j successfully receives the transmission from node v_i *iff*:

$$\frac{P_{v_i,v_j}}{P_n + \sum_{k \in \{1,\ldots,N\} \setminus \{i,j\}} P_{v_k,v_j}} > \gamma, \tag{3.18}$$

where, P_{v_i,v_j} is the received power from v_i to v_j and is computed using (3.17), P_n is noise power, v_k is one of the interfering nodes, $\sum_{k \in \{1,\ldots,N\} \setminus \{i,j\}} P_{v_k,v_j}$ is the accumulative interference power perceived by v_j, and γ is the SINR threshold which varies with transmission bit-rate.

Use of multiple bit rates: The success of every packet transmission over a hop is based on the SINR thresholds required. For the HTE, the bit rate is a operational parameter. However, for the PTE availability computation, since we do not consider any specific bit rate selection algorithm over the links, we consider a uniform rate selection on each hop of the path. However, other models can be easily applied here.

Medium access control (MAC): To remove protocol dependencies, we do not assume a specific MAC scheme. Instead, we use a parameter to characterize the interference that nodes perceive, which in turn reflects the interference resolving ability of the MAC in use. This simplified representation avoids modeling the operations of specific MACs. In essence, we assume that λ is the expected interference perceived by a node projected from another in unit time. If λ is low, it reflects an interference-managed MAC; else it represents a MAC where interference is not managed well.

Node distribution: The network consists of N uniformly distributed static nodes (v_i,

$i \in \{1, ..., N\}$).

Traffic pattern: Nodes send Poisson traffic, including their own packets and those to be simply forwarded.

The default parameter settings are listed in Table 3.2. Furthermore, without loss of generality, the values for the rates and SINR thresholds are adopted from 802.11a (see Table 3.3) [33].

3.5.1 TE Availability

We present TE availability when varying different network parameters.

Hop level TE availability: We vary the transmission bit-rate and packet length and compute HTE availability. Fig. 3.1 shows the numerical results. We see that for a fixed bit-rate, a smaller packet length always leads to higher HTE availability. This is because, with a smaller packet length, there is a smaller air time and thus, the chances of a packet being corrupted due to interference (either at the recipient or at the witnesses) is lower.

Furthermore, using lower bit rates (6, 9, 12, 18 $Mbps$) results in almost perfect TE availability (probability ≈ 1), when the packet length is small. When the rate increases (e.g. from $36Mbps$ to $54Mbps$) the TE availability decreases. With higher rates, packets are more susceptible to channel induced losses (require higher SINR thresholds) and thus, are able to only traverse shorter distances. This decreases the probabilities that (a) the recipient and (b) the overhearing witnesses are able to successfully receive the transmissions.

The above effects are more pronounced with larger packet lengths. We observe that when the packet length grows beyond 800 bytes, even the lower rates do not guarantee high TE availability any more. The above performance can be attributed to the longer air time of the packet (both due to the lower rate and the larger length). A longer air time will increase the likelihood of the packet

N (Total number of nodes)	10
P_t (Transmission power)	3.16E-2 *watts*
P_n (Noise power)	3.16E-10 *watts*
α (Path loss exponent)	2.0
λ (Interference level)	20 pkt/sec
R (Transmission range)	100 m
n_r (Retransmission limit)	7
Data overhead length	50 bytes
l_A (ACK length)	20 bytes
Rates and SINR thresholds	See Table. 3.3 for the SINR thresholds and rates.

Table 3.2: Parameter values in evaluation

Rate (Mbps)	6	9	12	18	24	36	48	54
SINR (dB)	6.02	7.78	9.03	10.79	17.04	18.8	24.05	24.5

Table 3.3: Rates and SINR thresholds in evaluation

being exposed to interference. The best rate to use is case-dependent. In the case we consider, the rate 18 (36) *Mbps* achieves the highest TE availability for a packet length of 800 (1500) bytes as seen from Fig. 3.1).

Path Level TE availability: We now vary the packet length and hop count; other parameters remain at default settings while recalling that for PTE we consider a uniform rate selection on the independent links constituting the path. We look at paths with hop count from 1 to 7. Especially, the path level TE with hop count 1 is the same to the hop level TE. The results are presented in Fig. 3.2. As we can see, shorter packet lengths yield a higher PTE availability. For example, PTE with a packet length of 200 bytes can be up to 3 times of that with a packet length of 1500 bytes, when the hop count is larger than one. This trend is consistent with that in HTE. Furthermore, as we can see the PTE decreases quite fast with increasing hop count. In particular,

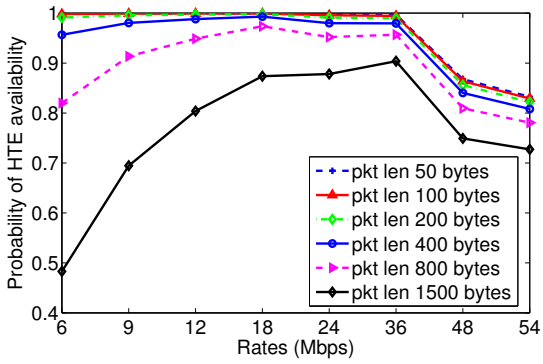


Figure 3.1: Hop level TE availability

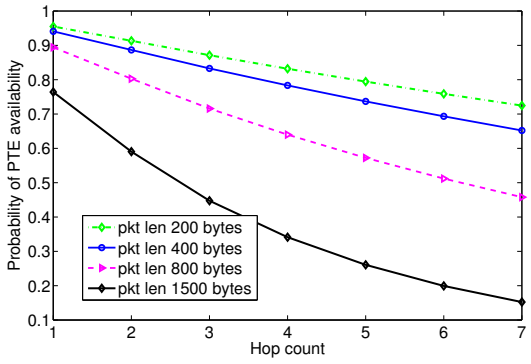


Figure 3.2: Path level TE availability

if the hop count increases by 1, PTE decreases by 10-20%.

3.5.2 Trade-offs between TE and Performance

Next we look at the trade-offs between TE and performance.

End to end delay: We examine the minimum achievable delay on a path given an TE requirement threshold β. Fig. 3.3 shows the minimum expected delay achieved as a function of β with a varying hop count H (1 to 7). The packet length is 800 bytes and the retransmission limit is

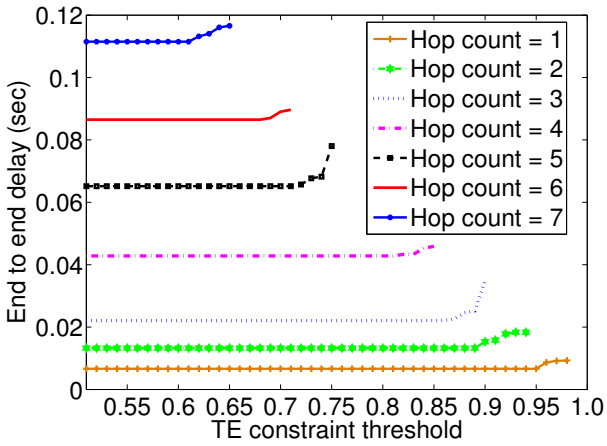


Figure 3.3: Minimum end to end delay under TE constraint

7 while β varies from 0.50 to 1.0[3].

As one can notice, every curve is limited by the maximum possible TE achievable under the corresponding settings (see Fig. 3.3). It is also seen that the optimization problem is rendered infeasible for specific combinations of parameters. Specifically, we also observe a tipping phenomenon; the end-to-end delay remains constant over a large range of values for β, however, for a small range just before the highest achievable β is reached, the delay increases abruptly.

End to end throughput: Next we examine the maximum throughput for various path level TE availability constraints (varying β). In Fig. 3.4 we depict the maximum throughput versus β. Similar with the delay, there is a maximum PTE requirement achievable on a path and the same tipping phenomenon arises. For example, with a two-hop path, there is a drop in the maximum achievable throughput (by more than 33%) when the requirement β approaches the maximum possible value for this configuration (i.e., 0.93). In other cases that such constraint is loose, the maximum achievable throughput does not get affected.

[3]For $0 < \beta < 0.5$, the TE constraint is too weak and thus, has no impact on the minimum delay.

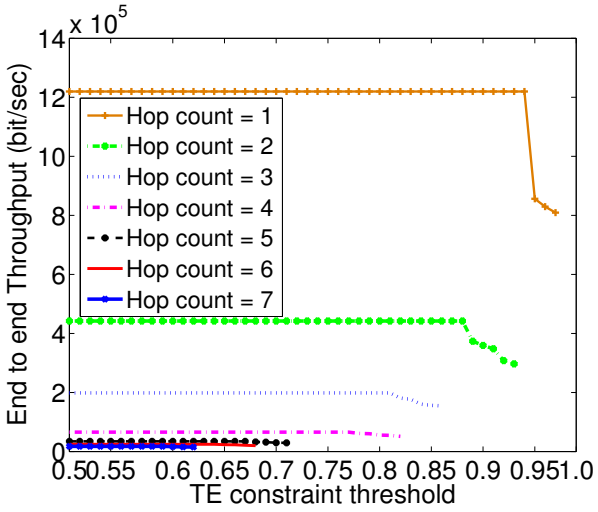


Figure 3.4: Maximum end to end throughput under TE constraint

Inference: These results suggest that upto a certain TE requirement threshold, different for different hop counts, the requirement has virtually no impact on the minimum achievable delay or the maximum achievable throughput. However, as the requirements almost reach the maximum achievable, the delay increases and the throughput drops; this suggests that in order to satisfy the high TE requirement regime the performance has to be degraded. These results can be explained with our observations that in most cases the highest TE availability and the highest performance are achieved at the same combination of rates used in the path. As we increase the rate (from the lowest rate), the maximum achievable throughput increases; this is because the air time decreases and the likelihood that the packet is hurt by interference decreases. Beyond a certain rate, the channel effects dominate and the throughput drops. Recall that, this is exactly what we observe with the TE availability as well.

In summary, there exists a trade-off between high TE and high network performance in throughput and delay, as we intuitively believe to begin with our work. According to the observation

results with our considered setting, unless the TE requirement is too stretching, there will not be a hit on either throughput or delay. As the TE constraint approaches the peak value of path level TE availability, the performance has to take a hit to satisfy this requirement.

These results suggest that operating at the regime in which the maximum throughput (or minimum delay) is achieved can also result in a high likelihood that TE exists. Trying to increase TE availabilities to extremely high level can render the network inoperable due to excessive delays or very low throughputs.

3.6 Conclusions

Transmission evidence is collected and maintained to aid network forensics. Building on our previous efforts on analyzing the availability of TE, in this paper, we develop analytical models to capture the trade-offs between provisioning TE and network performance in multi-hop wireless networks. Our results indicate that both delay and throughput can be affected when we require levels of TE availability close to the maximum possible (extremely high). As long as high levels of TE availability are not required, one can maintain high throughput and low delay.

Chapter 4

Directional Neighbor Discovery in 60 GHz Indoor Wireless Networks

Signal propagation in the 60 GHz band significantly differs from that in the 2.4 and 5 GHz bands. In particular, the signals are often reflected in indoor settings in this band. Directional antennas can help limit the impact of reflection, but make the process of neighbor discovery complex. We consider this problem in this chapter. We examine two approaches (a) direct discovery where each node explicitly discovers its neighbors and, (b) gossip-based discovery where nodes exchange information about their already discovered neighbors. We analyze the two approaches and validate our models via simulations. We examine the impact of system parameters such as varying beamwidth and node density.

4.1 Introduction

The unlicensed 60 GHz band has been the focus of recent attention as a candidate for materializing the transport of multimedia applications such as high definition (HD) video streaming over the wireless medium [34, 35, 36]. One may envision the construction of a wireless gigabit ethernet using the 60 GHz band. Compared with the maximum data rate of 54 Mbps supported by current wireless networks operating in the 2.4 GHz band (IEEE 802.11 b/g) or the 5 GHz band (IEEE 802.11a), the 60 GHz band can potentially support multi-gigabit data rates using 7 GHz of bandwidth; for the unlicensed 60 GHz band, the 57-64 GHz spectral regime in the US and the 59-66 GHz regime in Europe and Japan are reserved.

The 60 GHz band has distinct signal propagation characteristics as compared with the 2-5 GHz band. In particular, the propagation loss in the 60 GHz band (with the same transmitter and receiver antenna gains) is about 20-30 dB higher than in the 2-5 GHz bands. The penetration loss is also higher in the 60 GHz band. In addition, oxygen absorption is intensified as compared to the other bands. This absorption attenuates signals over distance by about 15 dB per km. The diffraction effects are much smaller in the 60 GHz band in comparison to the 2-5 GHz bands. The electromagnetic (EM) field of the 2-5 GHz signals is composed of diffracted and reflected waves and thus, has a complex structure with no traceable directions of arrival and departure of the signals. On the other hand, the EM field of a 60 GHz signal has a structure consisting of a few rays coming from the direct path (if available) and from first-order reflections; thus, the directions of arrival and departure are very close to what can be predicted with ray tracing (geometrical optics) laws. The experiments performed in indoor scenarios (such as in conference room, library, cubicle and aircraft environments) in [37, 38, 39] suggest that due to these reasons, communications between a transmitter and a receiver in this band are established via both direct and reflected beams.

The use of high-gain directional antennas can be especially attractive in the 60 GHz band,

given the above propagation effects. They can significantly increase communication range and this can be especially useful given the high propagation loss in typical indoor 60 GHz environments [36, 39]. Furthermore, the use of directional antennas can limit the number of reflected beams and thereby increase space reuse [37]. Given the higher spectral regime of the 60 GHz band the requirement on the size of the antennas is drastically relaxed [1] [35]; in this band, it is easy to fit many antenna elements on a small platform to create a high gain antenna array. In practice, an antenna array with more than 20 elements can achieve a gain of more than 13 dB [36]. In addition, the mitigated diffraction effects make it possible for highly directional antennas to focus most of the transmitted energy at the intended recipient; this drastically reduces the interference between links in the same geographic area.

We consider a 60 GHz wireless network in an indoor setting; we assume that nodes are equipped with antenna arrays that are capable of generating high gain focused beams. The distinct characteristics of the 60 GHz band make the problem of networking these nodes different from what has been considered before in the 2.4 and 5 GHz bands. In particular, one has to account for reflected beams when designing and evaluating protocols for neighbor discovery and media access control.

In this work, our objective is to analyze neighbor discovery in the 60 GHz band. Neighbor discovery in the considered setting is much different from that in traditional 2-5 GHz bands. *In this band, neighbors can be discovered not only via direct line-of-sight (LOS) beams, but also via reflected beams, which has not been considered before.* We consider two possible approaches for neighbor discovery viz., direct discovery and gossip-based discovery, which were previously introduced for neighbor discovery with directional antennas in wireless networks [40] (the setting considered in [40] however, did not account for reflections that arise due to operations in the 60 GHz band). With direct discovery, a node discovers a neighbor only when it successfully receives a transmission

[1] The elements of an antenna array should be separated by a distance that is of the order of the wavelength in use.

from that neighbor. With gossip-based discovery, a node can discover a neighbor either via direct discovery or from some other node (possibly a different neighbor) that has information about that neighbor. Our work provides a basis for establishing links in the 60 GHz regime and insights on the choice of operational parameters to employ under different conditions (such as different node densities).

In more detail, the major contributions of our work are as follows:

- **We develop analytical models for both direct discovery and gossip-based discovery in the 60 GHz band.** The proposed models take into account both direct and reflected beams. These models can effectively characterize the performance of the considered approaches in different settings; in particular, variations in node density, and directional beamwidth can be characterized.
- **We build a simulation framework that reflects operations in the 60 GHz band and validate our analysis through extensive simulations.** In particular, our simulation framework accounts for the presence of obstacles (such as exterior and interior walls) in an indoor wireless network. We use a ray-tracing method to simulate the interactions between signal propagation and the obstacles. Our implementation accounts for penetration and reflection with proper loss. The behaviors of directional antennas with varying beamwidth are also captured in the simulations.
- **We comprehensively examine the impact of various key parameters (such as node density and antenna beamwidth) on the performance of the neighbor discovery schemes.** Our extensive studies in some typical indoor scenarios reveal interesting trade-offs in performance, resulting from tuning the above parameters.

The rest of the paper is organized as follows. In Section 4.2, we discuss prior related work in 60 GHz band. Related work on the use of directional antennas in wireless networks is also dis-

cussed. In Section 4.3, we describe the two considered neighbor discovery approaches, namely, *direct discovery* and *gossip-based discovery*. In Section 4.4, we present our analytical models corresponding to the two neighbor discovery methods. The performance evaluations of the considered neighbor discovery methods via both simulations and using our models, are described in Section 4.5. We conclude in Section 4.6.

4.2 Related Work

Prior research efforts on the 60 GHz band have primarily focused on measuring and modeling the channel between a transmitter and a receiver in indoor and short range outdoor scenarios. In this section, we first review previous work on measurement and channel modeling; and then discuss related work on networking with directional antennas and in particular directional neighbor discovery in wireless networks.

4.2.1 Measurement Studies in the 60 GHz Band

We categorize the previous work on measurements as follows:

Path Loss and Coverage: Channel measurements in [37, 41, 42] indicate that transmissions in the 60 GHz band experience a fixed loss (loss in the first meter or so) that is greater than 70 dB. This imposes stringent restrictions on the range of 60 GHz transmissions. The path-loss exponent is around 2 or smaller, but obstacles can cause significant variations in loss (e.g. from 6 dB for plywood panels to up to 48 dB for a brick wall) [41, 43]. The range and thus, interference from other transmissions that impact a given transmission are heavily environment dependent. Longer ranges are possible in open space as compared to indoor areas. Reflections that bypass the obstacles can also result in better coverage. Consequently, a receiver that is further (say $2x$ m) away may have an excellent link to a transmitter whereas a closer receiver (say just x m away) may not hear the

transmitter. These artifacts have an impact on neighbor discovery.

Propagation mechanisms: Measurements in [37, 38, 39] show that in the 60 GHz band signal propagates via both a direct path and by means of reflections from objects. The work in [37] provides measurement results and analyzes the 60 GHz space-time channel in LOS room and hallway environments. Power delay profiles (PDPs) and power angle profiles (PAPs) are measured. The multipath structure retrieved from PDPs and PAPs demonstrates a strong correlation with the propagation environment. The measurement results in an aircraft [38] and in a conference room [39] reveal that a received 60 GHz signal is mainly a combination of the direct and first-order reflected (reflected only once) signals. The first-order reflected signals bounce off walls, the ceiling and the floor. Signals that penetrate through cubicle walls have been found to be sufficient for establishing communication links [39]. The work in [37] suggests that image-based ray tracing of first order reflections can be used for channel prediction in LOS applications. The work in [44] obtains a good match between the spatial and temporal characteristics of measured signals and those simulated using ray tracing.

