Mälardalen University Licentiate Thesis
No.48

Bandwidth Measurements in Wired and Wireless Networks

Andreas Johnsson

April 2005

MÄLARDALEN UNIVERSITY

Department of Computer Science and Electronics
Mälardalen University
Västerås, Sweden

ISSN 1651-9256
ISBN 91-88834-53-0
Printed by Arkitektkopia, Västerås, Sweden
Distribution: Mälardalen University Press

Abstract

In the Internet today, end-user applications cannot get bandwidth guarantees from the network. Instead, bandwidth measures, such as available bandwidth and link capacity, must be measured whereafter the application can adapt its send rate to the bandwidth measurement estimate. An example application that rely on bandwidth measurements is TV transmission in realtime over the Internet.

To measure the available bandwidth between two nodes in a computer network, such as the Internet, active measurement methods are used. These methods do not require prior knowledge about the network topology. Measurement data, divided into *probe packets*, is injected into the network with an initial packet separation. The packets are time stamped on the receiver side. By deploying analysis methods the available bandwidth and link capacity can be calculated using the initial separation and the time stamps as input.

The work presented in this licentiate thesis studies bandwidth measurement methods with two foci: a) *how can the interactions between probe packets and other traffic in the network be described?* and b) *how do existing measurement methods, designed for wired networks, behave in wireless networks?*

A framework has been developed to describe the interactions between probe packets and other network traffic packets. This framework also describes the differences between using the statistical mean and median operator on time stamps in an analysis.

A simplified version of the TOPP measurement method, called DietTopp, has been developed and implemented. DietTopp is evaluated and compared to other bandwidth measurement tools in both wired and wireless scenarios.

Results obtained from measurements in wireless 802.11b networks show important differences compared to measurement results obtained from wired networks. The origins to some of the observed differences are explained whereas some are left to future research.

To my family

Acknowledgements

First of all I would like to thank my two supervisors and friends, Mats Björkman and Bob Melander. Without your guidance, creative support and collaboration I never would have made it this far. I hope that you both will have the time and patience to continue the journey towards a doctoral degree with me.

I would like to thank Svante Ekelin, Annikki Welin and Rikard Holm at Ericsson Research for creative discussions and for helping me with many of the measurements presented in this thesis.

To Svante Ekelin, Jan-Erik Mångs, at Ericsson Research, Erik Hartikainen at Linköping University, Martin Nilsson at SICS and my supervisors Mats Björkman and Bob Melander: I hope our ongoing collaboration will result in fruitful research!

Furthermore I would like to thank Jonas Neander and Ewa Hansen, both part of the data communication group at Mälardalen University, for their support as fellow Ph.D. students.

I also want to send my great thanks to the rest of the personal at our department for interesting discussions, coffee break laughs and in general for making our department a creative and fun place.

I wish to thank the present and old CoRe-group members at Uppsala University for letting me visit their department once a week and for many fun and interesting discussions. Input and feedback from Christian Rohner and Henrik Lundgren have been rewarding, thank you.

I would like to thank my family who has always supported me and encouraged me to strive towards becoming a researcher. I also want to thank my friends for enriching my spare time. Finally, to my girlfriend Maaret, thank you for keeping me motivated in my studies and especially for making my life outside work full of joy. I love you.

Many great thanks to our generous sponsors: the Swedish Research Council, Vinnova and the KK Foundation. Financial support for travels, other visits and courses has also been granted from ARTES++.

Andreas Johnsson
Västerås, April 2005

Contents

List of Publications **1**

I Thesis **3**

1 Introduction **5**

1.1 Background and motivation 5
1.2 IP networks 7
1.3 Research area 10
1.3.1 Network measurements 10
1.3.2 Link capacity and available bandwidth 13
1.3.3 Problems 15
1.3.4 Testing and verification 16
1.4 Our research 17
1.4.1 Research problems 17
1.4.2 Research method 19
1.4.3 Contributions 20

2 Related work **21**

2.1 Introduction 21
2.2 Bandwidth measurement tools 22
2.2.1 Packet rate methods 22
2.2.2 Packet gap methods 24
2.2.3 Link capacity estimation 25
2.3 Measurements in wireless networks 26
2.4 Applications of bandwidth measurements 26

3 Summary of papers **29**
3.1 Paper A 29
3.2 Paper B 30
3.3 Paper C 30
3.4 Paper D 31

4 Conclusions and future work **33**

Bibliography **35**

II Included Papers 39

5 Paper A:
On the Analysis of Packet-Train Probing Schemes **41**
5.1 Introduction 43
5.2 Description of patterns 44
5.2.1 A multiple-hop model for route delay variation 44
5.2.2 Mirror pattern 46
5.2.3 Chain pattern 48
5.2.4 Quantification pattern 49
5.3 Testbed setup 51
5.4 Signatures 52
5.4.1 The independence signature 52
5.4.2 The mirror signature 52
5.4.3 The rate signature 53
5.4.4 Quantification signature 54
5.5 Mean and median analysis using patterns 54
5.6 Conclusions 56
Bibliography 56

6 Paper B:
A Study of Dispersion-based Measurement Methods in IEEE 802.11 Ad-hoc Networks **59**
6.1 Introduction 61
6.1.1 Ad-hoc networks 61
6.1.2 Dispersion-based measurements 62
6.2 Research problems 63
6.2.1 Variable link capacity 63
6.2.2 Moving nodes 66

6.2.3 Loss rate 66
6.2.4 Time control 67
6.3 Conclusions 69
Bibliography 69

7 Paper C:
DietTopp: A first implementation and evaluation of a simplified bandwidth measurement method 73
7.1 Introduction 75
7.2 TOPP: the original method 75
7.2.1 Probe phase 76
7.2.2 Analysis phase 76
7.2.3 TOPP complications 77
7.3 DietTopp 77
7.4 Implementation of DietTopp 78
7.5 The testbed 79
7.6 DietTopp Evaluation 80
7.6.1 Measurement results 80
7.7 Conclusions 84
Bibliography 84

8 Paper D:
Bandwidth Measurement in Wireless Networks 87
8.1 Introduction 89
8.2 Experimental setup 90
8.2.1 DietTopp 90
8.2.2 The testbed 92
8.2.3 Experiments 93
8.3 Experimental results 93
8.3.1 Measurement results in wired networks 93
8.3.2 Measurement results in wireless networks 94
8.3.3 Wireless measurement results discussed 99
8.4 Other observations 101
8.5 Conclusion 101
Bibliography 102

List of Publications

The following articles are included in this licentiate[1] thesis:

A. *On the Analysis of Packet-Train Probing Schemes*, Andreas Johnsson, Bob Melander and Mats Björkman, In proceedings to the International Conference on Communication in Computing, Special Session on Network Simulation and Performance Analysis, Las Vegas, June 2004.

B. *A Study of Dispersion-based Measurement Methods in IEEE 802.11 Ad-hoc Networks*, Andreas Johnsson, Mats Björkman and Bob Melander, In proceedings to the International Conference on Communication in Computing, Special Session on Network Simulation and Performance Analysis, Las Vegas, June 2004.

C. *DietTopp: A First Implementation and Evaluation of a Simplified Bandwidth Measurement Method*, Andreas Johnsson, Bob Melander and Mats Björkman, In Proceedings to the Second Swedish National Computer Networking Workshop, Karlstad, November 2004.

D. *Bandwidth Measurement in Wired and Wireless Networks*, Andreas Johnsson, Bob Melander and Mats Björkman, submitted for publication.

[1] A licentiate degree is a Swedish graduate degree halfway between M.Sc. and Ph.D.

Besides the above articles, I have (co-)authored the following scientific papers:

I. *On the Comparison of Packet-Pair and Packet-Train Measurements*, Andreas Johnsson, In Proceedings to the First Swedish National Computer Networking Workshop, Arlandastad, September 2003.

II. *Analyzing Cross Traffic Effects on Packet Trains using a Generic Multihop Model*, Andreas Johnsson, Bob Melander and Mats Björkman, MRTC Report, Mälardalen University, November 2003

III. *Bandwidth Measurements from a Consumer Perspective - A Measurement Infrastructure in Sweden*, Mats Björkman, Andreas Johnsson and Bob Melander, Presented at the Bandwidth Estimation (BEst) Workshop, San Diego, December 2003.

IV. *Modeling of Packet Interactions in Dispersion-Based Network Probing Schemes*, Andreas Johnsson, Bob Melander and Mats Björkman, MRTC Report, Mälardalen University, April 2004.

I

Thesis

Chapter 1

Introduction

1.1 Background and motivation

Internet has gained more and more popularity since the mid 1990's and is now an integrated part of our society. A large range of broadband providers and the development of new and more efficient Internet applications increase the possibilities to watch movies and TV, use IP-telephony and share files over the Internet. These applications create a need for large transmissions of data at high bit rates, which in turn consume network bandwidth. Since several users must share the common bandwidth capacity on the Internet, there will be locations in the network where the demand is higher than the capacity. This causes network congestion and has negative impact, from the user perspective, on the data transmission rate and quality.

The major part of the Internet is designed to forward data with equal priority, independent on the data source and destination. Hence, there is no trivial technique to guarantee a specific transmission rate between for example an Internet TV station and its viewers. In Figure 1.1, different users want to watch Internet TV, provided by the IP TV-Station. The video stream to user 1 can be controlled, since that user is located within the TV stations own network. However, user 2 and user 3 are located on an other end (a different network) of the Internet. Guarantees of transmission rate and quality can not be made since the IP TV-station only has control of its own network. Also, user 2 probably has a high speed connection while user 3 only has a slow 3G wireless connection to the Internet. The video quality and transmission rate must hence be adapted to suit the needs for each user.

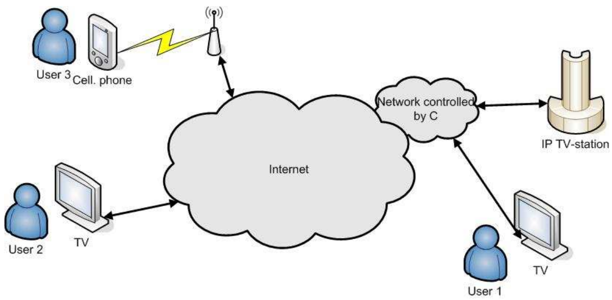

Figure 1.1:

If no guarantees of the transmission rate can be made by a data provider (such as the IP TV-station in Figure 1.1) it is desirable to at least be able to measure the current maximum transmission rate, called the *available bandwidth*, and hopefully also be able to make future predictions. Thereafter, the data provider can adapt both the amount of data to be sent and the transmission rate (e.g. a lower video quality requires less data to be transmitted per second).

Another scenario where it is important to measure the available bandwidth is when an Internet user wants to verify service level agreements for their broadband Internet access subscription. That is, “Do I, as a customer, get what I pay for from my service provider?”

The thesis discusses the problems of network measurements, especially how to measure the available bandwidth between two end-users on the Internet. The measurement problem is also studied in wireless topologies; What is the difference between available bandwidth measurements in wired networks (such as the Internet) and in wireless networks?

This thesis is divided into two parts. The first part of the thesis is organized as follows: Chapter 1 continues with an introduction to IP networks, which are the foundation of the Internet. Thereafter the bandwidth estimation research area is introduced. Measurement techniques, important definitions, measurement problems, testing and verification are issues that are discussed. The specific research questions addressed, the research method used and the contributions are then described. Chapter 2 describes some related work and chapter 3 is a summary of the papers included in the thesis. Part one ends with conclusions and future work.

The second part of the thesis contains four research papers. Three of them have been presented at conferences and the fourth is submitted for publication.

1.2 IP networks

This section gives a brief introduction to the underlying concepts of IP networks. A deeper introduction can be found in any text book on computer communications (e.g. [1]).

An IP network is essentially built up by end-hosts (such as laptops), servers (e.g. web-hosts) and routers (forward information from a source to a destination). In between these entities there are a set of connecting links that operate at different bit rates. An example of a simple network is illustrated in Figure 1.2. It contains two end-hosts (S and R), four routers (R1 - R4) and four subnetworks (network A - D).

