TNO-Information and Communication Technology

Connectivity

Faculty of Electrical Engineering, Mathematics and Computer Science

Network Architectures and Services

Capacity measurement in small-scale heterogeneous best-effort IP networks

F.P. Nicolai
(1067036)

Committee members:
Supervisors: dr. ir. F.T.H. Den Hartog, dr. ir. F.A. Kuipers
Member: dr. ir. R. Hekmat

July 02, 2009
M.Sc. Thesis No: PVM 2009 – 057

Abstract

Current home networks are not well designed to support quality of service (QoS). Successfully employing current QoS solutions in heterogeneous environments, such as home networks, requires compatibility between the many possible solutions for the different network technologies. Often the solutions need support from all devices throughout the network. Additionally, it is often difficult to configure QoS settings properly.

An alternative approach that adds some QoS support to home networks is admission control based on path quality assessment. The service delivering device decides if a new service can be admitted based on a quick path quality measurement. Currently there are no tools available for performing a fast and accurate path quality measurement in heterogeneous network paths.

This report describes the design and evaluation of a path capacity measurement tool for small-scale, heterogeneous, best-effort IP networks. A new probing method is developed that obtains the bottleneck capacity in paths consisting of different link layer technologies. Besides standard IP support, we do not put any requirements on the service receiving devices. The performance of the newly developed probing method is evaluated through simulations of Ethernet/802.11b networks in OPNET Modeler©.

Simulation results show that the path capacity estimation tool provides accurate (typically ±0.5 Mbps) estimates for the path capacity (typically within 5 seconds) if employed in paths with low to moderate cross traffic intensities. For higher cross traffic intensities, the performance of the tool decreases.

Key words: capacity estimation, capacity measurement, packet-pair probing, quality of service, home networks

Acknowledgements

My sincere appreciation goes out to my supervisors at TNO Information and Communication Technology, Frank den Hartog and Archi Delphinanto. Their willingness to help and their valuable feedback and suggestions really supported me during the research work and during the completion of the thesis. I am truly thankful for their guidance, their help and their patience.

I also like to thank Fernando Kuipers, from Delft University of Technology, who supervised on behalf of the Network Architectures and Services group. He guided me through the graduation process and provided me with useful comments and suggestions.

Special gratitude goes out to Dick van Smirren. The conversations that I had with him during my time at TNO-ICT really supported me and even changed my way of thinking and my view on several aspects of life.

Further, thanks go out to all colleagues and friends at TNO-ICT and at university who provided me with useful comments and who created a pleasant environment for me to work in.

Finally, I would like to thank my family because they always support me in the choices that I make.

Floris Nicolai
July, 2009

Contents

1 Introduction

1.1 Motivation and goal

Currently there is a process of convergence taking place in the domain of personal and home networks [1]. What can be observed is that many different devices will be able to inter-communicate over a plethora of different networks technologies. In order to provide satisfactory user experience of the services provided by these networks, a certain quality-of-service (QoS) has to be guaranteed by these networks. Current home networks are not well designed to support QoS. One cause for this lies in the many different QoS solutions that are available. Most of these solutions, such as 802.11e or DiffServ, are based on classification and prioritization of services. In some cases a form of network resource management is incorporated [2]. However, successfully employing these QoS solutions in heterogeneous environments such as home networks, if possible at all, requires compatibility between the different solutions for the different networking technologies used throughout the network. Additionally the solutions need to be supported by all devices within the network, which makes implementation of these solutions relatively expensive and excludes legacy devices. Further, many configuration parameters are left open to be specified by the end user, who often has a lack of knowledge or who is simply unwilling to do this.

In order to bypass all these problems, another approach is suggested in [3]. The solution adds some support of QoS to today's best-effort home networks by the addition of a resource manager (RM). The suggested location of this resource manager is in the residential gateway (RG), although it could be implemented anywhere within the home network.

The idea comprises the addition of device resource reservation to service discovery protocols. The central resource manager in the network decides whether a newly requested service can be admitted, based on a quick path quality measurement between the service providing device and the receiver. Additionally, open service sessions could be terminated as soon as available bandwidth in the network becomes insufficient. The idea is that the measurement can be performed from the device that wants to deliver the service, without requiring special support from the device that is the receiver of the service.

The main problem concerning this idea, is that currently no satisfactory path quality measurement tool exists, i.e. there is no tool that is able to perform a path quality measurement from a single node, in a heterogeneous network, in a fast and low-intrusive way. Therefore we are developing such a tool. Since available bandwidth is a key parameter for the path quality, the focus of this research lies on an IP level bandwidth estimation method. We try to find an answer on the question: how can we measure available bandwidth at IP level in a low-intrusive, low-convergence-time and accurate way without requiring modifications to currently existing IP clients. In this situation the device that acts as the end point of the measurement is regarded as the client, because it will eventually act as the receiver of the service.

The main goal of the thesis is to develop a tool for measuring available bandwidth between two nodes in small scale, heterogeneous best effort IP network which is a more formal description of a typical home network. The tool should only need to be implemented at the measuring node and should not require installation of extra hardware or software in the nodes to be measured. Additionally, the measurement should be of low intrusiveness, therefore not disrupting existing flows in the network. Finally, the tool should operate near real-time. The maximum measurement delays should be in the order of seconds. The outcome of the tool can be used for 'admission control' within the network or for determining the maximum possible service quality level so that the newly admitted service will not affect existing flows in the network.

1.2 Scenarios

Two example scenarios which are illustrated by figure 1.1, exemplify how the resource manager supporting path quality measurement could be implemented in reality.

Scenario 1

Generally, network operators have control over the network up to the point where the operator network connects to the home network. Therefore, it is currently the case that for services like IPTV a VLAN (Virtual LAN) in the public network is terminated on the RG, and a dedicated connection between the RG and the set-top box (STB) is required, to guarantee quality of the service within the home. This means that from the RG a separate cable needs to be drawn to the STB. However, it would be far more convenient if the user could connect the STB anywhere in the home network, without drawing separate cables, or in the case of wireless connections, without drawing any cables. Such a scenario is illustrated in figure 1.1. And having the STB in the same IP subnet as the other devices in the home means that the STB can also be used for other services than IPTV.

In the picture we see a home network where all devices are connected through wireless links, except the media server which has a wired connection to the router. User A is currently using part of the network resources for file sharing. User B is watching a movie from the media server. The movie is streamed over the wireless network connection. Now user C wants to watch a television show that is delivered through an IPTV service. At the moment that user C switches on the STB, the request for the new stream is noticed by the RG. The RG initiates a path quality measurement to the STB to determine whether there are sufficient resources available for smooth service delivery. If that is the case, then the IPTV stream will be admitted. If there are not enough resources available, the stream will be denied.

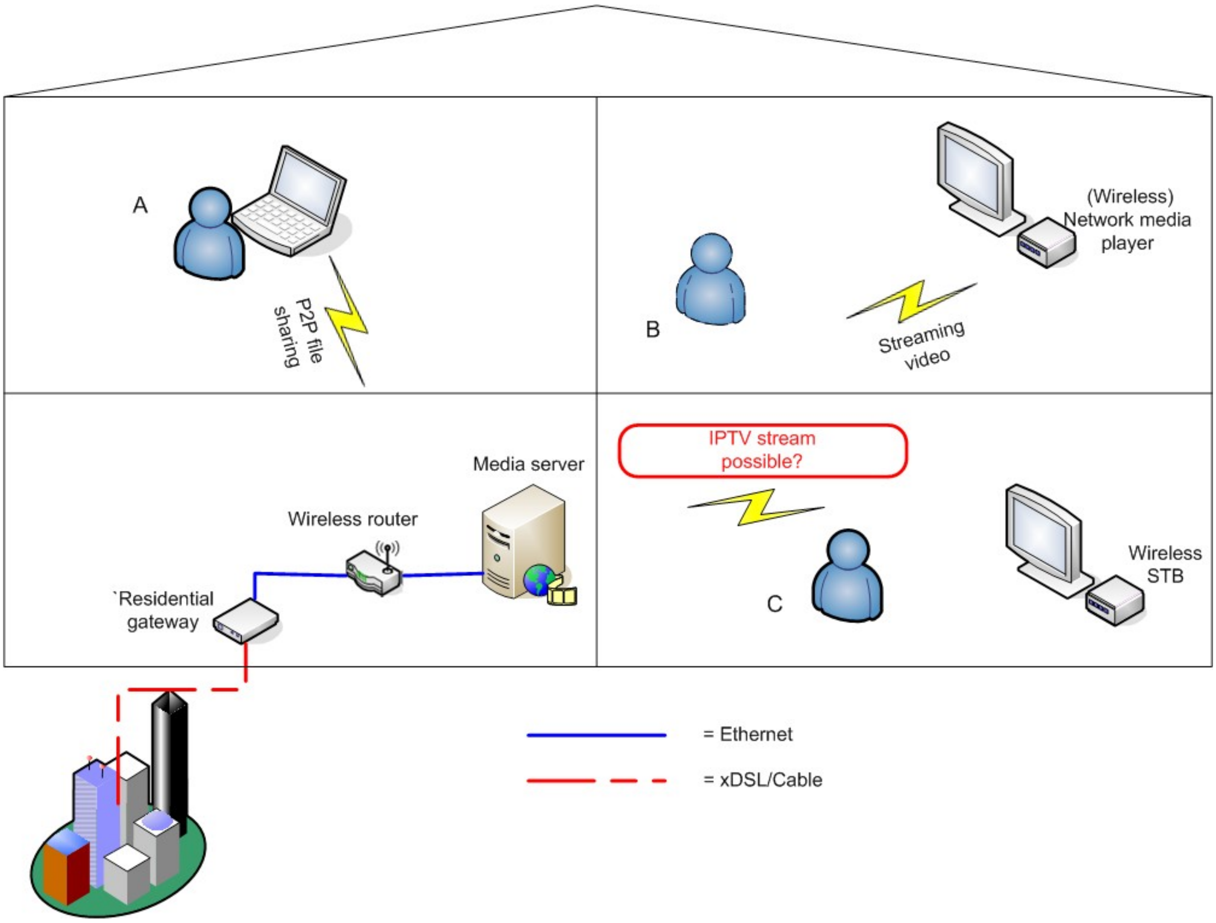

Figure 1.1: Example of a scenario for delivering public services over the best effort home network

Scenario 2

For the second scenario (Figure 1.2) we consider the same network as in the previous examples. Again user A is using part of the network resources for file sharing. However, in this scenario, user C is watching a television show delivered through the IPTV service. Now user B wants to watch a movie from the media server. As the (wireless) network media player does a request for the movie stream, a path quality measurement is initiated from the media server to the network media player. If the resources in the network are sufficient, the stream is admitted, if resources are insufficient the stream is denied. Another possibility would be that the stream quality is adjusted to the currently available resources.

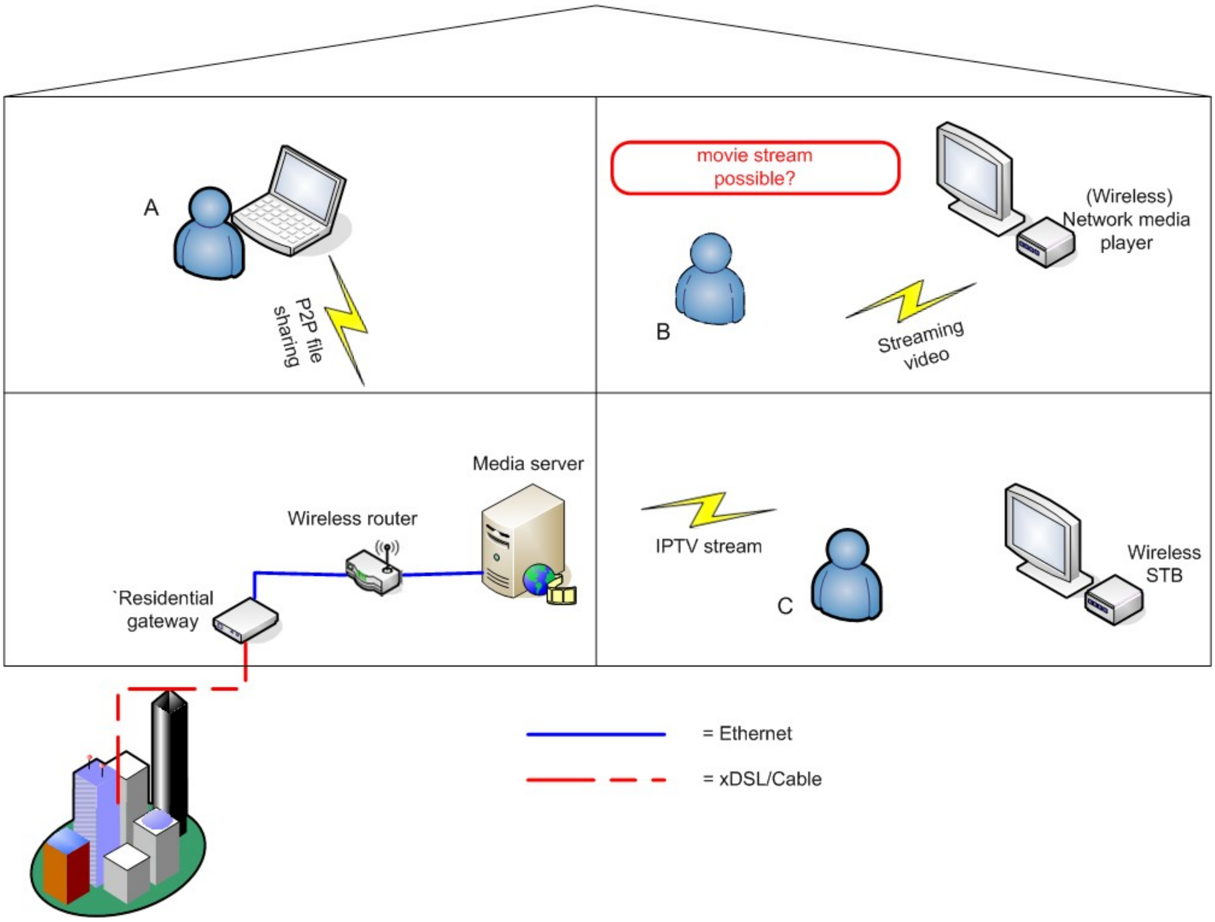

Figure 1.2: Example of a scenario for delivering privately owned services over the best effort home network

1.3 Methodology and thesis structure

We start our research with a thorough study of related work and of different measuring techniques. Based on the findings of this study we propose a new measuring concept. The new measuring concept is verified through simulation of increasingly complex networks in OPNET network simulation software. Through an iterative process of adapting the initial concept and verification through simulation we develop a new tool that meets all requirements mentioned in section 1.1.

The thesis is organized as follows. In chapter 2 we explain how different link quality metrics should be interpreted. Additionally we provide definitions for different bandwidth related metrics. Chapter 3 provides a brief overview of previous work related to bandwidth measurement in communication networks. Various different bandwidth measurement concepts are explained. Chapter 4 explains different methods and tools that have been used to perform our research. Chapter 5 gives a description of the newly suggested method for path capacity measurement which turns out to be the most important step in the path quality assessment procedure. The newly suggested method is analyzed through simulation and the results are presented. Subsequently in chapter 6 we propose and analyze a method to improve the accuracy of the developed measurement concept. Finally we end up with the conclusions of our work and suggestions for future work in chapter 7.

2 Introduction to link/path metrics (network quality metrics for in-home networks)

Terms like quality of service (QoS), capacity and available bandwidth are widely used but the meaning of these terms is mostly kept very vague. This is exemplified by documents such as RFC 2330 [4], RFC 3148 [5] and RFC 5136 [6] that are dedicated to defining various network metrics like “capacity”, “available capacity”, “bulk transport capacity” etc. As stated in [6], one objective of these documents is “to provide a common framework for the discussion and analysis of a diverse set of current and future estimation techniques”.

This chapter aims to clarify what is meant with various relevant terms. Further, the chapter provides several definitions for some bandwidth related metrics. Section 2.1 gives a short description of QoS and its relation to link quality. Section 2.2 gives definitions for capacity and available bandwidth. Additionally the terms 'narrow link' and 'tight link' are introduced. Section 2.3 provides a short description of other bandwidth related terms found in literature.

2.1 QoS introduction

Quality of Service is generally defined as the objective (measurable) performance level that is offered by the network to the user. By QoS provisioning, network resources can be better utilized and the deterministic behavior of traffic flows can be controlled better. Different services have different requirements regarding network performance. In the case of multimedia traffic, low latency and low jitter are of utmost importance. Additionally, multimedia traffic often requires relatively high bandwidths to be available for smooth delivery of the traffic flow.

The performance level at which a service can be delivered is strongly related with the path quality in the underlying network. Link/path quality is often expressed as a function of the following four parameters: available bandwidth, latency, jitter and packet/bit error rate. Therefore, in order to predict the maximum possible quality of a service over a path, it is necessary to have knowledge about these quality parameters.

2.2 Capacity and available bandwidth

Before we start measuring capacities or available bandwidths, it is necessary to have a clear definition of what is meant with these various terms. From the Oxford dictionary we find already two different definitions for bandwidth

Bandwidth

1 A range of frequencies, especially used in telecommunications
2 The transmission capacity of a computer network or other telecommunication system

The first one refers to a range of frequencies and is usually expressed in terms of hertz (Hz). The second definition refers to transmission capacity of (computer) networks and is usually expressed in bits per second (bps). In this work, where only digital communication networks will be considered, only the second definition of bandwidth will be used. The two are related to each other through Shannon's formula:

$$C = B \log_2(1+\frac{S}{N}) \tag{2.1}$$

where C is the maximum channel capacity at the physical layer in bits per second, B is the bandwidth in Hertz, S is the total signal power over the bandwidth, N is the total noise power over the bandwidth. S/N is also called the signal to noise ratio.

Nodes, links and paths

Before defining different link or path metrics, it is necessary to first have a notion about what actually links or paths are. In this work we use the same definitions as those given in the RFC 5136 [6] document.

"Nodes are defined as hosts, routers, Ethernet switches, or any other device where the input and output links can have different characteristics. A link is a connection between two of the network devices or nodes. A path P of length n is defined as a series of links (L1, L2, ..., Ln) connecting a sequence of nodes (N1, N2, ..., Nn+1). A source S and destination D reside at N1 and Nn+1, respectively. Furthermore, a link L is defined as a special case where the path length is one" [6].

Link Capacity at the IP layer

As mentioned in RFC 5136, capacity is only meaningful when defined relative to a given protocol layer in the network. When describing link layer technologies, often nominal physical link capacities are used. For example, 10 Mbps for 10BASE-T Ethernet or 11 Mbps for 802.11b wireless links. The nominal physical link capacity is the theoretical maximum amount of data that a link L can support, measured at the physical layer [6].

However, there are several factors which reduce the information carrying capacity of a link at the IP layer, as is described briefly in section 4.1. For IP layer capacity we use the following definition:

"Path/link capacity is the maximum achievable end-to-end throughput at IP-layer that can be sustained over the path/link in the absence of cross traffic, achievable with maximum sized packets

Cross traffic is here defined as any traffic in the same physical channel caused by other flows than the primary (probing) flow.

IP layer path capacity and narrow link

Generally the capacity of a path is determined by the slowest link in the path. The link with the lowest capacity in the path is called the narrow link. The narrow link is illustrated in figure 2.1. In the picture three consecutive links are shown. If H is the number of hops in the path and $C_{l,i}$ is the link capacity of link i then the capacity of the path C_p is

$$C_p = \min_{i=0...\mathrm{H}} C_{l,i} \tag{2.2}$$

This formula also holds in the case that networks contain shared media. It is very well possible in home networks that different links use the same shared channel. Consider for example a path between two wirelessly connected devices that both communicate with the same access point (AP). All traffic that goes from one wireless station to the other has to pass the wireless medium twice; first from the source device to the AP and second from the AP to the destination device. Because both links make use of the same wireless channel, the channel access is divided over the two. Therefore the capacity of the path will only be approximately[1] half the capacity of the individual links. If the full capacity of the channel C_{ch} is known then the capacity of a link in a path containing n wireless links sharing the same channel can be obtained from the following formula:

$$C_{l,i} = C_{ch} / n \tag{2.3}$$

It is important to realize that with this definition, the link capacity of a wireless link depends on how often the primary (probing) flow is passing the same wireless medium. This is because the primary flow is then acting as "cross traffic for itself", but this case is excluded from the definition of cross traffic as given before.

In section 4.1.2 we show what the consequence is for the capacity of paths that contain two 802.11b wireless links.

Available bandwidth and tight link

Our ultimate goal is to provide information to an application, about the maximum bandwidth that is available in the network, so that the application can judge if the network is able to handle the traffic flow without disrupting existing flows in the network and while maintaining acceptable quality of the service. Just like capacity, the term available bandwidth is only meaningful when defined for a given protocol layer in the network. We use the following definition for available bandwidth:

"Path available bandwidth is the maximum throughput that a path can provide to a data flow without causing significant degradation in service quality of other ongoing flows or to itself"

Similar to capacity, the available bandwidth within a path is often regarded to be determined by the link that has the lowest available bandwidth in the path, often called the "tight link". When two links in a path share the same medium, then the available resources of the medium will be divided over both links.

Superbottleneck link

The term superbottleneck link is used for a link that is both the link with the lowest capacity

[1] The capacity of the channel is dependent on the average random backoff and thus on the number of contending nodes.

and the link with the least bandwidth available. In other words, the link is both the “narrow link” and the “tight link” in a path. Figure 2.1 illustrates the different bottlenecks in a path.

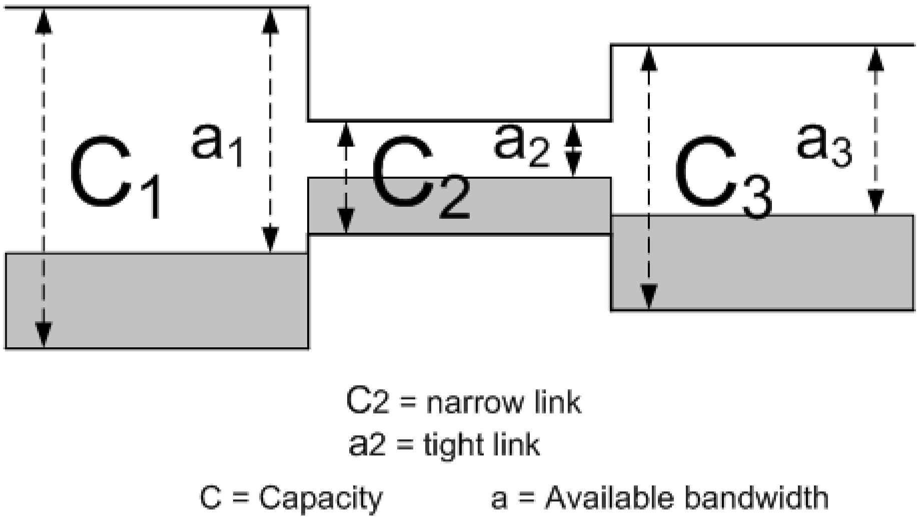

Figure 2.1: Narrow link and tight link

The link in the middle is the “super bottleneck” link. This link has both the lowest capacity (narrow link) and the lowest available bandwidth (tight link) of all links in the path.

2.3 Other bandwidth related metrics

Asymptotic Dispersion Rate

In literature sometimes the term asymptotic dispersion rate (ADR) is mentioned as a capacity related metric. This term was introduced by Dovrolis in [7]. The ADR refers to maximum achievable throughput that can be achieved when data is aggressively send over the channel, i.e without considering existing flows and therefore possibly disturbing these existing flows. As stated by Dovrolis, several early works that were claimed to measure link or path available bandwidth were actually measuring ADR.

Bulk transport capacity

Another network bandwidth related term that is regularly found in literature is bulk transport capacity (BTC). This metric is accurately described in the IETF RFC3148 document [5] as:
“Bulk Transport Capacity (BTC) is a measure of a network's ability to transfer significant quantities of data with a single congestion-aware transport connection (e.g., TCP). The intuitive definition of BTC is the expected long term average data rate (bits per second) of a single ideal TCP implementation over the path in question.”
In this report we are not interested in the maximum possible throughput using a congestion aware protocol, because our method is mostly to be used for services that use UDP (User Datagram Protocol) as a transport protocol. Our goal is to prevent congestion from happening by providing a number for the available bandwidth and thus for the maximum possible data

rate that is possible in the path according to the definition of available bandwidth.

3 Related work

This chapter provides an overview of various approaches to link/path bandwidth(s) measurement from previous works. Section 3.1 gives a brief overview about the history of bandwidth measurement. In section 3.2 several conceptually different methods for measuring link/path capacity and/or available bandwidth measurements are explained. Additionally, several state-of-the art measurement tools will be described, that are based on these methods. Finally, in section 3.3 the chapter is concluded with an analysis of shortcomings of existing tools.

3.1 History of bandwidth measurement

Up to 2004 most bandwidth measurement tools were targeted at measuring bottleneck bandwidth in Internet paths. In a paper from 1988 about congestion avoidance in TCP [8], Jacobson describes how the minimum spacing between packets in a path is determined by the bottleneck link. Later, in 1991, Keshav describes how this principle can be used to measure bottleneck bandwidth using packet pair probing.

In 1992, Bellovin publishes his paper [9] about the network performance model. In this work Bellovin describes how “variable packet size (VPS) probing” can be used to measure device packet handling capacity. This is done by investigating the relation between packet size and transit cost (total node forwarding time). This transit cost consists of a combination of “packet processing” and “packet forwarding” delay. The processing part is assumed to be constant while the forwarding part is assumed to be more or less linearly dependent of packet size. Actually, the forwarding rate of a device is determined by the interface link layer technology through which the device is forwarding the packets.

In 1996, Carter and Crovella present their measurement tool called “*bprobe” [10].* The tool exploits the packet pair “bottleneck spacing” effect to obtain an estimate of the capacity of a path. In the same publication Carter and Crovella present another tool, called “*cprobe”*, explicitly aimed at measuring congestion in a network path. To measure congestion on the bottleneck link *cprobe* exploits the “probe gap model” or “packet gap method” (PGM), which will be explained in subsection 3.2.2. By subtracting the congestion from the path capacity, Carter and Crovella obtain an estimate for the available bandwidth in the path.

In 1997, a tool called *Pathchar* is presented by Jacobson [11]. This tool measures hop-by-hop link capacities. In other words, the tool successively measures all link capacities in a path. The tool probes a link with varying packet sizes and from the relation between transmission time and packet size the capacity of the link is deferred.

Many capacity estimation tools would follow that make use of either the “bottleneck spacing effect” ([12], [13], [14], [15], [16], [17], [18], [19], [20], [21]) or “variable packet size probing” ([11], [22], [23]) as a basis for inferring link/path capacity.

In 2000 the available bandwidth measurement tool *TOPP* (Trains of Packet Pair) is introduced. The tool probes the network with “trains of packet pairs” at an increasing rate. The rate at which the overall delays of the pairs within a train start increasing is determined to be the available bandwidth of the tight link, and thus the available bandwidth of the path. The concept of determining turning points in the delays of increasing rate probing traffic is referred to as the “probe rate model” (PRM).

Both the PRM concept and the PGM concept lie at the basis of many other available bandwidth measurement tools (PRM:[24], [25], [26], [27], [28] PGM:[10], [29], [30], [31]).
In 2004 *CapProbe ([15], [16])* was the first capacity estimation tool based on packet pair probing, to claim suitability for heterogeneous paths. Also in 2004 *ProbeGap [32]* was introduced as a tool for measuring available bandwidth in “broadband access networks”, such as cable-modem and 802.11-based wireless access networks. *ProbeGap* aims to determine the utilization of the bottleneck link by determining the ratio between non delayed probe packets and probe packets delayed due to cross traffic in the bottleneck link. An estimate of the available bandwidth is obtained by taking the non-utilized share of the “narrow link” capacity.
In 2006 two other tools *DietTOPP* [26] and *WBest* [18], were introduced to measure available bandwidth over paths including wireless links. *DietTOPP* is a PRM based tool while *WBest* is a PGM based tool. Finally in 2008 a new tool called SLOT was introduced. This is a PRM based tool that uses active probing to measure available bandwidth in ad hoc wireless networks.
Starting from 2002 several tools ([33], [34], [35], [36], [37]) have been developed that are targeted at measuring bottleneck bandwidths in ad hoc wireless networks. Most of these tools make use of MAC layer information in order to obtain link state information. This information is used to obtain a capacity or available bandwidth estimate. These tools are not suitable for transport layer bandwidth measurement in heterogeneous network environments, and will therefore not be discussed any further in this report.

3.2 Methods for measuring path bandwidths

The objective of this work is to develop a bandwidth measurement tool that can be run by the source node at any moment in time. Therefore only active measurement tools are discussed in this work. This means that the tools actively send probes into the network in order to obtain information from the network. All existing active measurement tools are based on only a few different measurement concepts. These different concepts are described in this section. First the main path/link capacity measurement concepts will be explained. Subsequently different concepts for measuring available bandwidth will be explained. Additionally several state of the art tools based on these different concepts will be shortly discussed.

3.2.1 Measuring capacity

Variable packet size probing (VPS)

The time it takes to send the actual bits of a packet over a link depends both on the operational physical layer capacity of the link and on the packet size. This means that transmission time t_{tr} and packet size L are related to each other through the physical layer link capacity $C_{l,phy}$ as following:

$$C_{l,phy} = L / t_{tr} \qquad (3.1)$$

Usually it is difficult to measure the transmission time for a single packet, and therefore to determine the physical layer link rate. However, the transmission rate of a link (which is the same as the operational physical layer link capacity) can be determined by the difference in the delays of the transmission of two different sized packets using the following formula:

$$C_{l,phy} = \frac{(L_2 - L_1)}{(d_2 - d_1)} \qquad (3.2)$$

where L_2 and L_1 are the larger and smaller packet sizes and d_2 and d_1 are the delays of the larger and the smaller packets respectively.