4.2.2 Modelling the 60 GHz Channel

With plenty of measurements, the IEEE 802.15.3c group [45] developed a 60 GHz channel model for library environments [46]. Based on the generalized Saleh-Valenzuela channel model (or S-V model for short), this statistical channel model accounts for both LOS and non-LOS (NLOS) components, in both the time and angular domains. Most recently, a conference room channel model for 60 GHz WLANs was proposed in [47]. This model is partially based on experimental measurements in [39]. The proposed model structure again adopts the clustering approach (as with the S-V model) in both time and angular domains. Different types of antennas with diverse antenna patterns can be characterized by the model. Furthermore, the model allows for antenna

beamforming to be applied both at the transmitter and at the receiver. The proposed structure of the channel model can be applied to model other scenarios like cubicles and living rooms, if statistical characteristics specific to the scenario are available. The problem with the above two statistical models is that for each specific scenario, extensive experimental measurements are required. In contrast, our simulations are based on a simple partition based path loss model [48] and a ray tracing method used in conjunction. The partition based path loss model and simple deterministic ray tracing are reported to be efficient in terms of describing the general channel properties of the 60 GHz band [44, 37]. With this model, signal transmission between a transmitter and a receiver can be approximated as rays in different directions. For each specific ray, the total path loss can be calculated as the sum of the log-distance path loss and the partition attenuation factors (PAF), which correspond to the reflection or penetration losses.

4.2.3 Directional Antennas in Wireless Networks

While protocols for directional antennas in wireless networks have received a great deal of attention, most of the efforts are based on omni-directional receptions. There is limited work on fully directional communications [49][50]. As discussed earlier, the use of a directional antenna is desirable for 60 GHz operations since its beamforming ability can combat higher propagation losses and provide longer communication range. Some recent efforts address directional communications in the 60 GHz band [51, 52], however, either outdoor networks are assumed, as in [51], or only the blockage of beams by objects is considered as in [52]; reflections are ignored in these works. The key differences between prior efforts and our work are as follows: (a) all of the previous efforts assume that only a LOS path between a transmitter and its intended receiver is possible. However, NLOS communications are possible in the 60 GHz band due to reflections. (b) most previous efforts assume that a transmitted beam has a perfect conical shape and is of a fixed radial and angular range; any

receiver within this range will receive the signal in the absence of interference. However, in the 60 GHz band, there are reflected beams which suffer different levels of attenuation from reflectors and thus, the previous assumption is no longer satisfied.

4.2.4 Neighbor Discovery in Wireless Networks

There has been prior work on neighbor discovery (we refer to it as ND for short) using omni-directional and directional antennas [40, 53, 50, 54]. Previous efforts on developing a ND algorithm simply assume that the omni-directional footprint is a circle and the directional footprint is a perfect cone. In addition, only a single LOS communication link between a transmitter and a receiver is assumed. However, neighbor discovery becomes much more complicated in 60 GHz indoor settings since a neighbor can be discovered not only via a direct beam, but also with reflected beams. Furthermore, interference also arises due to both direct and reflected beams. In [55], a neighbor discovery algorithm for directional 60 GHz networks is proposed. The approach detects the presence or absence of a direct path between a pair of nodes by exploiting the polarization effects in LOS and NLOS environments. Based on this knowledge, the most effective direction for a transmitter to use for a specific receiver, is determined via neighbor discovery. Our work differs from that in [55] in that the latter focuses on exploiting the information obtained during neighbor discovery; such information includes condition of the channel (LOS or NLOS) and the best direction for a node to transmit to another node; whereas our work quantifies the effectiveness of the neighbor discovery process, and in particular examines the impact of reflected beams on the process in 60 GHz indoor networks.

4.3 Neighbor Discovery Methods

In this section, we provide an overview of neighbor discovery (ND) first. Two approaches for ND, direct and gossip-based discovery that are considered in this paper, are then introduced. As mentioned earlier, these approaches are derived from what was proposed in [40] for directional antennas in line-of-sight settings.

4.3.1 Background in Neighbor Discovery

ND is an essential process in the self configuration of multi-hop wireless networks; MAC, routing, and topology-control in wireless networks require the discovery of neighbors. The discovery process should be as rapid as possible and typically facilitates the bootstrapping of other protocols that utilize the knowledge derived from ND. There are trade-offs between using omni-directional and directional communications for ND. An omni-transmission can potentially reach a larger group of neighbors; however, the interference effects are likely to be more pronounced than with directional communications. With fully directional communications, a lower level of interference is incurred since the beamwidth used for ND is usually much smaller than 2π. However, ND is more difficult since a pair of nodes must align their antennas for successful discovery. Even if two nodes are within range, they cannot discover each other if their antennas do not align.

ND algorithms can be categorized in the following ways: (1) the way in which a node responds to a broadcast message [54] and (2) the way a node gains access to medium [56].

In the first category, ND algorithms can be further classified into one-way ND and handshake-based ND. In one-way ND, a node periodically transmits a broadcast packet containing its ID and possibly its location, to announce its presence. Neighbors are discovered by receiving broadcast packets from other nodes, and neighbor lists are then updated. Handshake-based ND is more complex; a receiver needs to provide an acknowledgement to an undiscovered transmitter upon receiving

a broadcast packet from that transmitter.

In the second category, ND methods can be further sub-divided into two classes: synchronized slotted ND and random access based ND. In synchronized slotted ND, a node chooses to transmit or receive at the beginning of each time slot. In contrast, with random access based ND a node receives for a random time interval. Upon interval timeout, the node transmits and then, returns to the receive mode.

The two ND methods that we consider in this paper, direct discovery and gossip-based discovery, belong to one-way and synchronized slotted ND classes.

4.3.2 Direct Discovery

At the beginning of each time slot, a node chooses to be in one of two states: **transmit** or **receive**. In the transmit state, a node transmits a broadcast packet with its identity, in a randomly chosen direction with a given beamwidth. In the receive state, a node listens for broadcast packets from a randomly chosen direction; if the received signal experiences a collision, the node fails to discover any neighbor. If the broadcast packet is received successfully and the transmitter is an unknown neighbor, the receiver records the angle of arrival (AoA) and the transmitter's identity. The transmitter is then said to be discovered. Furthermore, during this process, the most effective direction for transmission to the discovered node can be identified by comparing the received signal strength from all recorded AoAs, as suggested in [55].

Fig. 4.1 and 4.2 depict neighbor discovery with the direct discovery method. We see that the directional footprint is not a perfect cone[2] when there is a reflector (obstacle) within the footprint; instead, the footprint is *folded* at the reflector, forming a new *reflection area*. In Fig. 4.1, the receiver cannot discover transmitter A because its reception range does not cover transmitter

[2]We ignore sidelobes in this work for ease of discussion and tractability of our analysis.

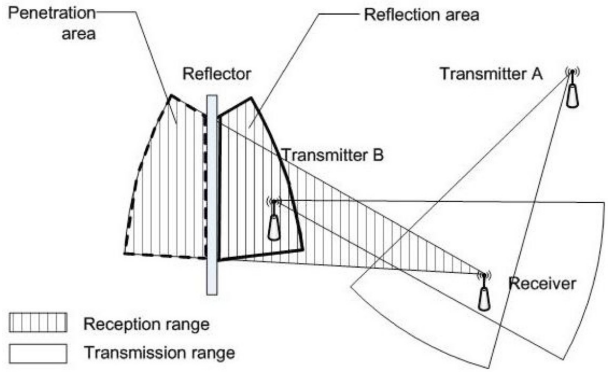


Figure 4.1: Neighbor discovery via direct beam

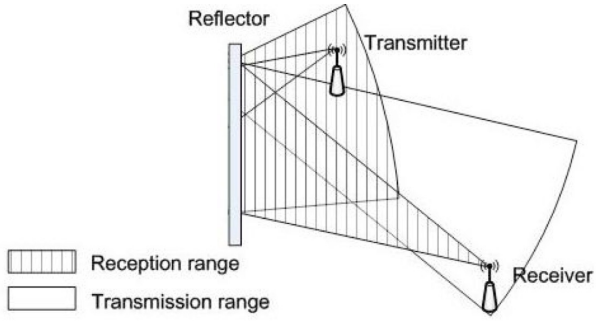


Figure 4.2: Neighbor discovery via reflected beam

A; note that this is the case even though transmitter A's transmission range covers the receiver. On the other hand, since transmitter B and the receiver are within each other's footprint, and since there are no interfering transmissions, the receiver discovers transmitter B. Fig. 4.2[3] illustrate the process of discovery via a *reflected beam*. We see that neither the transmitter nor the receiver's antenna directly point towards each other. However, the receiver can discover the transmitter in this case due to the fact that transmitter's reflected beam is aligned with the direction of antenna at the receiver.

[3]We do not show the signal that penetrates the reflector for clarity.

4.3.3 Gossip-based Discovery

Gossip-based discovery consists of two steps. In the first step, like direct discovery, a node chooses to transmit or receive at the beginning of each time slot. In this step, unlike direct discovery, a node in the transmit state not only broadcasts its own identity and location (possibly obtained with GPS), but also "gossips" about the neighbors that it has discovered so far. The gossip information includes the identity and location of each discovered neighbor. In this way, a node that successfully hears a transmission not only discovers the transmitter, but also learns about the neighbors that the transmitter has discovered so far. The node stores the information (discovered neighbors and their locations) obtained in this step, temporarily, in a table (called the gossip table). Note that nodes in the gossip table are *not yet* accepted as neighbors. In the second step, a node tries to find out whether these *indirectly* discovered neighbors are within its maximum transmission range. For each node in the gossip table, a node sends out probe packets in the direction of the discovered node (the direction is calculated based on the location information obtained in the previous step). If the response from the discovered node is obtained within a given number of tries, it is accepted as a *genuine* neighbor and is removed from the gossip table. Otherwise, it is just removed from the gossip table. The rationale for this step is that *not* every node in the newly acquired neighbor list can be guaranteed to be the recipient node's neighbor. This is because neighbor discovery depends on environment around the nodes; the presence of physical obstacles may even cause two geographically close nodes to not be "logical" neighbors. Note that this step is not required in direct discovery.

As evident from the above discussion, compared with the simple direct discovery method, the gossip-based discovery method raises some implementation challenges. First, the procedure of sifting genuine neighbors from the gossip (the second step above) inevitably incurs complexity to some extent in terms of implementation. In addition, the procedure can consume extra time in the discovery process. A second issue that could arise with gossip-based discovery is increased packet

size. In very dense network settings, the neighbor list will need to accommodate more neighbors, leading to increased packet size. As a result, the transmission of packets take longer times. This would impair the performance of discovery. In typical office and home settings however, given the short range of 60 GHz signals, this is unlikely to be a significant problem.

4.4 Analysis of Neighbor Discovery

In this section, we analytically model the considered neighbor discovery methods. The key performance metric for a ND algorithm should capture *the time it takes for nodes to discover their neighbors from scratch.* Thus, our analysis aims at deriving the expected fraction of neighbors discovered within a certain time. We begin with describing the system model under consideration and subsequently present our analysis for direct discovery and gossip based discovery.

4.4.1 System Model

We consider ad-hoc networks in indoor environments (e.g., offices or homes), where nodes are static and uniformly distributed. Each node is equipped with directional antennas with the same transmission power and antenna gain. Nodes transmit and receive directionally, and function in a half-duplex mode (i.e., it can *either* transmit or receive signals but cannot do both simultaneously). Nodes may or may not be equipped with GPS, as long as they have information about their locations and have some means to be synchronized. We assume that nodes are placed on a plane; the variation in the antenna beam pattern over the elevation angle is not considered (thereby, reflection occurs only at walls and not at the floor and ceiling).[4] The angular footprint of a directional antenna corresponds to a sector with a radius equal to the directional transmission range. Within its beamwidth the antenna has unit gain and a zero gain outside the beam. We assume that there are k available

[4] An extension of the analysis to consider three dimensional effects will be considered in future work.

non-overlapping directions ($k > 4$) for an antenna and thus, the beamwidth for each direction is $2\pi/k$.

As mentioned earlier, nodes do neighbor discovery in a synchronized slotted way. The clocks on all nodes are synchronized, as might be the case if the common clock source is GPS. In each time slot, nodes are in either in a transmit state or in a receive state. A node chooses the transmit state with probability P_t, and the receive state with probability $1 - P_t$. For transmitting or receiving, a direction for pointing the antenna is randomly chosen from among the k directions. We assume that reception failure at a receiver is caused only due to collisions (and not by signal degradation due to channel induced effects). A collision occurs at a receiver when two or more transmissions are simultaneously received. In Section 4.5, we show that the assumed collision model effectively captures indoor environments by comparing the analytical results to those obtained by simulating an indoor channel model (the partition based path loss model) driven by measurements in the 60 GHz regime.

Only first-order reflections are considered for the analysis. This is because, generally, higher order (more than two) reflections suffer from high degrees of signal attenuation to successfully establish communication links. We also ignore reflection after penetration of signals through obstacles for the same reason. Reflections occur at the boundaries of the deployment area (e.g., exterior walls) and at obstacles (e.g., interior walls). Table 4.1 summarizes the notation used in the following analysis.

4.4.2 Direct Discovery Analysis

Given a time slot, the event that node i discovers a particular neighbor j, occurs only when i's (the receiver) beam covers j (the transmitter). The probability $P_{i,j}$ that node i discovers a particular neighbor j is calculated by first conditioning on the probability that node i's beam covers

Notation	Description
k	Number of directions
f	Direction index ($1 \leq f \leq k$)
N	Number of nodes in the network with the exception of the node performing the neighbor discovery (total number of nodes = $N + 1$)
$c_{i,j}$	Expected number of directions in which node i's beam covers a particular neighbor j
A	Area of the network
$A_l(i, f)$	Area where nodes can communicate with node i using beam direction f only via a direct beam. See Fig. 4.3
$A_r(i, f)$	Area where nodes can communicate with node i using beam direction f only via a reflected beam. See Fig. 4.3
$A_b(i, f)$	Area where nodes can communicate with node i using beam direction f via either direct or reflected beams. See Fig. 4.3
$A_s(i, f)$	Area that a beam of node i covers with the antenna pointed in direction f (i.e., $A_s(i, f) = A_l(i, f) \cup A_r(i, f) \cup A_b(i, f)$)
P_t	Probability that a node transmits in a slot
$P^L_{i,j}(f)$	Probability that i discovers a neighbor j in area $A_l(i, f)$ in a slot by using beam direction f
$P^R_{i,j}(f)$	Probability that i discovers a neighbor j in area $A_r(i, f)$ in a slot by using beam direction f
$P^B_{i,j}(f)$	Probability that i discovers a neighbor j in area $A_b(i, f)$ in a slot by using beam direction f
$P_{i,j}$	Probability that i discovers a neighbor j in a slot
$D_{i,j}(T)$	Probability that i discovers a neighbor j directly within the first T slots
$I_{i,j}(T)$	Probability that i discovers a neighbor j indirectly (with gossip-based discovery) within the first T slots
$S_{i,j}(T)$	Probability that i discovers a neighbor j either directly or indirectly within the first T slots
$P_i(d, T)$	Probability that i discovers d neighbors within the first T slots
$F_i(T)$	Expected fraction of neighbors discovered by node i within time T

Table 4.1: Notation used in analysis

j:

$$P_{i,j} = P(i\text{'s beam covers } j) \cdot P(i \text{ discovers } j \mid i\text{'s beam covers } j). \tag{4.1}$$

Let u denote the number of directions (from the perspective of i) in which node i's beam covers a particular neighbor j. The value of u is determined by the locations of i and j, the positions of the reflectors, as well as the beamwidth. Let us consider as an example, a room with four walls (reflectors). Node i is located in the center as shown in Fig. 4.4(d). One can divide the room into four areas in terms of the value of u when a beamwidth of 30 is used. In order to find these areas, we consider a transmission by i in each of its sectors and use ray tracing tools [57] to find the coverage region for such transmission [5]. Parts of the room that are covered by the same number of transmissions are grouped into an "area". We assume that the wireless channel is reciprocal. In the example considered, we can find four such areas. If the transmitter is in area (1) or (3), node i can receive signals from directions 4, 7, 9 and 12, and u has a value 4 (Fig. 4.4(a)) [6]. Similarly, for area (2) (Fig. 4.4(b)) and area (4) (Fig. 4.4(c)), the values of u are 5 and 3, respectively. Then, the expected number of directions $c_{i,j}$ in this case can be calculated by

$$c_{i,j} = 4 \cdot \frac{\text{area of ((1)+(3))}}{A} + 5 \cdot \frac{\text{area of (2)}}{A} + 3 \cdot \frac{\text{area of (4)}}{A}. \tag{4.2}$$

Since there are totally k possible directions to choose from, $P(i\text{'s beam covers } j)$ in Eq. (4.1) is given by $\frac{c_{i,j}}{k}$.

Now, we derive $P(i \text{ discovers } j \mid i\text{'s beam covers } j)$. The event "node i discovers node j" is affected by the direction in which i points its antenna (we call it the beam direction of i) and the location of j. For each beam direction f, the beam areas (defined in Table 4.1) $A_l(i,f)$, $A_r(i,f)$ and $A_b(i,f)$, possibly containing different numbers of interfering nodes, are created. These

[5] A ray corresponds to the antenna boresight.

[6] For clarity, we show the direction of transmission as opposed to beams.