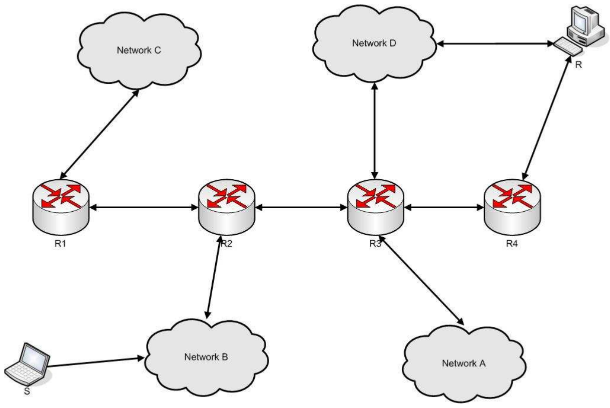

Figure 1.2: An example network with a sender S and a receiver R. S is directly connected to network B. The receiver R is connected both to network D and router R4.

All data that is sent from a sender to a receiver on the Internet is encapsulated into packets. The concept of encapsulating data is similar to that of a post card. Each packet contains a sender and a receiver address plus the information that is to be transmitted. However, in the Internet the encapsulation process is done in several steps, where each step adds an additional packet header, with information about the transmission between the sender and the receiver, to the existing packet (see Figure 1.3).

The first step is to add a transport header to the application data, this step is referred to as the transport layer. The transport layer contains among other things information so that the communicating processes on the source and des-

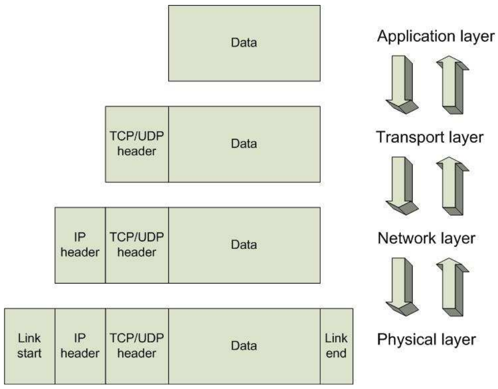

Figure 1.3: Application data is encapsulated in several steps at the sender side (arrows going down). On the receiver side the process is going in the reverse order (arrows going up).

tination computer can be determined. Other properties of the transport layer are described in more detail below.

Next, an IP header is added to the packet. This corresponds to the network layer. The IP header contains information about the source and the destination hosts, that is so the data can be correctly routed through the network. The bottom layer, the link layer, adds a link header. This header contains information on how to send the packet on an actual physical link between two computers. Depending on the link characteristics the link layer header can vary.

When an IP packet traverses a network, from S to R in Figure 1.2, the packet will encounter several routers in the path. A router is a machine that examines the destination address of the IP packet and then forwards the packet to the next router, one hop closer to the destination. This is repeated until the IP packet has traversed the path all the way to R.

A router, see Figure 1.4, is built up by several components: incoming links, the input queue, the switching fabric, output queues and outgoing links. A router has at least two link interfaces, one for incoming packets and one for outgoing packets.

Packets from different links can reach the router at the same time. In such cases packets are enqueued in the input queue. A common router queue discipline (which we also assume in our research) is the first-in-first-out discipline (FIFO). That is, a queued packet P must wait for all packets in front of it before

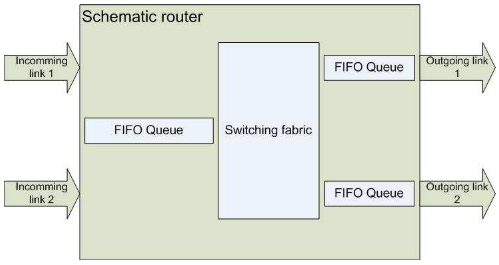

Figure 1.4: A schematic picture of a router. Packets from several incoming links are queued in a single FIFO queue. In the switching fabric the destination is decided. Thereafter the packet is put in the correct FIFO output queue.

P can be forwarded to the outgoing queue and thereafter to the outgoing link. The process of queue build up is called congestion. In the extreme case the queue will get full, then the next incoming packet is dropped (or some other packet - it depends on the drop policy).

The switching fabric (see Figure 1.4) is often a piece of hardware that uses a so called routing table to forward an IP packet to the correct outgoing link (i.e. to the next router in the path between the sender and the receiver). The switching fabric uses address information inside the IP header to determine the correct outgoing link.

The routing tables are either static (i.e. hardcoded by the network administrator) or dynamic (i.e. changes frequently depending on the state of the network). Static routing tables are often used in small networks where the network topology does not change. An example of such a protocol is the Routing Information Protocol (RIP). Dynamic routing protocols are used when the router administrator can not control the entire network, that is the network topology changes over time. Dynamic routing protocols exchange information on open and closed links and thus keep un updated view of the network. The two major routing protocols for the core Internet is the Open Shortest Path First (OSPF) and the Border Gateway Protocol (BGP).

Since a router can drop IP packets when a router queue gets full, a sender can not get guarantees that a packet will get through from the sender to the receiver. Such networks are called best-effort networks. The network does its best to forward each packet, but can not promise anything at all.

Due to the best-effort service, the end-hosts must deal with packet loss themselves. This is done by retransmission of lost packets. The IP layer does not provide retransmission mechanisms, instead a set of transport protocols have been developed (the transport layer in Figure 1.3). A widely used transport protocol is TCP which creates a connection between the sender and the receiver. TCP deals with packet loss, corrupt packets, packets out of order and so on. Further, TCP tries to be network friendly: it tries not to overflow the router queues nor the receiver buffer.

UDP is another common Internet transport protocol. UDP only provides an interface between the network layer and the correct user and process on the sending and receiving computer. That is, UDP does not provide TCP-like features such as retransmission of packets and packets out of order.

1.3 Research area

Having an overall understanding of the IP networks we now turn our attention towards measurements in such networks. A high level description of how to measure available bandwidth and link capacity is given in this Section.

There are two common types of measurement methods: passive and active methods. Both methods are presented in this section although the emphasis is on active measurements. Definitions of both link capacity and available bandwidth are made. Further, the difficulties of network measurements are discussed. The section ends with a survey of methods to verify active measurement results.

1.3.1 Network measurements

A vast variety of applications can benefit from estimates given by either passive or active measurement methods. In this section a brief description of passive methods is given, followed by a more in-depth description of active measurement methods.

Passive measurements

To measure network characteristics, such as the available bandwidth, the use of *passive measurement* methods is a possible strategy. Passive measurement methods and tools acts as observers inside a network and usually they will not interfere with other traffic. These methods most often also require control and administrative privileges of the underlying network infrastructure (i.e. access to routers and servers in the network).

MRTG [2] is an example of a passive measurement tool. It reports the traffic load (in bits per second) on outgoing and incoming links to a router. When the actual load of a link is measured, the available bandwidth is easily calculated if the link capacity of the measured link is known.

A simplified example of the use of MRTG information could be the following. Assume that S is doing a large data transfer to R, in Figure 1.2. In router R3, there is a passive measurement tool that monitors the traffic load (such as MRTG) on the link between R3 and R4, but also on the link between R3 and Network D. Depending on the outcome (i.e. the traffic load versus the link capacity), router R3 can decide to route the traffic through R4 or Network D to the destination R.

Another example of a passive measurement tool is the IPMON system [3]. It is, among other things, able to collect packet traces at several points in the network. A packet trace gives a more detailed picture, compared to the traffic load given by MRTG, of what happens on the monitored links. The packet traces are then used in analysis of traffic behavior. Using IPMON it is possible to study packet size distributions, the protocol type distribution (e.g. mail, http) et cetera.

Passive measurement methods are powerful in its context, but can typically not be used by others than those with network administrator priviledges. Furthermore, passive methods do not give full knowledge about what happens on an end-to-end path, between two end-users. Instead, these methods give a snapshot of the network status at a given time and link inside the network. Therefore another discipline of measurements has been developed: active measurements. These methods are described below.

Active measurements

Instead of using passive observers to measure network characteristics as described above we can deploy *active measurements* methods. Such methods inject so called probe traffic into the network at a traffic source and measure the network's influence at a probe traffic receiver. Hence, active measurement methods affect the network traffic itself, contrary to the passive measurement method family. Observe that active measurement methods only need access to two hosts, one traffic source and one traffic receiver. Such methods are called end-to-end measurement methods.

An example of an active measurement is the following. Assume that S in Figure 1.2 wants to know the average packet loss rate between S and R. If passive measurement methods are used there is a need for one or several passive observers inside the network core. This requires control and administrative privileges on the network path between S and R. In an active measurement S instead *probes* the network path between S and R by injecting a set of *probe packets* into the network (this is the active part of the measurement). These packets traverse the network all the way to R where the packets are collected (at least the packets that were not lost). Note that neither S nor R knows the exact route through the network. That is, the packets traverse a black box that you cannot study directly. When all packets have been collected by R, a ratio describing the number of probe packets lost is sent back to S. Thus, S has a snapshot describing the loss rate on the path between S and R.

Probing is the basic element of all active measurement methods, including measurements to gain information about link capacity and available bandwidth of a network path. There exist a variety of probing schemes. Below, two basic probing schemes are described: the packet-pair and the packet-train probing schemes.

Probing schemes for bandwidth measurements

To estimate end-to-end available bandwidth or link capacity by active measurements the first step is to probe the network path. This is done by injecting a set of probe packets with a pre-defined separation or *dispersion*. The dispersion is inversely proportional to the *probe rate* (measured in bits per second). A smaller dispersion between probe packets is equivalent to a higher probe rate. When the probe packets traverse the network, the pre-defined dispersion may change. This is due to competing network traffic or to the so called *bottleneck spacing effect* [4].

The most basic probing scheme is to divide the probe packets in *pairs*; each pair has a pre-defined dispersion that corresponds to the probe rate [5] [6] [7] [8] [9] [10] [11]. Each pair is sent through the network to the receiver where the packets are time stamped (usually at the application layer). The arrival-time stamps are used when the actual bandwidth estimate is calculated.

Instead of using pairs of probe packets many methods use *trains* of probe packets [9][10][12][13][14]. The dispersion between the probe packets inside the train can for example be equal or exponentially decreasing. There is a fundamental difference between using packet-pair and packet-train probing schemes. In packet-train probing schemes several probe packet delays may be dependent on each other, which is not the case in packet-pair probing schemes [15].

Having the arrival times of the probe packets, an analysis can be performed using the arrival times as input to obtain the estimate.

Analysis of active measurements

The analysis of results obtained from active measurements typically involves statistical operations on the dispersion values. To get an idea of how an analysis is performed, a simplified version of Pathload [12] is described since that algorithm is intuitive. A more in-depth description of probing analysis is given in Chapter 2.

A sender S is probing the path between S and R in Figure 1.2 using either a packet-pair or a packet-train probing scheme. An initial dispersion d_i (i.e. probe rate) is decided and thereafter the probe packets are injected into the network. The initial dispersion d_i may change on the path between S and R. The, by utilizing the statistical mean operator on the received dispersion values system noise is filtered out. The analysis idea is the following:

1. if the received dispersion mean is equal to d_i the probe rate is below the available bandwidth

2. if the received dispersion mean is greater than d_i the probe rate is above the available bandwidth

By sending probe pairs or probe trains at different probe rates (i.e. using different dispersion between probe packets) the available bandwidth can be determined by binary search.

1.3.2 Link capacity and available bandwidth

To be able to develop a deeper understanding of the models and tools in the area of bandwidth measurements, a strict definition of available bandwidth is needed. The following definition of link capacity and available bandwidth is taken from [16].

The capacity C of a single link (between two routers) is defined as the maximum IP layer transfer rate. When data is encapsulated into IP packets, overhead is induced. That is, the maximum IP layer transfer rate is lower than the actual raw link transfer rate. From an application point of view, the maximum IP layer transfer rate is more interesting than the actual raw link transfer rate since applications can not send data without data encapsulation.

The end-to-end link capacity C, having H successive links, is defined as the minimum capacity between the two nodes. That is,

$$C = \min_{i=1..H} C_i$$

where C_i is the capacity of link i.