Although this formula for calculating the physical layer capacity of a link is valid for a single link, it does not apply for the end-to-end capacity of a path consisting of multiple links. This is because the difference in the delays between different sized packets increases at each link, while obviously the size of the packets stays constant throughout all the links in the path. When formula 3.2 would be applied it would imply that the capacity of a path decreases with the number of links while, according to the definition, the path capacity is determined solely by the narrow link in the path. Therefore, to use VPS probing to determine path capacity, it is necessary to measure all links in a path separately and successively.

Several other problems exist concerning VPS probing for measuring path capacity. First of all, the VPS method measures the physical transmission rate of the medium instead of the throughput of “higher layers”. In the assumption that packet processing delays, media access delays etc., are independent of packet size, the packet size and transmission time are related solely through the physical rate of the link.

Since our goal is to measure the maximum achievable throughput at transport layer, it can be concluded that capacity estimation based on VPS probing is not a suitable solution for us.

Packet pair/train probing

The bottleneck spacing effect described by Jacobson [8] forms the basis of the packet pair/train probing concept for path capacity measurement. It is illustrated in Figure 3.1.

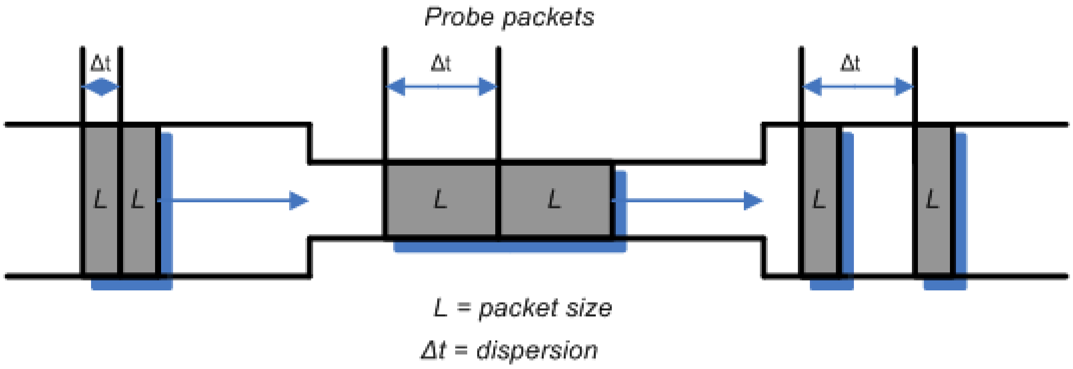

Figure 3.1: Illustration of capacity measurement using “packet pair probing”

Three consecutive links are shown. The link in the middle has the lowest capacity (narrow link). Two packets are sent back-to-back in the path (from left to right). The dispersion Δt is small since the initial link speed is relatively high. In the middle link, the packets are transmitted at a lower rate. Therefore Δt increases. In the last link, the packets are transmitted at a higher rate again, but this does not affect Δt anymore.

The illustration shows that the minimum spacing between packets that can be obtained after travelling a path, is determined by the narrow link of the path. Since path capacity is by our definition determined by the narrow link, the obtained minimum dispersion between packets is an indication of the "path capacity".

The capacity of the narrow link $C_{l,n}$ can be calculated with the following formula:

$$C_{l,n}=L/(\Delta t) \tag{3.3}$$

where L is the packet size and Δt is the dispersion measured at the destination node.

In contrast to VPS, where it is necessary to measure the one-way delays (OWDs) of the packets, the packet pair method only requires measuring the arrival times of the packets. This is much easier because it does not require clock synchronization between the sender and receiver. The dispersion can be obtained by:

$$\Delta t_r=t_{a2}-t_{a1} \tag{3.4}$$

where Δt_r is the dispersion between the packets at the receiver. t_{a1} and t_{a2} are the arrival times of the first and the second probe packet respectively. The dispersion at the receiver can be derived from the departure times of the packets and the delays of the packets using the following formula:

$$\Delta t_r=(d_2-d_1)-(t_{d2}-t_{d1}) \tag{3.5}$$

t_{d1} and t_{d2} are the departure times (from the sender) of the first and the second probe packet respectively.

An advantage of using the packet pair/train method is that in theory a single correct measurement of the dispersion (thus from only a single packet pair) could suffice to obtain the path capacity. However, due to cross traffic and other processes that might disturb the dispersion between packets, probing with multiple packet pairs/trains will be necessary in practice to obtain a valid value for the dispersion caused by the bottleneck link. One then assumes that the cross traffic has stochastic properties, so if one probes often enough, one of the pairs has been lucky and has not been disturbed by cross traffic.

A major problem of packet pair probing is that of the "post-narrow link" problem [7]. If a faster link is present in the path after the narrow link, then cross traffic in this "post-narrow link" can cause the dispersion to become less. This happens when the first packet of the pair is obstructed by cross traffic. This gives the second packet the opportunity to catch up with the first packet. Therefore the dispersion will decrease and the capacity will be overestimated. Because of this phenomenon one cannot simply use the minimum measured dispersion as the correct result of the measurement. The "post-narrow link problem" is illustrated in Figure 3.2.

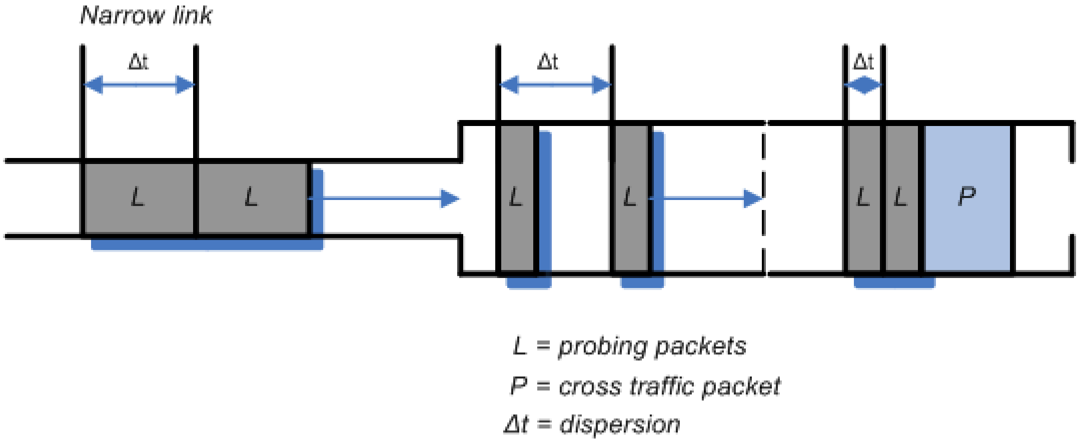

Figure 3.2: Decreased dispersion due to post-narrow link

Various other problems have been addressed in literature concerning packet pair probing [19]. One problem arises from non-FIFO (first-in-first-out) queuing in network devices. Then it can happen that the second packet of the pair arrives at the destination before the first packet of the pair. Similar problems could be introduced by "multichannel" bottleneck links [19] such as double-line-ISDN (integrated services digital network). If the two packets of a pair are transmitted over different channels, then the measured dispersion is not representative for the capacity of the combined channels. Other problems mentioned are limitations due to clock resolution and varying bottleneck bandwidth.

Packet pair probing forms the basis for capacity measurement in several existing tools. Especially *CapProbe* ([15], [16]), *Adhoc probe* [38] and *Wbest* [18] are relevant to mention at this point, since these tools perform capacity measurements over paths including wireless links.

CapProbe is one of the first tools that claims to measure paths including wireless links. It uses packet pair probing to perform either a sender-based (by forcing a reply from the receiver using Internet Control Message Protocol (ICMP) packets) or a receiver based (based on UDP) capacity measurement. If either packet of a pair is delayed due to cross traffic, then the sum of the measured delays of both packets will increase. By "minimum delay sum" filtering, the tool aims to filter out dispersion measurements that were distorted due to cross traffic. Additionally a convergence test is implemented that should improve the accuracy of the results.

AdHoc Probe is a receiver-based capacity measurement tool that uses CapProbe's probing method. The receiver-based version of the *CapProbe* probing method is used because the round-trip mode of *CapProbe* is inadequate in ad-hoc wireless networks. *AdHoc Probe* does not implement CapProbe's "convergence test", in order to keep the algorithm simple and fast. Similar to *CapProbe* it uses "minimum delay sum" filtering to filter out dispersion measurements that were distorted by cross traffic.

WBest is a receiver-based available bandwidth estimation tool based on the PGM. It aims to measure available bandwidth over paths with a wireless link as last hop. Before the available bandwidth measurement, *WBest* performs a very simple capacity measurement. For this purpose a train of at least 6 probe packets is sent to the destination node. In the destination node the capacity of the narrow link (usually the wireless link) is estimated using the median of the measured dispersion between the packets in the packet train. Since the narrow link is assumed to be in the last hop, the post-narrow-link problem will not exist.

3.2.2 Measuring available bandwidth

In section 3.1 three active probing concepts for measuring available bandwidths were briefly mentioned. They will be explained below.

Link idle time measurement

If a probe packet experiences higher than minimum OWD, the channel was probably busy sending probe traffic. This is the idea on which the link idle time estimation *ProbeGap [32]* is based. The sender sends a series of Poisson-spaced probes. 200 probe packets are sent with a size of 20 byte each over a time interval of 50 seconds. After probing the network, an algorithm performs a search for a turning point in the measured OWDs. Packets with delays below the turning point are assumed to have passed an idle link. For longer OWDs it is concluded that the link was busy. The knee in the cumulative distribution function (CDF) of OWD samples identifies the fraction of time that the channel is idle. If no turning point is found, the link is assumed to be 100% busy, and no bandwidth is available. The idle time fraction f_{idle} is multiplied by the narrow link's capacity to obtain the available bandwidth R_{AB}. Thus:

$$R_{AB}=f_{idle}\times C_{l,n} \tag{3.6}$$

To use this method for available bandwidth estimation, the narrow link capacity should be known in advance.

Probe rate model (PRM)

Probe rate model (PRM) is sometimes referred to as 'packet rate method' or 'packet rate model'. PRM based tools are based on the observation that the average rate of incoming probe traffic at the destination will be equal to the average rate of outgoing probe traffic at the source as long as the probing rate is lower than the available bandwidth in the path. As soon as the probing rate exceeds the available bandwidth in the path, the observed delays of probing packets and dispersion between probing packets will increase. The available bandwidth of a path can thus be determined by determining the point where probing packets delays or dispersions start showing different characteristics.

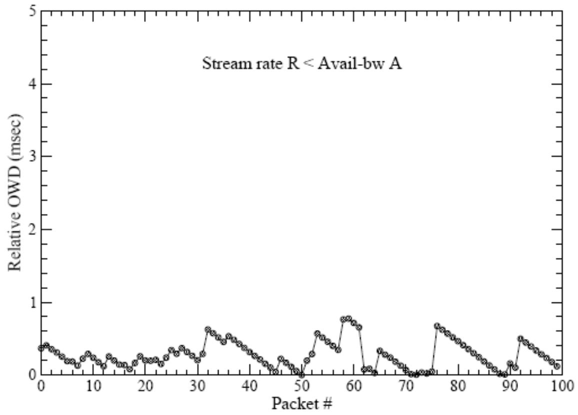

Figure 3.3: Measured OWDs for probing packets at a rate lower than the available bandwidth (source: [25])

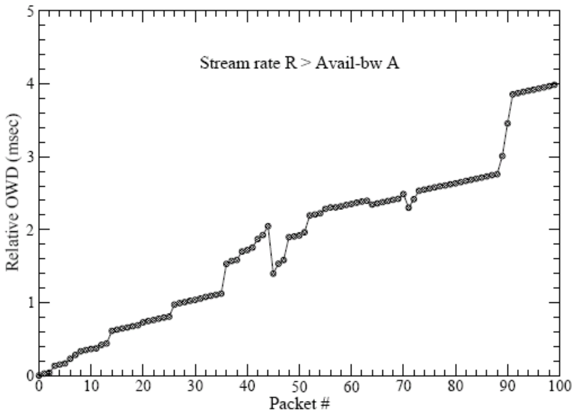

Figure 3.4: Measured OWDs for probing packets at a rate higher than the available bandwidth (source: [25])

The figures 3.3 and 3.4 show the OWDs of packets for different probing rates. In Figure 3.3 the probing rate is lower than the available bandwidth in the path. Therefore the overall trend is non-increasing OWDs. In Figure 3.4, the probing rate is larger than the available bandwidth. Therefore the overall OWDs show an increasing trend.

PRM based tools have several disadvantages. Since the turning point has to be found using varying probing rates, the amount of probing needed is often relatively large compared to other probing methods. Additionally, the network has to be probed with traffic rates higher than the available bandwidth. This implies that these tools are often very intrusive and obstructive to ongoing traffic in the network. Many existing available bandwidth measurement tools are based on PRM. *DietTOPP [26]* and *SLOT [39]* are two tools based on PRM which have been developed with special attention to measurement of paths including wireless links.

For several probe rates between r_{min} and r_{max}, *DietTOPP* sends multiple packet trains with equally sized packets. The tool observes the sending rate r_{in} and the receiving rate r_{out} of probe packets. If the ratio of r_{in}/r_{out} is plotted against r_{in}, then a turning point of the slope will be visible at the rate where r_{in} equals the path available bandwidth.

SLOT is a tool developed for ad hoc wireless networks. It is a combination of *TOPP [27]* and *SloPS [40]* and claims to be faster and less intrusive then either of the two techniques it is based on.

Probe gap model (PGM)

The PGM, also referred to as probe rate method, is used to measure the amount of cross traffic in the narrow link. If the path is probed at the rate of the path capacity, the probe packets will queue back-to-back in the narrow link. Cross traffic packets that interfere with the probe packets will cause the dispersion between probe packets to increase. If the output dispersion is twice the input dispersion, then the cross traffic rate is equal to the probing rate and thus equal to the narrow link capacity. Thus~~erefore~~ the amount of cross traffic can be inferred from the additional dispersion measured between the probe packets using the following formula.

$$R_c=(\frac{\Delta t_{out}}{\Delta t_{in}}-1)\cdot C_{l,n} \tag{3.7}$$

where R_c is the cross-traffic rate, Δt_{in} is the dispersion between probe packets at the source node, Δt_{out} is the dispersion between the probe packets measured at the destination node. The PGM concept is illustrated in Figure 3.5.

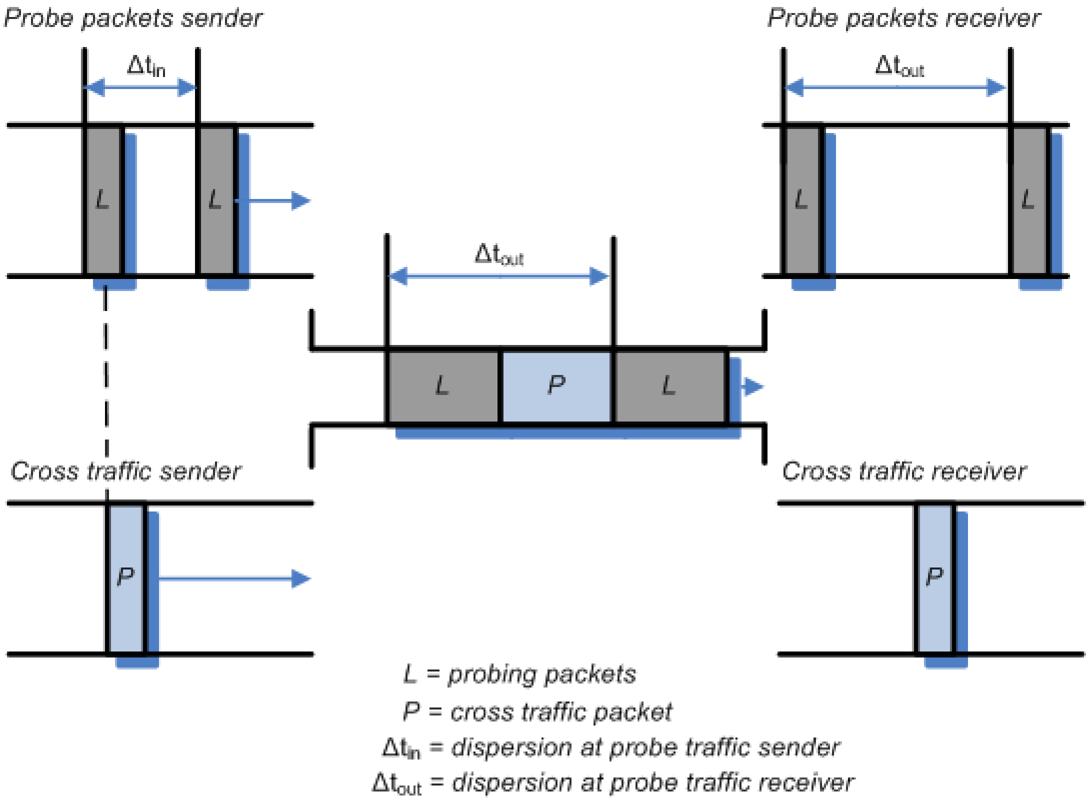

Figure 3.5: Illustration of the "probe gap model"

In the picture three consecutive links are shown. Packets are sent at the rate of the "narrow link". Without cross traffic Δt should not increase in the "narrow link" since it is able to handle packets at this rate. However, Δt is increased due to cross traffic which causes additional delay between two consecutive probe packets. From analysis of measured dispersions at the receiver the amount of cross traffic can be determined. Subsequently, the cross traffic rate R_c can be subtracted from the narrow link capacity $C_{l,n}$ to obtain the available bandwidth in the narrow link.

$$R_{AB}=C_{l,n}-R_c=C_{l,n}\cdot(2-\frac{\Delta t_{out}}{\Delta t_{in}}) \tag{3.8}$$

For the PGM to be valid it is assumed that the narrow link is also the tight link, sometimes called the "super-bottleneck link" [18]. Additionally, the narrow link capacity should be known beforehand. This means that it will often be necessary to perform a capacity measurement in advance of the PGM based available bandwidth estimation.

Since available bandwidth measurement based on the PGM is not an iterative process, we expect this method to converge faster and to be less intrusive compared to the PRM. Although PGM requires the path to be probed at the rate of the narrow link capacity, the intrusiveness can be kept relatively low by using short bursts of probing packets.

Available bandwidth measurement based on PGM has disadvantages also. Erroneous measurement results could be obtained if the narrow link is not the same as the tight link.

Wbest [18] is based on the PGM. Before the actual available bandwidth estimation the tool performs a very simple capacity estimate of the path, as is described in section 3.2.1. After the capacity estimation, *WBest* sends a sequence of probe packets at the rate of the previously measured capacity and estimates the amount of cross traffic from the measured dispersions at the destination based on formula 3.7. Further, the tool is receiver based only and it assumes the wireless link to be the ‘super bottleneck link’ and the last hop in the path.

3.3 Shortcomings of existing tools

In the previous sections is shown that there exist already a lot of tools aimed at measuring network bandwidths. However, none of the existing tools seems to comply to all our requirements, i.e. suitability for (small scale) heterogeneous networks, low intrusiveness, short convergence times and sender based.

Most of the bandwidth measurement tools that were developed previously where designed for measuring Internet paths. This means that these tools where not designed with heterogeneity of networks in mind (e.g. including wireless links and HomePlug AV). And most of these tools have not been evaluated in scenarios consisting of heterogeneous paths. Therefore there is only very limited information available about the performance of the tools under these conditions. Even in the case that tools have been evaluated in scenarios containing wireless links, the evaluation is often limited to accuracy and not considering convergence times or intrusiveness of the tools ([41], [42], [43], [26]).

It has been shown ([44], [45]) that especially PRM based tools tend to have long convergence times, from tens of seconds up to several minutes, even in “all wired” scenarios, i.e. scenarios without any wireless links. Additionally these tools need to probe the network with higher rates than the available bandwidth. It is therefore concluded that using PRM based methods will not provide a satisfactory solution for an available bandwidth estimation tool that will comply with all our requirements. In particular being low intrusive and providing short convergence times.

On the other hand, various experiments have indicated that PGM-based tools can potentially provide results within the order of seconds while not being intrusive. However, the only PGM-based tool aimed at measuring available bandwidth in heterogeneous paths is *WBest*. *WBest* has the limitations that the tool needs cooperation of both sender and receiver and that the “superbottleneck link” is assumed to be in the last hop of the path. Therefore the tool does not need to cope with the “post-narrow-link problem”. Further, PGM-based available bandwidth estimation needs knowledge in advance about the bottleneck capacity. Also *ProbeGap*, which uses bottleneck idle time estimation to obtain an estimate of the available bandwidth, is receiver based and needs knowledge in advance about the narrow link capacity.

Therefore, to use either a PGM based tool or a “link idle time measurement” based tool, it will be necessary to perform a path capacity measurement first. Additionally it will be necessary to develop a tool based on packet round-trip-time (RTT) measurement in stead of OWD measurements.

In section 3.2.1 two different concepts of path capacity measurement were presented. To use VPS-based measurement for measuring the bottleneck capacity, it is necessary to measure all links in the path. On the other hand, packet pair probing needs only a single correct measurement of the packet pair dispersion to obtain the bottleneck capacity of a path. For both probing schemes it will be necessary to send multiple probes into the network to be able to filter inconsistencies caused by cross traffic. Further, it was mentioned that VPS probing will provide the maximum physical rate of the links in the path, packet pair or packet train measurements provide throughputs at higher layers. Because of these reasons, packet pair/train based capacity measurement seems to be the best approach to develop a tool that complies with our requirements.

Therefore the rest of this work will be dedicated to the development of a packet pair/train capacity estimation tool which is based on packet RTTs and copes with heterogeneous home networks. The following step, to obtain the path available bandwidth, will be left for further research.

4 Methods and tools

This chapter provides background information on the various methods and tools that have been used to perform the research. Section 4.1 provides a brief overview of several networking link layer technologies. Additionally we analyze the theoretical IP layer capacities of Ethernet and 802.11b links. Section 4.2 explains IP fragmentation. Section 4.3 describes how the simulated networks are implemented in OPNET modeler. The settings necessary to reproduce the results are given in this section. In section 4.4 is described how simulation results are filtered in order to remove irrational outliers. This is necessary because the outliers significantly distort the mean outcome and standard deviation of the results.

4.1 Home networking link layer technologies

Many different technologies can be used in order to interconnect devices within the home. What we see in today’s typical home networks is that only a few link layer technologies are dominantly present. Especially “twisted pair Ethernet” (over UTP cables) and 802.11 wireless links are ubiquitous and to a lower extend HomePlug AV and HomePNA are found regularly in home networks. This section provides a brief discussion on the impact of the MAC and physical layers on the throughput of higher layer data. Therefore in section 4.1.1 a brief overview is provided of the Ethernet protocol followed by an analysis of the impact of the Ethernet MAC on the IP layer throughput. Subsequently, in section 4.1.2, a brief overview is provided of the 802.11 protocol together with a simple analysis of impact on higher layer throughput. Finally, section 4.1.3 discusses other typical home network link layer technologies. Analysis of the impact on higher layers will be omitted here, because these technologies are not further researched in this work.

4.1.1 10BASE-T/100BASE-T(X) Ethernet ([46], [47])

Originally, Ethernet was designed as a shared medium. All devices in the network were connected, initially, through a single (coaxial) cable (10BASE5). Later the coaxial cable was substituted by “twisted pair” cables and hubs (BASE-T(X)). Because of the shared nature, a scheme known as carrier sense multiple access with collision detection (CSMA/CD) was incorporated into the MAC. The shared medium and collision detection schemes would make an analysis of the higher layer throughput rather complex. However, today's Ethernet networks are generally fully switched with full duplex links. Therefore Ethernet cannot be regarded as a shared medium anymore. In our throughput analysis we will not consider Ethernet to be a shared medium.

Figure 4.1 is an illustration of the typical overhead introduced by the transmission of data using UDP over IP over Ethernet. In addition to the payload (including IP and UDP headers) an Ethernet frame contains a MAC header, typically 14 bytes in size. Further, the frame contains a 7 byte preamble, a 1 byte start frame delimiter (SFD) and a 4 byte frame check sequence. Finally, the Ethernet requires a 96 bit inter frame gap (IFG) after each frame. In total, the Ethernet MAC and lower layers add an additional 34 bytes or 272 bits to the IP layer data for each Ethernet frame.

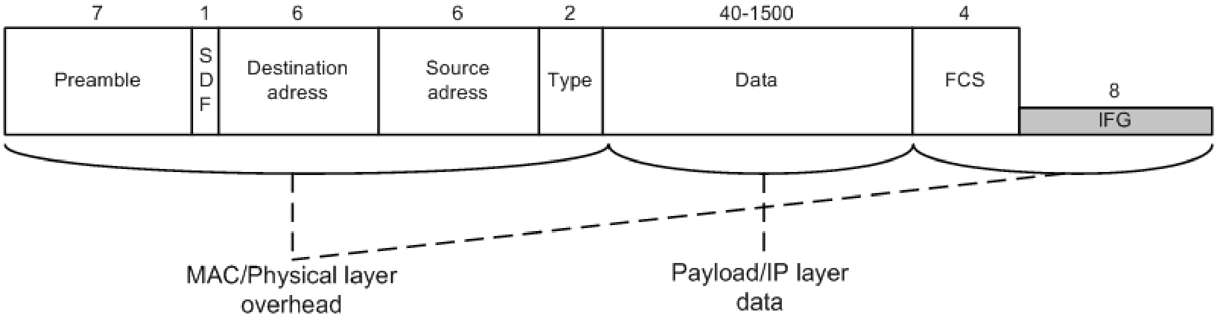

Figure 4.1: Ethernet overhead

Ethernet capacity analysis

Since the maximum payload in an Ethernet frame consists of 1500 bytes, the maximum achievable throughput of an Ethernet link can be calculated as following:

$$C_l = \frac{MACpayload}{MACpayload + overhead} \times R_{ph} \tag{4.1}$$

where C_l is the link capacity and R_{ph} is the nominal physical data rate of the Ethernet link. Because the MAC and physical overhead introduced by Ethernet consists of 272 bits in each Ethernet frame the expected IP layer capacity of the 10BASE-T and 100BASET(X) Ethernet links are 9.78 Mbps and 97.88 Mbps respectively.

4.1.2 802.11b

802.11b [48] is a standard for “wireless local area network” (WLAN) communication that operates in a frequency band around 2.4 GHz. The maximum specified data rate at the physical layer is 11 Mbps. In case of reduced channel conditions, lower data rates (5.5, 2 and 1 Mbps) can be used to improve signal to noise ratio. Therefore the channel capacity can fluctuate over time. However, rate adaptation mechanisms are not specified by 802.11 standards.

Analysis of 802.11b wireless network capacity has been done previously ([49], [50]). In this section we analyze the IP throughput based on the default settings for the 802.11b protocol in OPNET modeler.

The calculation of MAC and physical layer overhead for 802.11b is more complex than for Ethernet. Since all devices in a typical home network operate on a single wireless channel, the wireless medium can be generally regarded as a shared medium. Therefore the distributed coordination function (DCF), which is most commonly used to control media access on 802.11 channels, incorporates a so called carrier sense multiple access/collision avoidance (CSMA/CA) scheme. The total overhead of the MAC and physical layers added to each IP packet is illustrated in Figure 4.2.

In addition to the (IP layer) payload an 802.11b frame contains a MAC header, which is 24 bytes long for communication between a station and an AP. Additionally there is a frame check sequence (FCS) field of 4 bytes long. Further, the frame contains a physical layer convergence protocol (PLCP) preamble and a PLCP header of 144 bits and 48 bits long respectively. In contrast to Ethernet communication, all frames in 802.11b are acknowledged through an ACK frame. The ACK frame is preceded by a short interframe space (SIFS). The ACK also contains a PLCP preamble and a PLCP header. Additionally the ACK frame contains 14 bytes of (control) data.

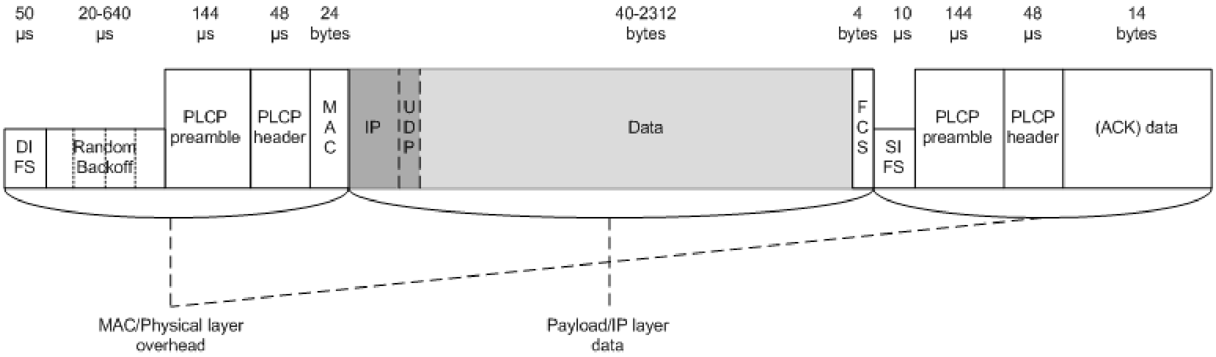

Figure 4.2: Overhead of 802.11b under DCF operation

From the figure it can be seen that the overhead added by the MAC and physical layers in 802.11b is much larger compared to the overhead in Ethernet. The CSMA/CA scheme is included in the media access control to avoid unnecessary bandwidth consuming retransmissions of 802.11 frames. In CSMA/CA a station listens to the channel for at least a period of DIFS (DCF interframe space). If the channel is idle during the full DIFS period then the station starts transmission directly after the end of the period. If the medium was sensed busy, then the station defers from transmission. After a busy period of the channel there is a relatively large probability that multiple stations have data to be transmitted. Only after an idle period of DIFS plus an additional random backoff period the station can start transmission. The random backoff period is included in order to lower probability that multiple stations start transmission simultaneously.