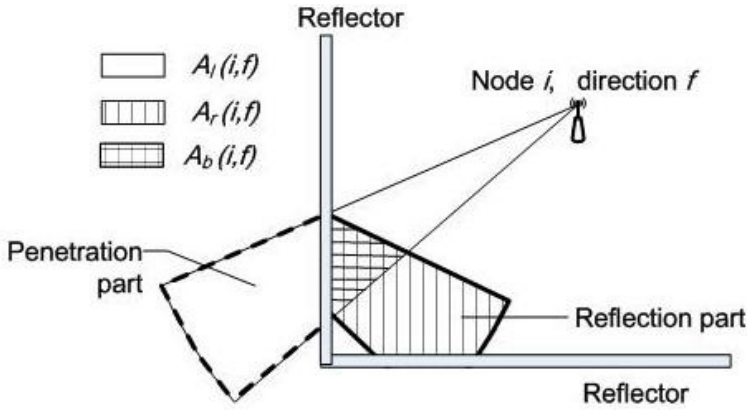
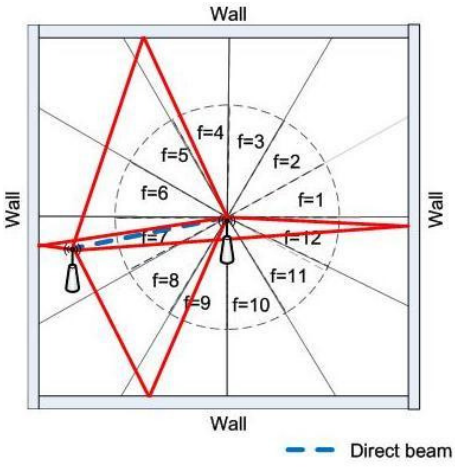


Figure 4.3: An illustration of $A_l(i,f)$, $A_r(i,f)$ and $A_b(i,f)$

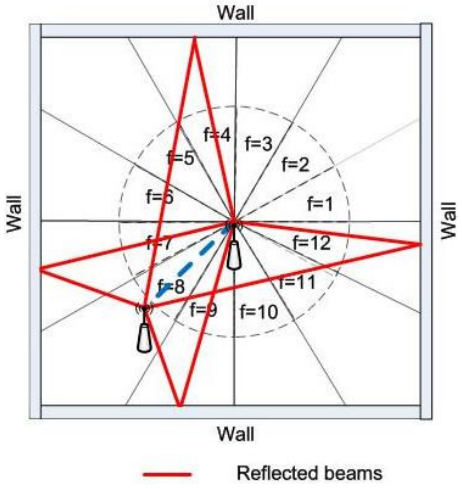
interferers affect the probability of interest. The location of j also affects this probability. If j is within $A_l(i,f)$ (or respectively $A_r(i,f)$), node i can discover j only via a direct beam (or respectively a reflected beam). If j is within $A_b(i,f)$, node i can discover j via either a direct or a reflected beam. Conditioning on the beam direction f, we have

$$\begin{aligned} &P(i \text{ discovers } j \mid i\text{'s beam covers } j) \\ &= \sum_{f=1}^{k} P(i \text{ points at direction } f) \cdot P(i \text{ discovers } j \mid i \text{ points at direction } f \text{ and } i\text{'s beam covers } j), \end{aligned} \tag{4.3}$$

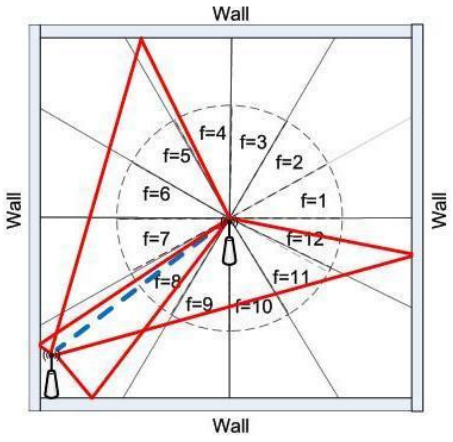
where $P(i \text{ points at direction } f) = \frac{1}{k}$ since the direction is randomly chosen. We condition on the location of j in the second term in the right-hand side of Eq. (4.3) and obtain:



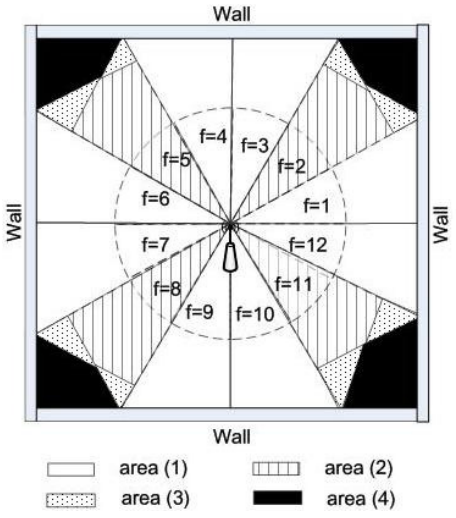
(a) Case of $u = 4$: j is in area (1) or (3)



(b) Case of $u = 5$: j is in area (2)



(c) Case of $u = 3$: j is in area (4)



(d) Different beam areas

Figure 4.4: Illustration of how $c_{i,j}$ is computed in terms of different transmitter positions

$$
\begin{aligned}
&P(i \text{ discovers } j \mid i \text{ points at direction } f \text{ and } i\text{'s beam covers } j) \\
&= P(i \text{ discovers } j \mid j \text{ is in } A_l(i,f)) \cdot P(j \text{ is in } A_l(i,f) \mid j \text{ is in } A_s(i,f)) \\
&+ P(i \text{ discovers } j \mid j \text{ is in } A_r(i,f)) \cdot P(j \text{ is in } A_r(i,f) \mid j \text{ is in } A_s(i,f)) \\
&+ P(i \text{ discovers } j \mid j \text{ is in } A_b(i,f)) \cdot P(j \text{ is in } A_b(i,f) \mid j \text{ is in } A_s(i,f)).
\end{aligned} \tag{4.4}
$$

Referring to the notation in Table 4.1, Eq. (4.4) can further be expressed as:

$$
\begin{aligned}
&P(i \text{ discovers } j \mid i \text{ points at direction } f \text{ and } i\text{'s beam covers } j) \\
&= P^L_{i,j}(f) \cdot P(j \text{ is in } A_l(i,f) \mid j \text{ is in } A_s(i,f)) + P^R_{i,j}(f) \cdot P(j \text{ is in } A_r(i,f) \mid j \text{ is in } A_s(i,f)) \\
&+ P^B_{i,j}(f) \cdot P(j \text{ is in } A_b(i,f) \mid j \text{ is in } A_s(i,f)).
\end{aligned} \tag{4.5}
$$

Given that j is within area $A_s(i,f)$, the probability that j is within $A_l(i,f)$ is $\frac{A_l(i,f)}{A_s(i,f)}$. Similarly, $\frac{A_r(i,f)}{A_s(i,f)}$ and $\frac{A_b(i,f)}{A_s(i,f)}$ are the corresponding probabilities for the cases where j is within $A_r(i,f)$ and $A_b(i,f)$, respectively. Substituting these expressions in Eq. (4.5), we have:

$$
\begin{aligned}
&P(i \text{ discovers } j \mid i \text{ points at direction } f \text{ and } i\text{'s beam covers } j) \\
&= P^L_{i,j}(f) \cdot \frac{A_l(i,f)}{A_s(i,f)} + P^R_{i,j}(f) \cdot \frac{A_r(i,f)}{A_s(i,f)} + P^B_{i,j}(f) \cdot \frac{A_b(i,f)}{A_s(i,f)}.
\end{aligned} \tag{4.6}
$$

Using Eqs. (4.3) and (4.6), $P_{i,j}$ in Eq. (4.1) can be expressed as:

$$
P_{i,j} = \frac{c_{i,j}}{k} \cdot \frac{1}{k} \cdot \sum_{f=1}^{k} \left(P^L_{i,j}(f) \cdot \frac{A_l(i,f)}{A_s(i,f)} + P^R_{i,j}(f) \cdot \frac{A_r(i,f)}{A_s(i,f)} + P^B_{i,j}(f) \cdot \frac{A_b(i,f)}{A_s(i,f)} \right). \tag{4.7}
$$

Note that $A_l(i,f)$, $A_r(i,f)$, $A_b(i,f)$ and $A_s(i,f)$ in Eq. (4.7) are easy to obtain when the direction of f, the coordinates of node i, the boundaries of deployment and the obstacles are given.

Next, we derive $P^L_{i,j}(f)$, $P^R_{i,j}(f)$ and $P^B_{i,j}(f)$ in Eq. (4.7). Following the direct discovery method described in Section 4.3, $P^L_{i,j}(f)$ can be expressed as:

$$P_{i,j}^{L}(f) = P(j \text{ transmits, } i \text{ receives, and } j\text{'s beam covers } i) \cdot P(\text{no collision}). \tag{4.8}$$

Since the three events in the first term on the right-hand side of Eq. (4.8) are independent of each other, we have:

$$P(j \text{ transmits, } i \text{ receives, and } j\text{'s beam covers } i) = \frac{1}{k} \cdot P_t \cdot (1 - P_t). \tag{4.9}$$

In addition:

$$\begin{aligned} &P(\text{no collision}) = P(\text{no transmission from } A_s(i,f) \text{ except from } j) \\ &= P(\text{no transmission from } A_l(i,f) \text{ except } j) \cdot P(\text{no transmission from } A_r(i,f)) \cdot P(\text{no transmission from } A_b(i,f)). \end{aligned} \tag{4.10}$$

For the calculation of the first term on the right-hand side of Eq. (4.10), let us suppose that there are m nodes (excluding j) in $A_l(i,f)$. These m nodes in $A_l(i,f)$ should not transmit with the beam pointing at i. This corresponds to:

$$P(\text{no transmission from } A_l(i,f) \text{ except } j) = \sum_{m=0}^{N-1} \left(1 - \frac{1}{k} \cdot P_t\right)^m \cdot P(\text{there are } m \text{ nodes in } A_l(i,f)). \tag{4.11}$$

The above expression can be computed if the distribution of nodes within the deployment area is known. The analysis can be applied with any distribution; we assume that the nodes are uniformly distributed in the area of interest, A. With this assumption, the probability that there are m nodes (excluding j) in $A_l(i,f)$, $P_{A_l(i,f)}(N-1,m)$, can be expressed as:

$$P_{A_l(i,f)}(N-1,m) = \binom{N-1}{m} \cdot \left(\frac{A_l(i,f)}{A}\right)^m \cdot \left(1 - \frac{A_l(i,f)}{A}\right)^{N-1-m}. \tag{4.12}$$

Substituting (4.12) into (4.11) gives:

$$P(\text{no transmission from } A_l(i,f) \text{ except } j) = \sum_{m=0}^{N-1} \left(1 - \frac{1}{k} \cdot P_t\right)^m \cdot P_{A_l(i,f)}(N-1, m). \quad (4.13)$$

The derivations of the following expressions are similar to the one above. Note that the maximum possible number of nodes in $A_r(i,f)$ (and in $A_b(i,f)$) later) changes in order to account for those nodes in $A_l(i,f)$:

$$P(\text{no transmission from } A_r(i,f)) = \sum_{n=0}^{N-m-1} \left(1 - \frac{1}{k} \cdot P_t\right)^n \cdot P_{A_r(i,f)}(N-m-1, n), \quad (4.14)$$

and,

$$P(\text{no transmission from } A_b(i,f)) = \sum_{q=0}^{N-m-n-1} \left(1 - \frac{2}{k} \cdot P_t\right)^q \cdot P_{A_b(i,f)}(N-m-n-1, q). \quad (4.15)$$

The probability that (i) i receives "no" transmission from $A_b(i,f)$ and, (ii) when there are q nodes in $A_b(i,f)$, is $\left(1 - \frac{2}{k} \cdot P_t\right)^q$. This implicitly assumes that in addition to the direct beam, there is only **one** beam received due to a first-order reflection. With small beamwidths ($< 45^o$) this is typically the case. For larger beamwidths this expression can easily be refined.

For small beamwidths, multiple reflected beams are incident on a receiver only when its beam covers the intersection of two adjacent walls (a corner) and the transmitter is at such a corner; the probability of this happening is small. For example, consider a scenario where i is at the center of the room with four walls around and a beamwidth of 30° is used, as shown in Fig. 4.5, where the dark shaded corner area is the area of interest. If the transmitter is located in such an area, it can reach i via two reflected beams with a high likelihood. These beams are created by pointing the antenna at either one of the two intersecting walls. In the scenario the receiver i's antenna

should be pointed towards the corresponding corner area (direction 2, 5, 8 or 11). The probability that a transmitter has 2 reflected beams to connect to i is the probability that a transmitter is at such a corner conditioned on the event that i is pointing its antenna towards this corner, and the transmitter is in i's reception range. In this specific scenario we consider, such probability is given by:

$$P(j \text{ has 2 reflected beams reaching } i) = \frac{S_{\text{corner area}}}{S_{i\text{'s reception range when pointing at direction 2, 5, 8 and 11}}} \cdot P(i \text{ points at direction 2, 5, 8 or 11}). \tag{4.16}$$

where S denotes the area. Assuming uniform node distribution, this probability is less than 0.065 as computed from geometrical considerations with ray tracing. Note that although Fig. 4.4 and Fig. 4.5 look alike, they actually serve different purposes: Fig. 4.4 depicts the different beam areas and aids the computation of the average number of directions in which i's beam covers a particular neighbor j, given that j's location is unknown; Fig. 4.5 is used to compute the area in which if j is located, it has two possible directions to produce reflected beams to cover i given that j is in i's reception range, from the perspective of j.

Substituting (4.11), (4.14) and (4.15) in (4.10), we have:

$$\begin{aligned} &P(\text{no transmission from } A_s(i,f) \text{ except from } j) \\ &= \sum_{m=0}^{N-1} \sum_{n=0}^{N-m-1} \sum_{q=0}^{N-m-n-1} \left(1 - \frac{1}{k} \cdot P_t\right)^{m+n} \cdot \left(1 - \frac{2}{k} \cdot P_t\right)^{q} \\ &\cdot P_{A_l(i,f)}(N-1, m) \cdot P_{A_r(i,f)}(N-m-1, n) \cdot P_{A_b(i,f)}(N-m-n-1, q). \end{aligned} \tag{4.17}$$

Substituting (4.9) and (4.17) into (4.8), we have:

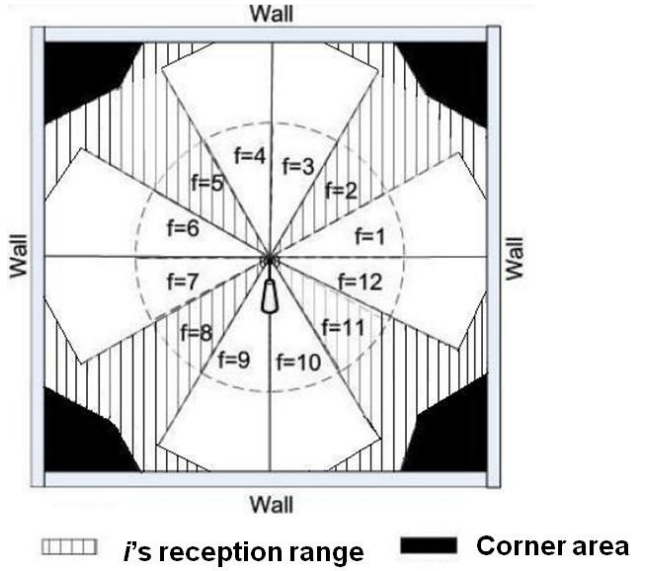


Figure 4.5: i's reception range with direction index 2, 5, 8, and 11 is the striped area PLUS the corner area. If a transmitter is in the corner area, it has two reflected beams to reach i when i points at the transmitter's corner.

$$P_{i,j}^{L}(f) = \frac{1}{k} \cdot P_t \cdot (1 - P_t) \cdot \left\{ \sum_{m=0}^{N-1} \sum_{n=0}^{N-m-1} \sum_{q=0}^{N-m-n-1} \left(1 - \frac{1}{k} \cdot P_t\right)^{m+n} \cdot \left(1 - \frac{2}{k} \cdot P_t\right)^{q} \cdot P_{A_l(i,f)}(N-1, m) \cdot P_{A_r(i,f)}(N-m-1, n) \cdot P_{A_b(i,f)}(N-m-n-1, q) \right\}. \tag{4.18}$$

Using an approach similar to what was used to calculate $P_{i,j}^{L}(f)$, we can obtain $P_{i,j}^{R}(f)$ and $P_{i,j}^{B}(f)$ as shown below.

$$P_{i,j}^{R}(f) = P_{i,j}^{L}(f) \tag{4.19}$$

$$P_{i,j}^{B}(f) = 2 \cdot P_{i,j}^{L}(f) \tag{4.20}$$

Substituting (4.18), (4.19) and (4.20) into (4.7), we finally obtain a closed-form expression for $P_{i,j}$. Using $P_{i,j}$ thus derived, the probability $D_{i,j}(T)$ that node i discovers a particular neighbor j within the first T slots is given by:

$$D_{i,j}(T) = 1 - (1 - P_{i,j})^T. \tag{4.21}$$

The event that node i discovers a particular neighbor within the first T slots is independent of the event that i discovers a different neighbor within the same T slots. Thus the probability $P_i(d,T)$ that i discovers d neighbors within the first T slots is given by:

$$P_i(d,T) = \binom{N}{d} \cdot D_{i,j}(T)^d \cdot (1 - D_{i,j}(T))^{N-d}, \qquad d \leq min(T,N). \tag{4.22}$$

Based on $P_i(d,T)$ obtained above, the expected fraction of neighbors (from among the N neighbor nodes) discovered by node i within time T is:

$$F_i(T) = \sum_{d=1}^{min(T,N)} \frac{d \cdot P_i(d,T)}{N}. \tag{4.23}$$

4.4.3 Gossip-based Discovery Analysis

Recall that with gossip-based discovery, i obtains information about a neighbor not only via a direct reception, but also indirectly from other neighbors. As discussed earlier, node i deems node k to be an indirectly discovered neighbor, if this node lies within the maximum coverage area of node i. It is however possible that this node (node k) is not reachable due to obstacles either directly or via reflections. However, as we show via simulations, in typical scenarios, the probability of such an event is very small (< 0.1 in two main scenarios that were simulated). Thus, for ease of analysis, we assume that i simply considers every node that is within its coverage, discovered indirectly via a directly discovered node, to be its neighbor. Let $D_{i,j}(T)$ and $I_{i,j}(T)$ denote the probabilities that i discovers a neighbor j within T slots directly and indirectly, respectively. Then, the probability $S_{i,j}(T)$ that i discovers j *either* directly *or* indirectly is given by:

$$S_{i,j}(T) = D_{i,j}(T) + (1 - D_{i,j}(T)) \cdot I_{i,j}(T), \tag{4.24}$$

where, $D_{i,j}(T)$ is given by (4.21). We calculate $I_{i,j}(T)$ by solving iteratively the following equation:

$$I_{i,j}(T) = I_{i,j}(T-1) + (1 - I_{i,j}(T-1)) \cdot \sum_{r \neq i,j} P_{i,r} \cdot S_{r,j}(T-1) \tag{4.25}$$

The rationale behind the recursion in Eq. (4.25) is as follows. If a node j is discovered by node i within T time slots, it can do so indirectly with help from a different neighbor (1) within the first $T-1$ slots or, (2) in the T^{th} slot. The first case corresponds to $I_{i,j}(T-1)$ in Eq. (4.25). In the second case, i could learn about j from any neighbor node r except j; r is to be discovered in the T^{th} time slot and should have j in its neighbor list. The probability of this event is $P_{i,r} \cdot S_{r,j}(T-1)$, where $P_{i,r}$ is the probability i discovers r in the T^{th} slot, and $S_{r,j}(T-1)$ is the probability that node r has discovered j within the first $T-1$ slots.

Replacing $D_{i,j}(T)$ with $S_{i,j}(T)$ in Eq. (4.22), we obtain the expressions for $P_i(d,T)$ and $F_i(T)$; these correspond to the probability that node i discovers d neighbors and the expected fraction of neighbors discovered by node i within time T, respectively, using gossip-based discovery.

The analytical results and the impact of the assumptions made, are validated via simulations in Section 4.5.

4.5 Performance Evaluation

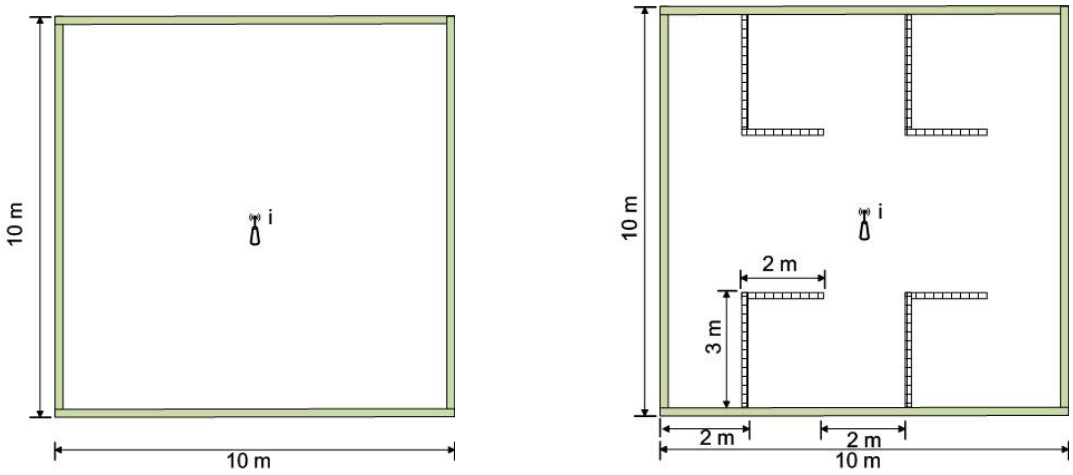
In this section, we evaluate the two neighbor discovery methods considered in this paper in the 60 GHz setting. To this end, we present both numerical and simulation results for room scenarios that we consider. In particular, the impact of a realistic 60 GHz channel and other key parameters on performance are investigated with numerical and simulation results.