The intuitive definition of the available bandwidth is the unused portion of the link capacity. However, this definition needs to be refined into a more strict definition. The definition from [16] builds on the observation that the transmission of bits (a bit is either 0 or 1) on a link is a discrete process. Either the link is busy sending a bit or the link is idle. The average utilization of a link can be described as

$$u(t_1, t_2) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} d(x)dx$$

where $d(x)$ is either 0 or 1 at a given time x. That is, the utilization of a link depends on the time interval.

The available bandwidth of a single link is defined as $A_i = (1 - u_i) * C_i$. Further, the end-to-end available bandwidth of a path containing H links is defined as the minimum available bandwidth:

$$A = \min_{i=1..H} A_i$$

where A_i is the available bandwidth on link i.

The utilization of a link depends on the *cross traffic*. Cross traffic is the regular traffic that flows through the network (i.e. all traffic but the probe traffic used in the measurement). Many parameters control the cross traffic behavior, some described below:

- *Intensity distribution* The dispersion between cross-traffic packets can be approximated by a probability function, such as rectangular (very smooth), poisson or pareto (very bursty). Depending on the cross-traffic source the intensity distribution may vary.

- *Packet size distribution* Depending on the cross-traffic source the size of the packets will vary. For example, a transfer of a large file uses large packets while a voice over IP application uses small packets.

- *Flow length* The cross-traffic flow between a specific sender and a receiver may vary from short web traffic flows (measured in seconds) up to long flows corresponding to a file download (measured in hours).

- *Aggregation* The cross-traffic on links within the core network is built up of several cross-traffic flows. A higher number of cross-traffic flows gives a higher degree of aggregation. Usually a high degree of aggregation gives a more smooth overall cross-traffic intensity distribution.

There as been many attempts to describe cross traffic behavior (e.g. in [17]). Such studies are most often done using passive measurements. The cross-traffic behavior changes over time, e.g., when new applications enter the market. It is important to understand the cross-traffic behavior when simulating cross traffic in network simulators and testbeds, discussed below.

1.3.3 Problems

One of the inherent complications in bandwidth measurement research is that the available bandwidth varies over time. This is a consequence of the variability of the cross traffic over time. In Figures 1.5 and 1.6, two examples are shown. In Figure 1.5, the cross-traffic load (y-axis) is plotted with a resolution of 5 minutes (x-axis) while in Figure 1.6 the resolution is 30 minutes (x-axis). The shaded area corresponds to the traffic load on one outgoing link while the line corresponds to the traffic load on one incoming link. The available bandwidth on each link is the link capacity minus the cross-traffic load.

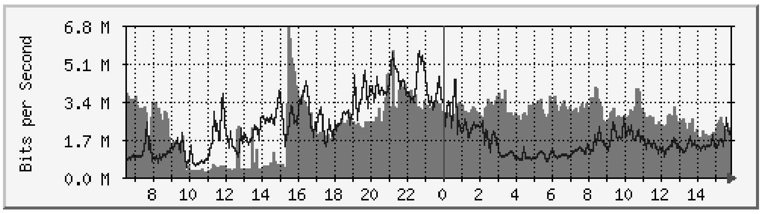

Figure 1.5: MRTG day trace. 5 minute intervals.

Assume that the available bandwidth is estimated. A fast measurement method can, for example, measure the available bandwidth during the large cross-traffic load peek at 15.5 (x-axis) in Figure 1.5. This will give a very low estimate of the available bandwidth. That value is not representative over other, longer time scales. Another method that probes a path during a longer period of time will get a higher estimate of the available bandwidth. However, in this case the cross-traffic peaks are invisible. Depending on the application that will use the available bandwidth estimates, different approaches must be taken.

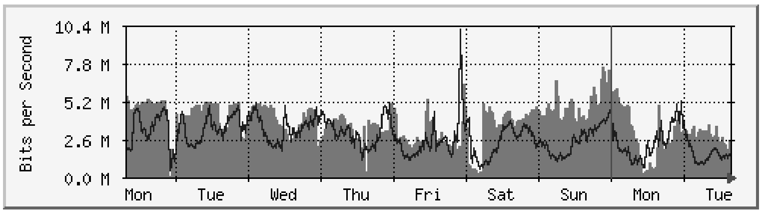

Figure 1.6: MRTG week trace. 30 minute intervals.

1.3.4 Testing and verification

It is important to be able to verify the bandwidth estimates obtained by both new and old measurement method. Three different techniques used for method testing and verification are discussed. Testing and verification by network simulations, by testbed measurements and by real network measurements. Each method is described and discussed below.

Network simulations

Network simulators (such as NS-2 [18]) are software that simulate computer networks. In such an environment, all parameters are known and hence are very well suited for testing and verification of available bandwidth measurement methods. Network topologies that are otherwise inaccessible to researchers can be simulated. In these environments a model of the measurement method can probe a simulated path and analyze the results to form a bandwidth estimate. Verification is simple, since the link capacity and available bandwidth are input parameters to the simulator. Cross-traffic is usually generated either by a cross-traffic generator or by traffic traces (a packet log collected from a real network).

The disadvantage of network simulations is that a simulation is a model that tries to represent the real world; in this case the simulation represents a computer network with its software and hardware. To create a model of a computer network many simplifications have to be made, hence there may be a gap between results obtained from network simulations and results obtained from real networks, especially if the scenario or the models are complex.

Furthermore, it can be difficult to make realistic scenarios. As one example, it is a known fact in the research community that it is hard to configure a realistic cross-traffic mix.

Testbed measurements

Testing and verification in testbed networks is a step towards testing and verification in real networks. In testbed networks an opportunity to run real software that implements the bandwidth measurement method is given. It is on the other hand expensive to create large network topologies. In testbed measurements the network topology, cross-traffic load and distribution et cetera can be controlled. Cross traffic is usually generated by cross-traffic generators or by traffic traces. The problem of creating realistic cross-traffic mixes is the same as in network simulations. Due to the control of all variables in testbed scenarios verification is simple.

Real network measurements

Usually, the developers of a new measurement methods want to run tests in a real network, such as the Internet. In large real networks it is hard to control the topology or the cross traffic. However, verification is still possible using passive measurements described in Section 1.3.1. For example, using MRTG [2] the current traffic load can be obtained on selected links on an end-to-end path (see Figures 1.5 and 1.6, both produced by MRTG). The cross-traffic load and the link capacity give the actual available bandwidth during the measurement. That is, we can verify the estimate produced.

The problem with passive measurements in real networks is of course the fact that most people do not have access to information collected in the core network. Another problem is that not all routers collect traffic load data to be presented in MRTG.

1.4 Our research

This section describes the research problems in the scope of this thesis, the research method and the contributions.

1.4.1 Research problems

Bandwidth estimation research has been in focus for quite some time and the area is starting to mature. However, there still exist several research problems that are unsolved or needs to be addressed more carefully. In the thesis the focus is on two important research problems. These are described below:

- It is important to understand the process when cross traffic and probe traffic interacts on a network path. This has to be studied both from the perspective of the probing method and from the perspective of the cross-traffic. How does the cross-traffic affect probe traffic and how is cross-traffic affected by probe traffic?

- Bandwidth measurements in wired networks, such as the Internet, has gained quite a lot of attention. When new network technologies starts to appear and are adopted (such as 802.11, ad-hoc networks, GPRS, 3G) it is important to investigate whether current bandwidth measurement methods are applicable in these new settings, with or without modification.

Other equally important research problems that need further attention are described in a set of bullets below[1]:

- Evaluation and comparison of existing bandwidth measurement tools. Is there a need for benchmarks to compare tools and methods against each other?

- There is a need to lower the amount of probe traffic send between a probe sender and a probe receiver but still keep the accuracy of the measurement method. This is due to the fact that probe traffic affect other traffic flows in the network, often negatively.

- Create stronger links between bandwidth estimation and TCP research.

- Continuous measurements of the available bandwidth of a network path: instead of measuring once and then use that estimate in an application there is a need for continuous measurements over large time periods.

- Coordination of several end-to-end measurements between different end-nodes in order to depict the current status of an entire network.

- Incorporation of bandwidth measurement research results into real applications. What are the applications that can benefit from results in the bandwidth measurement research area? What new problems and restrictions have to be considered in the specific case?

- Methods for bandwidth prediction.

[1]these research problems were discussed at the Bandwidth Estimation Workshop held in San Diego, 2004.

1.4.2 Research method

This section describes the research method used to go from the research problems above to the contributions described below.

The dominating research method in the computer science area discussed in this thesis is experimental. That is, the study of cause and effect. A set of hypotheses describing a system or a phenomenon are tested by conducting experiments. Thereafter conclusions are drawn which in turn will lead to new hypotheses.

In the thesis the interaction between cross-traffic and probe-traffic packets is studied. Further, the thesis presents an evaluation of tools and methods for estimating end-to-end available bandwidth and link capacity in new scenarios. These two research problems correspond to two branches of the experimental science. The first branch, to study packet interaction is more or less fundamental science while the study and evaluation of tools and methods are applied experimental research. Both branches will hopefully lead to new and better methods and tools.

Independent of the branch of experimental research methodology our primary task is to isolate the effect of single variables. The important variables in bandwidth estimation can be divided into two categories: system variables and method variables. A system variable is a variable that affect the network in-between a probe sender and a probe receiver. Such variables are for example cross-traffic load and distribution on each link between the sender and the receiver, router queues and policies. A method variable is a variable that is tightly bound to the actual method. Examples are the probing scheme, the total number of bits to be transferred, the analysis method et cetera. Depending on the verification method (described in 1.3.4), a subset of these variables can be fixed and controlled.

To obtain validity there is a need for verification of the experiments. Depending on the verification method (described in section 1.3.4) we can achieve everything from low to high validity.

Another aspect of experimental research methodology is the reproducibility of the study. By conducting experiments in network simulations and in testbeds and then documenting all parameters the experiments can be repeated with similar results. Experiments in real networks, such as the Internet, are harder to reproduce, however such measurements can be verified (as described in 1.3.4).

1.4.3 Contributions

This section presents the main contribution of the thesis. The results are divided into four papers, each individual paper is presented in the second part of the thesis. A summary of the papers can be found in Chapter 3.

- *Packet interaction framework:* A framework has been developed that describes, at the discrete packet level, how probe-packet trains and cross-traffic packets interact with each other when traversing a network path. Using this framework the differences between using the mean and the median operator on dispersion values obtained from active probing is discussed.

- *DietTopp:* Within the scope of the thesis DietTopp has been developed and implemented. DietTopp is a bandwidth estimation tool that is based on a modification of the previously not implemented TOPP method.

- *Combination of ad-hoc networks and bandwidth estimation:* The thesis illustrate and discuss four research problems associated with the combination of active bandwidth measurements and ad-hoc networks: variable measured link capacity, moving nodes, loss rate and time control.

- *Wireless measurements:* We show that both the measured link capacity and the available bandwidth decrease with decreasing probe packet size. Further, we show show that both the measured link capacity and the available bandwidth decrease with increasing cross-traffic rate.

Chapter 2

Related work

2.1 Introduction

Much work has been done in the bandwidth estimation area during the recent years. In this section the emphasis is on state-of-the-art bandwidth measurement tools, measurements in wireless networks and applications of bandwidth estimation. A short review of the more theoretic literature is given below.

Many studies of the underlying theories for probe packet interactions with routers, queues and cross traffic have been developed. In [8] a discussion of how the packet size affects the estimation of link capacity is made. Furthermore, they describe a model for probe packet delay variation. This model formalizes the packet-pair method concept.

In [9][19], theories and a discussions of link capacity measurements are given. What are the problems with variable packet size probing (described in subsection 2.2.1) and what are the alternatives? This work also describes Pathrate, a tool for measuring the end-to-end minimum link capacity. Further, [10][20] gives theoretical discussions about what packet-pair techniques measure, in the context of available bandwidth. This work also describes the available bandwidth estimation tool Pathload. In [16], common misunderstandings when designing bandwidth estimation tools are discussed.

Discussions about packet-pair methods have also been made in [11][13], which also describe the TOPP algorithm.

The problem of obtaining accurate time stamps for probe packets has been discussed in [21]. This is a very important problem, especially in high speed networks, since all analysis methods rely on accurate time stamps.