A more detailed description of all the fields in the figure can be found in [51].

802.11b single hop throughput analysis

With “single hop” is meant that there is only one wireless link in the path between the source device and the destination device. This could be for example the path between the media server and the network media player in the example scenarios described in section 1.2. For the 'single hop' maximum throughput analysis we assume that packets are sent in a long burst. Therefore the subsequent packets will be in the buffer, ready to be transmitted, before the transmission of the previous packet has been completed. Because the next packet is ready before completion of the previous packet there will always be a contention phase between the transmission of subsequent packets. Since there are no other devices communicating over the channel there will be no collisions. The average random backoff between subsequent packet will therefore be half of the contention window (CW) size. In 802.11b the default CW size is 31 time slots. A single time slot has a duration of 20µs. For the calculation we assume the use of the 'long preamble'. Therefore the total duration of the PLCP preamble and PLCP header is 192μs. Control information (ACK, PLCP preamble and header) is sent at a rate of 1 Mbps. Finally we assume that the link operates at the highest data rate of 11 Mbps. The calculation for the average duration of a frame transmission looks as follows:

DIFS		50μs
Avg. RB	$((1/2\times \mathrm{CW})\times \mathrm{slot\_time})$	310μs
PLCP*preamble*	$(144\,\mathrm{bits}/1\,\mathrm{Mbps})$	144μs
PLCP*header*	$(48\,\mathrm{bits}/1\,\mathrm{Mbps})$	48μs
MAC header and FCS	$((8\times 28\,\mathrm{bytes})/11\,\mathrm{Mbps})$	20μs
Payload	$((8\times 1500\,\mathrm{bytes})/11\,\mathrm{Mbps})$	1091μs
SIFS		10μs
PLCP*preamble*	$(144\,\mathrm{bits}/1\,\mathrm{Mbps})$	144μs
PLCP*header*	$(48\,\mathrm{bits}/1\,\mathrm{Mbps})$	48μs
ACK Data	$((8\times 34\,\mathrm{bytes})/1\,\mathrm{Mbps})$	112μs
TOTAL		**1977μs**

Thus the average dispersion between packets containing 1500 bytes of data is 1977 μs. Using formula 3.3 we get for the link capacity:

$$C_l = 1500\,\mathrm{byte}/1977\,\mu\mathrm{s}\,s = 6.070\,\mathrm{Mbps}$$

This is close to the values following from simulations performed in [49], [52], and [53] that vary between about 6.1 Mbps and 6.3 Mbps.

802.11b two hop throughput analysis

“Two hop” refers to a path that contains two wireless links between the source device and the destination device. Packets sent from the source devices to the destination device will thus travel the wireless medium twice. For example this could be the path in the example scenarios of section 1.2 between the laptop and the network media player, where the laptop will communicate with the media player through the AP.

For the throughput calculation of a path including two wireless links we make an assumption to simplify the calculation. The assumption is that the transmit buffers of the wireless devices will never be empty. Therefore both data transmitting stations, i.e. the source station as well as the intermediate AP, will always have a packet to transmit and will therefore always take part in contention for the medium. The average random backoff for this situation is determined with the use of a MATLAB script. This script simulates 10,000,000 contention phases with two nodes contending for media access. The code of the script can be found in appendix A. From the simulation we found that 303,307 times a collision occurred. Therefore $10{,}000{,}000-303{,}307=9{,}696{,}693$ times a packet was successfully transmitted. The average random backoff in the contention phases was 8.47 time slots or 169 μs. Therefore the calculation for the average duration of the successful transmission of a frame $\overline{d}_{suc}$ looks as follows:

DIFS		50μs
Avg. RB	(from MATLAB simulation)	169μs
PLCP*preamble*	$(144\,\mathrm{bits}/1\,\mathrm{Mbps})$	144μs
PLCP*header*	$(48\,\mathrm{bits}/1\,\mathrm{Mbps})$	48μs

MAC header and FCS	$((8\times 28\,\text{bytes})/11\,\text{Mbps})$	20µs
Payload	$((8\times 1500\,\text{bytes})/11\,\text{Mbps})$	1091µs
SIFS		10µs
PLCP*preamble*	$(144\,\text{bits}/1\,\text{Mbps})$	144µs
PLCP*header*	$(48\,\text{bits}/1\,\text{Mbps})$	48µs
ACK Data	$((8\times 34\,\text{bytes})/1\,\text{Mbps})$	112µs
TOTAL		**1836µs**

The average duration for a packet that has to be retransmitted takes a bit longer. This is because a device waits for the ACK_timeout (acknowledgement timeout) before it decides that a packet was not transmitted successfully. The duration of the ACK_timeout is defined through the extended interframe space (EIFS) and the DIFS. The calculation for the duration of the EIFS can be found in [48]. In this case EIFS is 264 µs. The ACK_timeout is equal to EIFS - DIFS = 214 µs. The average duration for a packet retransmission $\overline{d_{ret}}$ is 1836 µs + 214 µs = 2050 µs. Therefore, the calculation for the average duration for the transmission of packets over the double wireless link $\overline{d_{all}}$ looks as follows:

$$\overline{d_{all}} = \frac{9{,}696{,}693}{10{,}000{,}000} \times \overline{d_{suc}} + \frac{303{,}307}{10{,}000{,}000} \times \overline{d_{ret}} = 1843\,\mu s$$

Therefore, using formula 3.3. we get for the capacity of the wireless channel:

$$C_{ch} = 12000\,\text{bits}/1843\,\mu s = 6.511\,\text{Mbps}$$

Thus the maximum capacity of the wireless channel is 6.511 Mbps. As described in section 2.2, if a path between two devices contains n wireless hops, then each packet has to be transmitted n times over this medium to get from the source device to the destination device. Using formula 3.3 we find for the narrow link capacity in the “two hop” path:

$$C_l = C_{ch}/2 = 3.256\,\text{Mbps}$$

4.1.3 Others

Many other home networking link layer technologies exist today while several new link layer technologies are currently under development. An overview of home networking technologies can be found in [54]. This chapter gives an overview of several commonly used technologies. Although we did not include any of the technologies described in this section in our current research, the ultimate goal of the project is to develop an estimation tool that operates regardless of the link layer technology used. However, there are many problems to be solved before this goal is achieved.

802.11a

802.11a [55] is a wireless communication standard that differs from 802.11b on the physical layer. While 802.11b offers maximum physical layer rates of 11 Mbps and operates around 2.4 GHz, 802.11a offers a maximum data rate at the physical layer of 54Mbit/s and operates around 5GHz. Other supported rates are 48, 36, 24, 18, 12 and 6 Mbps. The higher data rates are possible due to orthogonal frequency division multiplexing (OFDM). Advantages of 802.11a over 802.11b are the higher data rates supported as well as the operation in the not so busy 5GHz band. Since it operates at this frequency there will be less interference from other devices such as cordless phones, microwave ovens and bluetooth devices. However, since the standard operates at a higher frequency, 802.11a devices will generally have a smaller coverage area than 802.11b devices.

802.11g

802.11g [56] is the successor of the 802.11b standard. Just like 802.11a, it offers maximum data rates at the physical layer of 54 Mbps by making use of OFDM. Like 802.11a, 802.11g supports data rates of 54, 48, 36 24, 18, 12 and 6 Mbps using OFDM modulation. Additionally the support of 802.11b data rates using accompanying modulations is mandatory for 802.11g devices. Like 802.11b, 802.11g operates around the 2.4 GHz frequency. Further, the standard offers backward compatibility with 802.11b. When 802.11g devices are deployed in a WLAN consisting of both 802.11b and 802.11g devices, the performance of the 802.11g devices will be limited.

802.11n

802.11n [57] will be a new standard for wireless communication that will operate around the 2.4 GHz band. The 802.11n standard will support physical layer data rates up to 600 Mbps combining OFDM, multiple input multiple output (MIMO) technology and channel bonding. MIMO technology enables the use of spatial multiplexing.

Eight different data rates are specified for each transmitter. With a maximum of 4 transmitters in use, the number of different data rates can be up to 32. Additonally, different spatial streams can use different modulations. Therefore it is possible, in theory, to use dozens more different data rates [58].

Although the final 802.11n standard has not yet been approved, there are already “Pre-N” devices available on the market that are based on a draft version of the standard.

Gigabit Ethernet

Gigabit [47] Ethernet refers to a set of standards that enables communication with data rates at the physical layer of 1 Gbps. Just like the Ethernet technologies described in section 4.1.1. Gigabit Ethernet supports half-duplex communication using hubs, although the common implementations of Gigabit Ethernet networks use full duplex switches. Like previous Ethernet standards, Gigabit Ethernet is defined for several different types of cabling, but the most commonly implemented technology in home networks is the Gigabit Ethernet over twisted pair cabling (1000BASE-T(X)).

HomePlug 1.0 and HomePlug AV ([59],[60])

The HomePlug Powerline alliance is an alliance of multiple companies that has as its purpose to provide high-quality networking over existing AC wiring within the home. The original HomePlug 1.0 specification allows half duplex operation with data rates at the physical layer of 14 Mbps. The newer HomePlug AV specification is an enhancement of the specification allowing full duplex communication [61] with maximum data rates of 200 Mbps (the capacity is shared by the flows in the two directions) at the physical layer using OFDM modulation. The 200 Mbps maximum theoretical data rate is for a channel where only 917 (out of a maximum of 1155 carriers) of the OFDM symbol carriers are used. Because certain carriers interfere with licensed bands, 917 is the number of carriers that are used in North America. If it were possible to use all 1155 carriers, then the theoretical maximum throughput is about 248 Mbps [62]. (Asymmetrical) full-duplex communication is possible by assigning parts of the carriers for the different communication direction.

Because of the heavily fluctuating noise, the capacity of the channel can be adapted several times within a single AC line-cycle period. A line-cycle period lasts about 17 ms or 20 ms in 60 Hz or 50 Hz areas respectively.

4.2 IP fragmentation

IP fragmentation provides a solution for the case where packets are sent that are larger than the maximum transfer unit (MTU) supported by the link layer technology. IP fragmentation is especially useful for devices such as routers that interconnect different link layer technologies. The MTU for the link layer at one side of the router can be different from the MTU at the link on the other side of the router. If a packet received at the incoming link is larger than the MTU of the outgoing link, then the payload of the IP packet is fragmented into two or more smaller IP packets. This is illustrated in Figure 4.3. Additionally the original IP header is copied into all fragments. A special bit is set in the flag field of the IP header to indicate that the fragments are separated parts of a larger IP packet.

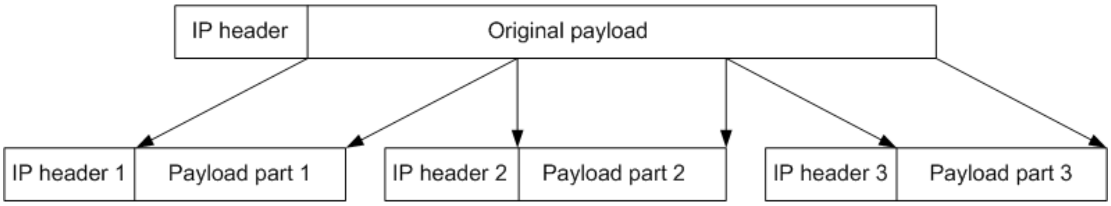

Figure 4.3: Illustration of IP fragmentation

If one of the intermediate devices in a path does not support IP fragmentation, then larger-than-MTU-sized packets will not be forwarded. Instead, the intermediate device will drop the packet and possibly reply with a ‘packet too large’ error message to the source node.

4.3 Implementation in OPNET

OPNET [63] technologies are widely used in industry and academic environments for simulating (communication) networks. They can be used for application performance management, network planning, engineering and operations and network R&D. For our works we had the “OPNET modeler” package to our disposal including the Wireless suite. In this section we describe the configurations and settings of the simulation environment.

4.3.1 Choices for implementation

For the implementation of the simulated networks we choose to use as much as possible standard available models and settings. We use the default Ethernet switch (*"ethernet4_switch_adv"*) and access point (*"wlan_ethernet_router_adv"*) models. For the cross traffic sender and cross traffic receiver(s) we use the default workstation (*"ethernet_wkstn_adv"* and *"wlan_wkstn_adv"*) models available in OPNET. The cross traffic is modeled with the use of the standard available IP traffic demand configuration (*"ip_traffic_flow"*) model. Finally all nodes within the network models are connected through the standard available 10M Ethernet, 100M Ethernet and 802.11b WLAN link models. An example is given in Figure 4.4. The figure shows how the network for the "two hop wireless network" as described in section 4.3.3 is implemented in OPNET. The blue line between the "cross traffic generator" and "wireless receiver" represents an "IP traffic flow". This "IP traffic flow" configuration object is used to specify the cross traffic in the network. Note that "switch 2" and the "cross traffic receiver (wired)" will not participate in this example configuration since there is no traffic flow to these devices.

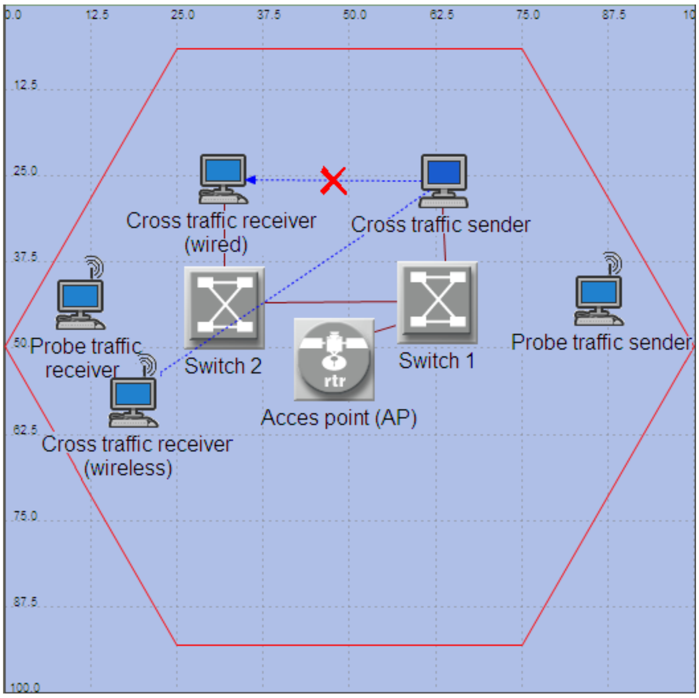

Figure 4.4: Example network in OPNET modeler

Only the node models for the 'probe traffic sender' and 'probe traffic receiver' are modified instances of the standard OPNET workstation models. How these models are implemented is described in section 4.3.2.

Three different types of cross traffic are specified for the the simulations. The settings for the three cross traffic types are shown in Table 1.

	Packet interarrival time	**Packet size distribution**
Type 1	Exponential	1500 byte
Type 2	Exponential	40-1500 byte uniformly distributed
Type 3	Exponential	300 byte

Table 1: Different cross traffic types

The rate parameter λ of the exponential distribution is deduced from the specified average $\bar{r}_d$ data rate and the average packet size $\bar{L}_p$ using the following relation:

$$\lambda = \bar{r}_d / \bar{L}_p \tag{4.2}$$

For the analysis of the capacity estimation tool it is necessary to monitor the round-trip times (RTTs) of all packets. From the RTTs we can calculate the dispersion between the packets. In order to be able to analyze the full round-trip process of packets, we also monitor the one-way delays or end-to-end delays (ETEs) of the packets; both from sender to receiver and vice versa.

4.3.2 OPNET models

For the implementation of the “probe packet sender” and “probe traffic receiver” we modify the existing “ethernet_wkstn_adv” and “WLAN_wkstn_adv” models found in OPNET modeler. We connect a single module to the UDP modules of these node models. Subsequently we specify the behavior of the new modules with our own process models. In this case the process model in the “probe traffic sender” receives and generates packets that are sent to the UDP module. The size and departure times of these packets can be specified through the node attributes. We also implemented a process in which the probe traffic receiver receives the probe packets that were send from the probe traffic sender. After the reception of a probe packet the node directly replies with an ICMP-packet sized UDP packet. Also the size of these reply packets can be specified through the node's attributes. We use UDP packets instead of ICMP because the ICMP protocol definitions are not fully implemented in OPNET modeler by default. We expect that the small UDP packets will behave similar to ICMP destination unreachable packets and thus will experience equal delays.

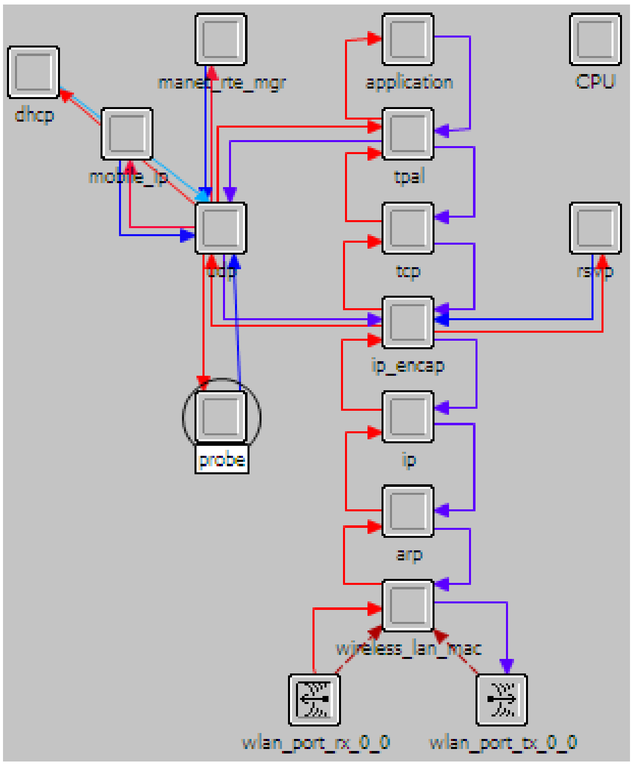

Figure 4.5: Example of a node implementation in OPNET modeler. The functionality of a node is fully described through several interconnected modules.

Since the newly added module ("probe") is connected to the UDP module of the existing node model, the new module only has to create data packets that represent the payload of UDP packets. Additionally this module should specify the source and destination addresses of the packets. An example of a modified node model that is built out of different modules is shown in Figure 4.5. When the packet is sent, the UDP, IP, and lower layer headers are added automatically while the packets pass through the 'UDP', 'ip_encap', 'ip', 'arp', 'wireless_lan_mac' and 'wlan_port_tx_0_0' modules. If a packet is received, all lower layer headers are removed and only the payload of the UDP packet is sent to the 'probe module'. The process model in the 'probe module' defines how this payload is processed. An example of a process model that describes the functionality of a single module is shown in Figure 4.6. The figure shows how a process is defined through a state diagram.

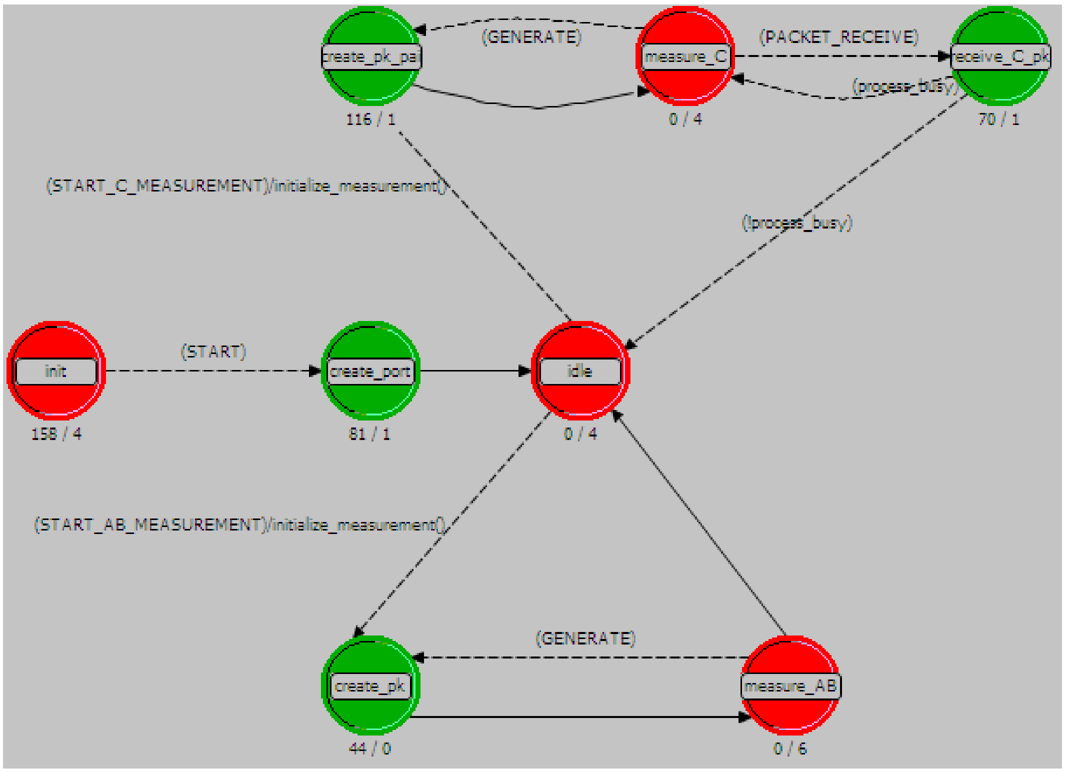

Figure 4.6: Example of a process model. A module's functionality is described through a process model.

4.3.3 Simulated networks

All wired network

Three different types of networks were simulated in order to verify the performance of our probing tool. We refer to the three different networks as the "all wired network", the " single wireless hop network" and the "double wireless hop network". The all wired network consists of six devices that are all connected via Ethernet links as shown in Figure 4.7. The two switches are interconnected through a 10M Ethernet link. All other connections consist of 100M Ethernet links. Therefore the link between the two switches is the "narrow link" in the path between the "probe traffic sender" and the "probe traffic receiver". The traffic from the "probe traffic sender" to the "probe traffic receiver" is disturbed by sending cross traffic from the "cross traffic sender" to the "cross traffic receiver".

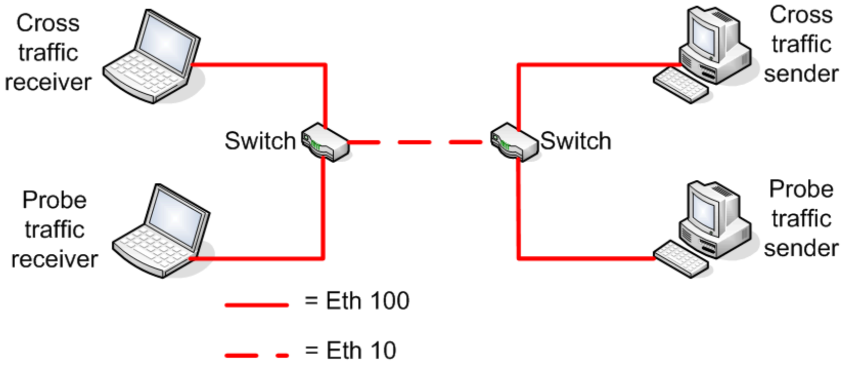

Figure 4.7: Simulated Ethernet network (all wired)

Single wireless hop network

The single wireless hop network consists of 802.11b links and 100M Ethernet links, as shown in Figure 4.8. Both the probe traffic receiver and the cross traffic receiver are connected with the AP through 802.11b (wireless) links. The probe traffic sender and the cross traffic sender are connected to the access point (wireless router) through 100M Ethernet links. The 802.11b link connecting the access point with the probe traffic receiver forms the "narrow link" in the path between the probe traffic sender and the probe traffic receiver.

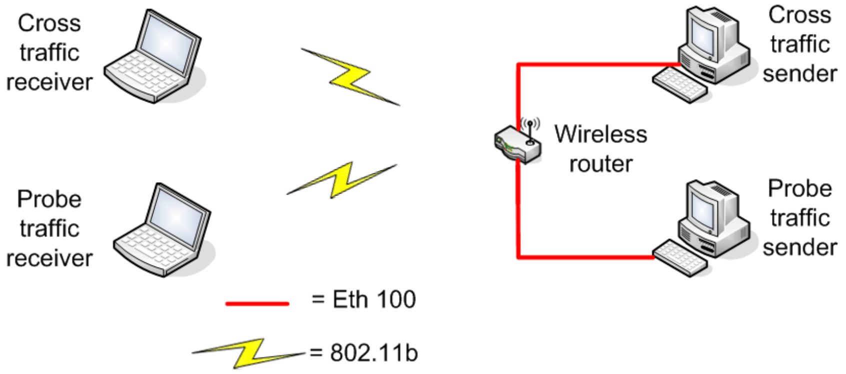

Figure 4.8: Simulated network with 802.11 bottleneck link ("single wireless")

Since the wireless medium is a shared medium, the "probe traffic receiver" and the "cross traffic receiver" share the capacity of the wireless medium.

Double wireless network

In the "double wireless network", two wireless links are present in the path between the probe traffic sender and the probe traffic receiver. Only the cross traffic sender is connected through a 100M Ethernet link to the access point (wireless router). Due to the complex nature of the 802.11 MAC, we expect this scenario to be the most challenging of the three.

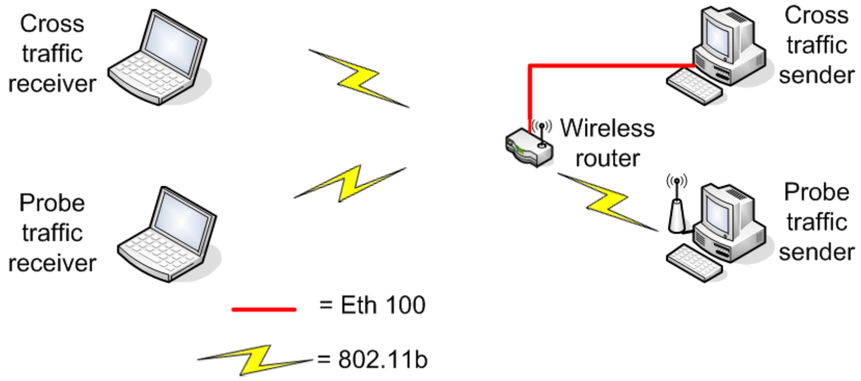

Figure 4.9: Simulated network with two 802.11 bottleneck links ("double wireless")

Since the wireless medium is a shared medium, the "probe traffic receiver", the "probe traffic receiver" and the "cross traffic receiver" need to share the capacity of the wireless medium.

4.4 Filtering of outliers

To evaluate the performance of the measurement algorithms, we analyze the mean outcome and variance for 50 measurements. Because some of the outcomes turn out to deviate several orders in magnitude from the outcomes of most of the other measurements, they significantly impact the mean outcome and the variance of the measurements. To filter out outliers like these, we evaluate only those outcomes that fall into a 95% confidence interval.
For this purpose we use only those 95% of the samples, in this case 48, that cover the smallest range of values. In this way the most extreme (most deviating) outcomes, including outliers, will not be used in the analysis of the performance. The filtering is illustrated in Figure 4.10.
In the picture we see the histograms for the estimated capacities for two different scenarios, using the peak estimation method as described in chapter 6. The histogram in Figure 4.10a has some severe outliers. Histograms b and c illustrate which values are filtered out using our filtering method with 95% and 90% confidence intervals. Let $\tilde{f}(x_j)$ be the relative frequency per unit interval as given in Figure 4.10, but normalized to one. Thus, $\sum_j \tilde{f}(x_j)=1$. We have then looked for the smallest number of (b-a) for which

$$\sum_{j=a}^{b} \tilde{f}(x_j) \geqslant 0.95 \tag{4.3}$$

x_j for $j>b$ and $j<a$ are then discarded.
Histogram d does not show really severe outliers. Nevertheless the most extreme results will be filtered out by our filtering method. This will only have a small influence on the mean and standard deviations of the results.

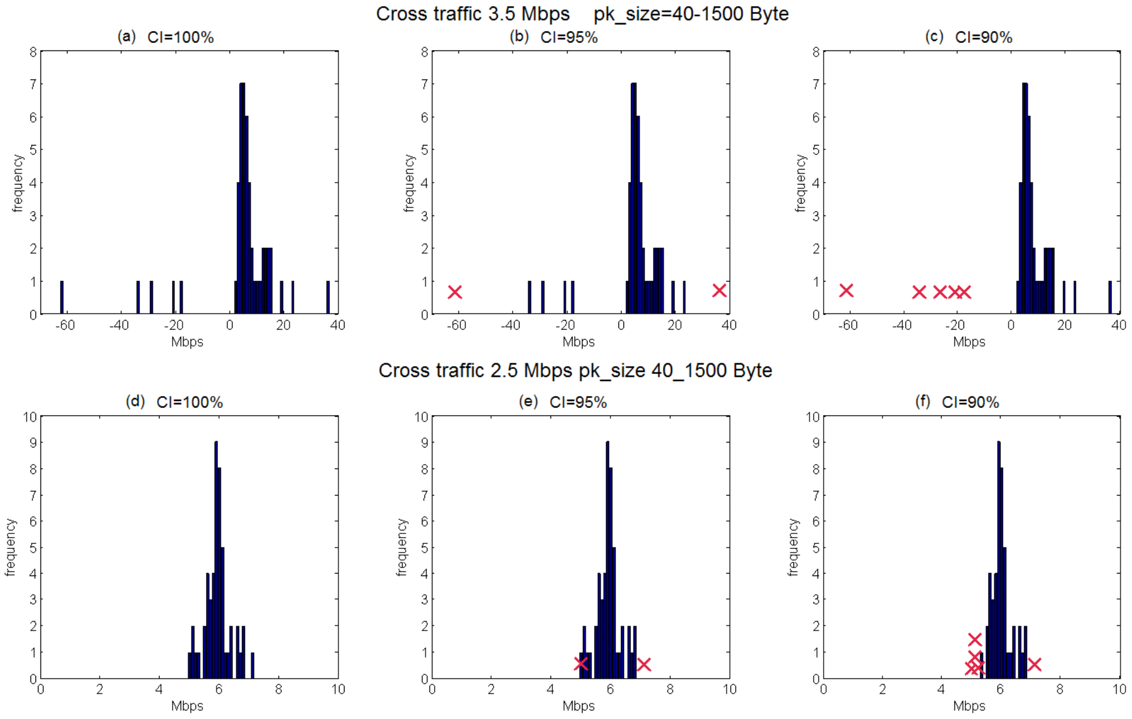

Figure 4.10: Illustration of filtering outliers

To indicate how the filter impacts the measurement results Table 2 shows the mean outcomes and standard deviations of the histograms of Figure 4.10.