4.5.1 Simulation Setup

Simulations are performed in OPNET version 14.5 [22]. For the evaluations, we mainly consider two scenarios, wherein the network of interest is deployed in a room of area 10 m by 10 m. In the first scenario, there are four exterior walls with orthogonal intersections but with no interior walls (see Fig. 4.6(a)); in the second scenario, four interior walls are also present and this represents a typical office environment with cubicles (see Fig. 4.6(b)). Our first set of evaluations are done with the scenario without the interior walls; later we consider the second scenario. Note that, for the second scenario, the thickness and materials used for the interior walls can affect the reflection and penetration losses. In our simulations, we assume that the thickness and materials result in a 10 dB reflection loss and a 15 dB penetration loss (as observed in measurements in [37]). For the first scenario, node i is fixed at the center of the room. We vary the location of node i in the second scenario. Time is slotted as discussed and the slot length is chosen such that all the signals transmitted in a slot are received in the same slot; in other words delay spread is ignored and a transmission does not interfere with communications in the next slot. By default, aside from the tagged node (node chosen for observation), there are 10 nodes in the network. Every node uses the same transmit probability P_t which is set to 0.5 (this is relaxed later to study the impact of varying P_t). In the default setting, we also assume that the number of directions, k, is 12, corresponding to a beamwidth of 30^o (this is also relaxed later for a study of the impact of beamwidth).

4.5.2 Results

Impact of a realistic channel model: In Section 4.4, we assume that multiple simultaneous transmissions in a receiver's beam range leads to a collision. However, realistically, this would depend on the signal strength the receiver perceives, from both the intended transmitter and the interferers. A packet loss occurs if and only if the ratio of received signal strength to the interference



(a) without interior walls

(b) with interior walls

Figure 4.6: Scenarios with and without interior walls

(i.e., SIR) is less than a given threshold. If the interferer's signal is via a reflected beam while the transmitter's signal is direct, a reception could be possible. Thus, the analysis may overestimate the number of collisions and thus, the discovery process may in fact be more efficient than what is predicted. To investigate the impact of this on the neighbor discovery efficiency, we perform simulations to obtain $F_i(T)$ over a number of time slots T with the SIR based model; the channel model incorporates the losses due to reflections as per the measurements reported in [37]. We compute an average over 30 runs, of random neighbor deployments. We set the transmit power to 10mW, antenna gains to 24 dBi at both transmitter and receiver, and the SIR threshold is set to 15 dB as in [51]. From Fig. 4.7, we observe that the $F_i(T)$ achieved with the analytical model is very close to what is achieved with simulations. This demonstrates the efficacy of our model in estimating the performance in realistic scenarios. Given this conformance, in what follows we selectively present either analytical or simulation results for clarity.

Impact of discovery method: Fig. 4.7 shows the fraction of discovered neighbors by

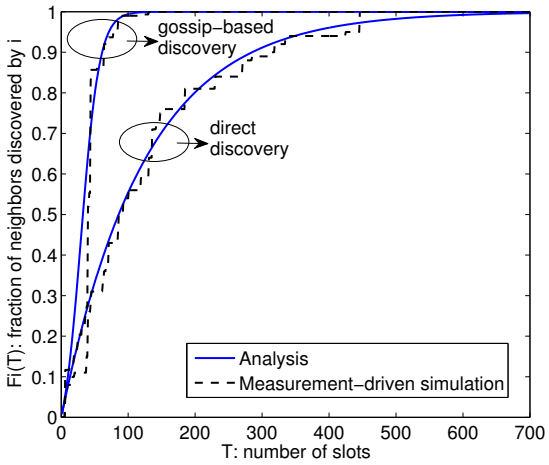


Figure 4.7: Efficiency of direct and gossip-based discovery

node i, $F_i(T)$, as a function of the number of time slots T, with the use of direct and gossip-based discovery. Both numerical results, based on the analysis of Section 4.4, and simulation results are plotted. The simulation results are averaged over 30 runs with different neighbor placements. We see that the analytical results conform with the simulation curves with both direct and gossip-based discovery.

Comparing the results, one can see that with gossip-based discovery a node finds its neighbors much faster than with direct discovery. At the end of 100 time slots, the fraction of neighbors discovered by direct discovery is only about 0.554 while the fraction with gossip-based discovery is 0.994. We see that with direct discovery, it takes 633 time slots to reach the fraction of 0.994; thus, direct discovery is over six times slower than gossip-based discovery. The reason is attributable to the contribution from indirect discovery (recall the gossip-based subsection in Section 4.4). We observe that (Fig. 4.8) the curve depicting $I_{i,j}(T)$, the probability that i discovers j indirectly within T slots, is above the curve depicting $D_{i,j}(T)$, the probability that i discovers j directly within T slots, and in fact, approaches the curve depicting $S_{i,j}(T)$, the probability that i

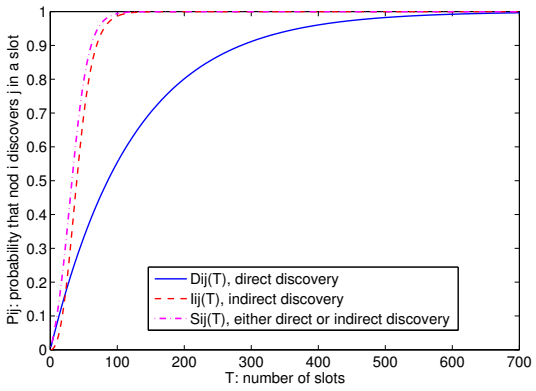


Figure 4.8: Contribution of indirect discovery to gossip-based discovery

discovers j directly or indirectly within T slots. This demonstrates that indirect discovery dominates the discovery process in gossip-based discovery. In our simulations, we observe that a node typically acquires about 2-4 neighbors in a single slot with gossip-based discovery; this contributes to its rapid discovery rate.

Impact of the probability of transmission P_t***:*** In order to investigate the impact of P_t, we plot the value of $P_{i,j}$ from the proposed analysis [7] while varying P_t from 0.0 to 1.0 in Fig. 4.9.

Irrespective of the network density, generally speaking, a higher P_t increases the probability that a node transmits. This increases the chance of being discovered, but interference also grows due to the increased number of transmissions. In particular, interference effects dominate if P_t is close to 1, leading to a discovery probability of zero. On the other hand, small values of P_t result in low levels of interference but also result in a lower chance of nodes being discovered. As one might expect, the discovery probability goes to zero as P_t gets close to zero. We see that P_t needs to be carefully chosen to balance the chance of being discovered and the level of interference generated.

We also find that the right choice of P_t (to induce a high $P_{i,j}$) also depends on the network

[7] Simulation results are observed to be consistent with the analysis results; we do not present simulation results for clarity of presentation.

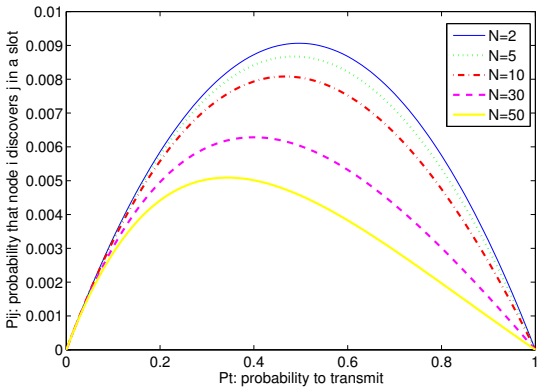


Figure 4.9: Impact of P_t on $P_{i,j}$ with different N

density. When the number of nodes N increased from 2 to 5, 10, 30 and 50, the value of P_t that results in the peak value of $P_{i,j}$ deceases from 0.51 to 0.50, 0.49, 0.43 and 0.38, respectively. This implies that as the network density increases, it is better for nodes to be less

Impact of network density: In Fig. 4.10, we plot $P_{i,j}$, the neighbor discovery probability for node i in a slot, while varying the number of nodes N in the deployed area, from 2 to 50. It is observed that as the number of nodes increases, the node discovery probability typically starts decreasing. This decrease in the probability is due to increased interference effects, which is captured in Fig. 4.11. In this figure we observe that the collision probability increases as the number of nodes grows. From these results, we also find that for a given N, an optimal operating P_t exists and needs to be carefully chosen for best performance.

We also investigate the impact of network density on the two considered discovery methods in Fig. 4.12 and 4.13, respectively. With direct discovery (Fig. 4.12) the performance gets worse as the number of nodes increases. This is attributed to increased interference levels in the network (nodes have to explicitly discover each neighbor). On the contrary, this phenomenon is not observed with gossip-based discovery (Fig. 4.13). The performance of neighbor discovery in terms of the

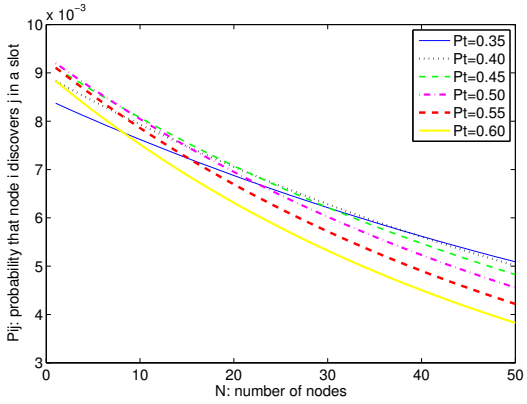


Figure 4.10: Impact of N on $P_{i,j}$ with different P_t

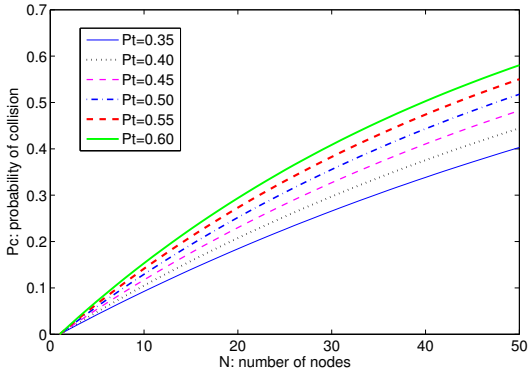


Figure 4.11: Probability of collision

fraction of discovered nodes, is the highest with $N = 50$. This is because an increase in node density increases the chances of a node discovering its neighbors indirectly. This gain with gossip-based discovery dominates the effects of increased interference (each success drastically increases the fraction of discovered neighbors).

Impact of beamwidth: To see the impact of beamwidth on the performance of neighbor discovery, we find the fraction of discovered neighbors as a function of slots while varying the

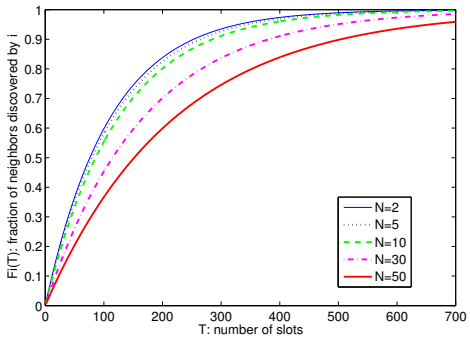


Figure 4.12: Impact of N on direct discovery

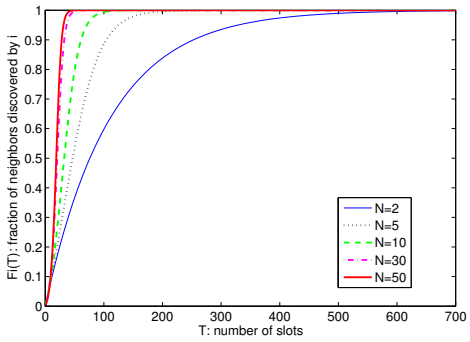
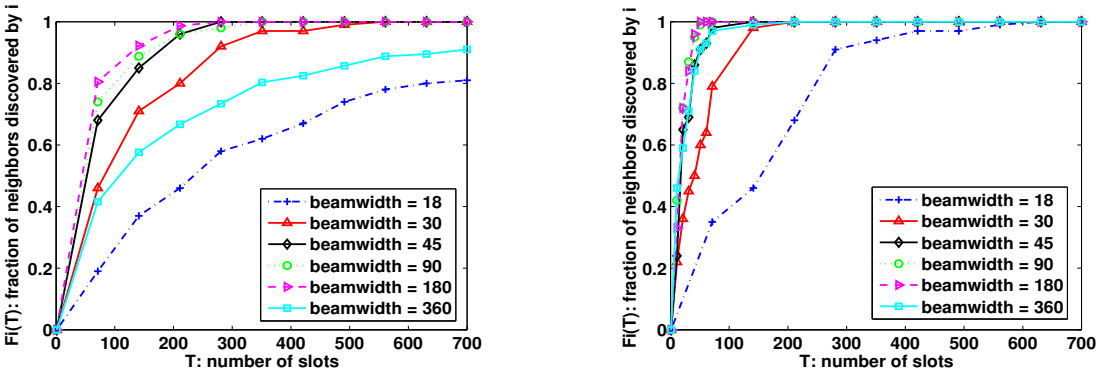


Figure 4.13: Impact of N on gossip-base discovery

beamwidth. We present simulation results[8] over six distinct beamwidths (18, 30, 45, 90, 180 and 360 degrees) for different network densities (5, 10, 15 nodes in addition to i). Note that a beamwidth of 360° corresponds to the omni-directional transmission-reception mode. We repeat the simulations twenty times for each configuration with both direct discovery and gossip-based discovery and compute the average values.

We observe a trade-off between "the chance of discovery" and "the level of interference" with changing beamwidth for both direct and gossip-based discovery. A wider (narrower) beamwidth

[8] Analytical results are similar.



(a) Direct discovery

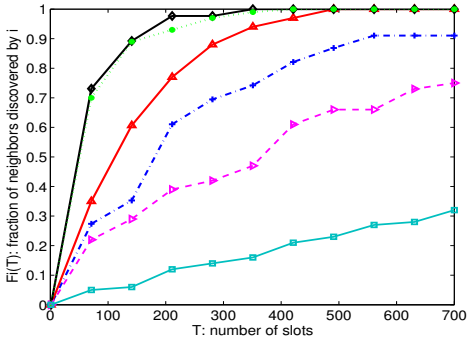
(b) Gossip-based discovery

Figure 4.14: Impact of beamwidth when neighbor number is 5

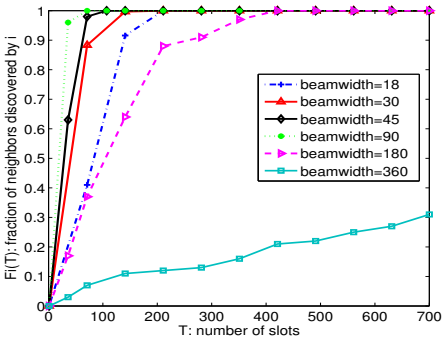
leads to a higher (lower) chance of discovering nodes, but results in a greater (smaller) chance of collisions. The effect of tuning the beamwidth might vary for different node densities. In other words, the best beamwidth to use depends on the node density.

We observe that (1) when i has 5 neighbors, as the beamwidth increases from 18° to 180°, the direct discovery efficiency improves, as shown in Fig. 4.14(a). With a beamwidth of 360°, the performance is poor. With gossip-based discovery, the beamwidth with the best performance is observed to be 180° (See Fig. 4.14(b)). The omni transmission-reception mode incurs a slightly lower efficiency than that of 180° beamwidth. (2) When the number of neighbors increases to 10, a beamwidth of 90° achieves almost the best performance for both direct (See Fig. 4.15(a)) and gossip-based discovery (Fig. 4.15(b)). Note that in both cases the performance of omni-transmission and reception mode (that is, 360°) is the worst due to a very high collision rate. (3) When the number of neighbors reaches 15, a smaller beamwidth of 45° yields the best performance for both direct and gossip-based discovery as shown in Fig. 4.16(a) and Fig. 4.16(b).

The observations above indicate that as the node density grows, the effects of interference

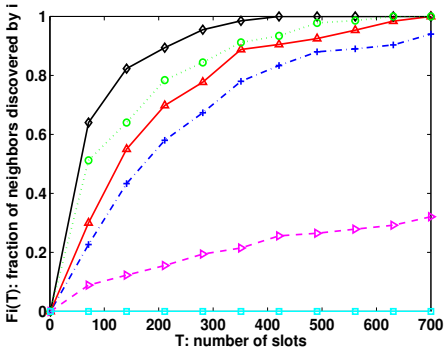


(a) Direct discovery

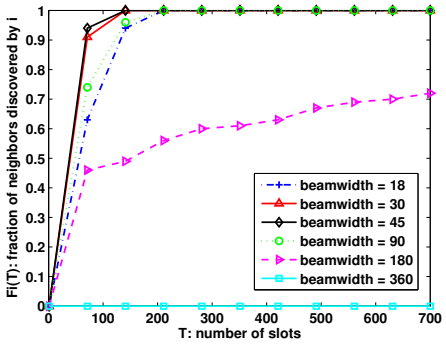


(b) Gossip-based discovery

Figure 4.15: Impact of beamwidth when neighbor number is 10



(a) Direct discovery



(b) Gossip-based discovery

Figure 4.16: Impact of beamwidth when neighbor number is 15

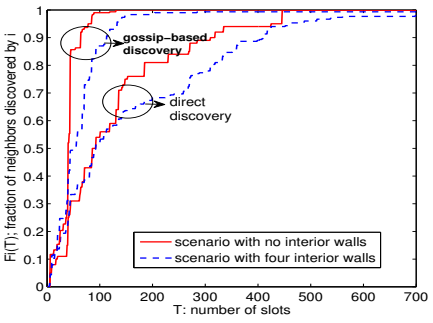


Figure 4.17: Impact of obstacles on discovery

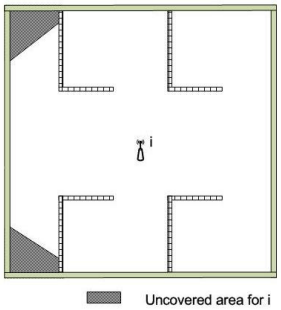


Figure 4.18: Uncovered area for node i

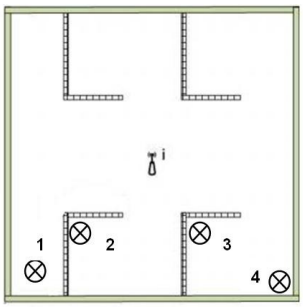


Figure 4.19: Spot 1, 2, 3 and 4

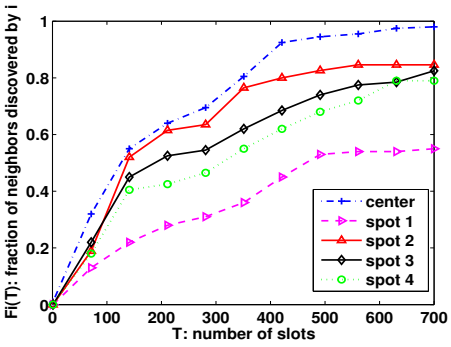


Figure 4.20: Impact of i's location on direct discovery

dominate "the chance of discovery". This causes the best beamwidth to be smaller as we increase density.

Impact of obstacles: The results thus far do not consider the presence of interior obstacles in the room. In order to study the impact of obstacles, we consider a series of scenarios with interior walls.