2.2 Bandwidth measurement tools

A large number of bandwidth measurement tools have been developed since Keshav's first attempt to measure the available bandwidth. In this section a description of the two main available bandwidth estimation categories are given along with a few examples. The main available bandwidth estimation categories are *packet rate methods* and *packet gap methods* (notation used in e.g. [22]).[1]

Also, a brief introduction to link capacity estimation is given. These methods either estimate the minimum link capacity of an end-to-end path, or the capacity of each hop. Here the emphasis is on a description of a hop-by-hop method.

It should be noted that a comparison of the method and tool performance is left out, since such a work is a study of its own. In [23], the problems of comparing bandwidth measurement tools is discussed.

2.2.1 Packet rate methods

The basis of the packet rate methods is so called self-induced congestion. The idea is the following: inject a set of probe packets (e.g. in pairs or in trains) with a predefined initial probe rate into a network. If the initial probe rate is higher than the available bandwidth, the probe packets will be queued after each other at the bottleneck and thus cause congestion. In such a case, the mean dispersion between the probe packets will increase, which is equivalent to a decrease in the received probe rate. However, if the initial probe rate is equal to the received probe rate it is assumed that the packets did not have to queue and thus the end-to-end path is not congested. That is, the initial probe rate is less than the available bandwidth.

There exist several tools that exploit this phenomenon, such as Bart [24], Pathchirp [14], Pathload [12] and TOPP [11]. To illustrate packet rate methods a deeper introduction to Pathload and Pathchirp is given below. A description of TOPP can be found in part two of this thesis.

Pathload

Pathload is an implementation of the Self-Loading Periodic Streams (SLoPS) methodology [12][20]. Using this methodology the end-to-end available bandwidth is measured. The basic idea of SLoPS is explained below.

[1] These categories are called iterative probing and direct probing, respectively, in [16].

A sender transmits a packet train with a pre-defined probe rate to a receiver. The sender time stamps the send time while the receiver time stamps the probe packet reception. The time stamp difference, defined as the *one way delay* (OWD) is calculated for each packet $(D_1, D_2, ..., D_K)$.

If the initial rate is higher than the available bandwidth, the router queue will grow when additional probe packets in the train are received to the router. Due to this fact, the OWD values will have an increasing trend (i.e. $D_K > D_{K-1} > ... > D_1$). When the initial probe rate is less than the available bandwidth, the router queues will not grow due to the injection of the probe train. Hence, the OWD will be more or less stable. Statistical tests are used to determine whether the OWD values are stable or increasing.

By sending probe trains at different pre-defined probe rates, the available bandwidth is found by binary search. If the initial probe rate causes an increasing OWD trend the probe rate is above the available bandwidth. The probe-train rate for the next train is decreased. On the other hand, if the probe train does not cause an increasing OWD trend, the initial probe rate is less than the available bandwidth. In this case the probe rate is increased. This process is iterated until a satisfactory accuracy of the available bandwidth is obtained.

Pathchirp

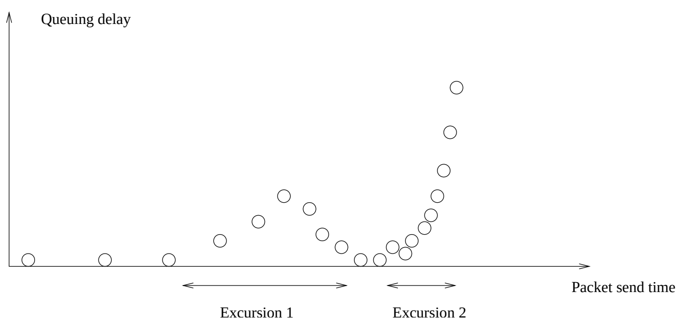

Figure 2.1: Packet sending time versus queuing delay by Pathchirp.

Pathchirp is another tool that measures the end-to-end available bandwidth and exploits the packet rate method. Instead of sending probe packet trains it sends *chirps* of probe packets. A chirp is essentially a packet train, but the dispersion between the probe packets are exponentially decreasing, $d_n = T\gamma^n, d_(n-1) = T\gamma^(n-1), ..., d_(2) = T\gamma^2, d_1 = T\gamma^1$, where d_i is the dispersion and T and γ are constants. By using chirps the end-to-end path is probed at different initial probe rates using just one train.

In the analysis the probe packet sending time is compared to the queuing delay that arises when the initial probe rate is above the available bandwidth. The queuing delay is derived from comparing probe packet send and receive times. In Figure 2.1 an example plot is shown, describing the measurement result obtained from one chirp. When the queuing delay is zero, the probe packets were transmitted without queuing. During the excursions, shown in the figure, the cross traffic is more intensive and hence causes queuing. The first few excursions go back to zero because the cross traffic is bursty (i.e. sometimes the cross traffic is absent during a chirp). The last excursion in the figure does not return to zero. This is when the send probe rate has exceeded the bottleneck link capacity.

By analyzing the excursions from many chirps, Pathchirp is able to form an estimate of the end-to-end available bandwidth.

2.2.2 Packet gap methods

When using packet gap models the network topology must be known to some extent, usually the bottleneck link (i.e. the link that determines the available bandwidth) must be known. Further, there can only be one bottleneck link between the probe sender and the probe receiver.

Packet gap methods solve the following equation:

$$A = C_{bl} - R_i(\frac{C_{bl}}{R_o} - 1)$$

where C_{bl} is the link capacity of the bottleneck link (that is known), R_i is the initial probe rate and R_m is the measured probe rate. To obtain values of R_i and R_o, the end-to-end path is probed using different initial probe rates. For each probe train or probe pair an estimate of the available bandwidth can be made.

Sometimes the equation above is expressed using time (in seconds) instead of rates (in bits / second). But the result is still the same.

Examples of tools exploiting the packet gap method are Delphi [25] and Spruce [22].

2.2.3 Link capacity estimation

The two measurement categories above describe how to estimate the available bandwidth. Much effort on developing link capacity measurement tools has also been made. Many such tools rely on variable packet size probing. The first tool to implement this method was Pathchar [26] which estimates the link capacity on each hop between two nodes. It should be noted that Pathchar does not need a receiver node. Other tools followed in the same track, such as pchar [27]. The basic idea of Pathchar is discussed next.

Pathchar

IP packets, and thus probe packets, contain a time to live (TTL) field. TTL is a number, specified by the sender, which is decremented by each router in a path. If the TTL is low, or the path contains many routers the TTL may reach zero. In such case the packet is dropped and an Internet Control Message Packet (ICMP time exceeded packet) is sent from the router to the sending node.

By sending probe packets with a TTL of n the round trip time (RTT) to router n can be estimated. The RTT to router n can be expressed as

$$RTT = \sum_{i=1}^{n} (\frac{p}{C_i} + l_i)$$

where p is the size of the probe packet, C_i the capacity of link i and l_i the latency of the ICMP packet on link i.

By sending probe packets with different TTL values an estimate of the RTT to both router n and $n-1$ is obtained. Further, by subtracting the RTT to router $n-1$ from the RTT to router n we obtain the RTT for the hop between n and $n-1$. In [26] it is stated that the relation between the minimum RTT (i.e. where probe packets did not have to queue in routers) and the probe packet size is linear for each link i in the path. The following equation describes that relation:

$$RTT = \alpha + \beta * p$$

where α is the sum of the ICMP latency and β is the sum of $\frac{1}{C_i}$ for each link i (i.e. the slope of the minimum RTT when p changes).

Thus, by sending probe packets with different sizes for each fixed n-value, the parameters of the straight line can be calculated. Having that relation, the capacity C_i of each hop is obtained.

New methods

In [19], it is discussed why the methods that rely on variable packet size probing often lead to inaccurate estimates. For example, it is not always possible to estimate the minimum RTT. Pathrate [28] is a modern link capacity tool that estimates the minimum link capacity on an end-to-end path. It requires both a sender and a receiver and hence is less flexible. However the link capacity is estimated with higher accuracy than previous methods. Pathrate relies on packet-pair and packet-train probing instead of using ICMP information from the routers.

TOPP [13] is another tool that estimates the bottleneck link capacity of an end-to-end path. This method also needs a sender and receiver part. A modification of TOPP is implemented in a tool called DietTopp which is described in paper C and paper D.

2.3 Measurements in wireless networks

To date, not many studies have been made on applying active bandwidth measurements in wireless networks. In [29] a description of the effect of variable probe packet sizes in wireless networks is given. In that study they have evaluated bandwidth measurement tools in a testbed scenario with simple cross traffic. That study is extended in paper D of this thesis.

[30] describes a model to calculate the available bandwidth between two nodes in an ad-hoc network[2]. However, the available bandwidth is calculated with the help of the intermediate nodes. In paper B a discussion of the problems of end-to-end available bandwidth measurement methods is given.

2.4 Applications of bandwidth measurements

There exist many potential applications of bandwidth measurement methods. For example: streaming media adaptation, server selection, network tomography, TCP improvement and service-level-agreement verification. This section discusses and describes these applications and the potential benefit that active measurements could give.

[2] An ad-hoc network is a wireless spontaneously connected network without pre-defined infrastructure.

Streaming media

The send rate and video quality adaptation of streaming media is important, especially when services like TV over the Internet is becoming more popular. Today, the adaptation is typically based on packet loss rate or delay. If there is a method to adapt, in real time, the send rate and video quality to the available bandwidth, the problem of congestion in the network may be minimized. Most current available bandwidth methods are not applicable here, since they measure the available bandwidth at one point in time. However, the tool Bart [24] continously measure the available bandwidth. Future research is needed to combine streaming media and available bandwidth measurement tools.

Server selection

Server selection is another important application. That is, which server gives me, as a user, the shortest download time? If the available bandwidth could be measured in a quick and easy manner for many mirror servers (i.e. servers that contain the same data) at the same time, a user would be able to choose the best alternative. Further research in this area may lead to less congestion if the total traffic load can be spread out among the servers.

Network tomography

Network tomography is about trying to describe and map a network by actively measuring the characteristics from many end-points at the same time. It is important to understand how available bandwidth bottlenecks are shared between different end-hosts and to study how the available bandwidth bottlenecks change and perhaps move to a different location over time. Network tomography is also interesting from a router perspective: if the routers have a global view of the available bandwidth bottlenecks they may be able to route the traffic around congested links.

Some research has been done in this area [31][32]. The research is promising but has to cope with slow measurement techniques. If several slow measurement techniques have to be synchronized, the entire network tomography measurement will in turn be very slow. Another problem is the identification of bottleneck links, that is how to determine that a bottleneck link is shared between two paths.

TCP improvement

TCP is a transport protocol that tries to adapt the send rate to the current available bandwidth. Many TCP versions have been developed, each with its own specific twist.

An example where active measurement techniques have been adopted is found in [33]. In this work the startup algorithm, usually the TCP slow start mechanism, uses bandwidth measurement techniques. The available bandwidth is estimated using the trains of packets that flow from the sender to the receiver in the initial phase. Observe that the packets are part of the data that is transferred. After just a few round trip times the TCP has an initial value of the available bandwidth. Thereafter the send rate can be adapted in the usual TCP way.

TPTEST

TPTEST [34] is a service-level-agreement verification tool that relies on active measurements from an end-user to a measurement server located inside the network. TPTEST uses TCP throughput as the metric. As described in for example [16], TCP throughput is not a good metric for available bandwidth. However, TPTEST is used as a tool for comparative analysis. That is, users should be able to compare results from using one network operator with results from using another operator. When all measurements are done using the same metric, it might be a fair comparison.

Chapter 3

Summary of papers

This section summarizes the main contribution of each paper included in this thesis. My contribution to each paper is also marked out. Bob Melander and Mats Björkman are co-authors of all papers. Both have guided me, discussed the research issues and written parts of each paper. Bob Melander has also co-developed the measurement tool DietTopp as well as the NS-2 framework. The following papers are included in the thesis:

3.1 Paper A

"On the Analysis of Packet-Train Probing Schemes", Andreas Johnsson, Bob Melander and Mats Björkman, In Proceedings to the International Conference on Communication in Computing, Special Session on Network Simulation and Performance Analysis, Las Vegas, June 2004.

Summary This paper describes probe packet and cross-traffic packet interactions at the discrete packet level. We identify three main interaction-types which we call mirror patterns, chain patterns and quantification patterns, respectively. Experiments have been performed in a testbed to understand and explore the behavior of these patterns and how they affect bandwidth measurement results. The paper ends with a description of the difference, using the patterns, between using the mean operator and the median operator on measurement values obtained during the probing phase.