Histogram position	Mean outcome (Mbps)	Standard deviation (Mbps)
a (CI 100%)	4.65	14.63
b (CI 95%)	5.37	10.42
c (CI 90%)	8.75	6.26
d (CI 100%)	5.99	0.44
e (CI 95%)	5.99	0.39
f (CI 90%)	6.04	0.34

Table 2: Mean outcomes and standard deviation of the histograms shown in Figure 4.10

5 Simulations of new capacity estimation tools

In this chapter we propose new path capacity measurement methods. The new methods and the simulation results are described.

5.1 Description of the new probing concept #1

From the literature research (chapter 3) we concluded that an available bandwidth measurement based on the probe gap model is the most promising approach to come to a low intrusive and relatively fast converging solution. Another possible approach is to use “link idle time measurement” for available bandwidth estimation. For both approaches it is necessary to have knowledge about the capacity of the path.

Li [18] obtains promising results with the *WBest* tool using the PGM model on a last hop wireless path. However, two important differences make our case more challenging compared to Li’s scenario. First off all we do not assume the “narrow link” to be at the last hop in the path. Secondly, we intend to build a sender-based measuring method, thus relying on round-trip measurements of probing packets.

Our idea for measuring path capacity looks similar to the probing scheme used in the *CapProbe* tool (see section 3.2.1). We send multiple packet pairs to infer the narrow link capacity from the measured dispersion between the pairs. The *CapProbe* authors suggest to use ICMP probe packets if installation on the destination node is not possible. With the use of ICMP packets it is possible to measure RTTs. If installation on both sender and receiver is possible, then the use of UDP probing packets is suggested. The use of UDP probing packets is preferred because the bandwidth critical applications such IPTV, are usually based on UDP flows also.

In contrast to *CapProbe*, we propose a sender based method based on UDP probing packets. To measure RTTs it is necessary that the receiver quickly responds with some reply packet. Therefore we suggest to send UDP probing packets to a (UDP) port number that is unlikely to be used. In this case, standard ICMP implementation requires that the receiver responds with an “ICMP port destination unreachable” (ICMP type 3, code 3) packet or 36 bytes [64](RFC 792). In contrast to ICMP echo packets, the “ICMP port unreachable” messages are usually much smaller in size than the original UDP packet. Because some ICMP implementations appear to use somewhat larger ICMP port destination unreachable packets than specified in [64] (larger ICMP error messages are allowed according to RFC112 [65]), we have used a packet size of 92 byte (20 byte IP header, 8 Byte UDP header and 64 byte payload) throughout our simulations.

The definition states that capacity is the maximum achievable throughput using maximum sized packets. Therefore we suggest to probe the path with packets that have a size equal to the path's maximum transmission unit (MTU). If a sender is unaware of the path MTU, than it should start the measurement with the MTU for the link through which it is connected. The path maximum MTU can generally be rapidly determined using path MTU discovery as described in [66] (RFC 1191). The new probing concept is illustrated by Figure 5.1.

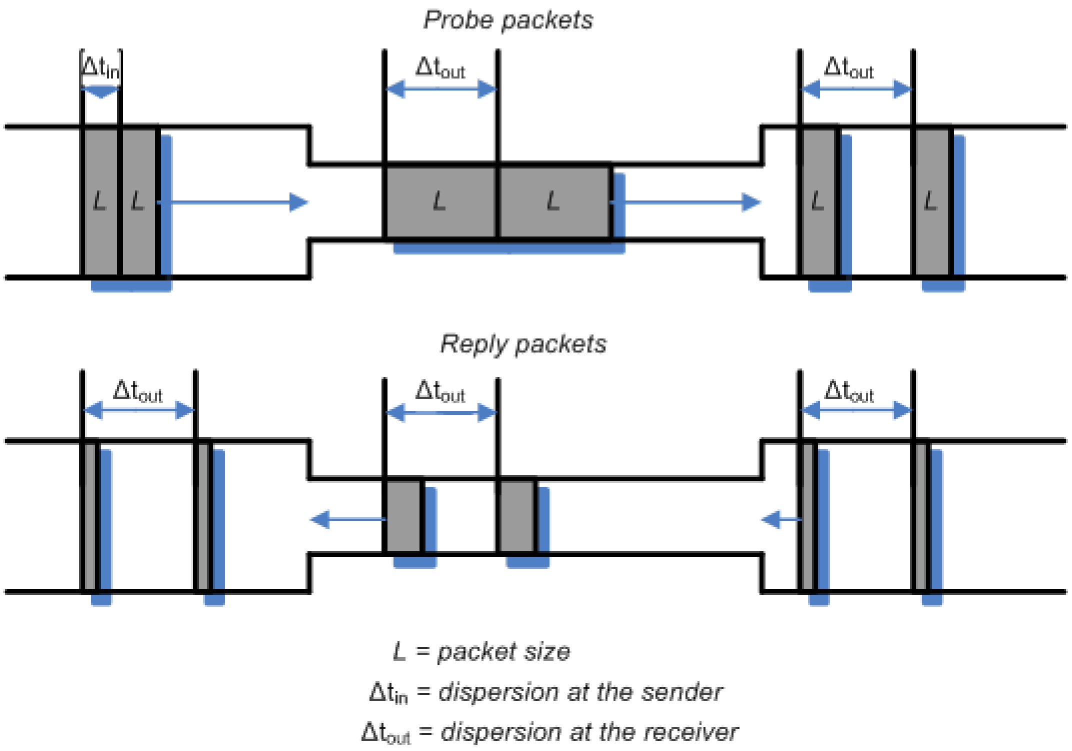

Figure 5.1: Illustration of the new probing method

In the upper part of the picture is shown how a packet is sent over three consecutive links. The link in the middle is the link with the lowest capacity (narrow link). Therefore, the packet will be transmitted at a lower rate over this link which causes the dispersion to increase Δt_{out}. Finally, the two packets are transmitted over the last link where the dispersion does not change.

In the lower part of the picture is illustrated how the reply packets are sent over the path. The dispersion between the reply packets is equal to Δt_{out} of the probe packets. Since the packets are smaller than the probe packet, the dispersion will not increase in any of the three links. Therefore, the dispersion measured after transmission of the reply packets is determined by the probe packet size and the capacity of the “narrow link”, i.e. Δt_{out}. Subsequently, the capacity of the narrow link can be derived from the dispersion using equation 3.3.

5.2 Packet pair probing in “Ethernet” networks

5.2.1 Wired scenario, no cross traffic

We start the investigation of our newly suggested probing scheme #1, by simulation of a relatively simple “all wired network” as shown in Figure 4.7. For this purpose we send 10000 packet pairs over the link with 50 ms delay between subsequent pairs. The 50 ms delay between the subsequent packet pairs is to make sure that the round-trips of both packets are completed before the next packet pair is send. In this way the packets of a particular pair will not interfere with packets of a subsequent pair. Both packets in the pairs have a packet size at IP layer of 1500 byte (payload without IP/UDP headers is 1472 byte). No cross traffic is inserted into the network in this scenario. Histograms of the obtained RTTs of the first and second packets of the pairs are shown in Figure 5.2a and b respectively.

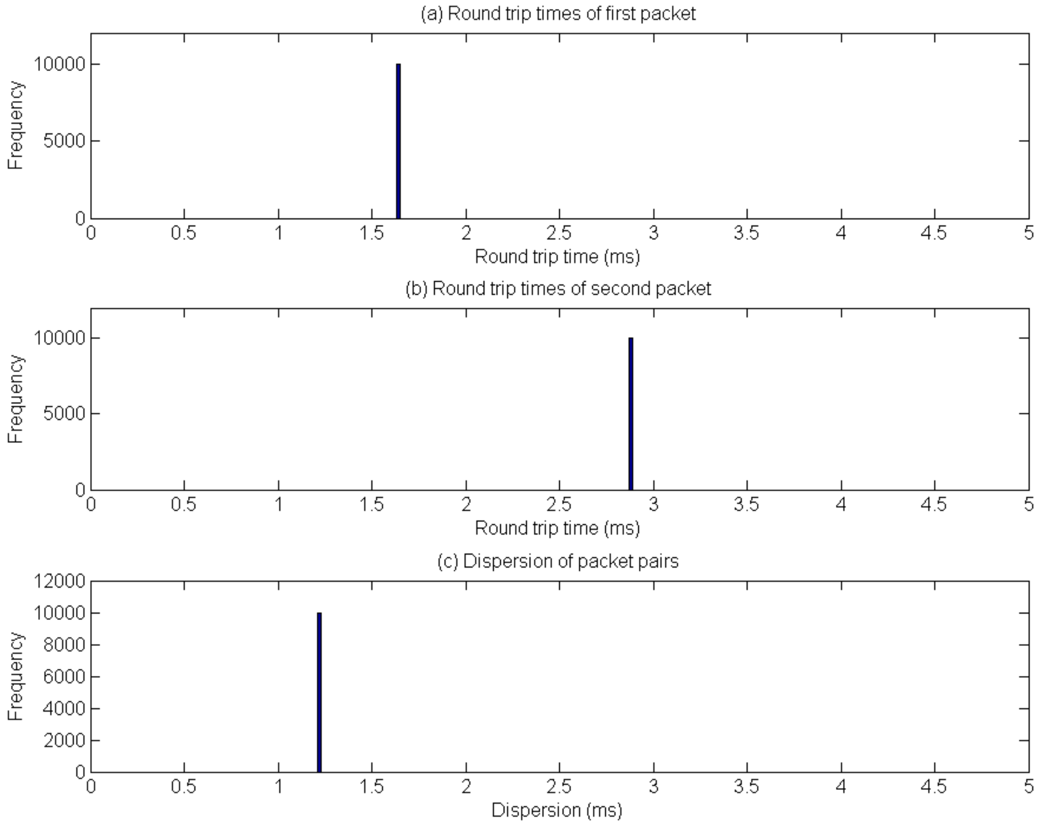

Figure 5.2: Measured RTTs for the first (a) and second (b) packets of the pairs and the dispersion (c) between the packets (wired network)

In Figure 5.2c we see a histogram of the measured dispersions of packet pairs over the "all wired" path without any cross traffic. The bin size of the histogram is 20 μs. The RTT histograms of both the first and second probe packets are single peaks with a height of 10000, which means that all the packets have equal RTTs.

The measured RTT of all first packets is 1.6589 ms. The measured RTT of all second packets is 2.8893 ms. Therefore, the dispersion measured between the packet of all pairs is 1.2304 ms. If we insert this numbers into equation 3.3 then we obtain an estimate for the narrow link capacity of 9.753 Mbps. This value is almost equal to the theoretical capacity of an 10M Ethernet link that was calculated in section 4.1.1. Apparently, OPNET adds 38 bytes of overhead to the IP payload because in that case equation 4.1 results exactly in the same capacity as the one obtained from the simulation. We did not yet determined what causes the additional overhead.

5.2.2 Wired scenario, with cross traffic

The next step is to evaluate the impact of cross traffic on the RTTs of probe packets. Therefore 5 Mbps of cross traffic is inserted into the network in two directions, from "cross traffic sender” to “cross traffic receiver” and vice versa. The packet size of the cross traffic is uniformly distributed between 40 to 1500 byte (including IP header); the inter-departure time of the cross traffic is exponentially distributed. This is one of the standard OPNET settings which more or less resembles average internet use [67], [68]. Both these streams will cross the “narrow link” as shown in Figure 4.7. Again 10000 packet pairs are sent over the link with a fixed delay of 50 ms between the pairs. In Figure 5.3 histograms are shown of the measured RTTs of the first (a) and second (b) packets of the pairs and the dispersions (c) measured between both packets in the pairs.

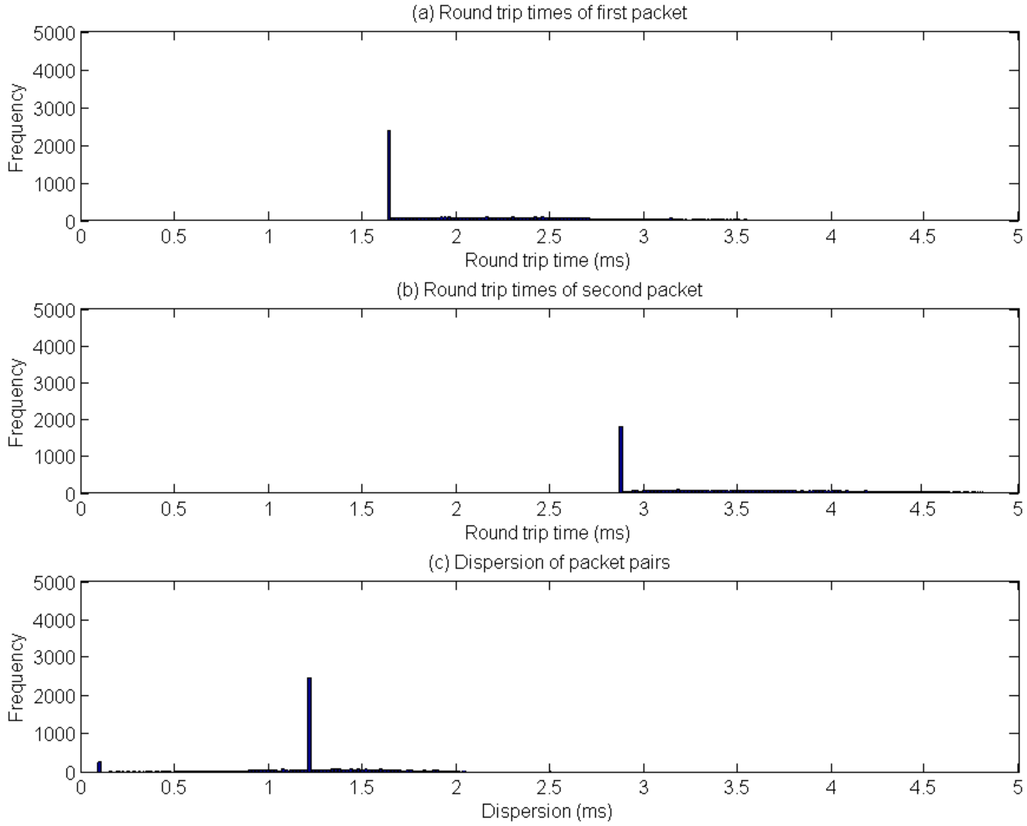

Figure 5.3: RTTs and dispersion for packet pairs on a 10 Mbps bottleneck link under 5 Mbps cross traffic of 40-1500 byte uniformly distributed packet size in two directions

Notice that only about 2000 “second packets” have a minimum RTT. The rest is distributed over longer times. Also notice that the measured dispersions are not constant. The dispersion measured varies between about 0.1 ms to about 2.1 ms. A small peak is visible at about the 0.1 ms and a second large peak is visible at about 1.25 ms.

The higher peak in the dispersion histogram (c) is at the same location as the peak in the dispersion histogram for the same network without cross traffic. This means that this mode of the measured dispersion still indicates the correct dispersion, i.e. the dispersion caused by the narrow link. The other small peak around 0.1 ms is caused due to a process similar to the “post narrow link” effect (see section 3.2.1). As can be seen in Figure 5.1, the smaller reply packets will not be sent 'back to back' from the probe traffic receiver. When these packets arrive at a link that is disturbed by cross traffic, which is the case for the narrow link in this scenario, there is a possibility that the first reply packet gets delayed due to cross traffic. In this case the second reply packet approaches the first reply packet, which causes the dispersion between the reply packets to decrease. The peak in the histogram is caused by those cases where the two reply packets end up back to back in the narrow link. This is the minimum possible dispersion for this scenario. This statement is confirmed by the following calculation:

We specified 92 bytes payload for the reply packets. The total overhead added by the Ethernet MAC is 34 bytes (see section 4.1.1). From equation 3.3 we derive that:

$$\Delta t = L / C_{l,n} \qquad (5.1)$$

Thus the minimum dispersion between the reply packet due to the narrow link is (126*8 bits)/10 Mbps = 0.10 ms, which is indeed equal to the minimum obtained dispersion.

For the 5 Mbps cross traffic scenario one could obtain the correct value of the measured dispersion by taking the second mode of the histogram. However, for even more challenging scenarios this can be difficult. Figure 5.4 shows the histograms of measured RTTs and dispersions for the same scenario with 8 Mbps of cross traffic in the narrow link and a reply packet size of 512 bytes. We increase the size of the reply packet to illustrate the problem of the “post narrow link”. For larger reply packets there is a higher probability that they will end up back to back in the narrow link. The RTTs and dispersions are measured and presented in the same way as Figure 5.3.

The minimum round-trip time for the first and second packets are now 2.017 ms and 3.2477 ms. This is slightly higher than the RTTs from figures 5.2 and 5.3 due to the additional delay for the transmission of the large reply packet. This time, a large peak is visible at about 0.45 ms and a second smaller peak is visible at about 1.25 ms.

This example illustrates that the largest mode in the dispersion histogram does not always indicate the correct value fort the dispersion. In the case of very heavy cross traffic, the first reply packet gets delayed relatively often. This process results in the relatively high mode in the dispersion histogram around 0.45 ms. It is even higher than the mode measured for probe packets that were not disturbed by cross traffic. Therefore we cannot simply use the largest mode of the dispersion histogram as an estimate of the dispersion over the narrow link. Other filtering techniques will be necessary as will be described in the next subsection.

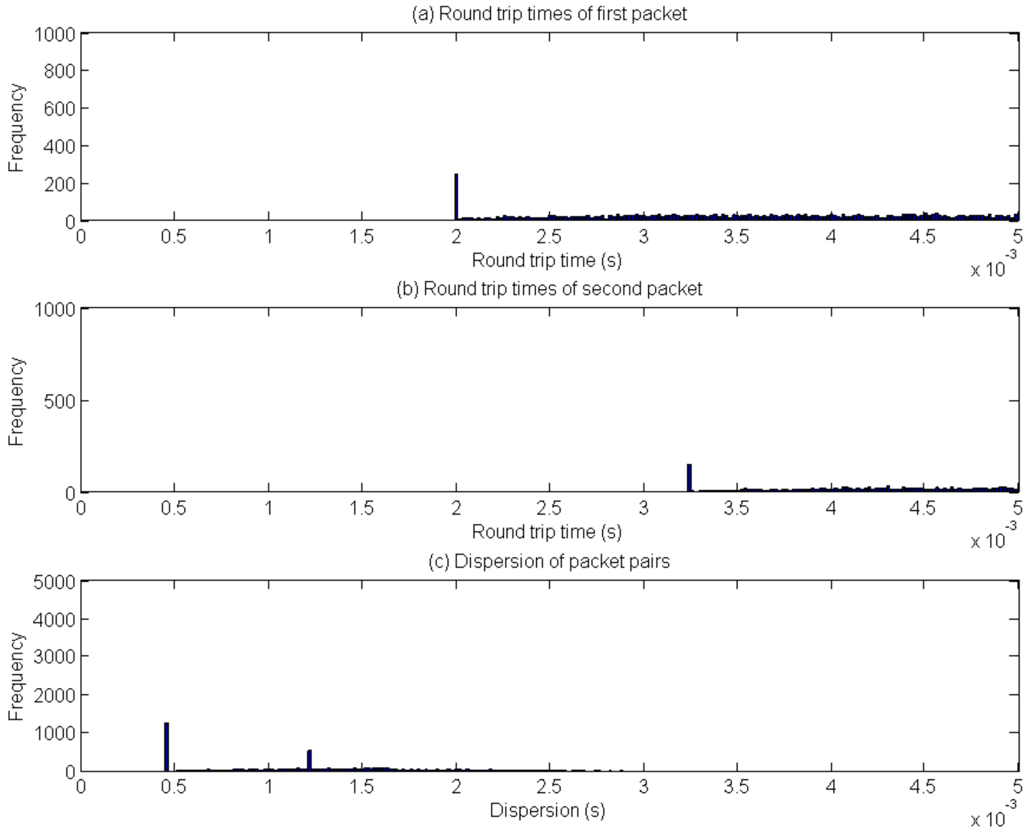

Figure 5.4: RTTs and dispersion for packet pairs on a 10 Mbps bottleneck link under 8 Mbps cross traffic of 40-1500 byte uniformly distributed packet size

5.2.3 Min-sum, median and min-min

In the previous section it is shown that it is not always possible to just take the minimum measured dispersion or the largest mode of the dispersion histogram to obtain the correct value for the capacity of the narrow link. Several filtering techniques are described in literature. In this section we compare the techniques described in [15] and [18] with a filtering method that we propose instead.

In [18] Li uses the median of the measured dispersions of a train of packet pairs to estimate the capacity of the narrow link. So, for a measurement consisting of N packet pairs, the dispersion d is found as follows:

$$d_{median} = median_{i=1 \ldots \mathrm{N}} \left(RTT_{2,i} - RTT_{1,i} \right) \tag{5.2}$$

With $RTT_{x,i}$ the RTT of the first ($x=1$) or the second ($x=2$) probe packet, and i the packet pair number. We include this method in the comparison because Li eventually obtains very promising results with his available bandwidth estimation method. However, Li publishes very little about the performance of the capacity estimation which he uses as an intermediate step in the available bandwidth estimation.

Another filtering technique is described by Kapoor in [15]. In this work it is assumed that the RTT will be minimal if a probe packet is not delayed by cross traffic. Therefore the dispersions are measured between multiple packet pairs. The dispersion of the pair of which the sum of the individual RTTs is minimal is used to estimate the capacity of the narrow link. So,

$$d_{minsum} = RTT_{2,i_{min}} - RTT_{1,i_{min}} \qquad \{ i_{min} | min_{i=1...\mathrm{N}} (RTT_{1,i} + RTT_{2,i}) \} \tag{5.3}$$

Our idea is simpler than the two described before. We also send multiple packet pairs. However, we use the difference between the minimum measured RTTs of the first and second packets of all pairs to estimate the capacity of the narrow link. So,

$$d_{minmin} = \min_{i=1...\mathrm{N}} RTT_{2,i} - \min_{j=1...\mathrm{N}} RTT_{1,j} \tag{5.4}$$

To get an indication of the performance of the different filtering methods, we analyze the capacity estimation results using the three filtering methods in the “all wired” scenario. First we split the 10000 RTT pairs (for 10000 packet pairs) into 50 'measurements' of 200 probes. Subsequently we calculate the capacities for the 50 measurements using the three methods. Thus, for example, for Li's method we take the median of 200 dispersions. Since there are 50 different 'measurements' we obtain 50 estimates for the narrow link capacity. We repeat these calculations for various number of probes per measurement (ppm), for example 50 measurements with only 100 ppm or 50 measurements with only 20 ppm. The mean values together with the standard deviation of 50 measurements are plotted against the number of ppm in Figure 5.5.

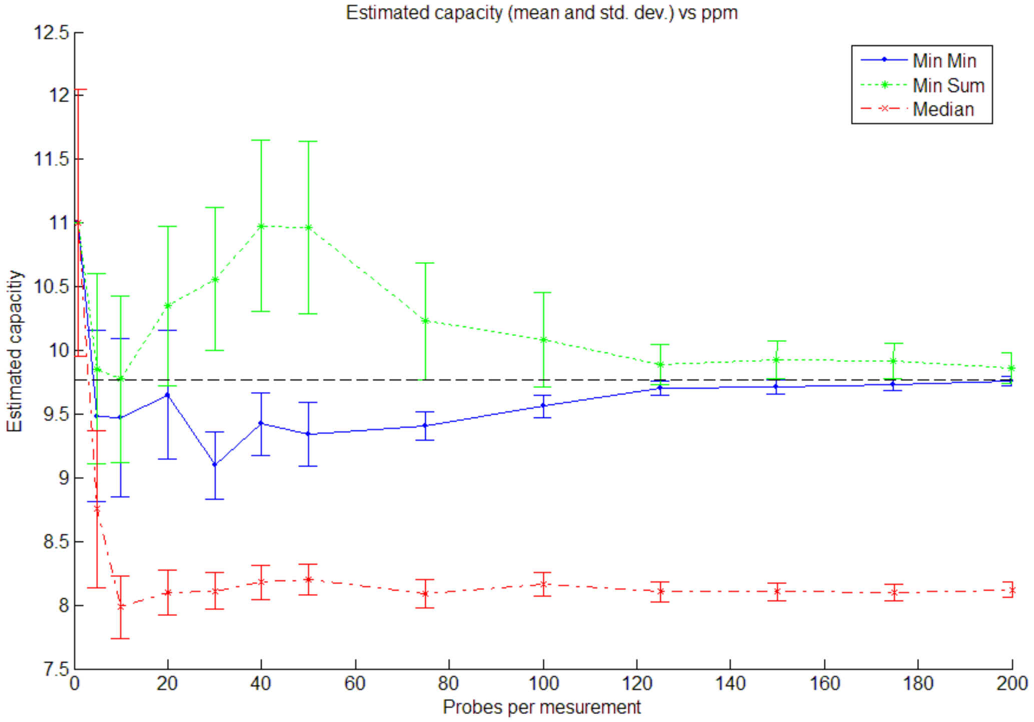

Figure 5.5: Estimated path capacities (with standard deviations) vs ppm under 8 Mbps cross traffic load on the bottleneck, for three different methods of filtering out post-narrow link values

The figure shows the measured capacities for the "all wired" path using three different methods for filtering the measured RTTs. The dashed black line indicates the theoretical capacity (see section 4.1.1). Standard deviations of all three results decrease when ppm increases. Further it is visible that the "min min" method shows the least deviation from the theoretically expected capacity of about 9.8 Mbps. Additionally the "min min" shows the lowest standard deviation of all three methods when ppm > 100. Finally it is noticeable that the mean outcome of the "median" method shows a large deviation from the actual capacity.

The results indicate that our "min min" filtering method outperforms the "median" and "min sum" filtering methods both on accuracy and consistency, at least for the wired network scenarios. The "min sum" result is slightly less accurate, but has the advantage that only one value (d) has to be maintained, whereas min min needs to keep track of RTT_1 and RTT_2 independently.

5.3 Probing with method #1 in wireless networks

In the previous section it has been shown that packet pair probing using "min min" filtering provides promising results regarding path capacity measurements in wired networks. Even under very challenging conditions, where cross traffic of about 80% of the bottleneck capacity flows in both directions, using a sufficient amount of probes leads to very good results.

However, the estimation tool should eventually work in various network scenarios that do not always consist of wired links only. Therefore we continue our simulations in a more challenging network where a single wireless (802.11b) link is included in the path between the “probe traffic sender” and “probe traffic receiver”.

5.3.1 Path with a single wireless link (802.11b) as narrow link

A schematic drawing of the network including a 'single wireless link' was shown in Figure 4.8. Again we start the investigation of the probing method #1 by sending 10000 packet pairs from the probe traffic sender to the probe traffic receiver. No cross traffic is added into the network in the initial simulation. All probe packets are 1500 byte in size (including IP header). The results for the measured RTTs and dispersions of the packet pairs are shown in Figure 5.6.

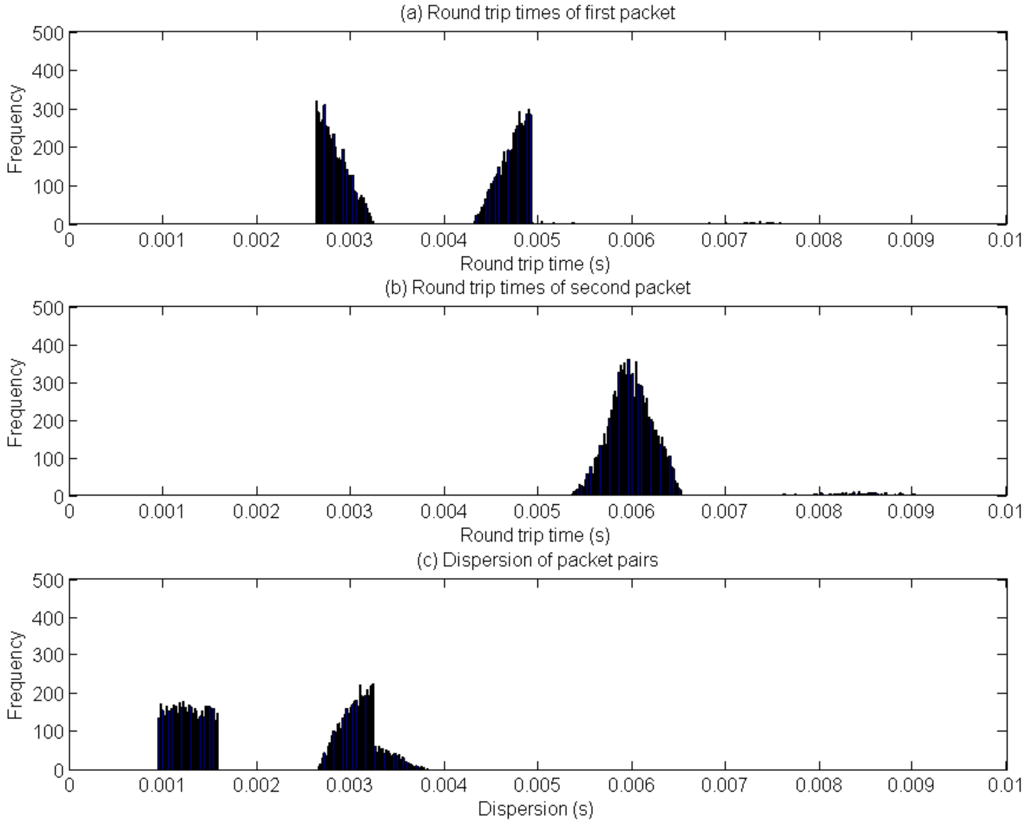

Figure 5.6: RTTs and dispersion of packet pairs in a “single wireless” scenario without cross traffic

Figure 5.6a shows the histogram for the measured RTTs of the first packets of the packet pairs. Clearly visible are two distinct regions where the RTTs concentrate. There are bins after 0.05 ms that have very low frequencies. This is caused by packets that need to be retransmitted due to a collision and thus experience much longer delays. Additionally it is noticeable that the peaks are not sharp peaks like in Figure 5.2-5.4, but are spread over multiple bins (binsize = 20µs) of the histogram. Figure 5.6b shows the histogram of the RTTs of the second packets of the packet pairs. Here we see a single region where the RTTs concentrate, although here also the RTTs spread over multiple bins. The lower part of the picture shows a histogram of the measured dispersions between the packet pairs. The

measured dispersions are also divided over two separated regions. The first region indicates a uniform distribution of the dispersions. The second region shows a remarkable distribution. The shapes of these distributions are further explained in the next section.
The estimated capacities that are obtained using the three different filtering methods are shown in Figure 5.7.