Case 1: Four interior walls; i is in the center. A 10 m by 10 m room where four interior walls are deployed as shown in Fig. 4.6(b). We perform 30 simulation runs for both direct and gossip-based discovery, and the results are compared with that with the empty room scenario in Fig. 4.17. The figure shows that the performance in terms of the fraction of neighbors discovered

(with both direct and gossip-based discovery) in the obstacle scenario is inferior to that in the empty room scenario. The reason for this can be explained as follows. Compared to the empty room scenario, with obstacles the number of communication links (via either direct or reflected beams) between any pair of nodes is likely to be reduced due to possible blockage by obstacles. For example, in the empty room, the number of paths between any pair of nodes is always five (including one direct beam and four reflected beams). However, in the room with obstacles, the number of paths varies from zero to five, depending on the environment around the pair of nodes. This results in a lower chance of discovering or being discovered. Note that as a by-product of the reduced number of communication links, the level of interference is also mitigated. The benefits due to the mitigated interference is however, observed to be insignificant when compared to the loss in connectivity.

We also observe that by the end of the simulation time (700 slots), with both direct and gossip-based discovery, not all the neighbors are discovered. Thus, with the considered schemes, the fraction of neighbor discovered is less than 1.0; in the empty room scenario, this fraction is 1.0. This is because in some scenarios, some of the neighbors do not have communication links with i due to the obstacles.

In particular, ray tracing reveals that in this scenario about 6 % of the area is uncovered (Fig. 4.18). If a node is in the uncovered area it is not a neighbor of node i. Here we point out that in our analysis we assumed that an indirectly reported neighbor is considered a neighbor. In this case, on average, 6 % of the reported neighbors are actually not the neighbors of i.

Case 2: Four interior walls; i's location varies. Thus far i's location had been fixed at the center of the room. Next we put i in some corners (spots 1, 2, 3 and 4 in Fig. 4.19) and see how its discovery efficiency changes. We refer to as corners, those spots that are surrounded by multiple obstacles and thus are isolated to the rest of the network to some extent. The simulation

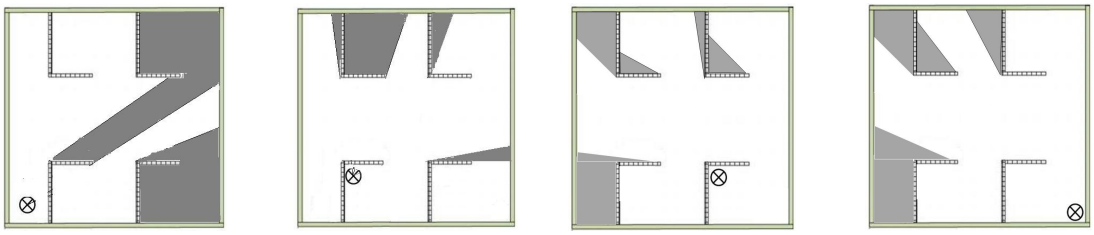


Figure 4.21: Uncovered area for spots 1, 2, 3 and 4 (from left to right)

results in Fig. 4.20 show that when i is at spots 2, 3 and 4, its discovery efficiency is brought down as compared to the case when i is located in the center, but not as much as when i is located at spot 1. This is because although spots 2, 3 and 4 are corners, i can still receive the transmission from most space of rest of the room (see the uncovered area in Fig. 4.21), due to reflection and penetration. However, spot 1 is an actual "dead" corner and if i is located there its reception is rather limited thus i is unable to discover other nodes even when they are physically close to it.

This implies that when deploying a 60 GHz indoor network, dead corners need to be carefully identified and one should avoid locating nodes in such corners. Reflections could however, render some of such seemingly "dead corners" useful in terms of coverage (spots 2, 3 and 4 in Fig. 4.19).

Case 3: Varying the number of interior walls and using a larger room size; i is at the center. We simulate cases where there are 2 and 6 interior walls, and (2) when the room size is enlarged to $20m$ by $20m$ with the number of interior walls fixed at 4 (see Fig. 4.22). The results shows that when other configuration settings remain the same, with the deployment of fewer walls or with a larger room size, the discovery performance is better. However the improvements are not dramatic. This is because signal reflections in 60 GHz band already renders neighbor discovery effective and cannot be improved much further.

In summary, the impact of obstacles on neighbor discovery efficiency in 60 GHz indoor

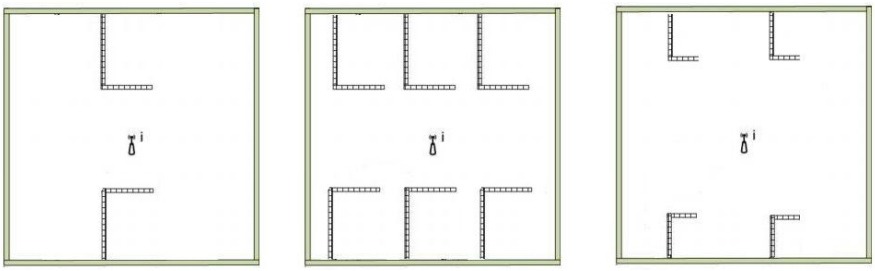


Figure 4.22: (left) 2 interior walls (middle) 6 interior walls (right) larger room

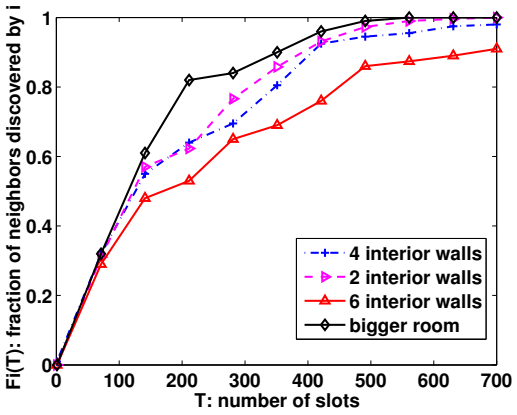


Figure 4.23: Impact of the number of obstacles and room size on discovery

networks is complicated in the sense that obstacles can block signals to eliminate links, while at the same time help establish links via reflected beams. The impact of obstacles depends on the deployment scenario and the properties of the obstacles, as well as the locations of nodes.

4.6 Conclusions

In this paper, we consider the problem of directional neighbor discovery in 60 GHz wireless networks. The existence of possible communication links due to reflected beams in the 60 GHz band makes neighbor discovery more complicated than if only direct line of sight paths exist (as considered

in previous work). We analytically model two neighbor discovery methods namely, direct discovery and gossip-based discovery. Our analysis accounts for reflections that exist in an indoor 60 GHz network. We validate our analytical models via simulations that incorporate a measurement based 60 GHz channel model. We examine the impact of various factors on neighbor discovery; our study reveals a number of interesting trade-offs that have to be considered when choosing system parameters.

Chapter 5

Covert Communication on Public Photo-Sharing Sites

The use of steganography to embed secret messages in images shared on photo-sharing sites has recently received some attention [58]. However, the establishment of a covert channel using uploaded photos or images is not straightforward. First, photo sharing sites often perform image processing on the uploaded images, thus destroying the embedded messages. Second, prior work assumes the existence of a secret out-of-band channel, using which the communicating users exchange meta data or secret keys a priori to make the covert channel viable.

In this chapter, we address both of these concerns in conducting covert communication via messages embedded in images uploaded on photo-sharing sites. In doing so, we make the following key contributions: (a) We provide an in-depth measurement study of the feasibility of hiding data on four popular photo sharing sites. (b) We identify the challenges in covertly communicating on these sites (e.g., Facebook) and propose ways of intelligently embedding secret messages while ensuring that the integrity of such messages is preserved. With the use of state of the art steganalysis

techniques, we demonstrate that our approach does not result in higher detection likelihood than commonly used stego techniques. (c) We present and evaluate an approach for bootstrapping the covert communication without an out-of-band channel, i.e., by exchanging keys via uploaded images.

5.1 Introduction

The explosion of photo-sharing on publicly available websites opens the door for secret communications on the Internet; users can use steganography [59] to embed information in the photos that they upload. As early as in 2001, US newspapers reported that terrorists use steganography to communicate in secret via the Internet [60]. The idea of hiding messages in user-generated content on photo sharing sites has recently received increased attention; as an example, Burnett et al. [58] suggest that the approach can be used to "chip away" at censorship firewalls.

However, while the idea of using steganography to embed secret information in shared content on photo-sharing sites is conceptually attractive, it is not straight forward. There are several challenges that exist in creating a viable covert channel of this type.

First, it is known that photo sharing sites often process uploaded images. The images could be either recompressed or processed to change features such as resolution or brightness [61]. While some of the processing functions are clearly specified on the photo-sharing sites [62, 63] (e.g., any photo exceeding a pre-specified size limit will be re-sized), not all such functions are publicly known. These (possibly unknown) processing functions often interfere with the use of steganography.

Second, steganography does not offer perfect secrecy [64]. Censors can try to read the embedded message by applying a list of extraction algorithms on every image. Thus, one will have to encrypt the secret information embedded in the shared photographs, to prevent exposure in the rare cases of interception. Encryption requires the establishment of secret keys between the communicating entities. Often, prior work assumes the existence of an out-of-band channel via which

such keys are established. However, in many cases (e.g., where people are trying to hide information from government-controlled censors), the creation of such an out-of-band channel may be difficult (e.g., e-mail or voice calls may be monitored).

Our goal in this work is to address the above challenges and provide a framework for covert communication on public photo-sharing sites. Towards accomplishing our goal, we take several steps.

First, to understand how secretly embedded messages are affected by processing done on photo-sharing sites, we perform an in-depth measurement study of photos uploaded on four popular sharing sites—Google+, Facebook, Twitter, and Flickr. Our measurement study involves the uploading and downloading of a large set of diverse images on these sites, and examining how they are affected. We consider both photos wherein secret information is embedded and photos without any such embedding. We observe that, while some sites preserve the original images and thus the integrity of the hidden messages, other sites perform various processing functions that result in a failure in the extraction of secret messages. Our study reveals several insights on the processing performed on the different sites, and provides an understanding of why secret content is affected. For example, Facebook removes the content in the EXIF field of images, a portion of the image where some stego tools hide messages.

Second, based on the understanding obtained above, we propose simple changes to the steganographic encoding process to ensure that the secret messages survive the image processing functions. We evaluate our approach by applying two state-of-the-art steganalysis tools. We observe that, for a fixed amount of secret data, the detection rate with our approach is similar (or even lower in some cases) as compared to prior common approaches of secret embedding.

Finally, as discussed above, encrypting the secretly embedded messages is a must. Therefore, to enable recipients of the shared photo to extract the raw data, a key exchange between the sender and recipients is essential. Towards this, we propose an approach for bootstrapping the covert

communication without any out-of-band channel (unlike what is assumed in prior work [58]). Our bootstrapping phase uses the very same channel, i.e., uploaded images, to exchange keys.

In summary, our contributions are as follows:

- We perform an extensive measurement study that provides an understanding of the feasibility of hiding data with image steganography on four popular photo-sharing sites, viz., Google+, Facebook, Twitter and Flickr. Our study reveals the impact of the image processing done on these photo sharing sites on secretly embedded content.
- We propose a new simple way of embedding secret information in photos to preserve the integrity of secret messages on popular photo-sharing sites. With the aid of steganalysis tools, we demonstrate that our approach does not result in higher detection likelihood as compared to other commonly used embedding techniques.
- We propose an in-band approach for bootstrapping secret conversations. Specifically, we use the same channel (i.e., secret embedding on uploaded images) as that used for covert communications to exchange keys and other essential metadata.

5.2 Background and Related Work

In this section, we first present relevant background information on image steganography and on online photo-sharing services. Subsequently, we discuss previous related work on using steganography to conduct covert communication on such sites.

5.2.1 JPEG Image Steganography

Steganography can be used to embed secret information in various types of files such as images, audio, video, or text files. The content in which the secret data is hidden is called the *cover*.

Image steganography is arguably the most popular among these, and JPEG (Joint Photographic Experts Group) [65] is one of the most popular image formats today. In many photo-sharing sites, even if the uploaded images are of other formats, they are converted to the JPEG format prior to storage. Steganographic techniques are typically developed to exploit the structure of JPEG format, and hence, we restrict our focus to this format in our work.

A JPEG image consists of a sequence of segments, each beginning with a marker [66]. A unique byte followed by a byte indicating what kind of marker it is. Some common JPEG markers include Start of Image marker, Application Use marker, Comment marker and so on. The JPEG encoding process typically consists of the following steps [67]: (1) transformation of the color space, (2) downsampling of the chrominance component, (3) application of the Discrete Cosine Transform (DCT), (4) quantization of the DCT coefficients, and (5) entropy encoding of the quantized coefficients using a lossless compression algorithm. From among these steps, (3) and (4) are where image compression is achieved because the actual DCT values are represented less accurately in fewer bits. The decoding process reverses the above steps, except (4) because quantization is irreversible. JPEG uses a lossy compression method for images. The compression ratio is adjusted at the quantization step of the encoding process; quantization provides a tradeoff between storage size and image quality. The final encoding step can be of two types. A "baseline" encoding is performed on a single scan of the image. With "progressive" encoding, data is compressed via multiple passes with progressively higher details. The main advantage of progressive encoding is that, when the Internet connection is slower, the user will be able to at least see a blurred version of the image to get a sense of what is loading. A baseline JPEG image can be displayed only after all of the image data has been downloaded and decoded.

There are many approaches and software available for JPEG-based steganography [68, 69, 70]. The naive ones are to hide the data after the end of image marker or in the comment field.

A lot of approaches hide data in the least significant bit (LSB)s of the DCT coefficients. JPEG steganographic algorithms can be categorized into several categories as follows.

Structure-based steganography simply exploits certain, usually optional, markers in the JPEG format without tampering the image itself. One way is to embed the secret message after the *End of Image (EOI)* marker in the image [71]; since the JPEG decoding program ignores anything that appears after the EOI marker, the image quality is not affected by this process. The stego decoder tool can examine the content after the EOI and extract the message. The secret message can also be appended to the image's Exchangeable Image File (EXIF) [72] data, which is used by digital camera manufacturers to store information such as the make and model of the camera, or the time when the picture was taken. The EXIF information is located within the *Application Use* marker. Similarly the message can be hidden within the *Comment* marker. Note that EXIF and Comment fields are for the purposes of storing optional metadata. When the images with the stego-encoding are displayed with photo-viewing programs, there is no discernable difference as compared to the original images since only metadata is modified.

Spatial domain techniques typically modify the Least Significant Bit (LSB) of the pixel values to embed the secret information [59]. These techniques exploit the fact that human perception is not sensitive to subtle changes in pixels. Information hiding in pixel information is however not reliable, especially when being used with lossy image compression schemes such as JPEG; these pixel values are likely to be modified in the downsampling or quantization phases. Steghide [73] is an available stego implementation in this category.

Frequency domain based methods replace the LSBs in the quantized discrete cosine transform (DCT) coefficients [74]. Since every DCT coefficient is a function of an image block of size 8×8 pixels, altering any single coefficient would affect all the 64 pixels in the decoding process. As the operation is in the frequency domain, instead of the spatial domain, there will be no noticeable

change in the cover image since these coefficients are handled with care by typical steganography tools. The JSteg algorithm [68] was among the first algorithms in this category. OutGuess [75] uses a pseudo-random number generator to select DCT coefficients. The F5 algorithm [76] embeds secrets only in non-zero DCT coefficients by decreasing the absolute value of the coefficient by 1.

Distortion-resistant algorithms are more robust to image processing. JPEG compression will cause bit errors for schemes relying on LSBs of either pixels or DCT coefficients. Image resizing and cropping will also result in bit losses or errors. Transformations in other domains (like with the Discrete Wavelet Transform or DWT), or the use of redundancy and/or masking techniques can bring down the bit error rate. Using erasure and error correcting codes provides protection against distortion caused due to JPEG compression, additive noise, and so on [77]. The YASS scheme [70] uses a redundancy parameter to control the number of times an information bit is repeated inside an image. This provides a significant advantage over most other stego methods available in the literature. Note that error-free recovery of the hidden message is not guaranteed.

5.2.2 Online Photo Sharing

Online social networks (OSNs) and other public photo-sharing sites allow users to post images and view those of others. In our work, we focus on four sites: Facebook, Twitter, Flickr, and Google+. Photo sharing in Twitter and Google+ is powered by Photobucket [78] and Picasa [79], respectively.

It is known to some extent that OSNs may process user uploaded images. The Google+ tutorial [80] says: "When you upload photos to the web, if you choose best for web sharing, photos larger than 2048 pixels by 2048 pixels will be automatically resized to that size. If you choose to upload images at their original size, the image will remain at the original resolution." The Twitter help center has the following regarding image processing [81]: "We'll scale the image for you to fit

into the displays on your Twitter.com timeline. We remove the EXIF data upon upload. It is not available to any consumers of your image." Flickr Help/FAQ/Photos section [62] says: "As a free account holder, no one (including you) has access to your original files, but if you ever upgrade to Pro, your high-resolution originals will be stored ... As you publish photos, they're compressed and resized by Flickr (if necessary)." The Facebook engineering blog [82] states: "Facebook is increasing the size of the photos stored from 720 pixels to 2048 pixels on the largest edge, for an 8 times increase overall. For better quality photos, check the High Quality box when you create an album".

This information clearly suggests that photo sharing sites modify user uploaded photos. However, due to a lack of official documentation, the specifications of image processing performed by these sites are unknown.

5.2.3 The Use of Steganography on Images Shared Online

It has been known for long that messages can be hidden in images posted to Internet [60, 83]. OSNs are the perfect setting for doing so. Billions of images are uploaded every day and thus, there is an abundance of "covers" to examine, if a censor seeks to do so. It is difficult or even not realistic to be able to monitor every piece of user-generated content. In addition, private control can allow a user to limit who can view her images and this makes steganalysis even more difficult. Provos et al. [84] analyzed two million images downloaded from eBay for the presence of hidden images. but not a single hidden message is found. Nevertheless, the use of social media and steganography to build a covert communication channel is recognized as promising in [85]. Zeljko et al. [86] implement SecretTwit, a Twitter client that hides secrets in tweets and images. An anti-censorship system proposed in [58] is built on the idea of two parties exchanging messages in images posted to Flickr. The idea of bots performing covert communication using images on Facebook, to form a botnet is suggested in [87]. Internet security specialists predict that social networking

botnets will become one of the biggest challenges [88].

Despite this attention, the feasibility of covert communication on OSNs or photo-sharing sites has not been fully explored in prior work. From their first hand experience, some Internet users have already realized that certain steganography techniques cannot be used on Facebook [89]. However, the reasons for this are not well understood given that these sites do not make public how they process uploaded images. So far there is no thorough investigation into the exact processing that public photo-sharing sites perform, and how it impacts different information hiding techniques. To the best of our knowledge, our work is the first to fill this gap. Moreover, we propose a simple yet practical method to ensure that secret messages can be indeed communicated in these sites.

5.3 Feasibility of Secret Embedding on Online Sites

In this section, we describe the results of our in-depth measurement study towards understanding the feasibility of embedding secrets in images uploaded on four online photo sharing sites.

5.3.1 Hiding Information on Different Online Sites

We use multiple representative steganography tools from each category described in Section 5.2 to hide messages in images. Then, we upload the images on to various sites and then attempt to retrieve the hidden messages from the downloaded images.

We use 100 images from a database made available by CMU [1]. These images are diverse and include categories such as sightseeing and social gatherings; they represent the kinds of photos that people are likely to share online.

Steganography tools: The steganography tools that we use are listed in Table 5.1.