My contribution I have co-developed the measurement tool used in the paper, I have performed all experiments and written most parts of the paper.

3.2 Paper B

"A Study of Dispersion-Based Measurement Methods in IEEE 802.11 Ad-hoc Networks", Andreas Johnsson, Mats Björkman and Bob Melander, In Proceedings to the International Conference on Communication in Computing, Special Session on Network Simulation and Performance Analysis, Las Vegas, June 2004.

Summary In this paper we study the effect of multiple-access networks on dispersion-based measurement methods. We especially focus on 802.11b networks. We discuss four important research problems in this area: variable measured link capacity, movement of wireless nodes, packet loss rate and timing issues. Using simulations in NS-2 we show that the measured link capacity depend on the current cross-traffic intensity. Furthermore we briefly discuss what happens when the path between the two end-nodes changes (i.e. when wireless nodes move around). We discuss the impact of packet loss in wireless networks and how available bandwidth measurement methods may adapt to that. Finally we discuss problems with jitter when sending probe packets in a multiple-access network. This paper is a work-in-progress paper and hence does not provide any experimental or measurement results.

My contribution I have co-developed the experiment platform in NS-2, I have performed all experiments and written most of the paper.

3.3 Paper C

"DietTopp: A First Implementation and Evaluation of a Simplified Bandwidth Measurement Method", Andreas Johnsson, Bob Melander and Mats Björkman, In Proceedings to the Second Swedish National Computer Networking Workshop, Karlstad, November 2004.

Summary This paper describes an implementation of DietTopp, which is a modification of the TOPP method for measuring end-to-end available bandwidth and link capacity. We evaluate the DietTopp implementation in a testbed where one of the links is congested during the measurement session. We compare our results to both Pathload (that measures end-to-end available bandwidth) and Pathrate (which measures the end-to-end link capacity) with promising results.

My contribution I have constructed the testbed, co-developed the DietTopp implementation, performed all experiments and written most of the paper.

3.4 Paper D

"Bandwidth Measurements in Wireless Networks", Andreas Johnsson, Bob Melander and Mats Björkman, Submitted for publication.

Summary In this paper we have used DietTopp to explore the effects of wireless bottlenecks on bandwidth measurement results. We show using experiments that both the measured link capacity and available bandwidth depends on the probe packet size when conducting bandwidth measurements in wireless networks. Furthermore we also show that the measured link capacity decreases with increasing cross-traffic rate on the wireless bottleneck. We have performed experiments with three different cross-traffic distributions, and with several cross-traffic sources to validate our findings. The obtained results are also compared to results obtained from Pathload (a tool that measures the end-to-end available bandwidth). We describe a simple method for identification of wireless bottlenecks. Finally, we discuss the observation that bandwidth measurement results will be application dependent.

My contribution I have constructed the testbed, co-developed the DietTopp implementation, performed all experiments and written most of the paper.

Chapter 4

Conclusions and future work

There are several foci in this thesis. The first is on enhancing the foundations of bandwidth measurement methods and the impact of cross traffic on probe packets. A framework for describing interactions, at the discrete packet level, between probe-train packets and cross-traffic packets has been developed. Using this framework the differences between using the mean and the median operator on obtained dispersion values are explained.

Further, an evaluation of bandwidth measurement methods in both wired and wireless networks has been performed. Several characteristics that differ between wired and wireless networks (including ad-hoc networks) have been identified from the available bandwidth measurement perspective. Using our own tool, DietTopp, we show that both the measured link capacity and the available bandwidth decrease with increasing cross traffic or decreasing probe packet sizes.

The differences between performing available bandwidth measurements in wired and wireless networks have been investigated in this thesis. Future work is to investigate whether old bandwidth estimation methods can be adapted to fit the needs in wireless networks. Perhaps, new algorithms and methods have to be developed. Further, the problems of probing in ad-hoc networks will be studied in more detail to give concrete solutions to the stated problems.

Investigation of active continuous bandwidth monitoring is also of great importance. Mathematical algorithms for continuous measurements will be tried out and evaluated. Continuous monitoring methods will for example be used in streaming video applications. Continuous monitoring will also be studied and evaluated in ad-hoc networks where the topology and available bandwidth may change both drastically and frequently.

A study of how probe traffic is treated by so called network shapers will be done. There is a belief that network shapers treat network traffic generated from the various applications differently. In such a case, there is not only one estimate of the available bandwidth of interest. Instead, the available bandwidth must be measured with the application protocol in mind.

Bibliography

[1] Larry L. Peterson and Bruce S. Davie. *Computer Networks*. Morgan Kaufmann, 2000.

[2] Mrtg - multi router traffic grapher. Web resource: http://www.mrgt.org.

[3] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diot. Packet-level traffic measurement from the sprint ip backbone. In *IEEE Network Magazine*, November 2003.

[4] Van Jacobson. Congestion avoidance and control. In *Proceedings of ACM SIGCOMM*, pages 314–329, Stanford, CA, USA, August 1988.

[5] Srinivasan Keshav. A control-theoretic approach to flow control. In *Proceedings of ACM SIGCOMM*, pages 3–15, Zürich, Switzerland, September 1991.

[6] Robert Carter and Mark Crovella. Measuring bottleneck link speed in packet-switched networks. Technical Report 1996-006, Boston University Computer Science Department, Boston, MA, USA, March 1996.

[7] Kevin Lai and Mary Baker. Measuring bandwidth. In *Proceedings of IEEE INFOCOM*, pages 235–245, New York, NY, USA, March 1999.

[8] Attila Pásztor and Darryl Veitch. The packet size dependence of packet pair like methods. In *Tenth International Workshop on Quality of Service (IWQoS 2002)*, Miami Beach, USA, May 2002.

[9] Constantinos Dovrolis, Parameswaran Ramanathan, and David Moore. Packet dispersion techniques and capacity estimation. In *IEEE/ACM Transactions on Networking*, December 2004.

[10] Constantinos Dovrolis, Parameswaran Ramanathan, and David Moore. What do packet dispersion techniques measure? In *Proceedings of IEEE INFOCOM*, pages 905–914, Anchorage, AK, USA, April 2001.

[11] Bob Melander, Mats Björkman, and Per Gunningberg. A new end-to-end probing and analysis method for estimating bandwidth bottlenecks. In *Proceedings of IEEE Globcom*, San Francisco, November 2000.

[12] Manish Jain and Constantinos Dovrolis. Pathload: a measurement tool for end-to-end available bandwidth. In *Passive and Active Measurement (PAM) Workshop 2002 Proceedings*, pages 14–25, Ft Collins, Co, USA, March 2002.

[13] Bob Melander, Mats Björkman, and Per Gunningberg. Regression-based available bandwidth measurements. In *Proceedings of the 2002 International Symposium on Performance Evaluation of Computer and Telecommunications Systems*, San Diego, CA, USA, July 2002.

[14] Vinay Ribeiro, Rudolf Riedi, Richard Baraniuk, Jiri Navratil, and Les Cottrell. pathchirp: Efficient available bandwidth estimation for network paths. In *Passive and Active Measurement Workshop*, 2003.

[15] Andreas Johnsson, Bob Melander, and Mats Björkman. Modeling of packet interactions in dispersion-based network probing schemes. Technical report, Mäalardalen University, 2004.

[16] Manish Jain and Constantinos Dovrolis. Ten fallacies and pitfalls on end-to-end available bandwidth estimation. In *Internet Measurement Conference*, October 2004.

[17] Yongmin Choi, Heung-No Lee, and Anurag Garg. Measurement and analysis of wide area network (wan) traffic. In *SCS Symposium on Performance Evaluation of Computer and Telecommunication Systems*, July 2000.

[18] The network simulator - ns-2. http://www.isi.edu/nsman/ns/.

[19] R.S. Prasad, C. Dovrolis, and Bruce A. Mah. The effect of layer-2 store-and-forward devices on per-hop capacity estimation. Proceedings of IEEE Infocom 2003, San Francisco, CA, April 2003.

[20] Manish Jain and Constantinos Dovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. In *Proceedings of ACM SIGCOMM*, Pittsburg, PA, USA, August 2002.

[21] Attila Pásztor and Darryl Veitch. Precision based precision timing without GPS. In *Proceedings of ACM SIGMETRICS*, Marina Del Rey, CA, USA, June 2002.

[22] Strauss, Katabi, and Kaashoek. A measurement study of available bandwidth estimation tools. In *ACM SIGCOMM Internet Measurement Workshop*, 2003.

[23] Federico Montesino-Pouzols. Comparative analysis of active bandwidth estimation tools. In *Passive and Active Measurement (PAM) Workshop*, 2002.

[24] Svante Ekelin and Martin Nilsson. Continuous monitoring of available bandwidth over a network path. In *Swedish National Computer Networking Workshop*, Karlstad, Sweden, 2004.

[25] Vinay Ribeiro, Mark Coates, Rudolf Riedi, Shriram Sarvotham, Brent Hendricks, and Richard Baraniuk. Multifractal cross-traffic estimation. In *Proceedings of ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, Monterey, CA, USA, September 2000.

[26] Allen B. Downey. Using pathchar to estimate Internet link characteristics. In *Proceedings of ACM SIGCOMM*, pages 241–250, Cambridge, MA, USA, August 1999.

[27] Bruce Mah. pchar: A tool for measuring Internet path characteristics, June 2001. http://www.employees.org/˜bmah/Software/pchar/.

[28] Pathrate - a measurement tool for the capacity of network paths. http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/pathrate.html.

[29] Karthik Lakshminarayanan, Venkata N. Padmanabhan, and Jitendra Padhye. Bandwidth estimation in broadband access networks. In *In Proceedings to the Internet Measurement Conference*, 2004.

[30] Hwa-Chun Lin and Ping-Chin Fung. Finding available bandwidth in multihop mobile wireless networks. In *Proceedings of the IEEE Vehicular Technology Conference, Tokyo, Japan*, 2000.

[31] Ningning Hu, Li (Erran) Li, Zhuoqing Morley Mao, Peter Steenkiste, and Jia Wang. Locating internet bottlenecks: algorithms, measurements, and implications. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, 2004.

[32] Alok Shriram and Jasleen Kaur. Identifying bottleneck links using distributed end-to-end available bandwidth measurements. First ISMA Bandwidth Estimation Workshop (BEst), San Diego, 2003.

[33] Ningning Hu and Peter Steenkiste. Improving tcp startup performance using active measurements: Algorithm and evaluation. In *In Proceedings to the 11th IEEE International Conference on Network Protocols*, 2003.

[34] TPTest. II-Stiftelsen, October 2002. http://www.iis.se/tptest/.

II

Included Papers

Chapter 5

Paper A: On the Analysis of Packet-Train Probing Schemes

Andreas Johnsson, Bob Melander and Mats Björkman
In proceedings to the International Conference on Communication in Computing, Special Session on Network Simulation and Performance Analysis, Las Vegas, June 2004

Abstract

With a better understanding of how probe packets and cross-traffic packets interact with each other, more accurate measurement methods based on active probing can be developed. Several existing measurement methods rely on packet-train probing schemes. In this article, we study and describe the interactions between probe packets and cross-traffic packets.

When one packet within a packet train is delayed, the dispersion (i.e. packet separation) of at least two (and possibly more) probe packets will change. Furthermore, the dispersions are not independent, which may bias calculations based on statistical operations. Many methods use dispersion averages, such as the mean, in the calculation of bandwidth estimates and predictions.

We describe cross traffic effects on packet trains. The interaction results in mirror, chain and quantification patterns. Experiments have been performed in a testbed to explore these patterns. In histograms of delay variations for adjacent probe packets, these patterns are manifested as different identifiable signatures.

Finally, we also discuss the effect of these patterns on the mean and median operations.

5.1 Introduction

Measurement of the end-to-end available bandwidth of a network path is getting increasingly important in the Internet. Verification of service level agreements, streaming of audio/video flows, and Quality-of-Service management are all examples of Internet activities that need or can benefit from measurements of available bandwidth.