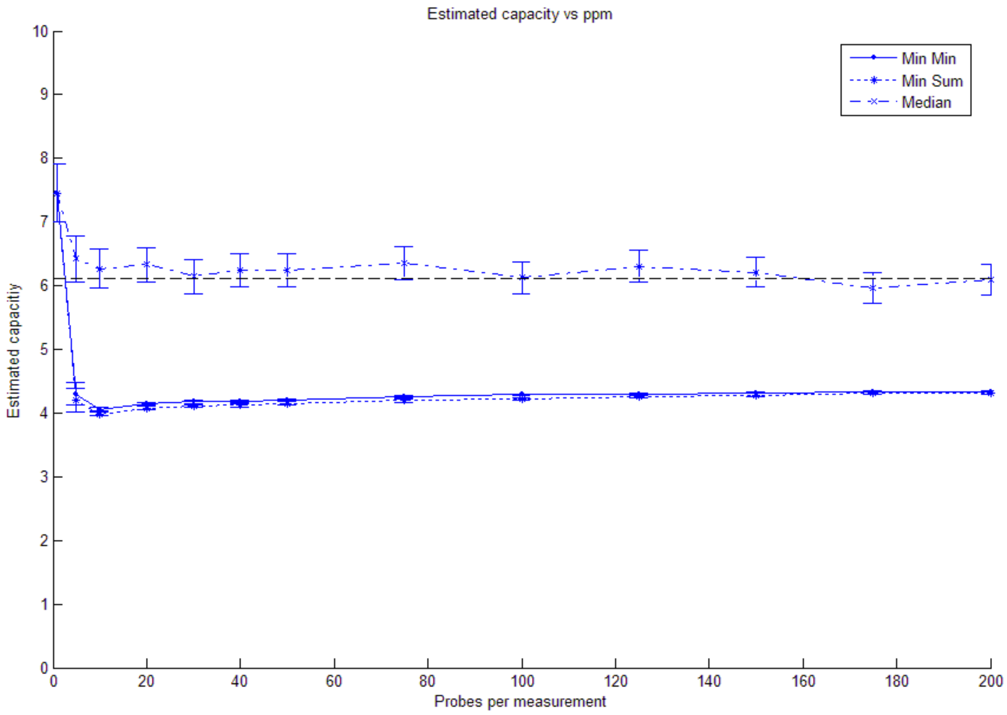

Figure 5.7: Estimated path capacities (with standard deviations) vs ppm for the "single wireless" scenario without cross traffic for three different methods of filtering out post-narrow link values

The figure shows the measured capacities for the single wireless scenario using the three different methods for filtering the measured RTTs. Again, the dashed black line indicates the theoretical capacity as determined in section 4.1.2. Both the 'min min' and the 'min sum' method rapidly (from about 10 ppm) approach a capacity of about 4.2 Mbps. The median method approaches a capacity of about 6.3 Mbps.
Although the median method shows slightly less consistency and a higher standard deviation in the results, the mean values are closer to the value expected from the capacity analysis in section 4.1.2. The 'min min' and 'min sum' methods show better consistency and lower standard deviations, but the estimated path capacity differs about 2 Mbps from the expected path capacity.
Figure 5.8 shows the path capacity estimation results if 2 Mbps of cross traffic is added in the path. The cross traffic is sent from the cross traffic sender to the cross traffic receiver (see Figure 4.8). The packet size of the cross traffic is uniformly distributed between 40 and 1500 bytes and the interarrival time of the cross traffic packets is exponentially distributed (type 2 in Table 1).

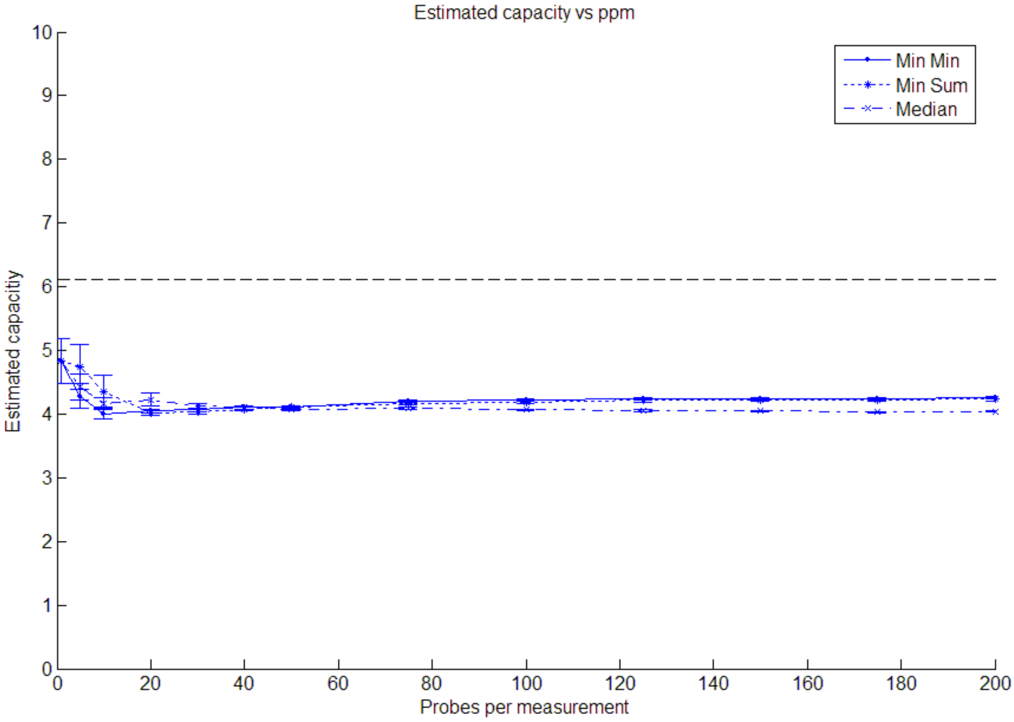

Figure 5.8: Estimated path capacities (with standard deviations) vs ppm for the "single wireless" scenario with 2 Mbps of cross traffic in both directions for three different methods of filtering out post-narrow link values

Figure 5.8 shows the measured path capacities for the 'single wireless' scenario using three different methods for filtering the measured RTTs. In this case all three methods rapidly (from about 10 ppm) approach a capacity of about 4.2 Mbps. Thus the 'median' method provides estimates for the path capacity that are strangely dependent on the amount of cross traffic in the wireless link.

From these results it can be concluded that packet pair probing with method #1 using any of the three filtering methods does not lead to correct capacity estimates over paths including wireless links. In the case that there is no cross traffic in the network, only the median filtering method shows reasonable results. The other two methods underestimate the capacity of the narrow link. If cross traffic is added in the wireless link, all three methods underestimate the capacity. Underestimation of the capacity means that the dispersion between the probe packets in the narrow link is overestimated. To explain this overestimation we analyze what causes the remarkable distributions of the RTTs and dispersions in Figure 5.6. Therefore we look at the ETEs and at the dispersion between the packets measured at the probe traffic receiver, thus halfway the round-trip. The results for the 'zero cross traffic' scenario are shown in Figure 5.9.

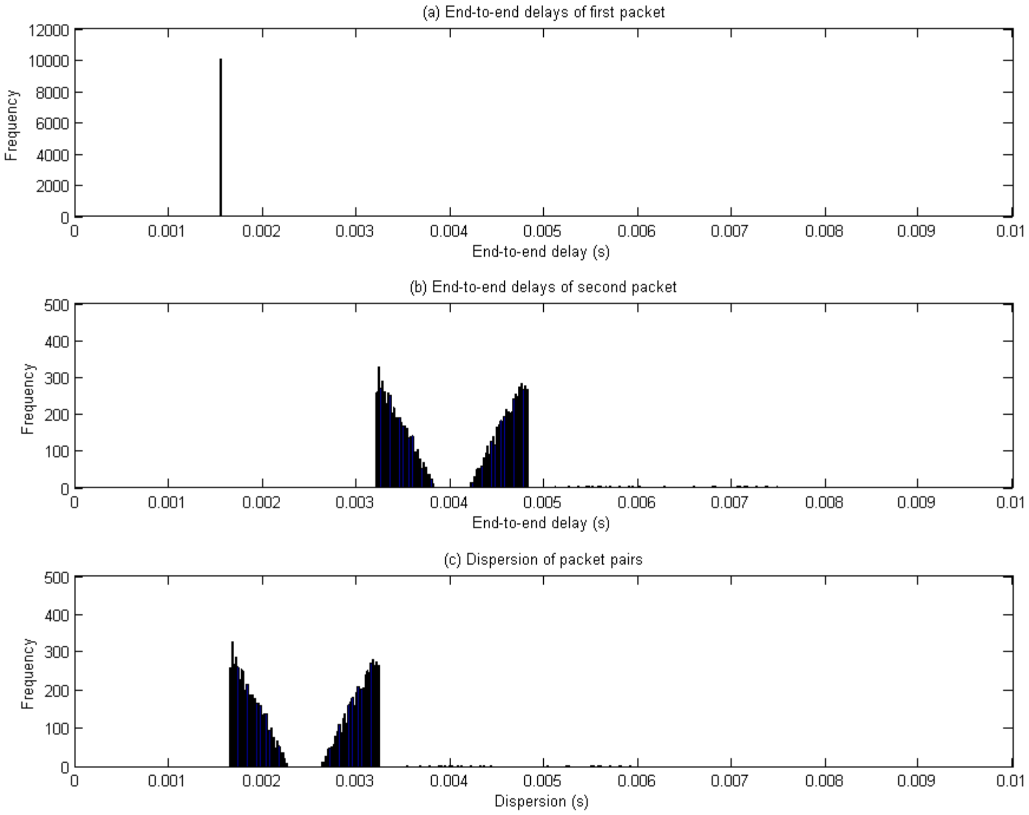

Figure 5.9: one-way delays and dispersions measured at the receiver ("Mob_rec_host") for a "single wireless" bottleneck path

Figure 5.9a shows a histogram of the ETEs of the first packets of the packet pairs. All first packets have an equal ETE which results in a single peak with a height of 10000. Figure 5.9b shows a histogram of the RTTs of the second packets of all packet pairs. We see two distinct regions where the RTTs are concentrated, although the RTTs are spread over multiple bins of the histogram. The first region shows linearly decreasing frequencies. The second region shows linearly increasing frequencies. Figure 5.9c shows the dispersion as measured in the probing traffic receiver. It shows the same distribution as the histogram of the second packets of the pairs.

Figure 5.9 shows the estimated path capacities when using the ETEs as measured at the 'probe traffic receiver'.

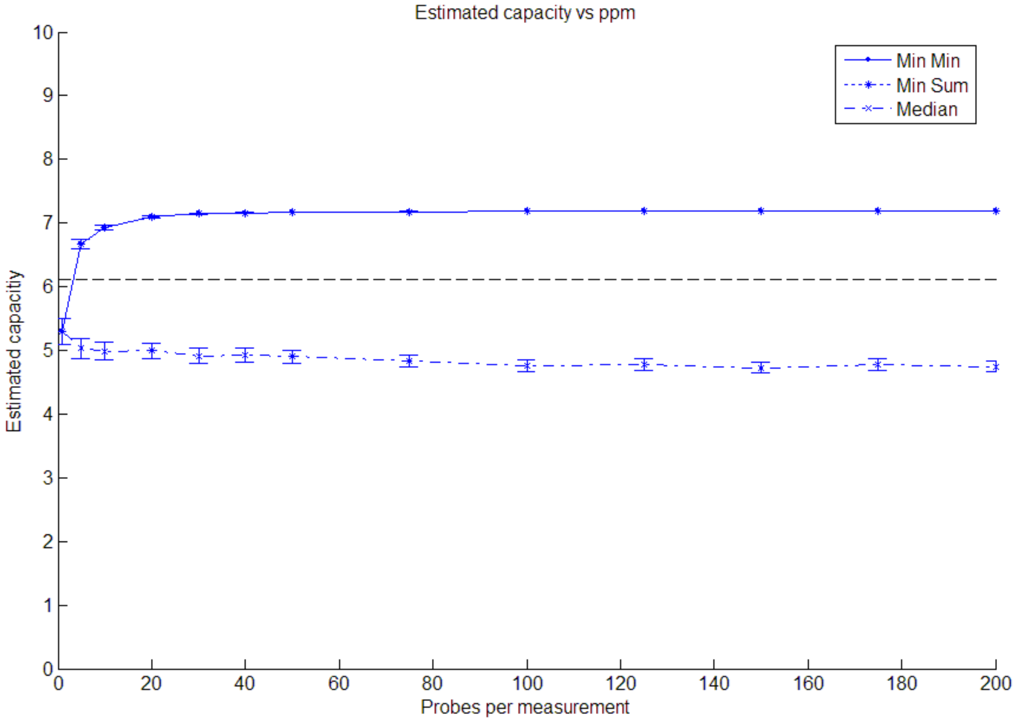

Figure 5.10: Estimated capacities from one-way delays over a “single wireless” path for three different methods of filtering out post-narrow link values

The figure shows the measured path capacities for the “single wireless” scenario using three different methods for filtering the measured ETEs or dispersions. Both the “min min” and the “min sum” method rapidly (from about 10 ppm) approach a capacity of about 7.1 Mbps. The median method approaches a capacity of about 5.0 Mbps. Additionally the “median” measurements show slightly lower consistency and a higher standard deviation. Thus, when using the ETEs for the estimation of the capacity, both the 'min min' end the 'min sum' significantly overestimate the capacity. The median method significantly underestimates the capacity.

5.3.2 Analysis of the results

In Figure 5.9 it can be seen that all first packets of the probe pairs have equal ETE. This means that all packets are sent from the AP to the probe traffic receiver without experiencing a random backoff period. In the histogram of the second packets it can be seen that the delays are spread over multiple 20 ms bins. This is because a random backoff period will be inserted before transmission of the second probe packet directly after the first probe packet. Although the random backoff times are uniformly distributed, in the figure of detected delays we see two distinct regions. The first region shows linearly decreasing frequencies and the second region displays linearly increasing frequencies.

This behavior is caused by the reply packet of the first probe packet. As soon as the first probe packet is received by the probe traffic receiver, this device will send a reply packet back to the probe traffic sender. However, the AP tries to send the second probe packet directly after the first probe packet is received by the probe traffic receiver. The randomly determined backoff in the two devices decides whether the AP wins the contention (and the second probe packet is sent) or that the probe traffic receiver wins the contention (and the first reply packet is sent). In case the AP wins, this results in an ETE of the second probe packet that falls into the first region of short delays in the histogram. The lower the random backoff, the higher the probability that the AP will win the contention. This explains why the left region shows linearly decreasing frequencies.

On the other hand, when the probe traffic receiver wins the contention, the reply packet is sent first. The second probe packet has to wait until the transmission of the reply packet has been completed. Only then the second probe packet will be sent (after the remaining backoff has been decremented). These cases lead to the region of longer ETEs of the second probe packet in the histogram. In this case the higher the random backoff, the higher the probability that the AP will loose the contention. This explains why the frequencies of the right-hand region in the histogram are linearly increasing. The case that the first reply packet wins the contention from the second probe packet is illustrated in Figure 5.11a.

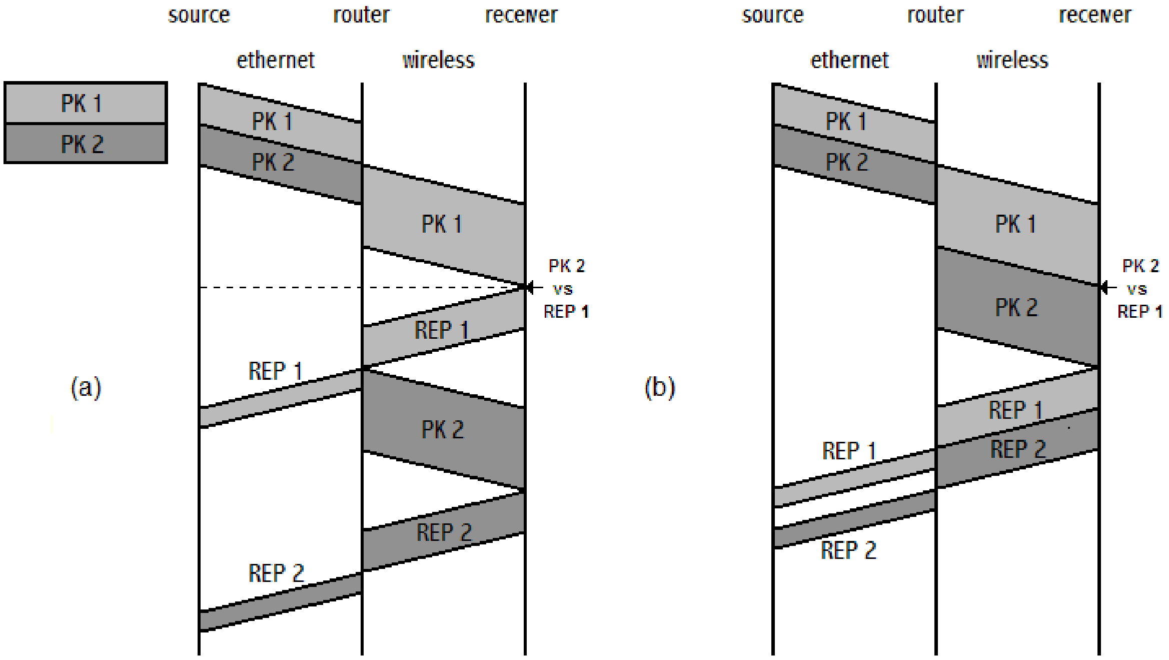

Figure 5.11: Traffic flow diagram for probing with method #1 in the "single wireless" scenario (a) when the reply packet wins from the second probe packet and (b) vice versa

Figure 5.11a illustrates how a packet pair round trip looks in a path consisting of an Ethernet link and a wireless link. As soon as packet 1 (PK1) has been sent from the AP (router) to the receiver, the AP wants to send PK2 to the receiver. At this moment the first reply packet (REP1) and PK2 will contend for the medium. What is shown in the picture is that the receiver wins the contention, so (REP1) is sent first. Only after this reply has been sent from the receiver back to the AP, PK2 can be sent from the AP to the receiver.

Thus the appearance of the two two regions in the ETE and RTT histograms of Figure 5.9b and 5.6a can be explained by the contention between the first reply packet and the second probe packet. This also explains why the path capacity is constantly underestimated by the 'min min' and 'min sum' methods if the estimation is based on round-trip measurements: PK2 is acting as cross traffic to REP1 and vice versa. And in case PK2 wins the contention, REP1 will obstruct REP2 instead. In any case, REP2 will arrive too late.

In a situation where UDP traffic is sent in the single direction from sender to receiver, then there will be no reply packets that cause a longer dispersion between subsequent packets. For a correct capacity estimation we need a value for the dispersion that is not increased due to reply packets.

5.4 Design and simulation of probing method #2 : small/large packet probing

5.4.1 Design of the small/large packet probing method

The problem concerning the extra delayed round trip of the second probe packet can be solved if the receiver does not respond with a reply packet on the first probe packet. This will yield the 'correct value' for the RTT of the second packet. In order to obtain a value for the dispersion, one then also has to measure the RTT for a single packet. This single isolated packet will be sent well before or after the packet pair and its reply mimics the avoided first reply packet of the packet pair. In this case the difference between the single packet RTT and the packet pair RTT will result in a correct value for the dispersion.

Thus if we can find a way to send a packet from sender to receiver so that the receiver will not respond with a reply packet, then we have a solution for the problem. We repeat that the eventual estimation tool should not require installation of special software on the receiving device. Therefore it is not possible to send, for example, UDP packets that are simply accepted by the receiver. If the packet cannot be delivered to some higher level application then ICMP standards require the node to respond with a “destination unreachable” packet. The solution therefore should be found within the standard IP implementation.

IP fragmentation, as described in section 4.2, turns out to provide a solution for the problem. The idea is to send an UDP packet of such size that it needs to be fragmented into two IP packets. First of all it ensures that both packets will be sent back-to-back to the lower layer two. Secondly, although the packet will be sent in two separate frames, there will only be one reply after reception of the second frame. This is because the receiver responds with a reply packet only after the full IP packet has been received. Therefore the dispersion between the two fragments will not be ”disturbed” by a reply message on the first packet. The new probing concept is illustrated in Figure 5.12.

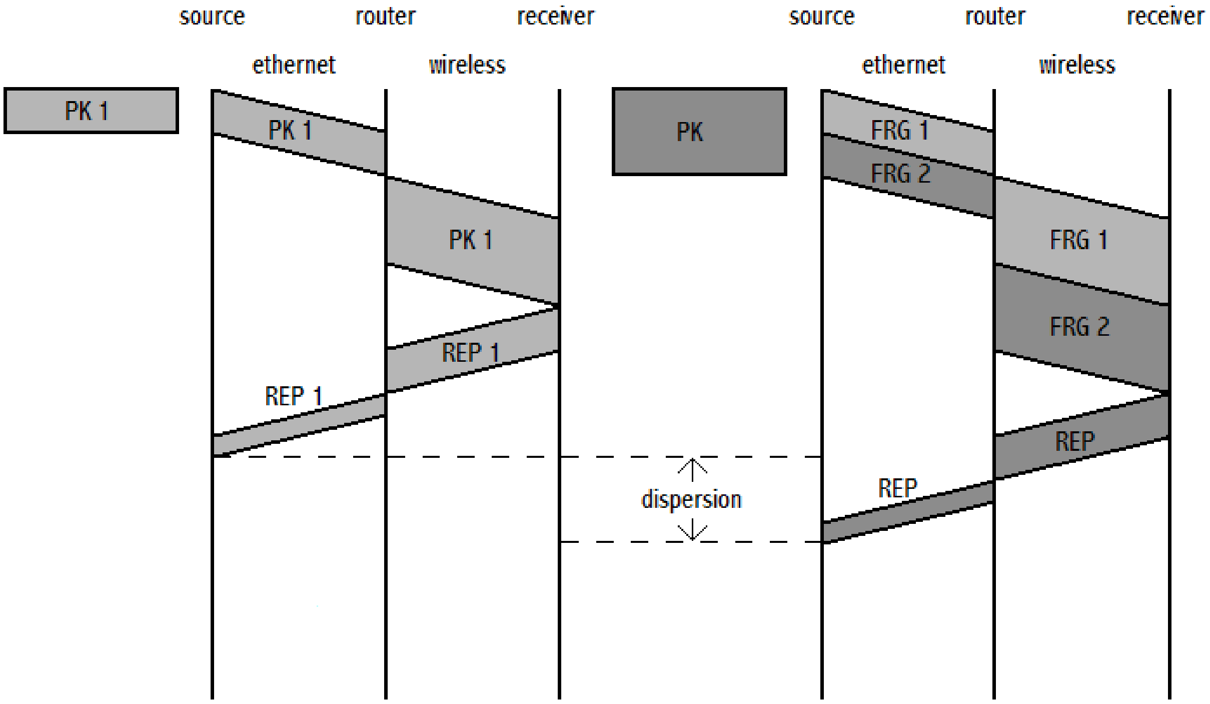

Figure 5.12: Traffic flow illustration of probing method #2 using small and large probe packets. Time goes from top to bottom

In the figure is illustrated how the new probing method should obtain a correct value for the dispersion caused by the narrow link. First, the RTT of a small packet, which fits into a single frame, is measured. The propagation of this small packet is illustrated in the left part of the figure. Secondly, the RTT of a large packet is measured. The packet needs to be fragmented into two smaller packets (FRG1 and FRG2) and thus two frames. Since the receiver will only reply with REP after both fragments of the packet have been received, there will be no contention between a second probe packet and a reply packet. The round-trip process for such a "large" packet is illustrated in the right part of the figure.

5.5 Simulation of small/large packet probing method without cross traffic

To accommodate probing method #2 we adjust the probing scheme of method #1 to send alternating small (equal to path's MTU) and large (double the path's MTU) probe packets. Additionally we adjust the scheme to send a new probe packet as soon as the round trip of the previous probe packet is completed. Thus the new probe packet is send directly after the reply packet on the previous probe packet is received. Since the average RTTs are generally much shorter/lower than 50 ms, this will cause the new scheme to send more probe packets in a shorter period (sending 200 pairs with interdeparture time of 50 ms would take 10 seconds). Further, it will also allow for extra time when needed. For instance, heavier cross traffic will result in the RTT of the probe packets to increase. The new probing scheme measurement time will therefore depend, among others, on the amount of cross traffic in the path. Figure 5.13 shows the simulated arrival histogram for probing the "single wireless link" network, without any cross traffic, with 10000 probe pairs (10000 small packets and 10000 large packets).

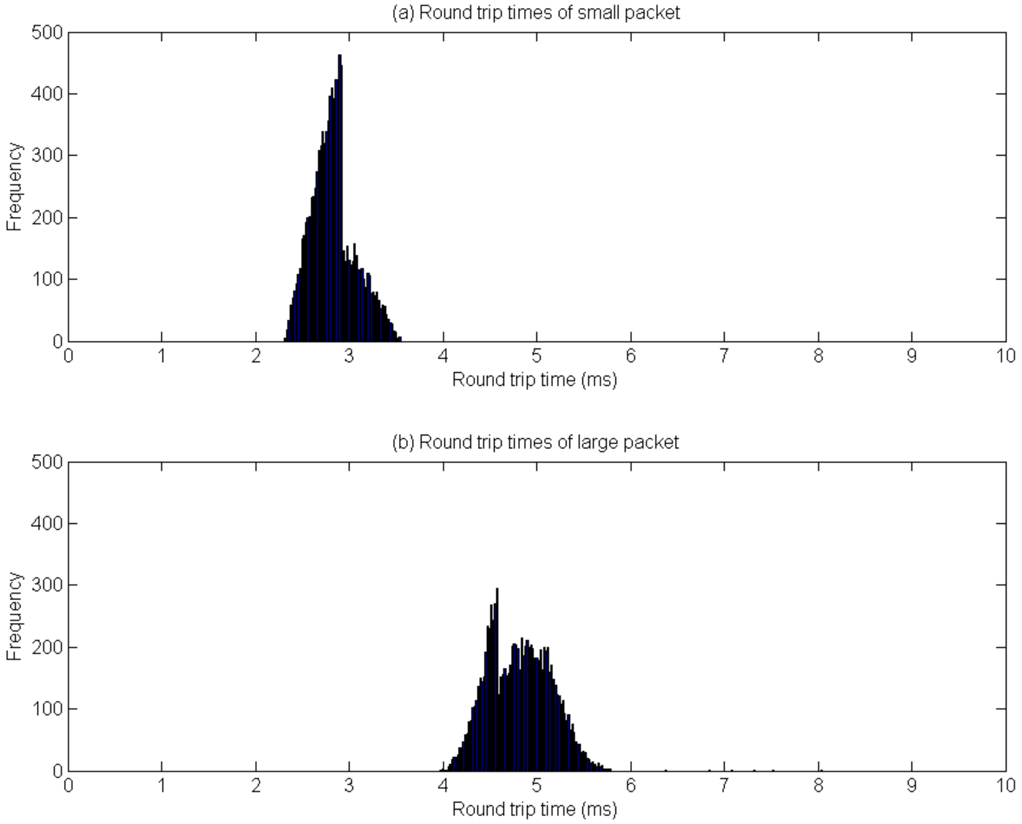

Figure 5.13: Measured RTTs in the "single wireless link" network using small/large packet probing

Figure 5.13a shows a histogram of the small probing packets. It still shows a remarkable distribution in contrast with probing method #1. The RTTs are now concentrated in a single region, though spread over multiple bins of the histogram. Most remarkable is that the distribution at the left half of the region has a much higher maximum than the distribution at the right half of the region. Figure 5.13b shows a histogram of the RTTs of the large probing packets. Again, RTTs are concentrated in a single region though spread over almost 2 ms. At the beginning of the feature a peak is visible which looks similar to the shape of the histogram of the small packet RTTs. The estimated capacity using 'min min' filtering on these 10000 pairs of packets is 7.2 Mbps, which is now an overestimation instead of an underestimation, but it is the value one expects with using the 'min min' filtering method. This method namely selects the RTT results that experienced minimum (i.e. zero) zero random backoff, whereas the expected path capacities as calculated in section 4.1.2 are based on average random backoff. Note that using the 'min sum' or 'median' filtering methods cannot be used anymore since the probing packets are not transmitted in pairs any longer.