[1] http://www.cs.cmu.edu/ ∼cil/v-images.html

These tools are chosen as they are widely used in the steganography community and publicly accessible. GhostHost simply appends the hidden message after the End-of-Image marker. Steghide, F5, and Outguess are probably the most used tools for benchmarking in academia. Yet Another Steganographic Scheme (YASS) is distortion-resistant in that it embeds data at randomized locations within an image and repeats an information bit multiple times inside the image. The redundancy (the number of times that a bit is repeated) is a tunable parameter. These tools have available implementations, though not all of them provide open source code.

Table 5.2 captures whether or not hidden messages can be retrieved using each tool, on each considered site.

Definition of terms: For ease of discussion, we define the following terms. (a) *Success* implies that the extracted message is equivalent to the hidden message. (b) *Failure* means that the retrieval effort does not yield any output. For example, upon failure the output of OutGuess decoding is: "extracting usable bits: 0 bits; Steg retrieve len: 27605; Extracted datalen is too long: 27605 > 0". Similarly, a failure causes the output of Steghide to be: "extracting data...could not extract any data". In the case of F5, the decoding process never completes or the extracted data is simply some random bit string. A failure is experienced even if only a part of the message is corrupted; a checksum may fail or metadata indicating the length of the message or the embedding locations could yield mismatches. In all of such cases, one cannot retrieve the original hidden message. (c) *Conditional success* only applies to YASS which uses redundancy to control the decoding BER. The decoding BER (the ratio of message bits in error to the total number of message bits in an image) of YASS depends on the redundancy parameter in use. We experiment with different redundancy parameters and the results are presented in Table 5.3. We observe that when the redundancy parameter is larger than 10, the BERs significantly decrease and approach 0. Similar results are reported in [87]; specifically, the work states that the BERs with YASS are image and redundancy dependent, and a

Tool	Details on approach
GhostHost [90]	Embedding after the EOI marker
Steghide [73]	Changing Pixel values
OutGuess [75]	Changing DCT coefficients (pseudo-random)
F5 [76]	Changing DCT coefficients (non-zero)
YASS [70]	Changing DCT coefficients (error correcting)

Table 5.1: JPEG steganography tools

Tool	Facebook	Twitter	Flickr	Google+
GhostHost	×	×	×	√
Steghide	×	√	×	√
OutGuess	×	√	×	√
F5	×	√	×	√
YASS	√*	√	√*	√

Table 5.2: Evaluation of steganography tools on photo sharing sites (× = Failure; √ = Success; √* = Conditional Success)

zero BER is likely with a high redundancy parameter (>12). We observe that the BERs experienced for a given (fixed) redundancy are similar on Facebook and Flickr.

Summary of the results: At first glance, we see that most of the tools (except YASS) fail on Facebook and Flickr but succeed on Google+ and Twitter. Google+ is the most generous platform and accommodates all the steganography tools. Twitter is the next best; GhostHost fails on Twitter but the other tools are able to successfully exchange hidden content. Facebook and Flickr show the least compatibility with steganography in that all the tools except YASS fail in successfully exchanging secret content.

To understand the above results, we next perform an in-depth examination of the processing that each site performs. Specifically, we examine the changes in the uploaded images at the bitstream level. We give explanations why certain steganography tools fail on certain OSNs.

Redundancy	2	6	10	14	18
Facebook	0.3442	0.1498	0.0411	0.0000	0.0000
Flickr	0.3491	0.1592	0.0456	0.0000	0.0000

Table 5.3: Average BER with YASS decoding with different redundancy levels

5.3.2 Impact of Processing on Hidden Messages

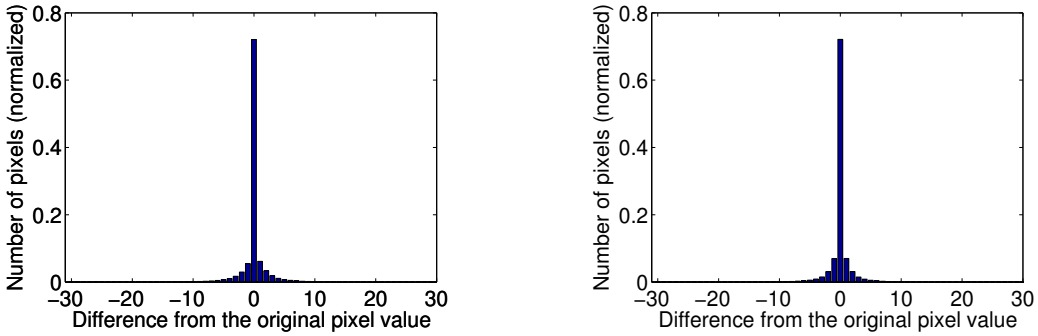
Google+: ***Image integrity is preserved.*** We carefully examine our sample data set of 100 photos. We observe that Google+ preserves the original just as they are, when their size are within 2048 pixels by 2048 pixels. Since the integrity of an image is preserved as long as the image adheres to the permitted resolution, any steganographic tool will work on Google+. This is reflected in the column for Google+ in Table 5.2.

Twitter: ***Metadata fields are cleaned up.*** Recall that inside the JPEG image structure, there are fields for storing metadata. We find that some of these fields, such as the COM (comment) and APP (application specific) fields, are rewritten by Twitter with its own data. In addition, anything that appears after the EOI (End-of-Image) marker is stripped off. The consequence is that tools that exploit metadata markers for embedding messages (e.g., GhostHost) will not work.

Except for this 'clean up' of the metadata fields, Twitter preserves the image integrity as long as the image size is no larger than 1024 pixels by 768 pixels. Exceeding this limit will cause a loss of integrity. Hence, as seen in Table 5.2, Twitter accommodates most of the steganographic tools.

Facebook: Similar to Twitter, Facebook removes the content in the Comment field and whatever appears after the EOI marker. However, in addition, Facebook applies the following processing functions.

Changes in pixel values. Typically each pixel in an image is represented by a byte,



(a) First dimension

(b) Second dimension

Figure 5.1: Distribution of the differences in the pixel values in two color dimensions.

with a value that lies in the range [0,255] in each of the RGB color dimensions. We find that, for a fraction of pixels, the values are changed on Facebook after the image is uploaded. Fig. 5.1 shows the distribution (probability density function) across the examined set of images of the deviation of the pixel values from the original values for two color dimensions. The distribution for the third dimension is similar and not shown. We observe that, while the majority of pixels ($> 60\%$) remain unchanged, the rest are modified. The maximum deviation can be up to 30. The exact distribution of the deviation differs from image to image, but in general seems to follow the Gaussian distribution.

The changes in pixel values can be due to JPEG's lossy compression method adopted by Facebook and/or other manipulations (discussed later). In either case, the steganography tools that rely on embedding the messages into the pixel values (e.g., StegHide) do not work as a result. Images larger than 2048 pixels in either the length or width dimensions get resized on Facebook. Resizing reduces the number of pixels and causes the pixels to shift from their original locations or even to be lost, thus destroying the integrity of embedded messages.

Changes in compression ratio. From the quantization tables (one for luminance and

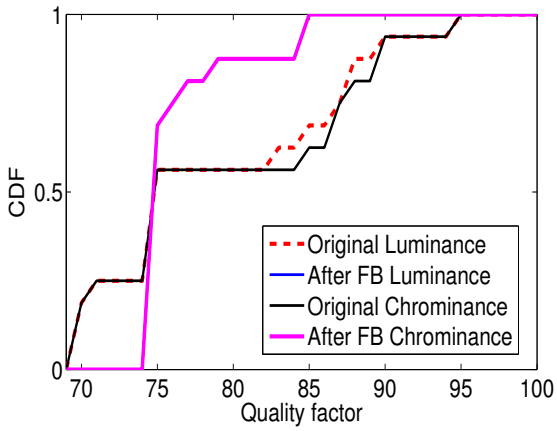


Figure 5.2: CDF of quality factor before and after upload.

one for chrominance) associated with the original images and the downloaded ones, we note that Facebook adjusts the compression ratios for many images. We compare with the tables published in the International JPEG Group standard (libjpeg [91]) and get the approximate quality factors before and after upload for each image, with a JPEG decoding utility JPEGsnoop [92]. Note that, though the quality factor is an integer in [0, 100], the calculated values are not necessarily so due to the fact that custom tables are used. We round each calculated number to the nearest integer. The results are shown in Fig. 5.2. We note that, while the quality factors for both the luminance and chrominance of the original images span 70 to 95, Facebook adjusts them to be at 75 in about 70% of all cases (this matches the observation in [87]). In addition, the same factor is used for both luminance and chrominance for a given image. Fig. 5.3 shows the cumulative distribution function (CDF) of the quality factor change for all images. We observe that, for about half of the images, Facebook uses a higher compression ratio (resulting in lower quality factors) than the original.

When a lower quality factor is used, it is likely that the pixel values will change due to compression. Interestingly, we observe that even for those images whose quality factors are

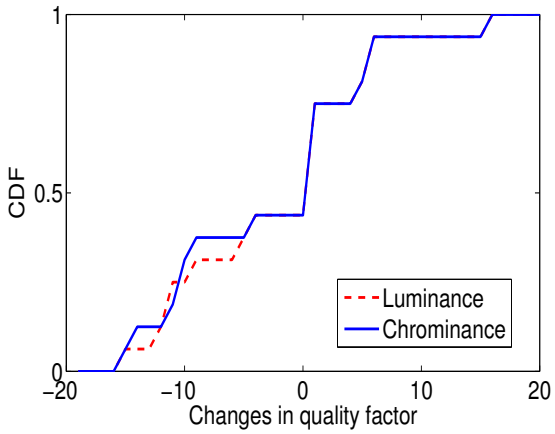


Figure 5.3: CDF of quality factor change before and after upload.

unchanged, the pixel values are modified. We conjecture that it may be the case that Facebook applies other image processing besides just compression. There is no official statement about what the exact processing is, but some online discussions suggest that a low pass filter is used [2].

Changes in DCT coefficients. We access the DCT coefficients using a JPEG dump utility [93] based on the libjpeg library. Fig. 5.4 shows the number of DCT coefficients having a specific normalized change as compared to the original value. We notice that, while the majority of coefficients remain the same (with no difference from the original value), about 20% are decreased or increased by one. It seems that the change is small (only one) and appears to occur in the LSBs; as one might recall, the least significant bits are exactly where the bits corresponding to the hidden message reside. Fig. 5.5 plots the changes in coefficients sequentially in an arbitrarily chosen image from our data set. We see that the changes are evenly distributed all over the image.

A careful inspection suggests that there are two potential reasons for the above changes in the DCT coefficients. First, Facebook uses a different set of quantization tables from that in the

[2]For example, see http://photo.stackexchange.com/questions/352/what-are-the-optimal-jpeg-settings-for-facebook-photos.

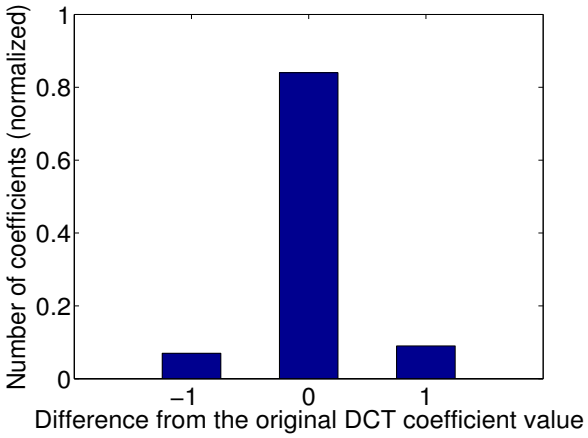


Figure 5.4: Change distribution of DCT coefficients

standard JPEG libraries. Second, it also changes the pixel values themselves, thereby compounding the effect. We were unsuccessful in preserving the DCT coefficients even when uploading images with the same quality factors seen in the images downloaded from Facebook. It is also possible that Facebook applies a watermark to the uploaded images; however, we were unable to verify this.

The above findings indicate that, embedding messages in the DCT coefficients (e.g., with F5 and Outguess) runs the risk of extraction failures when the images are uploaded on to Facebook and subsequently downloaded. Tools with error correcting capabilities (e.g., YASS) can lower the decoding BER and even eliminate errors in some cases. However, the BER depends on the message itself and where it is encoded within the image. Thus, it may not be possible to extract the message in all cases.

Finally, we wish to point out that the default DCT encoding in JPEG images is done with the baseline encoding; however, Facebook uses progressive encoding. However, this does not affect the values of DCT coefficients. The encoding scheme only determines if a specified band (i.e., a lower or higher part of the frequency spectrum) is encoded first, and if the most significant (and the

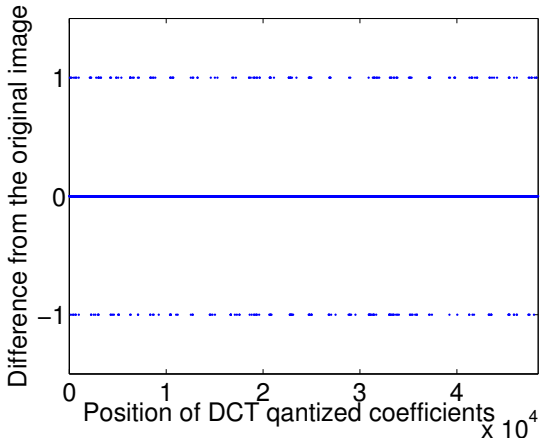


Figure 5.5: Change in each DCT coefficient in an image

number of) bits of the coefficients are encoded first [94]. This results in changes in the way that the coefficients are represented in the JPEG format, but not in their values. We have verified this by changing the encoding scheme on the original JPEG images.

Flickr: Flickr cleans up the metadata fields as other sites do. Unlike Facebook, it uses a constant quality ratio of 96 for both the luminance and the chrominance while re-compressing images. Our experiments show changes in the pixel and DCT coefficients in Flickr; however, for the same set of images, the distributions are different from that with Facebook.

Summary: (a) Some sites (Google+, Twitter) preserve the integrity of images to a large extent. Common steganography tools can be used directly on images uploaded on these sites. (b) Other sites (Facebook, Flickr) process uploaded images, thus making it difficult to use these tools directly.

5.4 Reliable Steganography for Facebook and Flickr

While Google+ and Twitter do provide forums for exchange of hidden content, Facebook is today the most popular OSN, e.g., as of January 2012, Facebook had over 800 million users as compared to Google+'s userbase of just 90 million [95]. Similarly, today, Flickr is considered as the top photo-sharing site [96]. Thus, we ask the question: can we provide a way wherein users can secretly communicate on Facebook and Flickr, in spite of the processing that is performed on these sites. Note that combating the resizing effect is not considered here because the case can be simply avoided by using images within the allowed size.

Towards this, we first considered a simple approach wherein we first upload an image onto the site, download it, and upload it back again. We repeat this process with the hope that after a few iterations, the images are not further manipulated. Even if one were to ignore the fact that repeatedly uploading and downloading the same image could potentially alert censors, we find that this approach does not help in preserving hidden messages. Therefore, we propose a new embedding scheme that preserves hidden messages with high probability, while simultaneously ensuring that the detection probabilities with steganalysis tools remain similar to that with common steganographic embedding.

5.4.1 Repetitive Uploads

When an image is uploaded for the first time on to Facebook or Flickr, it is exposed to fundamental changes like recompression as discussed earlier. We now ask the question: "What will happen if this image was downloaded, and then, uploaded again?", the basis being that the processing rendered upon the first upload may have caused the image to now conform to the photo-sharing site's requirements. In other words, we ask whether we can expect that the image undergoes no further processing. Towards answering this question, we experiment by performing such *repetitive*

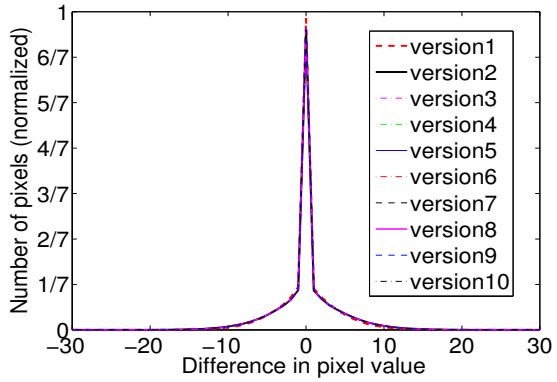


Figure 5.6: Differences in pixel values compared with original for a sample image

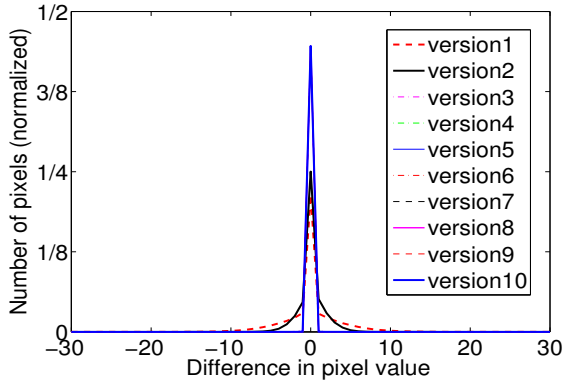
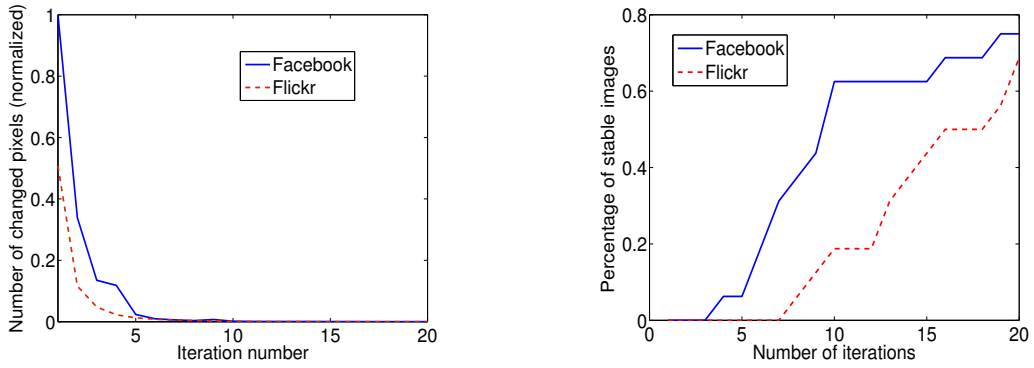


Figure 5.7: Differences in pixel values compared with previous version for a sample image

uploads of our images on Facebook and Flickr. We denote the original set of images as version '1'. After the first iteration of upload and download, we get a set of images which are referred to as version '2' images. Images of version x are the downloads of images of version $(x-1)$.

Changes in pixels and DCT coefficients converge with repeated uploads: When comparing each version to the original image, similar pixel change distributions are seen for all later versions (Fig. 5.6). However, when comparing each version to its previous version, the distribution has smaller variance, which approaches zero as the version numbers increase (Fig. 5.7). In other



(a) Number of changed pixels as a function of iterations

(b) Percentage of stable images

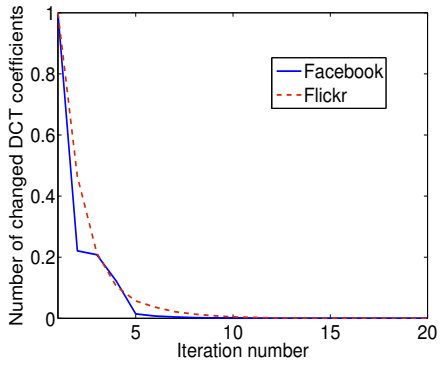
Figure 5.8: Pixel changes converge after repetitive uploads

words, each version incurs fewer changes from its previous version (as compared to changes from the original). Similar observations also hold with Flickr. Interestingly, if the same image is uploaded the download versions are identical, which means the changes to the same image is deterministic. We call a version of an image (say version x)*stable*, if there are no changes as compared to the previous version (version $x-1$).