Many methods that attempt to measure end-to-end bandwidth actively probe the network path by injecting probe packets in predetermined flight patterns. Common flight patterns include pairs of probe packets, so called *packet-pair probing schemes* and its extension into longer sequences of probe packets [1, 2, 3, 4, 5, 6], which we will refer to as *packet-train probing schemes*.

When the probe packets (independent of probing scheme) traverse the network path, the dispersion between successive probe packets will change. This is due to limited link capacity and interactions with other packets traversing the same path (so called cross-traffic packets).

To calculate an available bandwidth estimate an analysis of the dispersion values is made. The analysis relies on the probe packet dispersion at the sender side in combination with the probe packet dispersion at the receiver side.

In this article we describe the probe packet and cross-traffic packet interactions when using packet-train probing schemes. The interactions are manifested as patters. Further, we illustrate these patterns as identifiable signatures in histograms.

We describe the effect of the packet interactions when using mean and median operations to dispersion values obtained from packet-train probing schemes. All dispersion-based measurements has been performed in a testbed, which is described in the article.

The rest of this article is organized as follows: Section 5.2 describes three patterns that occur when probe and cross-traffic packets interact with each other. Section 8.2 defines a testbed that we have used for all of our measurements. Section 5.4 identifies four signatures that arise from the three patterns. Section 5.5 describes the impact of mean and median filtering when performing analysis of dispersion values obtained from packet-train probing schemes. The article ends with conclusions in Section 8.5.

5.2 Description of patterns

This section describes three patterns that arise when probing a network path using packet-train probing schemes. The patterns are described using a multiple-hop model for route delay variation [7]. The concepts of that model is described in the following subsection, while the patterns are described in subsections 5.2.2 - 5.2.4.

5.2.1 A multiple-hop model for route delay variation

This subsection describes the concepts of a multiple-hop model for route delay variation [7]. We use this model to describe the identified patterns in the following subsections.

In what follows, the definition of a *hop* is one router, its in-queue, and the outgoing link used by the packets. Hence, the arrival time of an arbitrary packet to hop $h+1$ is equal to the departure time from the previous hop h.

A packet P_i arrives to a hop h at time τ_i. After a queuing time $w_i \geq 0$ the packet begins its service time $x_i > 0$. Packet P_i leaves the hop at time τ_i^*. Thus, the one-hop delay for packet P_i is

$$d_i \equiv \tau_i^* - \tau_i = w_i + x_i + D, \tag{5.1}$$

where D is the link propagation delay, which is equal for all equally-sized packets traveling on the same link.

From Equation (5.1), a set of equivalences to compare two adjacent packets are derived:

$$\begin{aligned}
\text{inter-packet arrival time: } t_i &\equiv \tau_i - \tau_{i-1} \\
\text{inter-packet departure time: } t_i^* &\equiv \tau_i^* - \tau_{i-1}^* \\
\text{delay variation: } \delta_i &\equiv d_i - d_{i-1} \\
&= t_i^* - t_i \\
&= (x_i - x_{i-1}) + \\
&\quad\ (w_i - w_{i-1}).
\end{aligned}$$

The waiting time of a packet within an infinite FIFO buffer is described by Lindley's equation

$$w_i = max(0, w_{i-1} + x_i - t_i) + c_i, \tag{5.2}$$

c_i corresponds to the waiting time caused by cross traffic entering the hop between τ_{i-1} and τ_i.

A router queue can in principle operate in two states - busy and idle state. The busy state implies that the router is constantly forwarding packets from its in-queue, while in the idle state the in-queue is empty.

Probe packets (i.e. packets used for obtaining dispersion values) can consequently be divided into two categories, **I**nitial and **B**usy probe packets (adapting to the notation in [7]). The first packet of a busy period is by definition an initial probe packet. That is, an **I** probe packet is never queued behind another probe packet (i.e. w_i of Equation (5.2) is equal to $0 + c_i$). **B** probe packets are packets that *are* queued behind other probe packets.

With this categorization of probe packets, the delay variation δ_i is defined with respect to whether a probe packet P_i is **I** or **B**. From [7] we have

$$\begin{aligned} \textbf{I:} \qquad \delta_i &= (x_i - x_{i-1}) + (w_i - w_{i-1}) && (5.3) \\ \textbf{B:} \qquad \delta_i &= (x_i - t_i) + c_i && (5.4) \end{aligned}$$

where Equation (5.4) is derived from Equations (5.2) and (5.3).

Equations (5.3) and (5.4) are extended in [7] to describe multiple hops. These extensions are based on the following statements: if a probe packet is **I** or **B** at hop j and **B** at hop $j + 1$, δ_i is overwritten and replaced by δ_i from Equation (5.4). On the other hand, if the probe packet is **I** at hop $j + 1$, the right hand side of Equation (5.3) is added to the existing δ_i.

Hence, a probe packet that traverses an H-hop path, being **I** at every hop, has a delay variation

$$\delta_i = \sum_{h=1}^{H}(x_i^h - x_{i-1}^h) + \sum_{h=1}^{H}(w_i^h - w_{i-1}^h). \tag{5.5}$$

If a probe packet is **B** on at least one hop, it will be **B** for the last time at some hop in the path. Denote this hop s_i. The delay variation for such a probe packet is

$$\begin{aligned} \delta_i &= (x_i^{s_i} - t_i + c_i^{s_i}) + \sum_{h=s_i+1}^{H}(x_i^h - x_{i-1}^h) + \\ &\quad + \sum_{h=s_i+1}^{H}(w_i^h - w_{i-1}^h). \end{aligned} \tag{5.6}$$

5.2.2 Mirror pattern

In the following we will describe the characteristics of the mirror pattern. Hence, consider a probe packet train, containing at least three probe packets P_{i-1}, P_i and P_{i+1} that are all **I** at hop h. Further, assume that P_{i-1} and P_{i+1} are unaffected by cross traffic (i.e. $w_{i-1} = w_{i+1} = 0$). Then, if P_i is delayed, $w_i > 0$, its delay variation $\delta_i > 0$. Hence, P_i will have a delay variation $\delta_i = (x_i - x_{i-1}) + (w_i - w_{i-1}) = (w_i - w_{i-1})$, under the assumption of fix size probe packets.

Since both P_{i-1} and P_{i+1} are unaffected by cross traffic, the following holds

$$\begin{aligned}
\delta_{i+1} &= w_{i+1} - w_i \\
&= -w_i \\
\delta_i &= w_i - w_{i-1} \\
&= w_i \\
&\Longrightarrow \\
\delta_{i+1} &= -\delta_i, \qquad (5.7)
\end{aligned}$$

which we define as a *perfect mirror pattern*. An example of this phenomenon is shown in Figure 5.1. The vertical packets above the time line shows when in time a probe packet (white box) or a cross-traffic packet (shaded box) arrives to the hop. The arc indicates when in time all bits of the packet have been received to the router. When all bits have been received, the router transmit the packet on the outgoing link, if it is not delayed by another packet. The transmission from the router is shown below the time line in the same manner as above the time line. The horizontal packets describe the packet pattern on the out-going link. Probe packet P_i is delayed w_i time units, visualized by the horizontal arrow next to P_i in Figure 5.1. Since P_{i-1} and P_{i+1} are unaffected by cross traffic Equation (5.7) holds and we have a perfect mirror pattern.

In addition to the fact that probe packet P_i can be delayed, there is a possibility that one, or both of P_{i-1} and P_{i+1} are affected by cross traffic. This will cause changes to the mirror pattern as described below. It is obvious that this possibility grows with increasing cross traffic and/or increasing probe rate.

Assume, for instance, that both P_i and P_{i+1} are delayed by cross traffic, while P_{i-1} and P_{i+2} are unaffected. Then the mirror pattern is *divided* in a

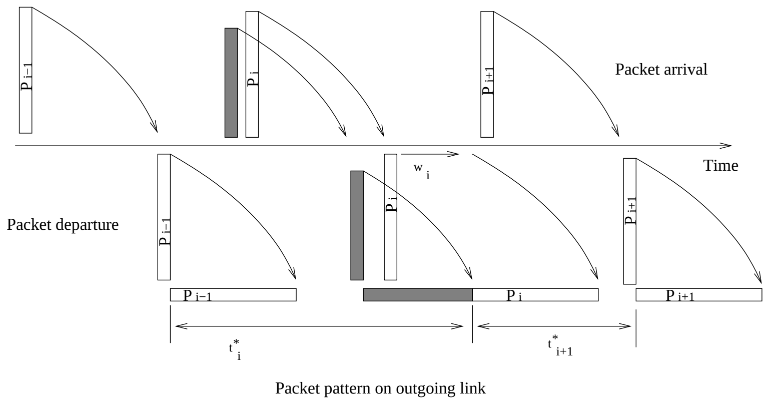

Figure 5.1: Arrival and departure times for cross traffic (shaded boxes) and probe packets (white boxes) entering a hop. The cross-traffic packet delays probe P_i in such a way that a *mirror pattern* arises.

predictable way. That is,

$$\begin{aligned}\delta_i &= w_i - w_{i-1} = w_i \\ \delta_{i+1} &= w_{i+1} - w_i\end{aligned}$$

since $w_{i+1} > 0$. Now, the next packet in the train, P_{i+2}, will have a delay variation

$$\begin{aligned}\delta_{i+2} &= w_{i+2} - w_{i+1} \\ &= -w_{i+1}\end{aligned}$$

since P_{i+2} has a waiting time $w_{i+2} = 0$. Hence, $-\delta_i = \delta_{i+2} + \delta_{i+1}$.

To generalize, assume that P_i and the following $(n-1)$ probe packets are delayed by cross traffic (not necessarily at the same hop), then we have a chain of divided mirror patterns. Their delay variations relate to each other in the following way:

$$
\begin{aligned}
\delta_{i+n} + \cdots + \delta_{i+1} &= (w_{i+n} - w_{i+(n-1)}) \\
&\quad + \cdots + (w_{i+2} - w_{i+1}) \\
&\quad + (w_{i+1} - w_i) \\
&= -w_i \\
&= -\delta_i \qquad (5.8) \\
&\Longrightarrow \\
\sum_{a=i}^{i+n} \delta_a &= 0 \qquad (5.9)
\end{aligned}
$$

since $w_{i+n} = 0$. That is, cross-traffic effects on packet trains will cancel out, to a certain degree, and hence not affect the mean value of the packet dispersion values (nor δ-values) obtained from the probe-packet train.

Mirror patterns are erased if probe packets in the packet train are transformed from **I** to **B**, as described by Equations (5.5) and (5.6).

When a probe packet train traverses an H-hop path, the delay variation of every probe packet is described by Equation (5.5) or (5.6) depending on whether the probe packets ever become **B**. We have extended the model presented in [7] to describe the relation between delay variation values obtained from packet-train probing.

5.2.3 Chain pattern

In this section we will describe the chain pattern. If a cross-traffic packet delays a probe packet, P_{i-1}, in such a way that at least P_i and P_{i+1} are transformed from **I** to **B**, and makes the involved probe packets P_{i-1}, P_i and P_{i+1} back-to-back after the hop, a chain pattern is visible.

This is the definition of a *pure chain pattern*. If other probe packets within the scope of the chain pattern are delayed by cross traffic, a quantification pattern will arise. Quantification patterns are described in Section 5.2.4.

An example of a chain pattern is shown in Figure 5.2, which is similar to Figure 5.1. When P_{i-1} is received, it must wait for the cross-traffic packet to complete its departure. The waiting time of P_{i-1} is w_{i-1}, shown in Figure 5.2. P_{i-1} is transmitted back-to-back behind the cross-traffic packet. P_{i-1} is in this example by definition **I** since it does not have to queue behind any other probe packet.

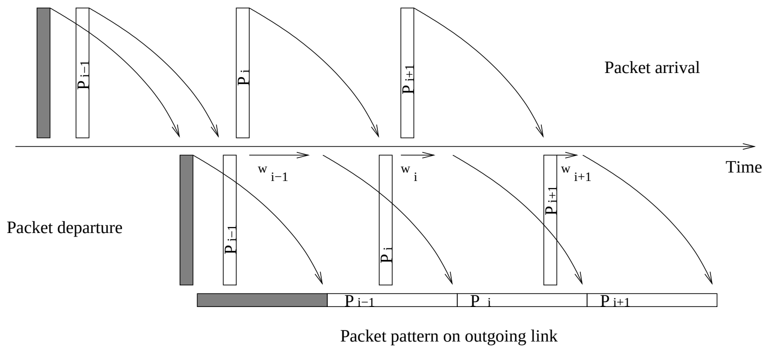

Figure 5.2: Arrival and departure times for cross traffic (shaded boxes) and probe packets (white boxes) entering a hop. The cross-traffic packet delays P_{i-1} in such a way that a *chain pattern* arises.