The remarkable distributions can be partly explained due to the fact that the probe traffic sender is connected to the AP with a 100M Ethernet link. After the reply packet has been received by the AP, it forwards the packet to the probe traffic sender. The probe traffic sender sends a new probe packet in the direction of the probe traffic receiver directly after the reply packet is received. Since the packet is transmitted over the 100M Ethernet link, this process happens so fast that the AP is still busy with acknowledging the reception of the reply packet to the probe traffic receiver. Therefore, unlike what we expect, this new probe packet will already experience random backoff in the forward path. However, the extra random backoff does not explain why the distributions are asymmetrical. We are not yet able to explain what causes the asymmetry in the distributions.

We expect that when the probe traffic sender is connected to the AP with an 10M Ethernet link, the AP should have sufficient time to acknowledge the reply packet before the new probe packet arrives. Figure 5.14 shows the histograms of the RTTs for a “single wireless link” scenario where the probe traffic sender is connected to the AP using a 10M Ethernet link.

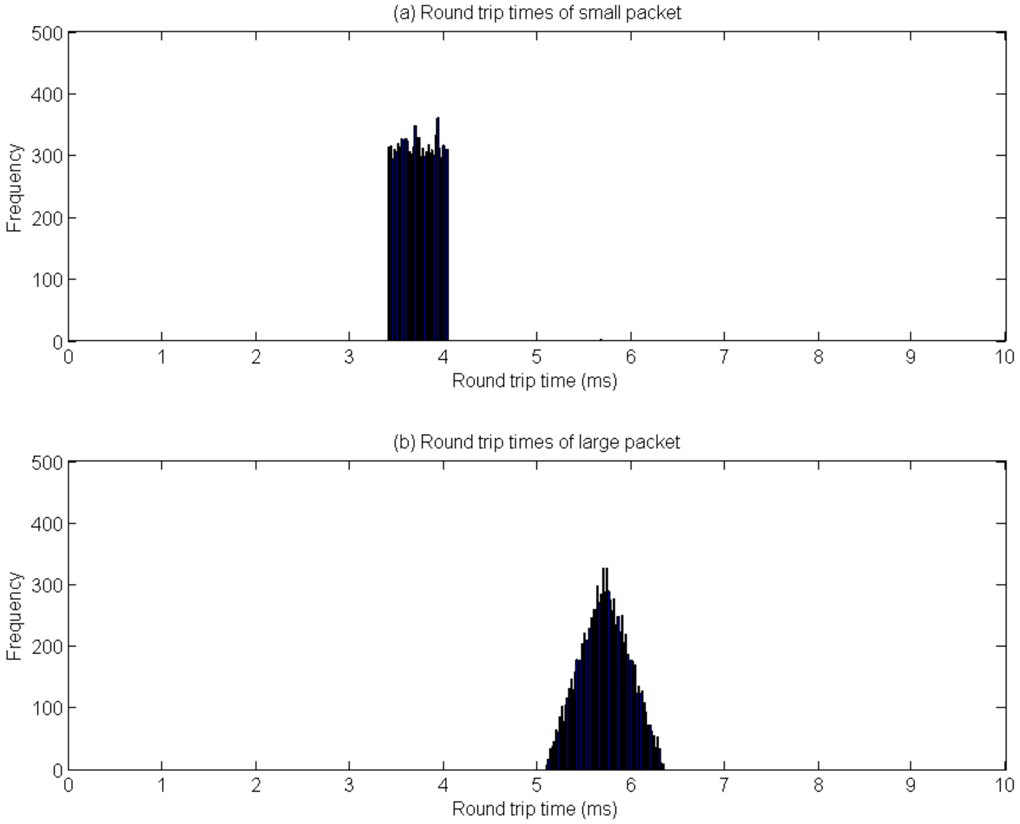

Figure 5.14: Measured RTTs using “small/large probing” and a 10 Mbps Ethernet link between “probe traffic sender” and the “access point”

Figure 5.14a shows a histogram of the small probing packets. It shows uniformly distributed RTTs over the region from 3.4 ms to 4.0 ms. This is exactly the distribution that we expect. The probe packets do not experience random backoff in the forward path. Only the reply packet will experience random backoff because the medium will still be busy with the ACK on the first fragment. Thus only one phase of random backoff will occur in the whole round trip. Therefore the RTTs of the packet are uniformly distributed over a region that has a width equal to the contention window (32 bins of 20μs wide).

Figure 5.14b shows a histogram of the RTTs of the large probing packets. The distribution has the shape of an upward pointing triangle and is located between 5.1 ms and 6.4 ms. This distribution also corresponds with our expectation. The first fragment of the probe packet will not experience random backoff in the forward path. However, the second fragments of the probe will experience random backoff because the medium will still be busy with the ACK on the first fragment. Also the reply packet will experience random backoff, since the medium will still be busy with acknowledging the second fragment of the probe packet. In total there will be two random backoff phases in the round-trip process of a large probe packet. Thus the distribution of the RTTs will show a convolution of two uniform distributions of 32 bins wide, i.e. a triangular distribution of 63 bins wide, like the distribution in the figure.

It can be concluded that using a 10M Ethernet link instead of a 100M Ethernet link between the probe traffic sender and probe traffic receiver prevents an extra random backoff phase in the round-trip process of the packets.

This additional random backoff phase could be prevented by adding a delay between the reception of the reply packet and the sending of the new probe packet when using a 100M Ethernet link. The delay should be 214 µs because it takes that amount of time to complete the transmission of the ACK (see section 4.1.2).

Anyway, it is expected that the extra random backoff phase will not have a large impact on the estimation of the capacity. This is because the random backoff adds equally to the delay of both the small probe packets and the large probe packets. An additional delay of 214µs between each probe packet would result in an increased convergence time of about 90 ms if 200 probe pairs (200 small + 200 large packets) are used. If the measurement is performed without the extra spacing of 214µs between the packets, then the extra random backoff adds about 120 ms (320µs*400) and about 70 ms (169µs*400) (see average random backoff times in section 4.1.2) to the measurement time for a "single wireless link" and a "double wireless link" configuration respectively. The additional delays due to the random backoff only occur in these network configuration, whereas the additional delay due to the extra spacing between probe packets would occur in all network configurations. Therefore we leave the probing scheme as it is, thus without the additional delay between subsequent probe packets. We prefer to use the 100M link instead of the 10M link because 100M Ethernet links are more common in today's home networks.

We want to note here that if the measurement is performed with the extra spacing between the packets, and therefore without the extra random backoff phase, the actual average RTT of the probe packets would be lower. This might be beneficial because it reduces the probability of probe packets to get disturbed due to cross traffic. We leave the comparison of these two methods for further research.

5.6 Small/large packet probing with cross traffic

Figure 5.15, Figure 5.17 and Figure 5.18 show the histograms of RTT measurements with different types of cross traffic in the single wireless (bottleneck) link. It is immediately obvious that the influence of cross traffic on the probe packet RTTs can be very different for different types of cross traffic. Figure 5.15 Shows the histograms for an RTT measurement of the ”single wireless” scenario with cross traffic that has a constant packet size of 1500 bytes. The histograms of both the small probe packet RTTs as well as the large probe packet RTTs show a repeating pattern. In the beginning of the histograms, patterns are visible that look similar to the distribution of the RTT results without cross traffic. The pattern repeats about every 1.9 ms. Further, the width of the patterns spreads over an increasingly larger time interval. Additionally, the left half of the distribution becomes less dominant over the right half of the pattern as it appears later in time.

From the measurements, without any cross traffic, we know that the average dispersion between the small and large packets is about 1.9 ms. Therefore we can conclude that it takes, on average, about 1.9 ms (see Figure 5.13) to send a 1500 byte packet over the wireless link. This explains why the histograms in Figure 5.5 show repeating patterns. Each interfering cross traffic packet will cause an additional delay of about 1.9 ms in the RTT of a probe packet. Also, additional contention phases will be introduced for each interfering cross traffic packet. Therefore, the patterns will increase in width at larger RTTs.

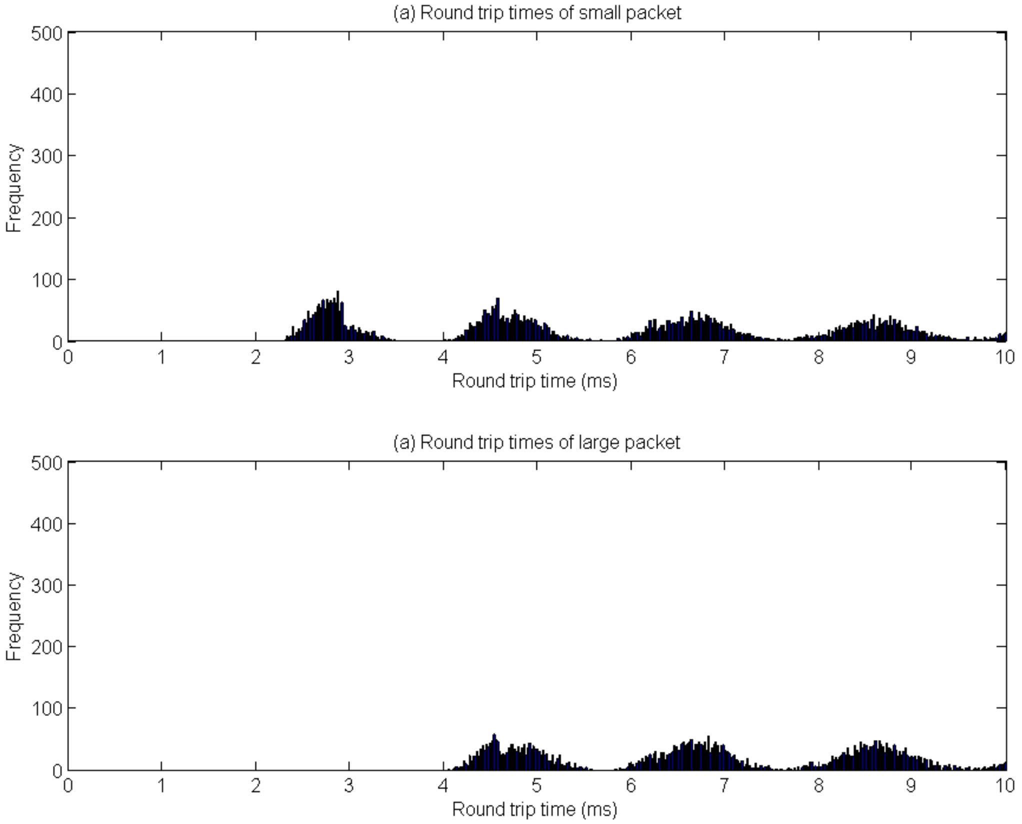

Figure 5.15: Measured RTTs in a "single wireless link" scenario with 3.5 Mbps cross traffic of 1500 Byte constant packet size

Figure 5.16 Shows the histograms for a measurement with cross traffic that has uniformly distributed packet sizes between 40 and 1500 bytes. Also in these histograms we can recognize the distribution at short RTTs that looks similar to the one with zero cross traffic. In contrast to the measurement with constant size cross traffic, we do not see a repeating pattern in the histograms. Instead, after the initial peak of the histograms we see that the RTTs are spread quite evenly over time. This can be explained by the fact that the cross traffic packet size is uniformly distributed between 40 and 1500 bytes. Therefore, various amounts of additional delay are added between probe packets due to the cross traffic.

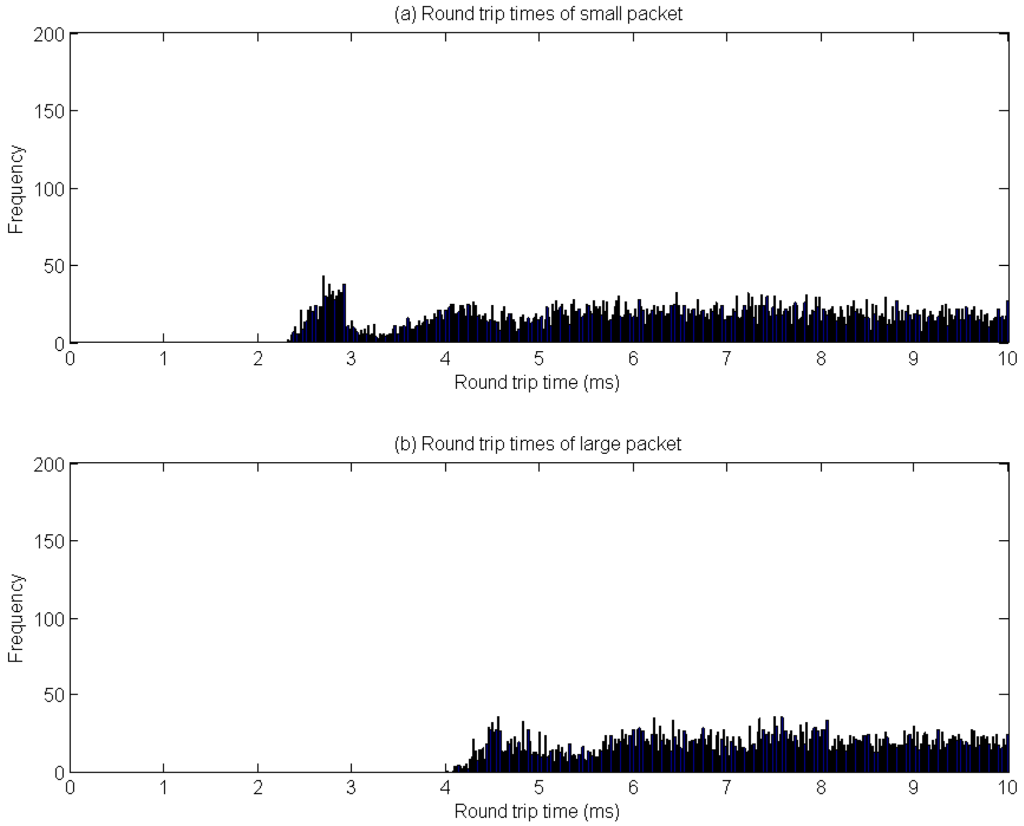

Figure 5.16: Measured RTTs on a "single wireless link" with 2.5 Mbps cross traffic of 40-1500 Byte uniformly distributed packet sizes

Finally, Figure 5.17 shows the RTT histograms for a measurement with 300 bytes constant sized cross traffic. Similar to the results with other cross traffic types, we see distributions at the beginning of the histograms that look similar to the ones from the measurement without any cross traffic. The pattern repeats about every 0.8 ms. However, due to overlap between the patterns, this repetition is less evident than in the case of the 1500 bytes packet cross traffic (Figure 5.15). From the distance between the repeating patterns it can be concluded that it apparently takes about 0.8 ms to send a 300 byte packet over the wireless link.

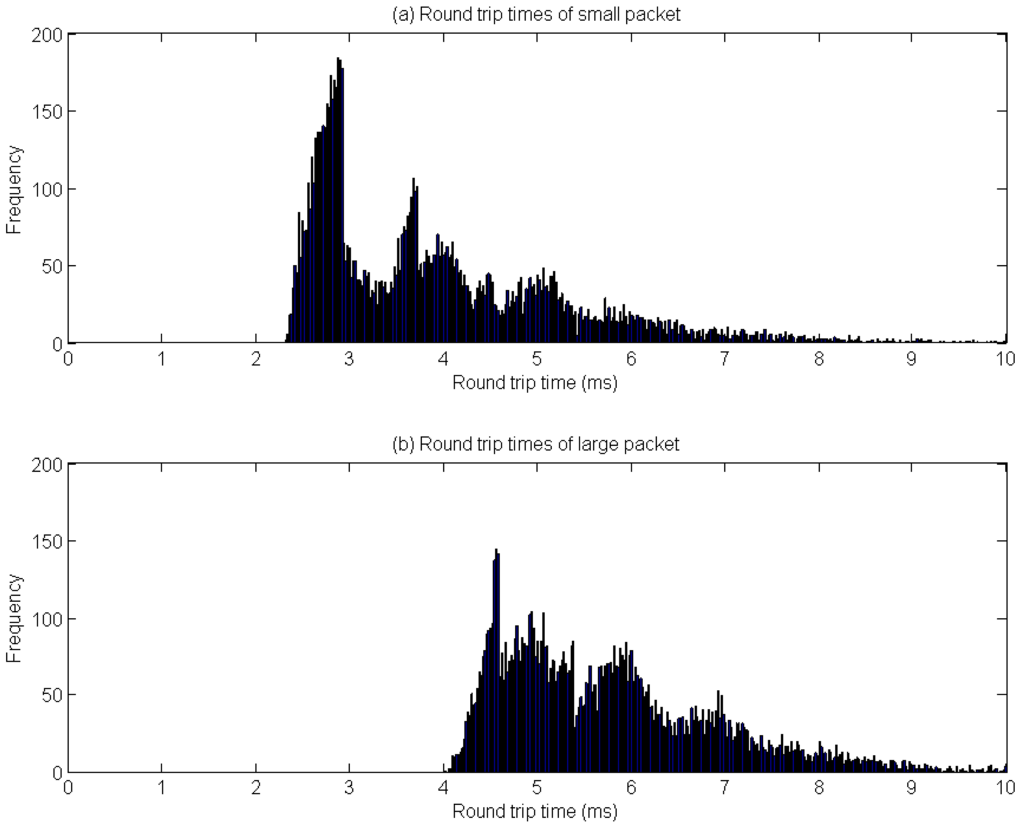

Figure 5.17: Measured RTTs on a "single wireless link" with 0.50 Mbps cross traffic of 300 Byte constant packet size

5.7 Small/large packet probing in a "double wireless link" scenario

In Figure 5.18, histograms are shown of the measured RTTs for probing a path with two wireless links. The network was illustrated in section 4.3.3. The figure shows the histograms for a measurement without any cross traffic.

Figure 5.18a shows a histogram of the RTTs of the small probing packets. The distribution has more or less the shape of an upward pointing triangle. The triangle is bound between 4.8 and 7.0 ms. Figure 5.18b shows a histogram of the RTTs of the large probing packets. It has a shape comparable to the one of Figure 5.18a. The triangle is bound between about 8.3 and 10.7 ms. Another distribution of RTTs with very low amplitude is visible after the initial peak between 11 ms and 13.5 ms.

The triangular shapes of the distributions are expected since both the small and the large probe packet will experience multiple random backoffs during the round trip. The distribution is therefore a multiple convolution of uniformly distributed random backoff times.

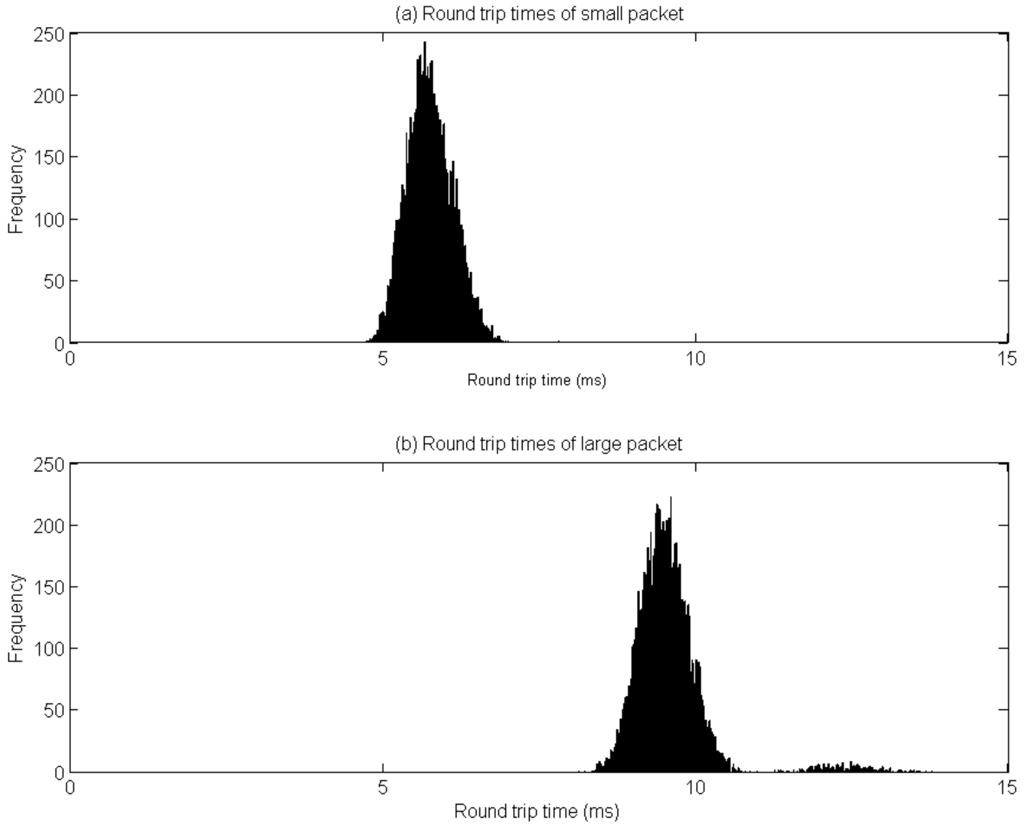

Figure 5.18: Measured RTTs in the "double wireless" network without cross traffic

5.8 Path capacity estimation results using probing method #2

This section presents the path capacity estimation results for the various wireless scenarios, with and without cross traffic, using the small/large packet probing method.

Figure 5.19 shows the estimated path capacities for a scenario with a single wireless link using probing method #2. These are the means and standard deviations after applying the outliers filtering as described in section 4.4. The results are shown for measurements using 100 ppm and for 200 ppm. For all three types of cross traffic the results lie around 6.6 Mbps. This deviates about 0.5 Mbps from the theoretical capacity of 6.1 Mbps as found in section 4.1.2. The value does not vary greatly as a function of cross traffic intensity.

The standard deviation increases as cross traffic intensity increases. In the case of favorable cross traffic, such as type 1 cross traffic (see Table 1), the standard deviation stays below 0.5 Mbps. However, the standard deviation gets larger than 2 Mbps in the scenario with 1.5 Mbps of cross traffic type 3. In the scenario with 3.5 Mbps of cross traffic type 2, both the accuracy of the estimations get so bad that they can not be properly depicted in the figure. The means for 100 ppm and 200 ppm become 3 Mbps and 9 Mbps respectively. The standard deviations become 21 Mbps and 10 Mbps. At this point the tool has become unusable for any kind of application.

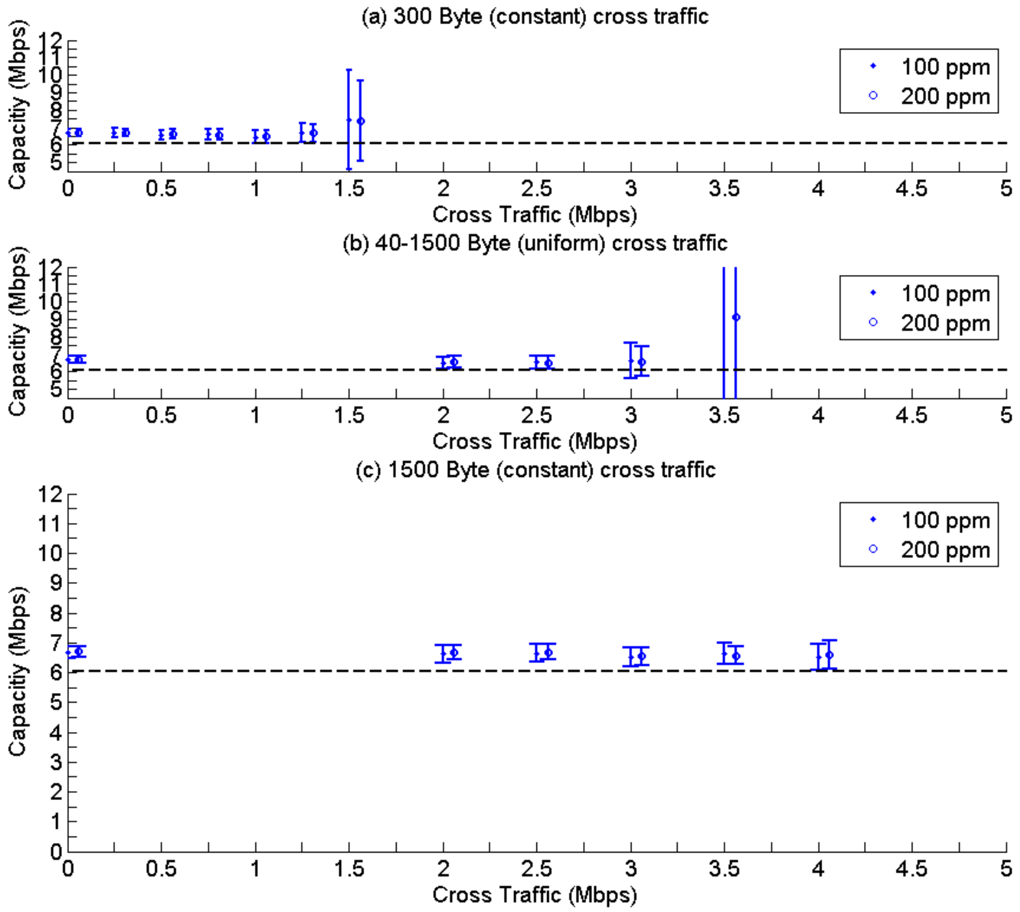

Figure 5.19: Estimated path capacities from probing method #2 for the "single wireless link" scenario, as a function of cross traffic intensity. (a) Cross traffic type 3; (b) cross traffic type 2; (c) cross traffic type 1

In Figure 5.20 the path capacity estimation results are shown for a path with two wireless links. From the analysis in section 4.1.2 it is known that the capacity should be about 3.26 Mbps. It can be seen that under favorable conditions, where cross traffic has a constant packet size of 1500 bytes, the results stay within 0.07 Mbps from the theoretical capacity, and lie around 3.2 Mbps for up to 2.5 Mbps of cross traffic. For this type of cross traffic the standard deviation stays within 0.2 Mbps. However, under less favorable conditions, where cross traffic has a constant packet size of 300 bytes, the obtained result decreases to about 2.6 Mbps at a cross traffic rate of 1.5 Mbps. The standard deviation in this case is about 0.7 Mbps.

It can be concluded that the capacity estimation using small/large packet probing shows good performance under various cross traffic conditions. When the cross traffic has a constant packet size of 1500 bytes, obtained results deviate less than 0.7 Mbps from the theoretical capacity for all scenarios. We know that this deviation is structural and largely caused by the fact that the 'min min' filtering method selects the cases with minimum random backoff, instead of averaging the effect of random backoff. This will be studied in more detail in the following chapter. The standard deviations mostly stay below 0.5 Mbps. However, the performance of the tool gets worse when conditions become less favorable. Especially cross traffic with small packet sizes has a large impact on the performance of the tool. Standard deviations then become quickly larger than 2 Mbps. Especially on a link with a theoretical capacity of about 6 Mbps, such a standard deviation is unacceptable. It would be impossible to make a good decision for admission control.

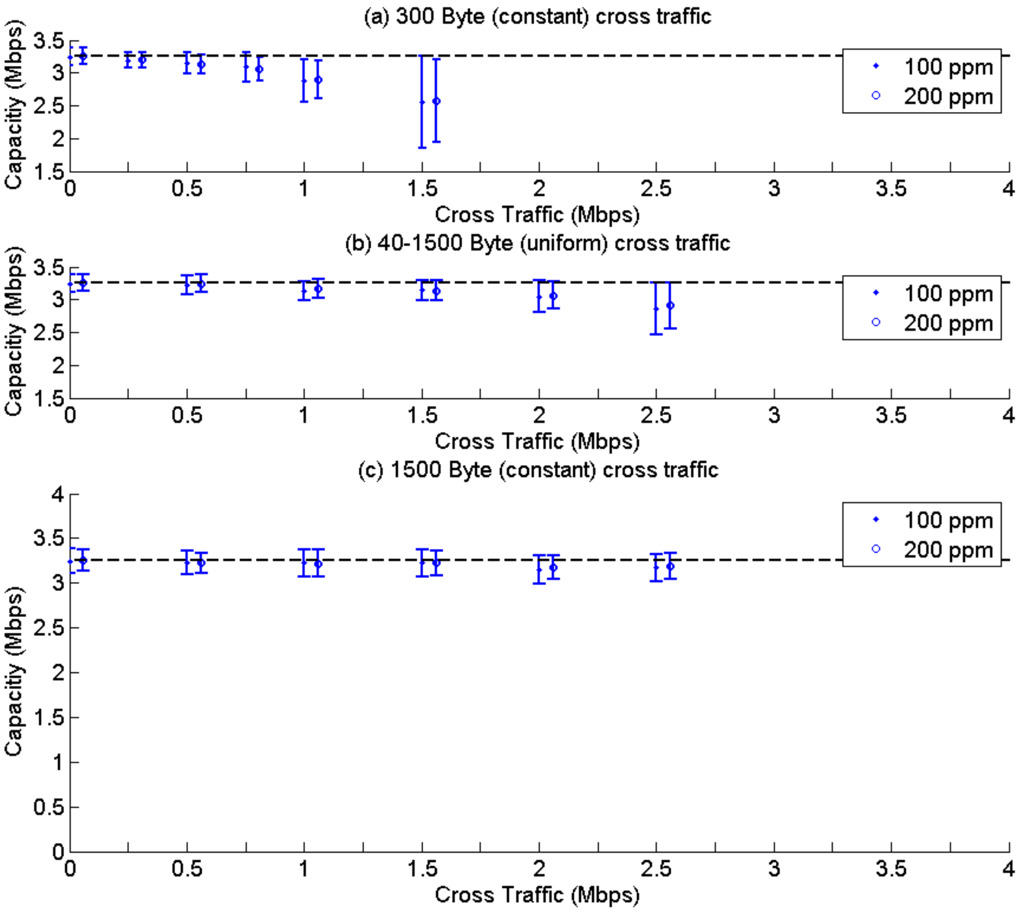

Figure 5.20: Estimated path capacities from probing method #2 for the "double wireless link" scenario, as a function of cross traffic intensity. (a) Cross traffic type 3; (b) cross traffic type 2; (c) cross traffic type 1

6 Improving accuracy of path capacity estimation

Chapter 5 was concluded with the results for the small/large packet probing method for both a path with single wireless link and a path with double wireless link. It was observed that the estimated capacities were about 6.6 Mbps for the single wireless hop scenario and about 3.2 Mbps for the double wireless link scenario. However, in section 4.1.2 it was found that the theoretical capacities for the single wireless link and double wireless link scenarios are about 6.07 and 3.26 Mbps respectively. In this chapter we explain what causes these differences and we suggest new filtering methods to improve the usability of the results.