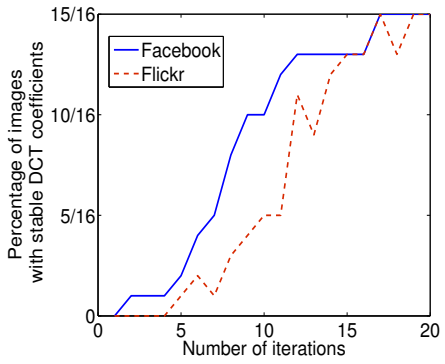
We experiment on a set of images, performing 20 iterations of repeated uploads. We find that after a few iterations, the number of changes significantly reduces; the convergence seems faster in Flickr than in Facebook (Fig. 5.8(a)). With Facebook, more than half of the examined images are rendered stable after about ten iterations; over 70% of images are stable by version '20' (Fig. 5.8(b)). Interestingly, while the convergence seems faster in Flickr, stable images are produced more slowly.

The trend seen in changes to DCT coefficients with repeated uploads is similar to that of pixel changes (Fig. 5.9).

In the above discussion, we used photographs without any form of steganographic embedding. Our goal was to primarily understand if the processing done at the photo-sharing sites



(a) Number of changes in DCT coefficients



(b) Percentage of images with unchanged DCT coefficients

Figure 5.9: DCT coefficient changes converge

stabilized after a certain number of iterations, and we were encouraged by the fact that it did. Next, we picked the stable images to embed hidden messages.

Embedding in stable images does not help: Unfortunately, when a message is embedded in a stable image, its inherent statistical properties change and thus the image is treated as new by Facebook. When these images are uploaded onto either Facebook or Flickr, we find that further processing is done and the hidden message is destroyed. Since the secret cannot be preserved during the repeated uploads/downloads, the approach of using stable images unfortunately does not work.

5.4.2 A Different Embedding Approach

Upon further examination, we realize that the failures experienced with common steganography tools is because the messages are embedded in the LSBs of either the DCT coefficients or the pixel values (which are more prone to processing changes). Thus, the only way of preserving secret content embedded in the LSBs is to use robust error correction codes (ECC) as with YASS. However,

since these bits are often subject to processing changes, the overhead incurred will be high, thereby reducing the secret carrying capacity (shown later). Furthermore, as we also show later, the use of high degrees of redundancy is one factor that increases the chances of dectecting the presence of a secret message via steganalysis.

Based on the above, we ask the question: "Are there locations within an image that remain relatively unaffected after processing on Facebook or Flickr?" If there are such locations, we could then embed secrets in such locations, possibly with much weaker ECC.

Recall that the maximum change in the pixel values is about 30 for almost all images (Fig. 5.1), and the maximum change observed in the DCT coefficients is 1 (Fig. 5.5). Intuitively this suggests that embedding the message in the higher significant bits of a DCT coefficient, as opposed to embedding it using the LSB, could protect it during the processing operations on the photo-sharing site. However, this approach poses a potential pitfall. To evade the detection using steganalysis of a message hidden within an image, there is an inherent tradeoff between preserving integrity by changing higher-order values and keeping the detection likelihood low.

Let us consider a simple example with a given color image, wherein a pixel is represented by 3 bytes, one each for the RGB dimensions. If two bytes only differ in the LSB, the represented colors are virtually indistinguishable to the human eye. A variation at the start of each byte results in more drastic noticeable color differences. In addition, to detect hidden messages, a steganalysis tool could examine the colors of adjacent "pixel pairs" and determine how close they are to each other. It could examine the rare occurrences of abrupt color changes within the image and flag the image if such occurrences are observed; changing the higher order bits cause more drastic color changes resulting in easier detection.

In fact, several sophisticated steganalysis tools have been developed to combat steganographic embedding; such tools are able to detect the embedding done by common steganographic

algorithms with differing fidelities. As an example, one modern image steganography detection tool [97] adopts a machine learning tool that is trained to distinguish between the original and "stego-ed" images. The algorithm is sensitive to steganographic embedding changes, but is insensitive to the original image content. It also captures many dependencies among individual DCT coefficients; there is an increased likelihood that at least some of these dependencies will be disturbed by embedding. Because of this, common steganographic embedding tools typically hide data using the LSBs. Needless to say, there seems to be a race between the development of new steganographic embedding solutions, and steganalysis tools to combat such approaches.

The above implies that in pursuit of achieving the integrity of message, embedding at higher significant bits may not be a bad idea if the chances of being detected is not significantly higher that embedding at LSBs. We will discuss and evaluate such possibility in the following.

Our approach: Given this, we propose the embedding of secret information in the 2^{nd} least significant bit (2-LSB) in the DCT coefficients; this, in our view, could provide (and as shown later, does provide) the best tradeoff between detection evasion and preserving the integrity of the hidden message when images are shared on Facebook and Flickr [3]. We modify the open source stego tool F5 [76], which embeds the secret message bits in the LSBs of *non-zero* DCT coefficients. We embed the message bits in the 2-LSB of these coefficients instead. In typical images, with both a length and width of about 1000 pixels, there are about 10,000 non-zero coefficients. The number of these usable coefficients is called "image capacity". As an example, when the image capacity is 10,000 bits, by using 10% of the capacity to embed secret information, we can embed 125 bytes or characters. This translates to approximately 25 words (based on statistics that there are about 4.5 characters per word on average [98]). In Fig. 5.10, we show images with and without secret embedding; we see that visually it is hard to tell the difference between the original image and

[3]We do not pursue embedding in 3-LSB and above because we find that 2-LSB suffices to preserve messages while keeping detection likelihood low.

Method	**BER**	**ECC overhead**	**Detection likelihood (ensemble classifier)**	**Detection likelihood (StegAlyzerAS)**
LSB	0.15239	0.0	0.44	0.69
2-LSB	0.00968	0.0	0.50	0.63
LSB+ECC *[15,13]*	0.09375	0.1333	0.45	0.69
LSB+ECC *[15,11]*	0.01125	0.2667	0.48	0.69
LSB+ECC *[7,3]*	0.0	0.5714	0.53	0.72
2LSB+ECC *[15,13]*	0.0	0.1333	0.51	0.63

Table 5.4: Comparison between 2-LSB and LSB stego methods

stego-ed images with either the LSB or the 2-LSB methods, when a reasonable amount of data is embedded (10% of the image capacity).

We upload two sets of stego-ed images, using the 2-LSB and LSB methods (the message length is 10% of the image capacity) respectively, to Facebook and then download the images. We calculate the bit error rates (BER) when the messages are retrieved.

BER behaviors: From Table 5.4 (see Column 1, Row 2), we see that embedding information in the 2-LSB of the DCT coefficients encounters *much fewer* bit errors ($\approx 1\%$) as compared to using the LSB ($\approx 15\%$). This is because, when the DCT coefficients are changed by 1 (recall Fig. **??**), the LSBs are altered and so are the embedded data bits (if the embedding is done in the LSB). Note that with a unit change in the LSB, the 2-LSB may be sometimes altered due to a carrier overflow or a borrowing from the LSB. However, though using the 2-LSB does not provide complete error-free decoding, it comes really close.

Adding ECC: Next, we seek to eliminate errors by applying ECCs of varying capabilities to the hidden message. Considering the small BER induced by the 2-LSB method, we expect the overhead to be minor. We experiment with the Reed-Solomon code [99] with different error-

Figure 5.10: Visual comparison of original image (left), stego-ed with LSB (middle) and with 2-LSB (right) (10% image capacity used)

correcting abilities. A Reed-Solomon code is a linear block code and is usually represented in the form [n, k]; n is the length of the code word and k is the length of the message. The redundancy is (n - k) and, in general, up to $(n - k)/2$ errors can be corrected.

We experiment with three settings—with *[15,13]*, *[15,11]*, and *[7,3]* codes—with both the LSB and 2-LSB methods. The error-correcting abilities of the three settings are in increasing order. The results are in rows three to six in Table 5.4. We note that using the weakest code *([15,13])* protects the 2-LSB method from bit errors, while the LSB method needs the strongest code of all—*[7,3]*—to achieve the same result. In terms of delivering the same amount of error-free secret data, the ECC overhead is about 13% for the 2-LSB method and is about 58% for the LSB method. Note that the settings we use are adopted from commonly used codes and may not be optimal.

Summary: By comparing the BERs between the conventional LSB stego method and our 2-LSB approach on Facebook, we find that embedding secret information in the 2-LSB jointly with a weak ECC is sufficient. It outperforms the LSB method in terms of ECC overhead when the same message capacity is delivered.

5.4.3 Evaluation with Steganalysis

Next, we use state-of-the-art steganalysis techniques to evaluate the detection likelihood of a secret embedding with our 2-LSB approach. We compare this with the detection likelihood in cases where traditional steganography tools, which embed information in the LSBs of the DCT coefficients, are used. Our goal here is to show that our approach does not significantly increase the detection likelihood, and thus, is viable in practice.

Steganalysis techniques in use: The goal of steganalysis is to detect the presence of embedded data in an image. Note that steganalysis does not attempt to extract the embedded message itself. To date, the most advanced steganalysis methods do supervised classification using machine learning tools like SVM or ensemble classifiers [100, 97]. For example, the method in [100] extracts feature sets to build the detector after being trained using a set of normal and stego-ed images. We use their ensemble classifier implemented in Matlab [101] along with the 548-dimensional CC-PEV features [102] [4]. As to commercial steganalysis products, Steganography Analyzer Artifact Scanner (StegAlyzerAS) developed by Steganography Analysis and Research Center (SARC) [104] is probably the best available steganalysis software in the market today. We use a limited time trial version with full functionality from their site.

Methodology: When using the ensemble classifier, we use a training set of 100 (each) normal and stego-ed images, respectively. The stego-ed images are produced by the 2-LSB and the traditional LSB stego tools; an equal number from both sources are considered. For both the normal and stego-ed images, we uploaded and then downloaded from Facebook.

Next, we apply the trained steganalysis tool on a test data set. The test set consists of different 100 (each) normal and stego-ed images. The false alarms on normal images contribute to the computed false positive rate, and missed detections on stego-ed images contribute to the

[4] CC-PEV was first proposed in [103] and its analysis is based on the use of an extensive set of DCT and other features.

computed false negative rates. When using StegAlyzerAS, we simply scan the folders containing normal and stego-ed images. We experiment on different sets of stego-ed images with embedded message lengths that consume 10%-50% of the image capacity. The DCT coefficients to be modified are pseudo-randomly chosen, spreading out evenly in the image.

Results and interpretation: Tables 5.5 and 5.6 present the detection accuracy for the academic and commercial tools that we use. We observe that the ensemble classifier is able to detect more stego-ed images constructed with the 2-LSB method than with the LSB method; however, the difference is insignificant. In particular, in typical regimes of interest (when the used image capacity is 10% or 20%), steganalaysis on the images created with the two methods exhibits very similar detection rates. As we aggressively embed more data into the images, the detection likelihood increases with both methods. Surprisingly, the commercial tool StegAlyzerAS categorizes a larger number of stego-ed images with the LSB method correctly, than those with the 2-LSB method. This is probably because the ensemble classifier uses machine learning while the StegAlyzerAS relies on known stego signatures and identifiable patterns of particular steganography tools.

Most importantly, from columns 4 and 5 in Table 5.4, we observe that when delivering the same amount of secret data, the proposed 2-LSB approach with a weak ECC (2LSB+ECC *[15,13]*) is less likely to be detected by *both* steganalysis tools than the traditional LSB approach with a stronger ECC (LSB+ECC *[7,3]*). While embedding information in the 2-LSB can increase detection likelihood, so can increased redundancy; the latter results in a higher number of changes to an image to deliver the same amount of secret data. We observe here that, due to the reduction in the level of redundancy needed, the 2-LSB scheme can in fact out-perform the traditional embedding method when used on online photo-sharing sites. In fact, due to the low redundancy required by 2-LSB, the detection likelihood with StegAlyzerAS when using this approach is practically identical with and without ECC.

Capacity used	*0.1*	*0.2*	*0.3*	*0.4*	*0.5*
LSB	0.44	0.62	0.75	0.87	1.00
2-LSB	0.50	0.66	0.81	0.94	1.00

Table 5.5: Detection accuracy of ensemble classifier on stego-ed images (false positive rate 0.23)

Capacity used	*0.1*	*0.2*	*0.3*	*0.4*	*0.5*
LSB	0.69	0.77	0.83	1.00	1.00
2-LSB	0.63	0.75	0.79	0.81	0.81

Table 5.6: Detection accuracy of StegAlyzerAS on stego-ed images (false positive rate 0.19)

Finally, we observe that the two steganalysis tools have a false-negative rate of 40–50% in cases where only 10% of the image capacity is used for secret embedding. Note this is almost equivalent to random guessing. Furthermore, as seen in Tables 5.5 and 5.6, the false positive rates are also fairly high ($\approx$ 20%). The detection accuracy will be even lower if lesser image capacity (say 5%) is used for secret embedding. Thus, if the users are careful enough not to embed secret information in the majority of the photographs that they upload, this suggests that the likelihood of detection is very low.

Summary: In summary, our experiments in this section suggest that embedding secret information in the 2-LSB of images first provides a decrease in the observed bit error rate. While embedding in a higher order bit inherently increases the likelihood of detection with steganalysis, the above property decreases the level of redundancy required (with 2-LSB) as compared to LSB embedding; this in turn decreases the likelihood of detection via steganalysis. Finally, we observe that the state-of-the art steganalysis tools only offer about a 40% - 50% likelihood of detection in the common cases wherein 10% of the image capacity is used for steganographic embedding; moreover, the false positive rates are about 20%. This suggests that 2-LSB embedding is a very practical way of embedding secret content on images uploaded onto Facebook, and our preliminary experiments on Flickr show similar results.

5.5 Bootstrapping the Covert Communication

Steganography can never be fully secure. The plain text may be extracted using exhaustive search. Thus, one will have to encrypt the secret information before embedding the information in images, in order to avoid exposure to interception. Encryption requires the establishment of secret keys between the communicating entities. In prior work, the existence of an out-of-band channel is assumed [58], via which such keys are established. However, in cases where censor authorities have massive access to information, the availability of such an out-of-band channel may be difficult. In this section, we first propose an approach for bootstrapping the covert communication without any out-of-band channel, i.e., bootstrapping of communication too is done by uploading images to exchange keys. Subsequently, we discuss the security properties provided by our approach.

5.5.1 Bootstrapping the Covert Channel

There are a few things that any user that would like to participate in covert communication by default agree to. If a user is open to be contacted by any friend in a covert way, his public key must be found in his newest profile photo. The steganography tool used for encoding/extracting message, and the handshake signaling need to be agreed on in prior.

In order to establish covert communication channels via photo sharing sites, we propose that a user first embed her public key (using steganography) in her profile photograph (without loss of generality, we assume an OSN such as Facebook for this discussion). By uploading new profile photos, the public key can be changed. Now, with this in place, we describe how two entities establish the covert channel below.

Let us consider a scenario wherein users A and B are friends on Facebook and have embedded their public keys on their profile pictures. A and B also install our common bootstrapping software that includes a steganography tool (which consists of an encoder and a decoder). Now let

us assume that B wishes to initiate the establishment of a covert channel with A. The bootstrapping phase consists of the following steps:

1. First the software on B's device fetches A's profile photo and extracts the first L_k 2-LSB bits from the DCT coefficients, where L_k corresponds to the length of the key (the length is configured in the software). At this point, B does not know that what she has is A's public key or has simply extracted some arbitrary string (since A may not have embedded a public key in her photo). The string of length L_k that is extracted is called K_A^{pu}.

2. B encrypts a signal ("request") with K_A^{pu} and embeds it in an uploaded image. The request could also contain metadata that indicates the length of the message (how many 2-LSB bits it consumes), and timestamp or nonce such that it is not constant.

3. If K_A^{pu} is a random string, and not A's public key as assumed, A does not respond to the communication. If A has indeed embedded his public key in his profile photo, he may extract the hidden message (depending on when he views the image). At this point, he decrypts the request signal from B's image, using his private key K_A^{pr}, thereby learning of B's intent to communicate.

4. A then retrieves B's profile photo and obtains her public key K_B^{pu}. Note that, at this moment, A knows for certain that he has a valid key (K_B^{pu}) and not just some random string.

5. A then creates a signal ("ack"), attaches a symmetric key k_s to the ack, and then encrypts this content with K_B^{pu}. The (secret) encrypted message is then embedded in an uploaded image.

6. B extracts the encrypted message from A's image, decrypts it using her private key K_B^{pr}, and gets both the ack and k_s. Note that a decryption failure at this stage indicates that A did not respond to her request.

7. B then encrypts a new signal ("ack2") with k_s and embeds this in a new image, which she then uploads.

8. A extracts the secret from B's image, decrypts the message using k_s, and obtains the signal "ack2". At this point, A has validated that B has the secret key k_s, and thus, the covert channel is established.

At the end of this sequence of steps, all the secret messages exchanged between A and B can be encrypted using k_s. In addition, k_s can be used as the seed to generate the pseudo-random series of DCT coefficients chosen to carry the secret message.

5.5.2 Security Properties

Next we discuss the security properties of our proposed bootstrapping approach. These properties come from the inherent privacy control and photo-sharing service provided by OSNs.

A ***Man-in-the-middle (MIM) attack*** can potentially occur during a typical key exchange wherein an eavesdropper, say C, somehow intercepts the communication between A and B. C could send his own public key and mislead B into believing that she has A's public key (and similarly deceive A into believing his public key is that of B). In the scenarios of interest, such cases are not possible since the only source of a user's public key is his own profile photo, which is under the complete control of the user. Thus, unless a user's account is compromised, the communication cannot be intercepted, and the MIM attack is implicitly avoided.

Detection of the existence of embedded key: As discussed earlier, steganalysis tools are far from perfect. As seen in Tables 5.5 and 5.6, the likelihood of false positives and detection misses are high. In fact, if only 10% of the image capacity is used for embedding secret information, our study suggests that the tools yield a detection miss rate of about 50%; this is equivalent to randomly guessing whether or not a secret is embedded. This, combined with the high false positive

rate, makes steganalysis difficult, if not impossible, if encrypted content is hidden.

However, beyond relying on the low probability of detection by steganalysis, users can also explicitly control access to their profile photos. Facebook offers flexible controls using which any user can control access to his profile photo by applying the proper privacy settings. When the audience selector is set to "friends only", the stored profile photo along with the embedded public key, if any, can only be accessed by the friends of the user. Others (such as friends of friends) can only access the thumbnail version. The thumbnail version of an image is produced by resizing the image (unless the original image has fewer pixels than the thumbnail size). Resizing downsamples the pixels in the image using scaling algorithms, and hence, the embedded message is typically not preserved in the thumbnail (the message was preserved in no case in our experiments). This ensures that only a user's friends can have access to his public key. Note that a covertly communicating user can always set his privacy settings such that his full profile photo can be accessed only by his friends. Since more Facebook users are beginning to be concerned about their privacy [105], there is no strong tie between limiting profile photo access and participation in covert communication.