During the waiting time of P_{i-1}, the next probe packet P_i enters the router. P_i has to wait w_i time units in the queue for P_{i-1} to complete its departure, and is therefore **B**. After the waiting time, P_i is sent back-to-back behind P_{i-1}. The same procedure is repeated for P_{i+1}.

After the service time of P_{i+1} has elapsed, P_{i-1}, P_i and P_{i+1} travel back-to-back after each other on the link. Also, P_i and P_{i+1} have been transformed from **I** to **B** since both packets had to queue behind other probe packets. Hence, a chain pattern is visible after the hop.

The relation between delay variation values from probe packets involved in a chain pattern can be described by Equation (5.8), similarly to mirror patterns. The difference is that the mirror pattern involves **I** probe packets while the chain pattern involves probe packets that change from **I** to **B**.

Chain patterns are preserved to some degree in an H-hop path if the hop where the patterns arise is the last hop where the probe packets are **B**. Of course, this pattern is blurred by mirror and quantification patterns if there are hops downstream hop s_i with cross-traffic. This is described by Equations (5.5) and (5.6).

5.2.4 Quantification pattern

The last pattern identified in this paper is the quantification pattern. This pattern is described below.

Let us assume that the probe packet generator is sending probe packets at a high rate (i.e. the probe packet dispersion is less than the service time x_{CT} of a large cross-traffic packet). When a cross-traffic packet enters the router queue between the arrival time of two probe packets the probe packets will become separated by the service time x_{CT} of that cross-traffic packet. Hence there is no idle time gap in the router between the probe packets. This separation is hereafter referred to as a *quantification pattern*. The term quantification is used since the traffic consists of discrete transmissions, rather than a continuous flow of bits.

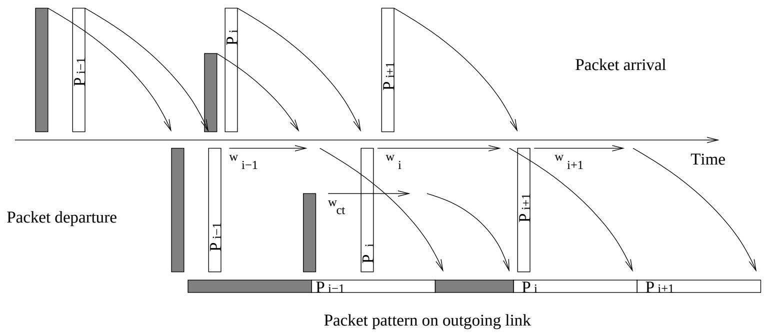

Figure 5.3: Arrival and departure times of cross traffic (shaded boxes) and probe packets (white boxes) entering a router. The upper left cross-traffic packet causes a chain pattern. The smaller cross-traffic packet causes a *quantification pattern*.

An example of the quantification pattern is shown in Figure 5.3, which is similar to Figure 5.1. In this example we see that the leftmost cross-traffic packet creates a chain pattern (equivalent to Figure 5.2). The second cross-traffic packet entering the hop between probe packet P_{i-1} and P_i will be sent directly after P_{i-1}, while P_i is sent back-to-back with the second cross-traffic packet. Hence, P_{i-1} has to wait w_{i-1} time units (stemming from the large cross-traffic packet), while P_i has to wait w_i time units (corresponding to the small cross-traffic packet and a portion of the the big cross-traffic packet). That is, a quantification pattern has arisen.

The delay variation relation of values obtained from probe packets involved in a quantification pattern can be described by Equation (5.8), similarly to mirror patterns.

Quantification patterns are preserved to some degree in an H-hop path if the hop where the pattern arise is the last hop where the probe packets are **B**. This pattern is blurred if there are hops downstream hop s_i with cross-traffic noise. This is described by Equations (5.5) and (5.6).

5.3 Testbed setup

The testbed network (see Figure 5.4) used in the experiments consists of three router nodes, one Black Diamond (BD) and two Torrent routers (T1 and T2) [8]. There are also a probing generator (PG), a probing receiver (PR), a traffic generator (TG) called IP Traf Gen (internal product of Ericsson, *www.ericsson.com*) and a traffic receiver (TR) which is an IXIA 1600 Traffic Generator/Analyzer. The links are all 100 Mbps except between the three routers, where the links are limited to 10 Mbps.

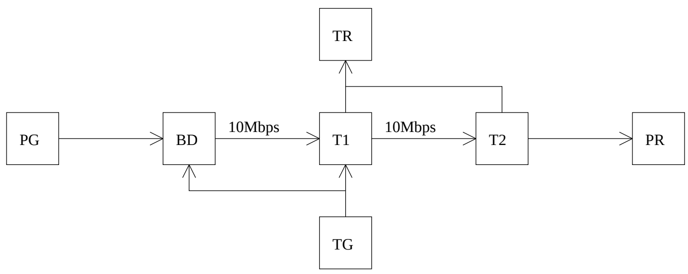

Figure 5.4: Testbed setup

Depending on the experiment setup the cross traffic can enter and leave the router chain at different positions. The cross traffic flow can be one or several of the following: TG → T1 → T2 → TR (flow 1), TG → BD → T1 → TR (flow 2) and TG → BD → T1 → T2 → TR (flow 3).

All cross traffic is exponentially inter-packet spaced. Four different cross traffic sizes where used: 64, 148, 482 and 1518 bytes. In the experiments all of these could be used at the same time or just a selection of them. When all sizes are used, 46% is of size 64, 11% of 148, 11% of 482 and 32% of size 1518 bytes. This distribution of packet sizes has its origin from findings in [9].

The cross traffic intensity is variable within the testbed, in steps of arbitrary size.

The probe traffic is sent through the path PG → BD → T1 → T2 → PR. The probe packet size is 1500 bytes and consists of 32 packets. Normally 5 trains are sent per test run.

5.4 Signatures

When cross traffic affects a packet train, the train will suffer from different patterns, as discussed in Section 5.2. In this section we illustrate these patterns using real packet trains which interact with cross traffic. The patterns, alone or in combination with each other, correspond to different signatures. The signatures are described and illustrated in delay variation histograms below.

Four signatures are defined and discussed in this section. They are: independence, mirror, rate and quantification signatures. In [7] similar signatures have been identifies for packet-pair probing schemes. However, we have shown that packet-pair and packet-train probing schemes are fundamentally different [10]. Thus, it is important to identify the signatures for packet trains as well. Examples from the described testbed are shown.

The probe rate for each diagram in Figure 5.5 is 1 Mbps in the upper left diagram, 2.9 Mbps, 4.8 Mbps and 6.7 Mbps in the bottom right diagram. The probe packets are 1500 bytes, and are sent in 5 trains consisting of 32 packets in each. The cross traffic rate is 5 Mbps, exponentially distributed and consist of 4 different packet sizes. The cross traffic uses flow 1 as described in Section 8.2.

5.4.1 The independence signature

The independence signature is visible in scenarios where there is no or very little cross traffic interfering with the packet train. That is, no or very few probe and cross traffic packet interactions.

The signature arises from the fact that most probe packets traverse the network path unaffected by cross traffic, thus independent of other traffic. This means that the delay variation δ for such packets are near 0 as seen in the upper left diagram in Figure 5.5. The peak at $\delta = 0$ is called the independence peak.

5.4.2 The mirror signature

The mirror signature is a signature that arises due to the mirror pattern. If there is very little cross traffic and the probe rate is relatively low, there will arise mirror patterns in the values obtained from packet-train probing. That is,

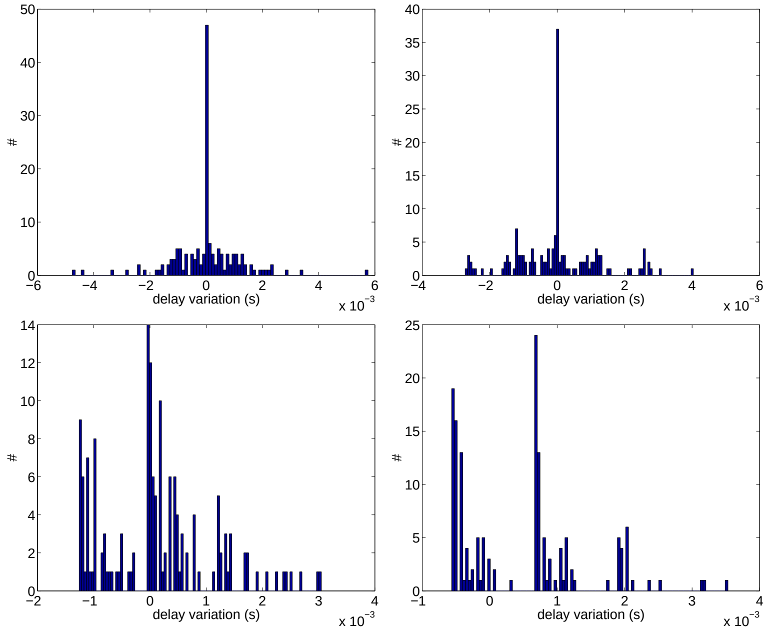

Figure 5.5: Using 4 packet sizes and cross traffic flow 1, with a rate of 5 Mbps. The probe rate increases from approximately 1 Mbps (upper left) to 6.7 Mbps (lower right).

for each positive delay variation value there is a corresponding negative delay variation value.

The upper right diagram of Figure 5.5 shows a distribution of delay variation values around the diminished independence peak (compared to the upper left diagram).

5.4.3 The rate signature

The rate signature is a peak that arises from the fact that several probe packets traverse the network path back-to-back, because of the chain patterns. The rate signature corresponds to the link rate of the link creating the chain pattern.

The rate peak grows in size when the probe rate increases, since more probe packets will travel back-to-back in such cases.

The rate peak is visible in the bottom left diagram of Figure 5.5. The rate peak is the leftmost peak. The delay variation value for that peak can be converted to the link rate.

5.4.4 Quantification signature

The quantification signature arises from the quantification pattern described above. The quantification signature corresponds to the gap between probe packets. In the bottom right diagram of Figure 5.5 there are 3 clearly separated regions of peaks. The leftmost peak corresponds to the rate signature.

The locations of the quantification peaks depend on the size of the cross traffic packets and the number of packets of each size that are interfering.

5.5 Mean and median analysis using patterns

An important step in the analysis phase is to reduce noise and effects stemming from cross-traffic burstiness. One way to do this is to mean or median filter the dispersion samples. It is therefore important to understand the impact and properties of these statistical operations.

In a previous simulation-based study [10] we have shown that the patterns described in Section 5.2 affect the mean and median. However, they are affected in different ways. These differences can be illustrated and explained in terms of the packet interactions described as patterns in Section 5.2.

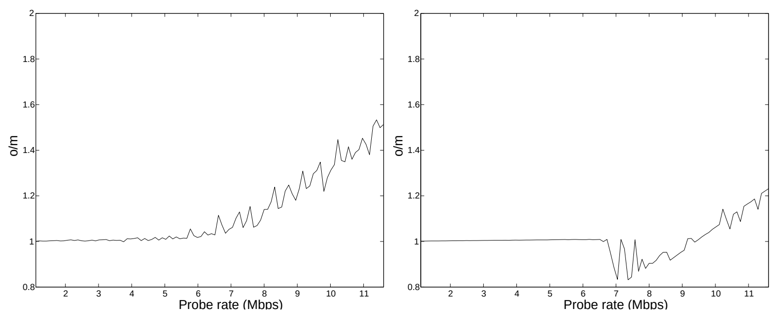

Figure 5.6: Mean offered / measured probe rate to the left. Median offered / measured probe rate to the right. Cross traffic = 3 Mbps on a 10 Mbps link.