6.1 Average random backoff vs minimum random backoff

In the calculation for the maximum achievable throughput of wireless links, the value of the average random backoff was responsible for a significant part of the average dispersion between packets. However, when taking the minimum RTTs of both the small and the large probing packets, the delays caused by the random backoff are not taken into account. While for the double wireless link scenario this difference may be dismissed because it is smaller than the standard deviation, for the single wireless link scenario it may not.

From Figure 4.1, Figure 5.14 and Figure 5.18, it can be seen that the value of the average random backoff differs for the small and large probe packets. In fact, the value for the average random backoff of the small probe packet is smaller than the average random backoff of the large probing packets. This is because the larger packet is fragmented into two smaller packets and thus encapsulated into two separate MAC layer frames. The second frame will always experience an extra random backoff period. This extra random backoff period will add to the (average) random backoff in the RTT of the large probing packets.

To determine the maximum achievable throughput for a typical UDP-based application we need to know the average dispersion between subsequent packets sent over the path (in the absence of cross traffic). To obtain an accurate estimate of this average dispersion between packets we should use the difference between the average RTTs of the smaller and larger packets for a scenario without cross traffic.

If we calculate the dispersion from the 10000 probing pairs from the zero cross traffic scenarios described in section 5.5 we obtain the following results:

- Single wireless link (AP connected to 10BASE-T Ethernet link):
  5.724 ms – 3.748 ms = 1.976 ms → (6.072 Mbps)
- Single wireless link (AP connected to 100BASE-TX Ethernet link):
  4.8338 ms – 2.8419 ms = 1.992ms → (6.025 Mbps)
- Double wireless link:
  9.600 ms – 5.7560 ms = 3.843 ms → (3.122 Mbps)

These capacities are very close to the theoretical capacities from section 4.1.2. This supports the idea that using the average RTTs instead of minimum RTTs to estimate the average dispersion leads to more accurate results. However, in practice there will be cross traffic in the network most of the time. The cross traffic will cause the average RTTs of the packets to increase. Taking the average RTTs to estimate the dispersion in a network with cross traffic will result in incorrect estimates. Therefore it is necessary to filter out the RTTs of the packets that were disturbed by cross traffic. In the next sections several filtering approaches will be described.

6.2 Delay by random backoff vs cross traffic

First we investigate the differences between the arrival characteristics of probe packets that were not interfered by cross traffic and probe packets that were interfered by cross traffic.
The histograms of the RTTs for scenarios without cross traffic show a single sharp peak in the case of wired networks (see figures 5.2 - 5.4). For networks containing wireless links the peaks will be wider because the RTTs are spread over multiple bins due to random backoff. When packets are delayed due to cross traffic this will most often result in an RTT that falls outside this region. The higher the cross traffic intensity, the more the histogram will be dominated by packet arrivals delayed by cross traffic. This is illustrated in Figure 6.1.

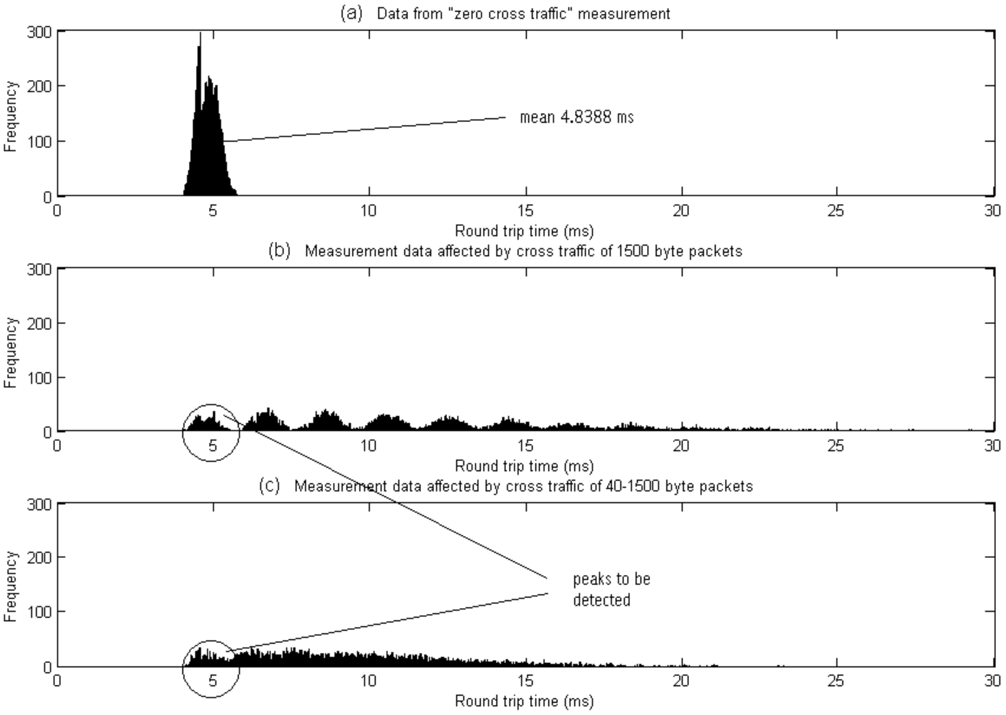

Figure 6.1: RTTs delayed by random backoff become less distinguishable the more cross traffic is present in the network. (a) Large packets over single wirelss link without cross traffic; (b) same scenario with 3 Mbps of cross traffic type 1 ; (c) same scenario with 2.5 Mbps of cross traffic type 2

Figure 6.1a shows a histogram for RTTs measured when there is no cross traffic present in the network. Figures 6.1b and c show histograms of RTTs of probe packets disturbed by different types of cross traffic.
It can be concluded that the RTTs for non disturbed packets and RTTs for disturbed packets show different characteristics. The RTTs for non disturbed packets will concentrate in a “random backoff region” at the left side of the histogram. In order to obtain an estimate of the

average RTT of non disturbed packets it is necessary to distinguish this region from the histogram. Subsequently the average value of the RTTs of non-delayed packets can be estimated from this region.

6.3 Detection of the “random backoff region”

An estimate for the average RTT can be obtained after filtering the region of non disturbed RTTs out of the histograms. Two approaches for this filtering will be described in this section.

6.3.1 Direct histogram peak detection

The peak in the random backoff region of the histogram may be detected directly from the obtained histogram. With 'direct peak detection' is meant that no prefiltering of the histogram is applied. The peak is detected with the use of a peak detection algorithm. The algorithm searches for the first peak in the histogram that meets several criteria. These criteria are specified heuristically.

To determine these criteria, it is necessary to investigate the characteristics of the “random backoff region” in more detail. We know from theory and from measurements that the region either shows a single sharp peak (in wired networks), or a peak shaped by a convolution of multiple uniform distributions. It is also known that the peak is located at the beginning of the histogram. For low to medium levels of cross traffic these peaks can be identified easily, as is suggested in Figure 6.1.

However, the histograms we have shown so far consist of 10000 probe packet pairs. When average RTTs are in the order of 3 ms and 5 ms (as in Figure 6.1) the full measurement will take about a minute to complete. For scenarios with cross traffic, where average RTTs can be a factor 3 or 4 larger, measurements will take in the order of minutes to complete. Therefore, the peak should be identified from a lower number of probes. But when cross traffic levels increase, the peak tends to get less distinguishable. For higher levels of cross traffic the peak can hardly be identified, even when using many ppm. Therefore we determined for all scenarios the minimum number of ppm for which the peak is just becoming visible. Additionally we specified numbers of ppm at which the region starts to show really clearly. These numbers were determined by visual inspection of histograms. Examples of such histograms are shown in Figure 6.2. For the scenario that is illustrated in the figure, the minimal ppm was determined to be 3000 ppm. The threshold at which the peak starts to show really clearly, was determined to be 4000 ppm. Although the determination of these numbers is slightly arbitrarily, the numbers only need to provide an indication for the maximum performance of a peak estimation algorithm. The results are shown in the data sheet in Appendix B. The numbers are used to predict how many ppm will be necessary for a peak detection algorithm to get a reliable estimate.

The numbers that were found indicate that a simple “direct peak detection” method will not give satisfactory performance. First, the peaks can not be determined when higher amounts of cross traffic are present in the path. Secondly, even for moderate amounts of cross traffic, thousands of ppm need to be used in order for a peak to become visible.

Therefore more advanced filtering methods will be necessary to detect the “random-backoff regions” in the histograms.

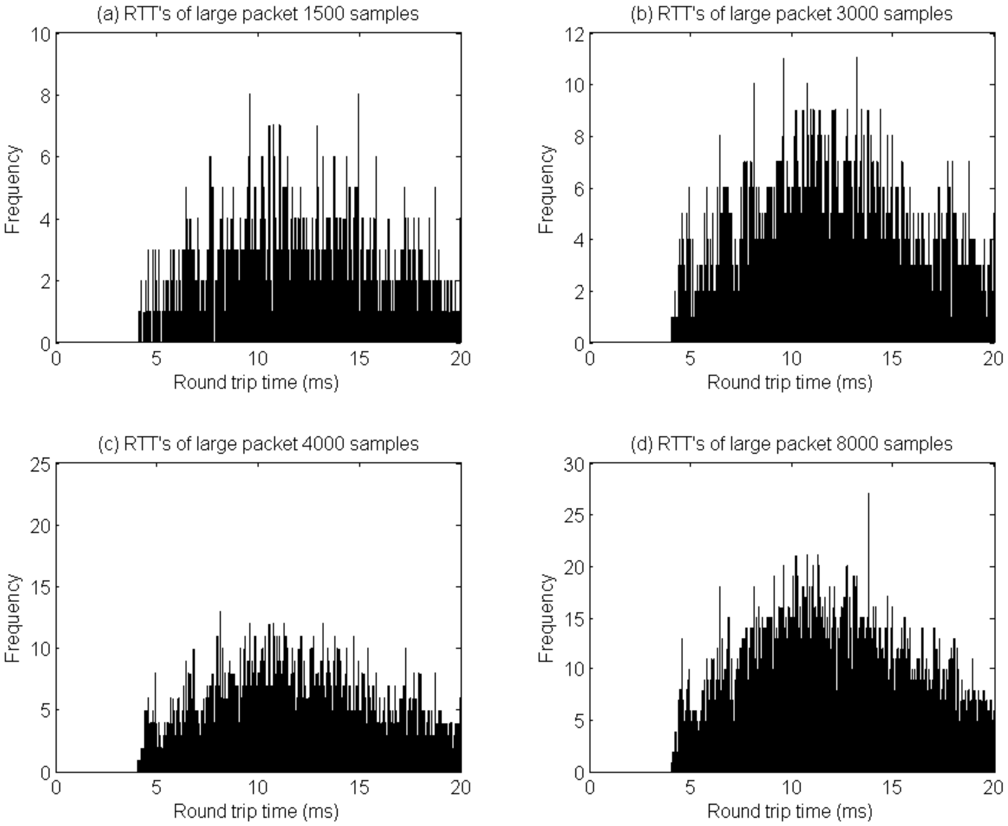

Figure 6.2: Histograms of measured RTTs for various ppm measured for the "single wireless link" scenario with 3 Mbps of 40-1500 byte uniformly distributed packet size cross traffic; the peak to be located lies around 4.8 ms.

Figure 6.2 illustrates how the ppm limits were determined. Histograms for varying ppm were investigated for all scenarios.

6.3.2 Histogram prefiltering

In this section we investigate prefiltering of the histograms to filter out RTTs of packets that were disturbed by cross traffic. In order to do the filtering we need to use the knowledge that we have about the different characteristics of RTTs of undisturbed packets and disturbed packets.

We know that the peaks to be detected are always located at the beginning of the histogram. This is either a very sharp peak, for a path with wired links only, or it is a distinct distribution for a path that also contains wireless links.

In both cases it is possible to filter the histograms using a pre-defined mask that is based on the distributions of the RTTs of non-disturbed packets. When this pre-defined mask is used as a reference in a correlation filter, then the output of this filter will show a distinct peak located at the point where the mask has the highest correlation with the histogram. This should be at the location where we expect to find the average RTT after random backoff. Different types of pre-defined masks may be used in the correlation filter. Preferably, a single mask should be able to filter out any type of cross traffic for measurements performed in any kind of network. This is because in practice, ideally, we do not require any knowledge in advance about the topology of the network. Neither do we require preknowledge about the type of cross traffic that is present in the links.

The design of one or several pre-defined masks is left for further research. We give several suggestions and considerations that can be used for the design of such "pre-defined masks" in section 7.2.

To get an indication of the potential performance of path capacity estimation using prefiltering of the histograms, we applied masks that were obtained from the "zero cross traffic" histograms. This is illustrated in Figure 6.3.

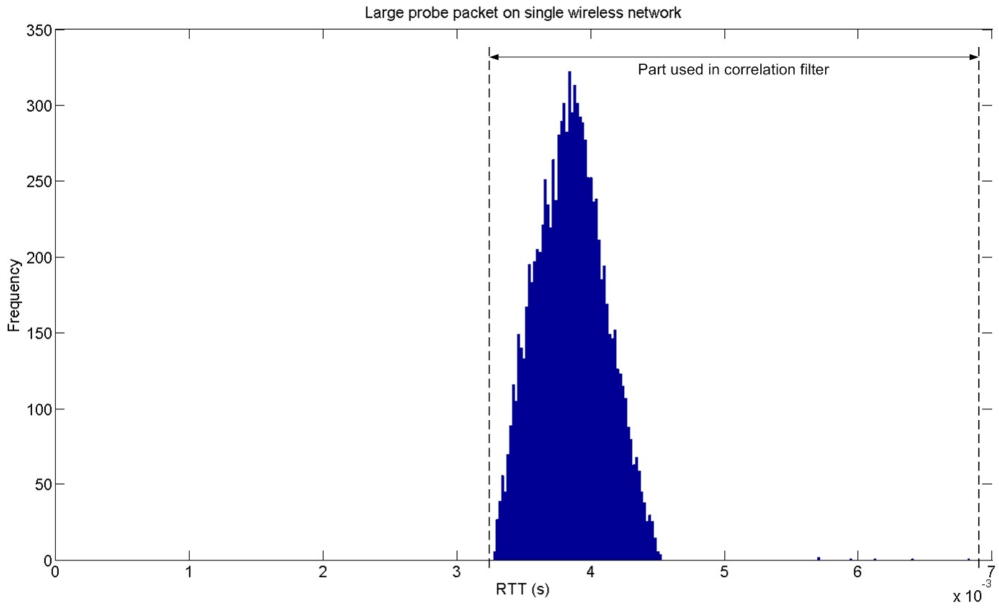

Figure 6.3: RTT histogram for large probe packets for zero cross traffic measurement in the single wireless link scenario.

The figure shows the histogram of the large probe packet RTTs in a single wireless link scenario without cross traffic. The part of the histogram that is used in the filter lies between the two dashed vertical lines. Notice that just before the right vertical line there is a non-zero entry (of height 1).

We calculate the normalized cross correlation between the sample (of 200 probes) and the reference histogram (from the zero cross traffic measurement), with formula 6.1. $R(i)$ is the functional representation of the reference histogram and $\bar{R}$ is the mean value of R. $F(i)$ represents the histogram of the sample measurement where frequency is a function of bin number i. $\bar{F}_n$ is the mean value of the sample F_n from F. F_n has the same size as R.

After applying normalized cross correlation a new histogram is obtained represented by the function $N(n)$. In the new histogram we expect to see a peak at the same location as the peak in the reference histogram if the narrow link is of the same technology as the reference histogram is taken from.

$$N(n)=\frac{\sum_i \left[F(i)-\bar{F}_n\right]\left[R(i-n)-\bar{R}\right]}{\left\{\sum_i \left[F(i)-\bar{F}_n\right]^2\left[R(i-n)-\bar{R}\right]^2\right\}^{0.5}} \tag{6.1}$$

Figure 6.4 illustrates how an original histogram looks and how the resulting histogram looks after applying the cross correlation filter. These are histograms for a measurement over a “single wireless link” path. Figure 6.4a shows a RTT histogram for the small probing packets and Figure 6.4b shows the RTT histogram for the large probing packets. The histograms are composed from only 100 RTTs. Figures 6.4c and d show how the histograms look after performing the normalized cross correlation with the reference histogram of Figure 6.3. From the zero-cross-traffic measurements we know that the average RTTs are 2.842 ms and 4.834 ms for the small and large probing packets respectively. As can be seen in the figure, the peaks of the filtered histograms are located at approximately the right times.

-

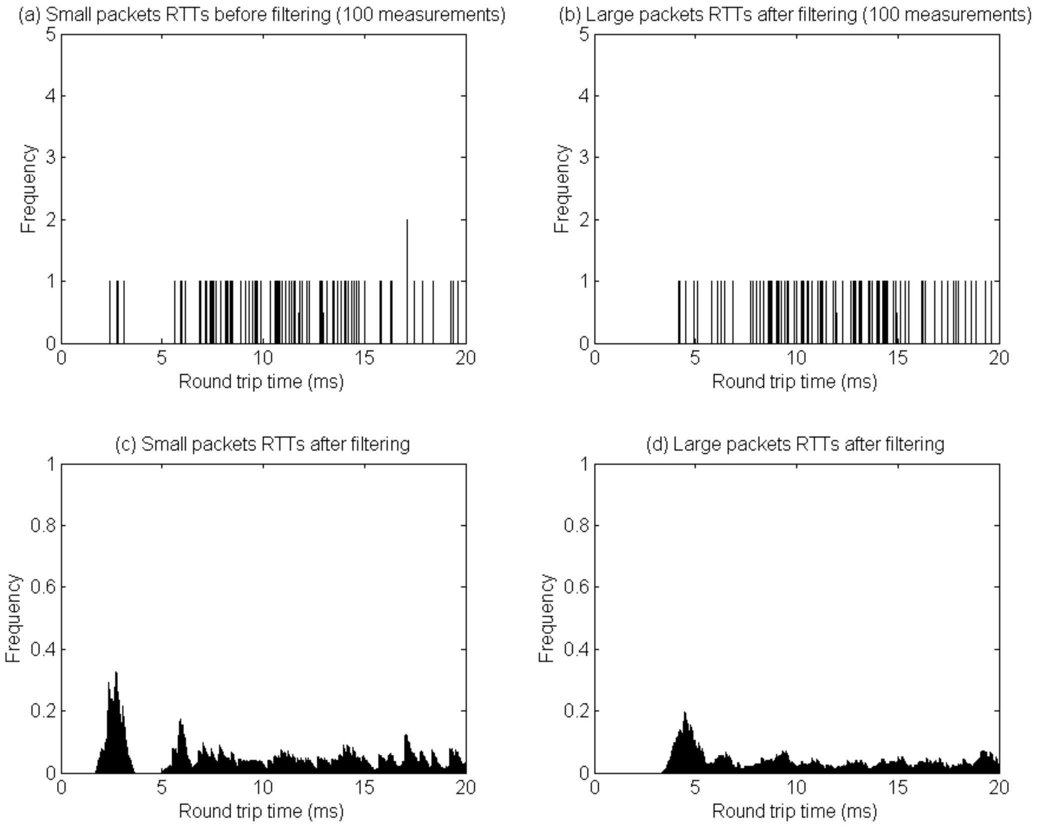

Figure 6.4: Histogram of measured RTTs before and after the use of a “correlation filter” for a “single wireless link” scenario with 3 Mbps 40-1500 byte uniformly-distributed-packetsize cross traffic

In the example of Figure 6.4, we can estimate the average RTTs simply from the global maxima of the filtered histograms. However, this is not always the case. Figure 6.5 shows the initial and filtered histograms for a different scenario. In this case the measurement is done for a "two hop wireless" path. From the zero-cross-traffic measurements we know that the average RTTs of non-delayed packets should be about 5.756 ms and 9.60 ms for the small and large probing packets respectively. In both filtered histograms we do indeed see peaks at approximately these locations. However, in the filtered histogram of the large probing packets it is not the global maximum.

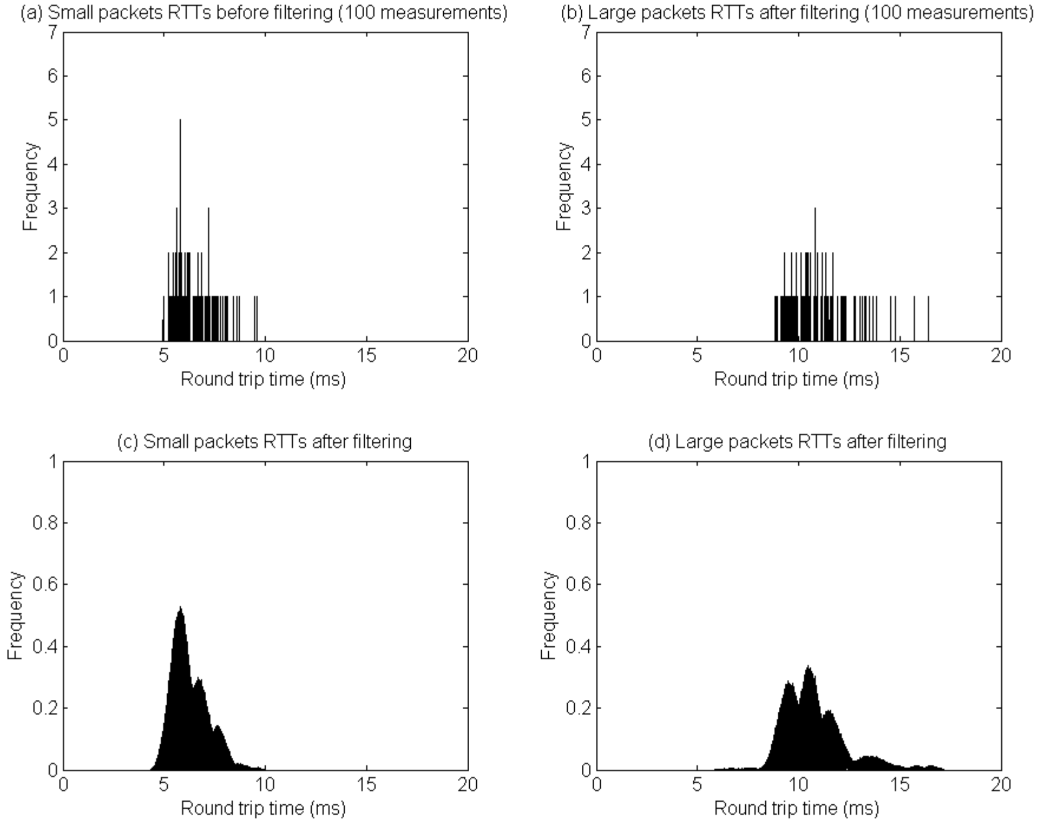

Figure 6.5: Histogram of measured RTTs before and after the use of a "correlation filter" for a "double wireless link scenario with 0.25 Mbps 300 byte constant packet size cross traffic

As was explained in section 5.6, RTT histograms can show a repeating pattern under certain cross traffic conditions. This is especially true for scenarios with cross traffic of fixed packet size. What we can conclude from Figure 6.5 is that in some cases it can happen that one of the later patterns of the histogram shows a higher correlation with the reference histogram. However, it is still possible to estimate the average RTT from the earlier local maximum.

In Figure 6.5d a region with low amplitude values is visible between 6 and 8 ms. This area contains several local maxima. We want the peak estimation to detect the first larger peak in the histogram (at about 9 ms in Figure 6.5d), Therefore the peak that we want to locate should be the first peak in the normalized histogram that meets certain criteria.

The design of a high performance peak estimation method is left for further research. In this work we use a very simple peak detection algorithm that is based on heuristically determined criteria. These criteria assure that the peak has a certain minimum width and height (so that it will ignore local maxima in the "low amplitude region"). The peak detection algorithm is shortly described in the following pseudocode.

```
1   find next local peak (start search at beginning of histogram)
2   if peak meets specified criteria
3       peak position = average RTT estimate;
4       break;
5   else
6       find next local peak;
7       go back to step 2;
```

The criteria that were specified look as follows:

1. The peak should be the maximum over a range of 600 μs
2. The height of the histogram should be 0.005 points lower than the peak value at the limits of this range

After using the algorithm on a filtered histogram we obtain an estimate for the average RTT. Finally, when average RTTs have been found for both the small and large probe packets, then the estimate for the path capacity is calculated using the following formula:

$$C_p = L_s / ((\overline{RTT_l}) - (\overline{RTT_s})) \qquad (6.2)$$

In this formula, L_s is the packet size equal to the path MTU and thus equal to the size of the small probe packet. $\overline{RTT_l}$ is the estimate of the average RTT of large probe packets and $\overline{RTT_s}$ is the estimate of the the average RTT of small probe packets.

6.4 Estimation results

In this section we present the capacity estimation results after filtering of the histograms and peak detection. We will refer to this method as 'peak estimation' method. The results are compared with the estimation results from the small/large packet probing method using the 'min min' filtering.

Single wireless link

In Figure 6.6 the results are shown for the path capacity estimation in a network with a single wireless link. Again, each node in the figure represents the mean and the standard deviation for 50 capacity estimates after applying the outliers filtering as described in section 4.4. The results were obtained from 200 ppm. 'Peak est' refers to the 'peak estimation' method.

From the figure it can be seen very clearly that the 'min min' filtering method always gives higher values for the path capacity than the peak estimation method. This is due to the tendency of the 'min min' method to underestimate the dispersion, as was explained in section 6.1.

Also it can be seen that the standard deviations do not differ much between the two methods, at least not for the cases with cross traffic of type 1 (see Table 1 on p.29) and for cross traffic of type 2 up to 3.0 Mbps. The means of the estimates in these cases lie around 6.0 Mbps and 6.6 Mbps for the 'peak estimation' method and the 'min min' method respectively. From 3.5 Mbps of type 2 cross traffic, the performance of both methods becomes very low. In this case the capacities become 5 Mbps and 9 Mbps for the 'peak est' and 'min min' filtering methods respectively. The standard deviations for the two methods become 10 Mbps and do not fit into the plotted window. In this case both methods become unusable for any application.

Also for the scenario with 1.5 Mbps of type 3 cross traffic, the standard deviation of both methods becomes really high, namely 2.8 Mbps and 2.3 Mbps for the 'peak est' and 'min min' methods respectively. The capacity estimates for this scenario are 6.7 Mbps and 7.4 Mbps.

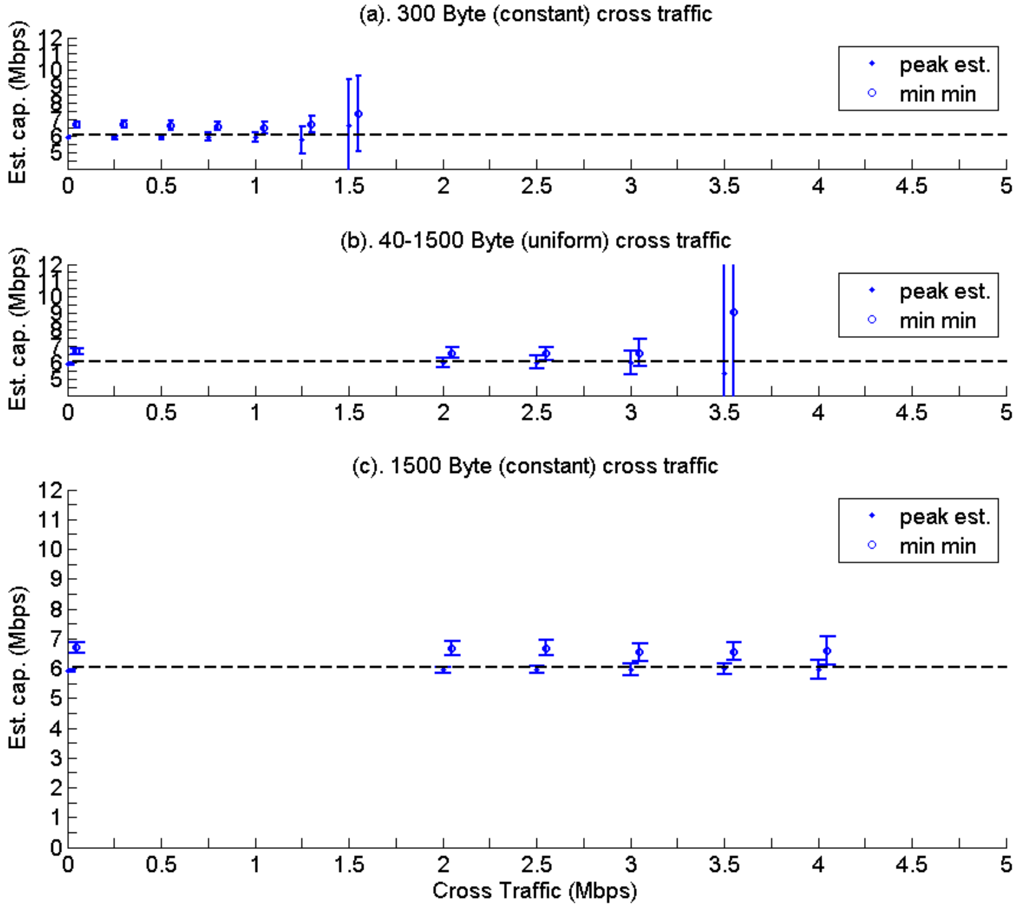

Figure 6.6: Estimated path capacities with standard deviations for the "single wireless link" scenario for different types of cross traffic. (a) Cross traffic type 3; (b) cross traffic type 2; (c) cross traffic type 1

Double wireless link

In Figure 6.7 the results are shown for path capacity estimation in a path with a double wireless link. Again, the 'min min' filtering method gives higher capacity estimates than the 'peak estimation' filtering method. In case of cross traffic type 2 and cross traffic type 3, the precision/consistency of the 'peak estimation' method decreases more rapidly with the amount of cross traffic compared with the precision of the 'min min' method. This can be seen from the faster growing standard deviations in case of the 'peak estimation' method. Additionally, the values of the path capacity estimates of both filtering methods decrease with the amount of cross traffic. This can be explained due to the higher probability for the large probe packet to be delayed compared with the small probe packet. This makes overestimation of the dispersion (and thus underestimation of the capacity), more likely.