Like Facebook, Google+ too enables users to restrict access to the full version of their profile photo; any Google+ user can specify the circles that are permitted to download her profile photo. In contrast, Twitter does not yet provide fine-grained privacy control on users' profile photos, and it can be accessed by anyone. However, it provides "Tweet" privacy controls such that only previously approved followers can see one's tweet which may have images associated with it. One can bootstrap conversations by sending such private tweets. Flickr, with a focus more on photo sharing as opposed to social functions, does not have a profile photo feature. However, Twitter and Flickr offer privacy controls to limit access to tweets and photo albums. Therefore, the bootstrapping of covert communication channels on Twitter and Flickr can instead be performed by embedding a user's public key in the first photo that the user ever uploaded to these services. When a user installs

our software, it can update the first photo that the user had shared on the service by embedding the user's public key in it. A second option is that users can bootstrap their covert channels on Facebook or on Google+, and then use Flickr or Twitter for covert communications. We defer further examination of these issues with Twitter and Flickr to future work.

If undesired entities are limited to having access only to the thumbnail of a user's profile photo, we find that the detection probability decreases further. To determine the reduction in the detection likelihood, we embed a 1024 bit RSA public key (about 10 %-15% of the image capacity) in 100 images using our proposed 2-LSB method. We upload them to the profile album on Facebook and download the thumbnail versions. We also upload the original images and download their thumbnail counterparts. We run the ensemble classifier and StegAlyzerAS on the downloads. We then choose those images that were flagged by the steganalysis tools and run the same tool on the thumbnail versions of the images. With the ensemble classifier, we find that the false negative rate (which was 50%) further increases; only 28 photos out of the 50 photos that were flagged to begin with are now flagged, i.e., the false negative rate increases to 72%. We find that the StegAlyzerAS tool is somewhat more robust to the reduction in size; however, we still notice an increase in the false negative rate. Of the 63 photos that were flagged originally, only 53 of them are still flagged with the thumbnail version, thus increasing the false negative rate to 47%.

Next we show that by appropriately choosing the profile photo, one can lower the detection likelihood. We experiment with a set of 10 images embedded with the public key, upload them and download both the full and thumbnail versions. One out of ten full-version images is flagged as non-stego by both steganalysis tools. Two out of ten thumbnail-version images are flagged as non-stego, one of which is the one surviving the full-version detection.

This simple experiment demonstrates that if one uses the right profile photo as carrier, there is a good chance that either his friends or anyone other than his friends will be unable to find

out if he has embedded something in his profile photo using steganalysis. Note that considering the large number of steganalysis tools available and the diverse techniques they use, the right profile photo for one tool may be detected by another tool.

Discussion: Finally, we discuss two issues relating to improving the efficiency of the bootstrapping phase.

Key length: In our implementation of the above key exchange process, we use a publicly available implementation of the RSA algorithm to generate the public key (of length 1024 bits). It is widely known that generating a public key component with Elliptic Curve Cryptogrpahy (ECC) is a better alternative to RSA [106]; with ECC, the key length is much shorter for providing similar security. For example, a 160 bit key generated with ECC provides equivalent security as a 1024 bit RSA key. The reduction in the key size directly translates to the embedding of a shorter secret message in the profile picture. Since shorter messages are harder to detect, this in turn will lead to a further decrease in the detection likelihood with steganalysis tools.

Embedding multiple messages in the same image: Let us suppose that B wants to send a "request" to other users besides A (step 2). B can encrypt copies of the "request" using these users' public keys and embed all the ciphers back to back in the same image. For instance, with a 160 bit key generated with ECC, if 10% of the image capacity is to be used, 6 requests can be packed in an image. Since the ciphers will have the same length, the entire secret message is composed of segments of the same length. After a recipient extracts the composite message (consisting of the segments) from an image, he can decrypt each of these segments using his private key. It does not matter that there are segments which are not meant for him. As long as he sees the signal "request" in one of these segments, he would know that he is one of the intended recipients. In this way, B can bootstrap the communication with multiple users at the same time. A similar approach can be used for responding to request and ack (in steps 5 and 7). We will consider such implementations

in future work.

5.6 Conclusions

In this work, we demonstrated how to establish a covert communication channel using uploaded photos on popular public photo-sharing sites. While using steganography for this purpose had previously received some attention, many nuances were ignored. We first showed via an in-depth measurement study that the processing performed by the online sites on the uploaded photos destroys the secret message in many cases. Our study also reveals the reasons for this loss. Based on the understanding developed with our study, we proposed a new way to embed information that can ensure the integrity of the message, while at the same time reducing the likelihood of detection via steganalysis tools. Finally, we also proposed and implemented an approach wherein users do not have to rely on an out-of-band channel to exchange keys, which are then used to encrypt the embedded messages in the photos.

Bibliography

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

[2] K. P. McGrath and J. Nelson. Monitoring & Forensic Analysis for Wireless Networks. In *Conf. on Internet Surveillance and Protection*, 2006.

[3] K. N. Ramach, E. M. Belding-royer, and K. C. Almeroth. Damon: A distributed architecture for monitoring multi-hop mobile networks. In *IEEE SECON*, 2004.

[4] S. Yang, S. Vasudevan, and J. Kurose. Witness-based detection of forwarding misbehaviors in wireless networks. In *UMass Computer Science Technical Report UM-CS-2009-001*, 2009.

[5] Wireless Open-Access Research Platform. http://warp.rice.edu/.

[6] K. P. McGrath and J. Nelson. Flux: A forensic time machine for wireless networks. In *IEEE INFOCOM 2006 Poster and Demo Session*, 2006.

[7] A. Adya, P. Bahl, R. Chandra, and L. Qiu. Architecture and techniques for diagnosing faults in ieee 802.11 infrastructure networks. In *ACM MOBICOM*, 2004.

[8] L. Qiu, P. Bahl, A. Rao, and L. Zhou. Troubleshooting wireless mesh networks. *ACM SIGCOMM Computer Communication Review*, 2006.

[9] J. Yeo, M. Youssef, and A. Agrawala. A framework for wireless lan monitoring and its applications. In *ACM workshop on Wireless security: WiSe*, 2004.

[10] Y.-C. Cheng, J. Bellardo, P. Benko, A. C. Snoeren, G. M. Voelker, and S. Savage. Jigsaw: Solving the puzzle of enterprise 802.11 analysis. In *SIGCOMM*, 2006.

[11] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Analyzing the mac-level behavior of wireless networks in the wild. In *SIGCOMM*, 2006.

[12] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *ACM MOBICOM*, 2004.

[13] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *ACM MOBICOM*, 2004.

[14] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-based models of delivery and interference in static wireless networks. In *ACM SIGCOMM*, 2006.

[15] L. Qiu, Y. Zhang, F. Wang, M. Han, and R. Mahajan. A general model of wireless interference. In *ACM MOBICOM*, 2007.

[16] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *ACM MOBICOM*, 2006.

[17] S. S. Yau, Y. Yin, and H. G. An. An adaptive approach to optimizing tradeoff between service performance and security in service-based systems. In *International Journal of Web Services Research*, 2011.

[18] T. S. Rappaport. *Wireless Communications: Principles and Practice (2nd ed.)*. Prentice Hall, 2001.

[19] C-K. Chau, M. Chen, and S. C. Liew. Capacity of large-scale csma wireless networks. In *ACM MOBICOM*, 2009.

[20] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, New Jersey, 1992.

[21] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw–Hill, New York, 1991.

[22] Opnet User's Documentation. http://www.opnet.com.

[23] howpublished="http://networks.cs.ucr.edu/testbed/" The UCR testbed.

[24] J. Ning, S. Singh, K. Pelechrinis, B. Liu, S. V. Krishnamurthy, and R. Govindan. Forensic analysis of packet losses in wireless networks. In *IEEE ICNP*, 2012.

[25] G. Kipper. *Wireless Crime and Forensic Investigation*. Auerbach Publications, 2007.

[26] Y. Guo and M. Simon. Network forensics in manet: Traffic analysis of source spoofed dos attacks. In *NSS*, 2010.

[27] K. Shanmugasundaram, N. Memon, A. Savant, and H. Bronnimann. Fornet: A distributed forensics network. In *International Workshop on Mathematical Methods, Models and Architectures for Computer Networks Security*, 2003.

[28] V. Corey, C. Peterman, S. Shearin, M. S.Greenberg, and J. VanBokkelen. Network forensics analysis. In *IEEE Internet Computing*, 2002.

[29] M. Soini, J. Kukkurainen, and L. Sydänheimo. Security and performance trade-off in kilavi wireless sensor network. In *ICCOMP*, 2010.

[30] W. Zeng and M-Y Chow. A trade-off model for performance and security in secured networked control systems. In *IEEE ISIE*, 2011.

[31] A. A. Abdullah, F. Gebali, and L. Cai. Modeling the throughput and delay in wireless multihop ad hoc networks. In *GlOBECOM*, 2009.

[32] K. G. Murthy. *Linear Programming*. Wiley, 1983.

[33] T. S. Kim, H. Lim, and J. C. Hou. Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks. In *ACM MOBICOM*, 2006.

[34] P.Smulders. Exploiting the 60 GHz band for local wireless multimedia access: prospects and future directions. *IEEE Communications Magazine*, 40(1):140–147, 2002.

[35] N. Guo, R. C. Qiu, S. S. Mo, and K. Takahashi. 60-GHz Millimeter-Wave Radio: Principle, Technology and New Results. *EURASIP Journal on Wireless Communications and Networking*, 2007.

[36] M.Park, C.Cordeiro, E.Perahia, and L.L.Yang. Millimeter-Wave Multi-Gigabit WLAN: Challenge and Feasibility. In *IEEE PIMRC*, 2008.

[37] H. Xu, V. Kukshya, and T. S. Rappaport. Spatial and Temporal Characteristics of 60 GHz Indoor Channels. *IEEE J. Sel. Areas Commun.*, 20(3):620–630, 2002.

[38] M. Peter, W. Keusgen, A. Kortke, and M. Schirrmacher. Measurement and Analysis of the 60 GHz In-Vehicular Broadband Radio Channel. In *Proc. of IEEE Vehicular Technology Conference*, 2007.

[39] A.Maltsev et al. 60 GHz WLAN Experimental Investigations. IEEE doc.802.11-8/1044r0.

[40] S. Vasudevan, J.Kurose, and D. Towsley. On neighbor discovery in wireless networks with directional antennas. In *Proc. IEEE INFOCOM*, Mar. 2005.

[41] N.Moraitis and P.Constantinou. Indoor Channel Measurements and Characterization at 60 GHz for Wireless Local Area Network Applications. *IEEE Trans. Antennas and Propagation*, 52(12), 2004.

[42] T.Zwick, T.J.Beukema, and H.Nam. Wideband Channel Sounder with Measurements and Model for 60 GHz Indoor Radio Channel. *IEEE Trans. Veh. Tech.*, 54(4), 2005.

[43] C.R.Anderson and T.S.Rappaport. In-building Wideband Partition Loss Measurements at 2.5 and 60 GHz. *IEEE Trans. Wireless. Comm.*, 3(3), 2004.

[44] J. Perez B. Neekzad, K. Sayrafian-Pour and J. S. Baras. Comparison of Ray Tracing Simulations and Millimeter Wave Channel Sounding Measurements. In *IEEE PIMRC*, 2007.

[45] IEEE 802.15 WPAN Task Group 3c (TG3c). Millimeter Wave Alternative PHY. http://ieee802.org/15/pub/TG3c.html.

[46] A. Davydov, A. Maltsev, and A. Sadri. Saleh-Valenzuela Channel Model Parameters for Library Environment. 802.15-06-0302-02-003c, July, 2006.

[47] A. Maltsev et al. Conference Room Channel Model for 60 GHz WLAN Systems - Summary. IEEE doc. 11-09/0336r0, March, 2009.

[48] T.S.Rappaport. *Wireless Communications: Principles and Practices. Second Edition.* Prentice Hall, New Jersey, 2002.

[49] T.Korakis, G.Jakllari, and L.Tassiulas. A MAC Protocol for Full Exploitation of Directional Antennas in Ad-hoc Wireless Networks. In *ACM MOBIHOC*, 2003.

[50] G.Jakllari, W.Luo, and S.V.Krishnamurthy. An Integrated Neighbor Discovery and MAC Protocol for Ad hoc Networks Using Directional Antennas. *IEEE Transactions on Wireless Commuications*, 2007.

[51] R.Mudumbai, S.Singh, and U.Madhow. Medium access control for 60 GHz outdoor mesh networks with highly directional links. In *IEEE INFOCOM 2009 Mini Conference*, 2009.

[52] S.Singh, F.Ziliotto, U.Madhow, E.M.Belding, and M.Rodwell. Millimeter Wave WPAN: Cross-Layer Modeling and Multihop Architecture. In *IEEE INFOCOM Minisymposium*, 2007.

[53] S.Vasudevan, D.Towsley, D.Goeckel, and R.Khalili. Neighbor Discovery in Wireless Networks and the Coupon Collector's Problem. In *ACM MOBICOM*, 2009.

[54] G.Pei, M.A.Albuquerque, J.H.Kim, D.P.Nast, and P.R Norris. A neighbor discovery protocol for directional antenna networks. In *MILCOM*, Oct. 2005.

[55] F.Yildirm and H.Liu. A Cross-Layer Neighbor-Discovery Algorithm for Directional 60-GHz Networks. *IEEE Trans. Veh. Tech.*, 2009.

[56] Z.Zhang. Performance of neighbor discovery algorithms in mobile ad hoc self-configuring networks with directional antennas. In *MILCOM*, Oct. 2005.

[57] K.Pahlavan, G.Yang, T.Holt, Y.Xu, P.Krishnamurthy, and J.Beneat. Ray tracing tools. http://www.cwins.wpi.edu/project/scripts/ray-tracing.html.

[58] S. Burnett, N. Feamster, and S. Vempala. Chipping away at censorship firewalls with user-generated content. In *USENIX Security*, 2010.

[59] Neil F. Johnson and S. Jajodia. Exploring steganography: Seeing the unseen. In *IEEE Computer*, 1998.

[60] J. Kelley. Terror groups hide behind Web encryption. http://www.usatoday.com/life/cyber/tech/2001-02-05-binladen.htm.

[61] Facebook photo upload compression. http://scottwyden.com/facebooks-photo-upload-compression/.

[62] Flickr-Help-Photos. http://www.flickr.com/help/photos/.

[63] Facebook Help Center. http://www.facebook.com/help/?page=132417053500408.

[64] J. Zollner, H. Federrath, H. Klimant, A. Pfitzmann, R. Piotraschke, A. Westfeld, G. Wicke, and G. Wolf. Modeling the security of steganographic systems. In *International workshop in Information Hiding*, 1998.

[65] The JPEG committee home page. http://www.jpeg.org/.

[66] JPEG marker code assignment (ISO/IEC 10918-1: 1993(E)). http://www.digicamsoft.com/itu/itu-t81-36.html.

[67] W. B. Pennebaker and J. L. Mitchell. *JPEG still image data compression standard (3rd ed.).* Springer, 1993.

[68] JSteg. http://zooid.org/ paul/crypto/jsteg/.

[69] A. Westfeld. F5-a steganographic algorithm. In *International Workshop on Information Hiding*, 2001.

[70] K. Solanki, A. Sarkar, and B. S. Manjunath. YASS: yet another steganographic scheme that resists blind steganalysis. In *Intl. Workshop on Information Hiding*, 2007.

[71] A. Cheddad, J. Condell, K. Curran, and P. M. Kevitt. Digital image steganography: Survey and analysis of current methods. In *Elsevier Signal Processing*, 2010.

[72] P. Alvarez. Using extended file information (EXIF) file headers in digital evidence analysis. In *Intl. Journal of Digital Evidence, Economic Crime Institute*, 2004.

[73] StegHide. http://steghide.sourceforge.net/documentation/manpage.php.

[74] C. C. Chang, T. S. Chen, and L. Z. Chung. A steganographic method based upon JPEG and quantization table modification. In *Information Sciences*, 2002.

[75] OutGuess. http://www.outguess.org/.

[76] F5. http://wwwrn.inf.tu-dresden.de/ westfeld/f5.html.

[77] K. Solanki, N. Jacobsen, U. Madhow, B. S. Manjunath, and S. Chandrasekaran. Robust image-adaptive data hiding using erasure and error correction. *IEEE Trans. Image Process*, 2004.

[78] Photobucket. http://photobucket.com/.

[79] Picasa. http://picasa.google.com/.

[80] Google+ help. http://support.google.com/plus/?hl=en.

[81] Twitter Help Center. https://support.twitter.com/articles/20156423.

[82] The Facebook blog. http://blog.facebook.com/blog.php?post=432670242130.

[83] Secret Messages Come in .Wavs. http://www.wired.com/news/politics/0,1283,41861,00.html.

[84] N. Provos and P. Honeyman. Detecting steganographic content on the internet. Technical report, In ISOC NDSS, 2001.

[85] Steganographic Command and Control: Building a communication channel that withstands hostile scrutiny. http://www.irongeek.com/i.php?page=security/steganographic-command-and-control.

[86] SecreTwit: Social Steganography. https://code.google.com/p/secretwit/.

[87] S. Nagaraja, A. Houmansadr, P. Piyawongwisal, V. Singh, P. Agarwal, and N. Borisov. Stegobot: a covert social network botnet. In *Intl. Conf. on Information hiding*, 2011.

[88] K. Noy. A look at Steganography. http://www.dosarrest.com/en/news/90-a-look-at-steganography.html.

[89] Steganography in Social Media. http://www.greatplay.net/essays/steganography-in-social-media.

[90] K. Wilson. Previous address: http://pages.prodigy.net/robyn.wilson/.

[91] JPEG Group. libjpeg. http://www.ijg.org/.

[92] C. Hass. JPEGsnoop 1.5.2. http://www.impulseadventure.com/photo/jpeg-snoop.html.

[93] JPEG analysis utilities. http://www.theonlineoasis.co.uk/scriptslibraries/jpegutils.html.

[94] E. M. Schwalb. *iTV Handbook: Technologies and Standards*. Prentice Hall, 2003.

[95] Facebook versus Google+. http://www.foxnews.com/scitech/2012/01/21/facebook-vs-google-whats-best-social-network/.

[96] Flickr Popularity. http://web.appstorm.net/roundups/media-roundups/top-20-photo-storage-and-sharing-sites/.

[97] J. Kodovsky, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security*, 2012.

[98] J. R. Pierce. *An Introduction to Information Theory: Symbols, Signals and Noise*. Dover Publications, 1980.

[99] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 1960.

[100] J. Kodovsky and J. Fridrich. Steganalysis of JPEG images using rich models. *SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics*, 2012.

[101] Ensemble classifier. http://dde.binghamton.edu/download/ensemble/.

[102] Feature Extractors for Steganalysis. http://dde.binghamton.edu/download/feature-extractors/.

[103] T. Pevny and J. Fridrich. Merging markov and DCT features for multiclass JPEG steganalysis. *SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents*, 2007.

[104] SARC: Steganography Analysis and Research Center. http://www.sarc-wv.com/.

[105] R. Dey, Z. Jelveh, and K.W. Ross. Facebook users have become much more private: A large-scale study. In *Intl. Workshop on Security and Social Networking*, 2012.

[106] The case for elliptic curve cryptography. http://www.nsa.gov/business/programs/elliptic_curve.shtml.