With zero cross traffic, the results of the 'peak estimation' and 'min min' methods are about 3.06 ± 0.4 Mbps and 3.25 ± 0.12 Mbps respectively. For cross traffic type 1, the results become 3.01 ± 0.10 Mbps and 3.18 ± 0.14 Mbps. With 2.5 Mbps of type 2 cross traffic, the results of the 'peak estimation' and 'min min' methods become 2.6 ± 0.8 Mbps and 2.9 ± 0.4 Mbps respectively. Finally, for the scenario with 1.5 Mbps of type 3 cross traffic, the results are 2.5 ± 0.9 Mbps and 2.6 ± 0.6 Mbps respectively.

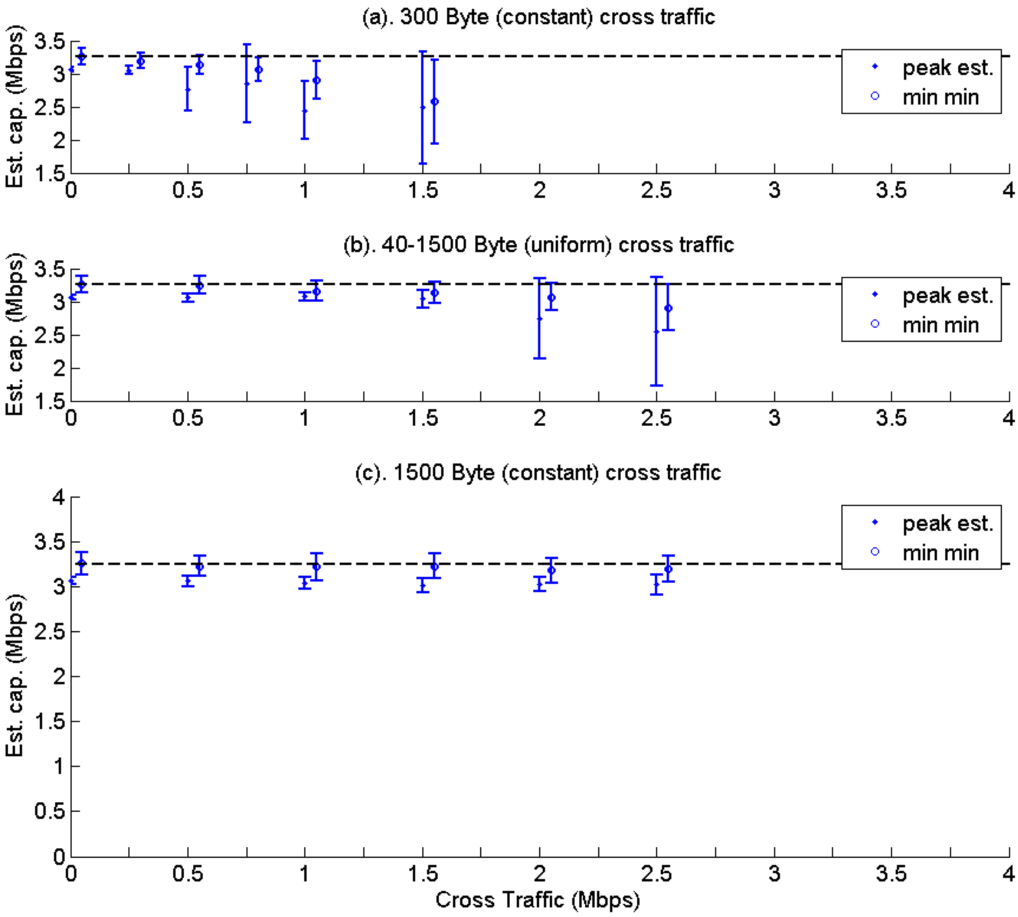

Figure 6.7: Estimated path capacities for the "double wireless link " scenario for different types of cross traffic. (a) Cross traffic type 3; (b) cross traffic type 2; (c) cross traffic type 1

An overview of all results, including the average convergence times, for 'min min' filtering and 'peak estimation' filtering is given in the tables 3 ("single wireless" path) and 4 ("double wireless" path). The tool converges within 5 seconds for almost all types and amounts of cross traffic. These convergence times are much shorter than convergence times of currently existing tools [69], which can be in the order of minutes for comparable amount of cross traffic.

Single wireless link						
		'min min'		'peak estimation'		Convergence time for 200 ppm (seconds)
Cross traffic type	Cross traffic rate (Mbps)	Mean (Mbps)	Standard deviation (Mbps)	Mean (Mbps)	Standard deviation (Mbps)	
Type 1	0	6.7	0.18	5.92	0.05	1.54
	2	6.67	0.25	5.94	0.12	2.26
	2.5	6.68	0.26	5.96	0.13	2.57
	3	6.54	0.29	5.96	0.18	2.97
	3.5	6.56	0.31	5.99	0.19	3.61
	4	6.6	0.48	5.97	0.32	4.46
Type 2						
	2	6.57	0.33	5.97	0.3	2.8
	2.5	6.52	0.39	5.99	0.39	3.63
	3	6.58	0.84	5.97	0.72	5.05
	3.5	9.11	10.21	5.37	10.42	8.4
Type 3						
	0.25	6.69	0.21	5.88	0.1	1.72
	0.5	6.62	0.27	5.88	0.13	1.96
	0.75	6.59	0.28	5.94	0.27	2.29
	1	6.46	0.36	5.92	0.3	2.75
	1.25	6.68	0.49	5.75	0.85	3.46
	1.5	7.37	2.29	6.66	2.78	4.79

Table 3: Summary of the results of the small/large packet probe method using the 'min min' and the 'peak estimation' filtering methods, for "single wireless link" scenario

Double wireless link						
		'min min'		'peak estimation'		Convergence time for 200 ppm (seconds)
Cross traffic type	Cross traffic rate (Mbps)	Mean (Mbps)	Standard deviation (Mbps)	Mean (Mbps)	Standard deviation (Mbps)	
Type 1	0	3.253	0.123	3.059	0.038	3.07
	2	3.224	0.113	3.059	0.061	3.34
	2.5	3.218	0.150	3.032	0.064	3.65
	3	3.225	0.138	3.009	0.076	4.04
	3.5	3.176	0.134	3.025	0.081	4.51
	4	3.189	0.142	3.021	0.110	5.14
Type 2						
	2	3.244	0.133	3.054	0.069	3.46
	2.5	3.160	0.146	3.072	0.066	3.96
	3	3.134	3.119	3.039	0.139	4.64
	3.5	3.063	0.206	2.739	0.610	5.60
Type 3						
	0.25	3.195	0.122	3.051	0.064	3.45
	0.5	3.132	0.147	2.764	0.331	3.92
	0.75	3.057	0.181	2.845	0.585	4.56
	1	2.898	0.283	2.445	0.445	4.56
	1.5	2.575	0.630	2.487	0.847	9.50

Table 4: Summary of the results of the small/large packet probe method using the 'min min' and the 'peak estimation' filtering methods, for "double wireless link" scenario

6.5 Reliability calculation

For a well-informed decision with respect to admission control, the probing device should not only yield a value for the available bandwidth, but also the precision (standard deviation) of the value. So far we have determined the standard deviation by performing 50 measurements of 200 ppm.

In practice, the capacity measurement tool will only perform a single measurement. Therefore it should estimate the reliability of the results from a single measurement. We have tried various ways of determining a standard deviation from the convergence rate, and compared the results to the values following from 50 measurements, but did not find any correlation. This is therefore left for further research.

7 Conclusions and future work

The main objective of the research is to develop and test a low intrusive and fast available-bandwidth measurement tool for heterogeneous, best-effort, small-scale IP networks. This implies that the tool should be able to measure available bandwidth of a path irrespective of the link layer technologies used throughout the network. Application is aimed at the typical home network, and therefore the expected number of hops in the path is about two. Additionally, the tool should only require additional implementation at the measuring device (sender). This means that there must not be any further requirements on the devices on the other side of the path (responder) beside the presence of a standard IP stack.

7.1 Conclusions

Measuring available bandwidth from a single node in a heterogeneous environment has proven to be a challenging task. The most difficult step in measuring the available bandwidth is determining the path capacity when cross traffic is present. This work has shown how path capacities can be measured involving a new tool implemented at a single node. It takes typical a few seconds or less for a measurement to complete. The performance of the tool in terms of accuracy and convergence time is strongly depending on the level and type of cross traffic throughout the path. Additionally the performance is heavily depending on the link layer technologies present in the path. Therefore, the focus of the research went to networks containing wireless links.

The “peak estimation” method indeed shows hardly any bias in the estimations compared to the 'min min' method. Determining the path capacity from the minimum round-trip time of probe packets (the "min min" method) leads to a structural overestimation. This could be tackled by introducing the "peak estimation" method, which filters the RTTs of the packets that are delayed by random back-off only from the RTTs that are delayed by cross traffic also, after which the average RTT is determined instead of the minimum. The filtering can successfully be performed by correlating the measured RTT histograms with various zero-cross-traffic histograms.

The precision of both methods is found to be better than 1 Mbps for cross traffic intensity ratios up to 50% (this ratio is the amount of cross traffic on the narrow link divided by the path capacity). The precision also depends on the type of cross traffic. The larger the packet size of the cross traffic, the smaller the standard deviation in the path capacity estimation. We found that, mostly the peak estimation method shows both, better accuracy and smaller standard deviation under favorable cross-traffic conditions in comparison with the “min min” filtering method. However, when cross-traffic conditions get more challenging, then the “min min” method outperforms the “peak est” method in terms of accuracy. A disadvantage of the "peak estimation" method is that it requires more resources.

An accuracy of about 1 Mbps should be sufficient for e.g. controlling IPTV streams in the network but for lower bitrate streams, such as VoIP streams, the resolution is insufficient.

It has been shown that for wired networks using the minimum obtained RTTs of both packets in a packet pair is a promising and simple approach for filtering results delayed by cross-traffic and avoiding the post-narrow-link problem. Through simulations it was shown that this approach performs better than existing techniques such as estimating the dispersion from the pair with the minimal “sum of RTTs” or estimating the dispersion from the median of the RTTs.

It has been shown that path capacity estimation using standard packet pair probing is not applicable for sender-based measurements on paths containing shared media. The cause for this was found to be the contention between the second probe packet and the reply on the first probe packet in shared media.

A solution to this problem was found by using two different sized probe packets. First the RTT for a MTU sized packet is determined. Subsequently the RTT for a larger than MTU sized packet, that is fragmented into two MTU sized packet because of the MTU limit, is determined. The difference between the minimal obtained RTTs is used for the “narrow-link capacity” estimation.

Further it became clear that there are no generally accepted definitions for the metrics of capacity and available bandwidth in network paths. Therefore, it is not always clear which exact metric various tools try to obtain from their measurements.

7.2 Future work

The reliability of the results of the newly developed path capacity estimation tools is determined by performing 50 measurements. In practice, only one measurement of about 200 ppm can be done. A method needs to be developed to determine the precision of the results from this single measurement.

In this work, the performance of the concept has been evaluated through simulation of different networks. The simulated networks have ideal performance at some points, which will be different in real networks. For example, the simulated “probe traffic sender” and “probe traffic receiver” nodes do not implement any protocol stack processing time. Also, the wired and wireless media are error free. Since the differences in real networks will probably affect the RTTs of packet, the performance of the tool needs to be evaluated in real networks too.

In order to do the histogram prefiltering in the 'peak estimation' method we use as a mask the RTT histogram obtained from a measurement without cross traffic. In practice it will not always be possible to obtain such a mask. Therefore it will be necessary to use one or more pre-defined mask(s).

The pre-defined masks could be designed such that they look similar to various expected RTT distributions, depending on the link technologies used throughout the path. Either pre-knowledge of the path can be used to decide what filter to use, or a decision could be made based on the comparison of the results from multiple pre-defined masks. The predefined masks may be remotely managed by a service provider.

The evaluated networks in this work are all simulated under static conditions. For example, the rate of operation of the wireless links are 11 Mbps during the full simulations. In practice, technologies, such as those from the 802.11 family and Homeplug AV, incorporate rate adaptation schemes. It might be possible that during a measurement the operational rate of a link changes. Further research is necessary to determine the impact of these rate adaptations schemes. Improvement of the measurement tools might be necessary to overcome possible problems caused by rate adaptation schemes.

Also, the tool only provides a snapshot of the state of a path. How frequent a measurement needs to be repeated depends on the characteristics of the cross traffic and the demands of the application. Not much is known yet about the typical characteristics of cross traffic in home networks.

The networks that have been simulated during this work, consisted of symmetrical links only. This mean that the capacity is equal in both directions. However, network technologies such as HomePlug AV support asymmetrical links also. Although we expect that our path capacity estimation tools can cope with asymmetrical links, verification is necessary.

Many of the newer home network technologies provide solutions for QoS based on classification and prioritization of services. Although we expect that our tool will be able to measure path capacity in such QoS enabled networks. Verification of the performance of the tool in such networks is necessary. Service classification and prioritization based QoS schemes will possibly affect available bandwidth for services with different priority, but the capacity of the path will probably not be affected.

Appendix A. MATLAB script for determination of average random backoff

```
%Determine  average  random  backoff.  Two  saturated  nodes  are  constantly
contending for the medium.
%The  initial  CW  is  32.  After  each  subsequent  collision  this  window  is
doubled, with a maximum of 1024. (Thus 5
%collisions  in  a  row.  The  probability  that  this  happens  is  about
1/32*1/64*1/128*1/256*1/512*1/128 = 2.8e10-14

tic %start time measurement
clear CW
clear CW64
clear CW128
clear CW256
clear CW512
clear CW1024
clear RB_array

%Calculate in two stages so that memory has to maintain an array of maximum
%10000 entries long
N = 1000;
O= 10000;
%generate initial values from uniform distributed integers between 0 and 32
RB1 = floor(random('unif',0,32));
RB2 = floor(random('unif',0,32));

%This variable serves to keep track the number number of retransmission
%in order to adjust the CW correspondingly
CW=32;      %CW=32

%Just to keep track of the number of occurrences of the larger CWs. If CW >
%32  (retry_count  >=  1)  then  a  collision  occurred.  We  need  to  know  the
number of
%collisions to calculate the average number of packet retransmission.
CW64=0;     %CW=64
CW128=0;    %CW=128
CW256=0;    %CW=256
CW512=0;    %CW=512
CW1024=0;   %CW=1024

for j=1:O
    %generate arrays with uniform distributed integers between 0 and 32
    RB_array1 = floor(random('unif',0,32,[1,N]));
    RB_array2 = floor(random('unif',0,32,[1,N]));
    for i=1:N
        %if RBs are equal, a collision occurs. The CW is doubled. When a
        %collision occurs, no packet has been successfully
        %transmitted. Also, because CW is not equal to 32, generate two new
        %random integers that fall into the new CW.
        if RB1==RB2;
            %CW cannot become larger than 1024
            if CW > 512
```

```
                CW=1024;
            end
            RB_array(i)=RB1;
            CW = CW*2;        %if transmission, then CW is doubled
            RB1 = floor(random('unif',0,CW-1));
            RB2 = floor(random('unif',0,CW-1));
        %If one node wins the contention, then the CW can be maintained (or 
put
        %back) to 32. In this case a packet is successfully transmitted.
        elseif RB1>RB2
            RB_array(i)=RB2;
            RB1=RB1-RB2;
            RB2=RB_array2(i);
            CW=32;
        %If one node wins the contention, then the CW can be maintained (or 
put
        %back) to 32. In this case a packet is successfully transmitted.
        else
            RB_array(i)=RB1;
            RB2=RB2-RB1;
            RB1=RB_array1(i);
            CW=32;
        end
        if CW ~= 32
            if CW == 64
                CW64=CW64+1;
            elseif CW == 128
                CW128=CW128+1;
            elseif CW == 256
                CW256=CW256+1;
            elseif CW == 16
                CW512=CW512+1;
            else
                CW512=CW512+1;
            end
        end
    end
    end_array(j)=mean(RB_array);
end
end_array2(k)=mean(end_array)


CW64
CW128
CW256
CW512
CW1024
mean(end_array)
toc      %calculation of 10.000.000 RB phases takes about 90 s on an AMD64 
4800+
```

Appendix B. Minimum numbers of ppm for peak detection

	Single wireless link		Double wireless link	
Cross traffic type	Cross traffic rate (Mbps)	Minimum ppm	Cross traffic rate (Mbps)	Minimum ppm
Type 1	0.0	10	0.0	15
	2.0	20	0.5	25
	2.5	30	1.0	30
	3.0	30	1.5	225
	3.5	70	2.0	300
	4.0	170	2.5	350
Type 2				
	2.0	170	0.5	200
	2.5	500	1.0	1000
	3.0	3000	1.5	4000
	3.5	X	2.0	X
Type 3				
	0.3	150	0.3	600
	0.5	200	0.5	800
	0.8	400	0.8	6000
	1.0	900	1.0	X
	1.5	5000	1.5	X

Bibliography

[1] Venkatesh, A., Kruse, E. and Chuan-Fong Shih, E., "*The Networked Home: An Analysis of Current Developments and Future Trends*", CRITO, (2001)

[2] HAVi organization, "*HAVi, the A/V digital network revolution*", White paper, (1999)

[3] Delphinanto, A., Koonen, A.M.J. and den Hartog, F.T.H., "*Improving Quality of Experience by Adding Device Resource Reservation to Service Discovery Protocols*", IEEE ICC'08 International Conference on Communications, Beijing, (May 2008)

[4] Paxson, V., Almes, G., Mahdavi, J. and Mathis M., IETF RFC 2330: "*Framework for IP Performance Metrics*", (May 1998)

[5] Mathis, M. and Allman, M., IETF RFC 3148: "*A Framework for Defining Empirical Bulk Transfer Capacity Metrics*", (July 2001)

[6] Chimento, P. and Ishac, J., IETF RFC 5136: "*Defining Network Capacity*", (February 2008)

[7] Dovrolis, C., Parameswaran, R. and Moore, D., "*What do Packet Dispersion Techniques Measure?*", IEEE INFOCOM'01, Anchorage, AK, USA, (2004)

[8] Jacobson, V., "*Congestion Avoidance and Control*", SIGCOMM'88 Symposium on Communication Architectures and Protocols, Stanford, CA, USA, (August 1988)

[9] Bellovin, S.M., "*A Best-Case Network Performance Model*", (1992)

[10] Carter, R.L. and Crovella, M.E., "*Dynamic Server Selection using Bandwidth Probing in Wide-Area Networks*", Technical Report, (March 1996)

[11] Jacobson, V., "*Pathchar: A Tool to Infer Characteristics of Internet Paths*", University of California at Berkeley, (April 1997)

[12] Chen, L.-J., Sun, T., Yang, G., Sanadidi, M.Y. and Gerla, M., "*Ad hoc probe: path capacity probing in wireless ad hoc networks*", WICON'05, Los Angeles, CA, USA, (July 2005)

[13] Dovrolis, C., Ramanathan, P. and Moore, D., "*Packet-dispersion techniques and a capacity-estimation methodology*", IEE/ACM transactions on networking, Volume 12, p.963-977, Atlanta, GA, USA, (December 2004)

[14] Goutelle, M. and Primet, P.V.B., "*Study of a non-intrusive method for measuring the end-to-end capacity and useful bandwidth of a path*", ICC'04, Lyon, France, (June 2004)

[15] Kapoor, R., Chen, L.J., Lao, L., Gerla, M. and Sanadidi, M.Y., "*CapProbe: a simple and accurate capacity estimation technique*", ACM SIGCOMM Computer Communication Review, (October 2004)

[16] Kapoor, R., Chen, L.J., Lao, L., Gerla, M. and Sanadidi, M.Y., "*CapProbe: a simple and accurate capacity estimation technique for wired and wireless environments*", ACM SIGMETRICS Performance Evaluation Review, (June 2004)

[17] Lai, K. and Baker, M., "*Measuring bandwidth*", INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings.

IEEE, (March 1999)

[18] Li, M., "*Using Bandwidth Estimation to Optimize Buffer and Rate Selection for Streaming Multimedia over 802.11 Wireless Networks*", Ph. D. Thesis, (2006)

[19] Paxson, V, "*Measurements and Analysis of End-to-End Internet Dynamics*", Technical Report, (1997)

[20] Di Pietro, A. Ficara, D.,Giordano, S., Oppedisano, F. and Procissi, G, "*PingPair: A Lightweight Tool for Measurement Noise Free Path Capacity Estimation*", ICC'08. IEEE International Conference on Communications, Beijing, (May 2008)

[21] ZiXuan, Z., Lee, B.S. Fu, C.P. and Song, J., "*Packet triplet: a novel approach to estimate path capacity*", IEEE Communications Letters, (2005)

[22] Mah, B.A., "*Pchar: Child of pathchar*", DOE NGI testbed workshop, Berkeley, CA, (1999)

[23] Pásztor, A. and Veitch, D., "*Active Probing Using Packet Quartets*", ACM SIGCOMM Workshop on Internet measurement, Marseille, France, (2002)

[24] Ningning Hu; Steenkiste, P., "*Improving Quality of Experience by Adding Device Resource Reservation to Service Discovery Protocols*", IEEE Journal on Selected Areas in Communications, Volume 21, p.879 - 894, (August 2003)

[25] Jain, M. and Dovrolis, C., "*Pathload: A Measurement Tool for End-to-End Available Bandwidth*", In Proceedings of Passive and Active Measurements Workshop, (2002)

[26] Johnsson, A., Melander, B. and Björkman, M., "*Bandwidth Measurement in Wireless Networks*", Springer Boston, (July 2006)

[27] Melander, B. Bjorkman, M and Gunningberg, P., "*A new end-to-end probing and analysis method for estimating bandwidth bottlenecks*", GLOCOM'00, San Francisco, CA, USA, (November 2000)

[28] Ribeiro V.J., Riedi R.H., Baraniuk R.G., Navratil J. and Cottrell L., "*pathChirp: Efficient Available Bandwidth Estimation for Network Paths*", San Diego, CA, USA, (April 2003)

[29] Hu, N. and Steenkiste, P., "*Estimating Available Bandwidth Using Packet Pair Probing*", (Sep. 2002)

[30] Ribeiro, V., Coates, M., Riedi, R., Sarvotham, S., Hendricks, B. and Baraniuk, R., "*Multifractal cross-traffic estimation*", ITC Specialist Seminar on IP Traffic measurement, (2000)

[31] Strauss, J., Katabi, D. and Kaashoek, F., "*A Measurement Study of Available Bandwidth Estimation Tools*", ACM SIGCOMM'03, (2003)

[32] Lakshminarayanan, K., Padmanabhan, V.N. and Padhye, J., "*Bandwidth Estimation in Broadband Access Networks*", ACM, Taormina, Sicily, Italy, (2004)

[33] Kazantzidis, M., Maggiorini, D. and Gerla, M., "*Network Independent Available Bandwidth Sampling and Measurement*", Springer, (2003)

[34] Davis, M., "*A Wireless Traffic Probe for Radio Resource Management and QoS*

provisioning in IEEE 802.11 WLANs", ACM, Venice, Italy, (2004)

[35] Lee, H.K., Hall, V., Ki, H.Y., Kyoung Ill, K. and Eun Jung, K., "*Bandwidth Estimation in Wireless Lans for Multimedia Streaming Services*", Advances in Multimedia, Volume 2007, p.9-9, (2007)

[36] Lee, H., Kim, S., Lee, O., Choi, S. and Lee, S.J., "*Available Bandwidth-Based Association in IEEE 802.11 Wireless LANs*", ACM, Vancouver, British Columbia, Canada, (2008)

[37] Sarr, C., Chaudet, C., Chelius, G. and Lasous, I.G., "*A node-based available bandwidth evaluation in IEEE 802.11 ad hoc networks*", IEEE, (July 2005)

[38] Chen, L.-J., Sun, T., Yang, G., Sanadidi, M.Y. and Gerla, M., "*Ad hoc probe: path capacity probing in wireless ad hoc networks*", Los Angeles, CA, USA, (July 2005)

[39] Amamra, A. and Hou, K.M., "*SLOT: A Fast and Accurate Technique to Estimate Available Bandwidth in Wireless IEEE 802.11*", IEEE, (April 2008)

[40] Amamra, A., Hou, K.M. and Chanet, J.P., "*Evaluation of the performance of SloPS: Available Bandwidth Estimation Technique in IEEE 802.11b Wireless Networks*", Springer Netherlands, (2007)

[41] Bredel, M. and Fidler, M., "*A Measurement Study of Bandwidth Estimation in IEEE 802.11g Wireless LANs Using the DCF*", Springer Berlin / Heidelberg, (2008)

[42] Sundaram, N., Connor, W.S. and Rangarajan, A., "*Estimation of Bandwidth in Bridged Home Networks*", IEEE, (April 2007)

[43] Johnsson, A., Bjorkman, M. and Melander, B., "*An Analysis of Active End-to-end Bandwidth Measurements in Wireless Networks*", IEEE, (April 2006)

[44] Botta, A., Pescape, A. and Ventre, G, "*On the performance of bandwidth estimation tools*", IEEE, (August 2005)

[45] Urvoy-Keller, G., En-Najary, T. and Sorniotti, A., "*Operational comparison of available bandwidth estimation tools*", ACM, (January 2008)

[46] IEEE Std 802.3-2005, "*Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*", (2005)

[47] http://www.ieee802.org/3/

[48] IEEE Std 802.11b-1999, "*Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Higher-Speed Physical Layer Extension in the 2.4 GHz Band*", (1999)

[49] Choi, S., Del Prado, J. and Sherman, M., doc. IEEE 802.11-01/055: "*802.11a and 802.11b Maximum Throughput for Simulation Model Conformance*", (January 2001)

[50] Heusse, M., Rousseau, F., Berger-Sabbatel, G., and Duda, A., "*Performance Anomaly of 802.11b*", IEEE INFOCOM 2003, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, (March 2003)

[51] IEEE Std 802.11-1999, "*Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*", (1999)

[52] Garroppo, R.G., Giordano, S., Lucetti, S. and Russo, F., "*IEEE 802.11b Performance Evaluation: Convergence of Theoretical, Simulation and Experimental Results*", 11th International Telecommunications Network Strategy and Planning Symposium. NETWORKS 2004, (June 2004)

[53] De Vendictis, A., Vacirca, F. and Baiocchi, A., "*Experimental Analysis of TCP and UDP Traffic Performance over Infra-structured 802.11b WLANs*", (2008)

[54] Hendrix, B.S.E, den Hartog, F.T.H., van der Vlag, H.A.B. and Baken, N.H.G., "*In-home Video Distribution for Telecom Operators*", (2007)

[55] IEEE Std 802.11a-1999, "*Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY)specifications - High-speed Physical Layer in the 5 GHz Band*", (1999)

[56] IEEE Std 802.11g-2003, "*Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment 4: Further Higher Data Rate Extensionin the 2.4 GHz Band*", (2003)

[57] http://www.ieee802.org/11/Reports/tgn_update.htm

[58] CISCO Systems, "*802.11n: The Next Generation of Wireless Performance*", White Paper, (2007)

[59] HomePlug Powerline Alliance, "*HomePlug 1.0 Technology White Paper*", White Paper, (2002)

[60] HomePlug Powerline Alliance, "*HomePlug AV White Paper*", White Paper, (2005)

[61] Internal communication Delphinanto, A.

[62] Hazen, M.E., "*The Technology Behind HomePlugAV Powerline Communications*", Computer, Volume 41, p.90-92, (June 2008)

[63] www.opnet.com

[64] Postel, J., IETF RFC 792: "*Internet Control Message Protocol*", (1981)

[65] Braden, R., IETF RFC 1122: "*Requirements for Internet Hosts -- Communication Layers*", (1989)

[66] Mogul, J. and Deering. S., IETF RFC1191: "*Path MTU Discovery*", (1990)

[67] Thompson, K., Miller, G.J. and Wilder, R., "*Wide-Area Internet Traffic Patterns and Charasteristics*", IEEE Network, Volume 11, p.10-23, (Novemebr 1997)

[68] Fraleigh, C., Moon, S., Lyles, B., Cotton, C., Khan, M., Moll, D., Rockell, R., Seely, T. and Diot, S.C., IEEE Network, Volume 17, p.6-16, (November 2003)

[69] Li, W., Zeng,B., Zhang, D. and Yang, J., "*Performance Evaluation of End-to-End Path Capacity Measurement Tools in a Controlled Environment*", Springer Berlin/ Heidelberg, (2008)

List of abbreviations

ACK	Acknowledgement
CSMA/CA	Carrier sense multiple access/collision avoidance
CSMA/CD	Carrier sense multiple access/collision detection
DCF	Distributed coordination function
DIFS	DCF Interframe Space
ETE	End-to-end
FCS	Frame check sequence
ICMP	Internet control message protocol
IFG	Interframe gap
IPTV	Internet protocol television
IP	Internet protocol
ISDN	Integrated services digital network
LAN	Local area network
MAC	Media access control
MTU	Maximum transmission unit
OFDM	Orthogonal frequency division multiplexing
OWD	one-way delay
PLCP	Physical layer convergence protocol
ppm	probes per measurement
RTT	round-trip time
SIFS	Short interframe space
TCP	Transmission control protocol
UDP	User datagram protocol
VLAN	Virtual local area network
WLAN	Wireless local area network