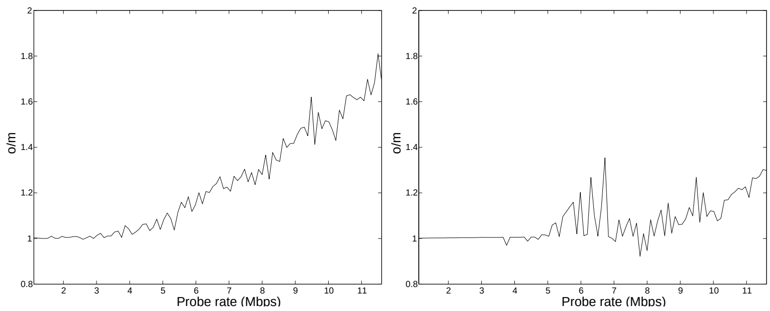

Figure 5.7: Mean offered / measured probe rate to the left. Median offered / measured probe rate to the right. Cross traffic = 5 Mbps on a 10 Mbps link.

Figures 5.6 and 5.7 show measurements from two different scenarios with increasing amounts of cross traffic. The topology is described in Section 8.2 with only cross traffic flow 1 being active. Both graphs in each figure show the ratio between offered and measured dispersion rate on the y-axis. The x-axis is the probe rate. In the left side graphs mean filtering is used whereas median filtering is used in the right hand graphs. A ratio close to 1 corresponds to an underload situation. In the part where a curve deviates from 1, the slope using mean filtering is inversely proportional to the link capacity [5].

From the graphs it is apparent that the mean and median filtering give different results. When the probe rate increases the number of probe and cross-traffic packet interactions will also increase. These interactions will occur as mirror patterns followed in turn by chain and quantification patterns. In the case of mean filtering, the asymmetry effects of the mirror and the chain patterns will be cancelled. Hence, they will have no effect on the offered / measured ratio. The quantification patterns that arise when the link is full (i.e. when the available bandwidth is reached) cause the slope of the curves seen in the graphs regardless if mean or median filtering is used.

The symmetry effect of the mirror pattern is also cancelled with respect to median filtering. However, the asymmetry of chain patterns is not cancelled. The reason is that number of back-to-back probe packets constituting the chain pattern outnumbers the initial probe packet dispersion. Thus, the median is shifted towards the dispersion of back-to-back packets. Since the back-to-back chain corresponds to the measured probe rate whereas the offered rate correspond to the initial probe packet dispersion there will be a dip near 7 Mbps in the median ratio curve in Figure 5.6.

Using the mathematical definitions of chain patterns it can be shown that chain patterns can not occur until probing at 7 Mbps on a 10 Mbps link with an MTU of 1500 bytes.

Figure 5.7 shows an even more striking behavior when median filtering is applied. The median filtering curve begins to rise at 5 Mbps because of quantification patterns that occur when the channel is full. At 7 Mbps there is a drop in the ratio curve caused by chain patterns. As before, these chain patterns are invisible when using mean filtering. When the probe rate is increased further the quantification patterns will again dominate and hence cause a new rise of the median curve.

The two median curves has a chain saw shape near the available bandwidth (7 Mbps in Figure 5.6 and 5 Mbps in 5.7) compared to the mean curves. Exactly why this happens is ongoing research. We also study if there is a possibility to combine the information from mean and median curves to make better estimates of the available bandwidth.

5.6 Conclusions

In this article we have identified and described patterns that arise when probe packets and cross-traffic packets interact with each other. Further, we have identified signatures that originate from these patterns. We have studied how these patterns and their corresponding signatures affect mean and median operations that are used in the analysis phase of available bandwidth measurement methods. All measurements in this article has been made in a testbed.

Our goal is to further study the patterns and the corresponding signatures derived from dispersion values obtained from packet-train probing schemes. We will study mean and median operations and especially the combination of them to make better estimates of available bandwidth and link capacity on an end-to-end path.

Bibliography

[1] Robert Carter and Mark Crovella. Measuring bottleneck link speed in packet-switched networks. Technical Report 1996-006, Boston University Computer Science Department, Boston, MA, USA, March 1996.

[2] Constantinos Dovrolis, Parameswaran Ramanathan, and David Moore. What do packet dispersion techniques measure? In *Proceedings of IEEE INFOCOM*, pages 905–914, Anchorage, AK, USA, April 2001.

[3] Srinivasan Keshav. A control-theoretic approach to flow control. In *Proceedings of ACM SIGCOMM*, pages 3–15, Zürich, Switzerland, September 1991.

[4] Kevin Lai and Mary Baker. Measuring bandwidth. In *Proceedings of IEEE INFOCOM*, pages 235–245, New York, NY, USA, March 1999.

[5] Bob Melander, Mats Björkman, and Per Gunningberg. Regression-based available bandwidth measurements. In *Proceedings of the 2002 International Symposium on Performance Evaluation of Computer and Telecommunications Systems*, San Diego, CA, USA, July 2002.

[6] Manish Jain and Constantinos Dovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. In *Proceedings of ACM SIGCOMM*, Pittsburg, PA, USA, August 2002.

[7] Attila Pásztor and Darryl Veitch. The packet size dependence of packet pair like methods. In *Tenth International Workshop on Quality of Service (IWQoS 2002)*, Miami Beach, USA, May 2002.

[8] Rikard Holm. Probe-based load measurement. Master thesis, Ericsson Research, Sweden, 2003.

[9] Yongmin Choi, Heung-No Lee, and Anurag Garg. Measurement and analysis of wide area network (wan) traffic. In *SCS Symposium on Performance Evaluation of Computer and Telecommunication Systems*, July 2000.

[10] Andreas Johnsson, Bob Melander, and Mats Björkman. Modeling of packet interactions in dispersion-based network probing schemes. Technical report, Mäalardalen University, 2004.

Chapter 6

Paper B: A Study of Dispersion-based Measurement Methods in IEEE 802.11 Ad-hoc Networks

Andreas Johnsson, Mats Björkman and Bob Melander
In proceedings to the International Conference on Communication in Computing, Special Session on Network Simulation and Performance Analysis, Las Vegas, June 2004

Abstract

In this article we discuss how dispersion-based measurements are affected by multiple-access networks, such as IEEE 802.11-based ad-hoc networks. We study four topics: variable link capacity, movement of wireless nodes, loss rate and timing issues.

We show by simulations that the link capacity in a wireless topology is changing due to cross traffic generated from nodes using the same radio frequency channel. Further, we address the problem of moving nodes. Moving nodes may cause route changes during a dispersion-based measurement session. Since all dispersion-based measurement methods are running during a time interval, moving nodes will cause problems.

We extend the term *available bandwidth* to cope with losses in shared medium access networks. Losses will lower the available bandwidth that can be used by applications.

Finally we study the problem of time jitter induced between successive probe packets by multiple-access networks.

6.1 Introduction

Active measurements of end-to-end characteristics are getting increasingly important in best effort networks, especially in the Internet. Such measurements are by definition made between two or more end hosts with no knowledge of the network in-between. Characteristics measured by current tools are path capacity and available bandwidth (both terms are defined in subsection 6.2.3). Many applications could benefit from having an estimate of the available bandwidth between the end hosts. Such applications are for example streaming media players and route decision algorithms.

The measurement methods have matured and evolved over the past years. Today there exist a variety of tools to measure network path characteristics. Examples are found in [1, 2, 3, 4, 5].

However, these methods were developed having the traditional wired Internet in mind. Researchers, corporate organizations, rescue teams and the military are deploying wireless ad-hoc networks in a higher degree than before. To this reason, dispersion-based network measurements must be adapted to such environments.

This article discusses research problems that should be considered when applying dispersion-based network measurement methods designed for the Internet to wireless ad-hoc networks. However, since this article is an initial study it does not give concrete answers.

6.1.1 Ad-hoc networks

This section describes ad-hoc networks on a high level. More information can be found in any text book on the subject, e.g. [6].

An ad-hoc network can be defined as a set of spontaneously connected nodes that need to communicate with each other for some reason. Since the nodes are connected on demand, ad-hoc networks usually rely on wireless communication techniques, such as the IEEE 802.11 standard [7, 8].

In ad-hoc networks there is no pre-configured infrastructure. This means that the basic networking functionality (e.g. routing and forwarding) must be supported by each node within the network. In Figure 6.1, node 1 wants to communicate with node 4. To be able to do this, the messages from node 1 must be routed by either node 2 or 3 to reach node 4. This is because the wireless device of node 1 can not send its message directly to node 4 due to limited radio coverage.

Ad-hoc networks are usually dynamic in their nature. This means that the

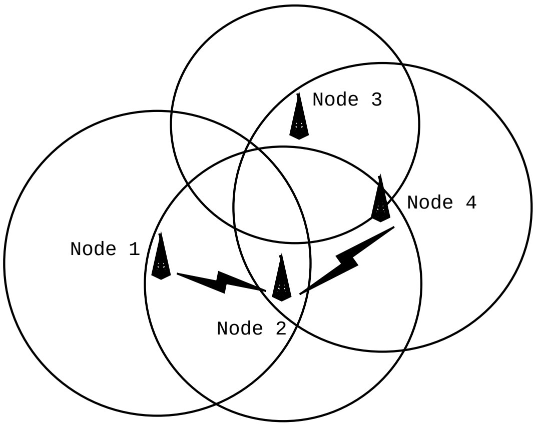

Figure 6.1: Wireless ad-hoc network with 4 nodes. Node 1 is sending a message to node 4 via node 3 since the radio in node 1 can not reach node 4's position.

nodes will move and thus the route used between two nodes will change. There are several routing techniques for ad-hoc networks [6].

6.1.2 Dispersion-based measurements

Active end-to-end measurements of available bandwidth are usually divided into two phases; the data collection phase and the analysis phase.

In the data collection phase, a probe generator injects probe packets along the path to be measured. The probing scheme (i.e. the dispersion or separation between successive probe packets) is predefined by the sender. Two common probing schemes are the injection of probe packets in pairs or in a train, respectively [9, 10, 3, 11, 4]. The initial probe packet dispersion is proportional to the probe rate.

The dispersion between successive probe packets changes when the probe packets traverse the network path. It either changes due to limited link capacity or due to interaction with other packets traversing the same path (such packets are called cross traffic). Limited link capacity will increase the dispersion between probe packets while packet interactions may increase as well as decrease the dispersion.

The probe packets are received by a probe receiver. Upon reception, the probe packets are time stamped. Using these time stamps the probe packet dispersion at the receiver is calculated.

The second phase, the analysis, uses the dispersion values obtained from the data collection phase to produce an estimate. The difference between the initial probe packet dispersion and the received probe packet dispersion is used to produce an estimate of the path capacity or the available bandwidth.

The data collection and the analysis phases combined is called a measurement session.

6.2 Research problems

In this section we discuss properties of multiple-access networks that differ from single-access networks. Multiple-access networks are for example based on the 802.3 (Ethernet) or IEEE 802.11 (wireless) standards. In this article we focus on wireless networks. In particular, we study IEEE 802.11b (hereafter called 802.11) wireless ad-hoc networks, which is one of the common standards for radio communication. All topics are discussed from the perspective of dispersion-based probing schemes.

Dispersion-based probing schemes make implicit assumptions that the network properties, such as link capacity, cross traffic and the route between end nodes are the same during the measurement (especially during the data collection phase). However, dispersion-based measurements are not instantaneous. They run for some time period. Since wireless ad-hoc networks tend to be very dynamic, we must understand and compensate for the dynamic properties. In the following subsections we discuss how the link capacity may change due to cross traffic or due to node movement. We also address route changes, loss rate and time control issues.

6.2.1 Variable link capacity

In single-access networks, the link capacity on a link is determined by physical characteristics of the link itself and is usually static. However, in 802.11b networks the link capacity is typically changing dynamically over time. This variation is due to changing distance between nodes due to their mobility, radio noise, weather conditions and physical obstacles. However, the link rate variability is also due to cross traffic using the same radio frequency channel within the ad-hoc network.

Below, we show simulation results that illustrate that the link capacity is changing due to cross traffic. The simulation topology is shown in Figure 6.2 and the simulation tool is ns-2 [12]. Two nodes generate cross traffic between each other while the other two nodes are measuring network link characteristics using dispersion-based measurement methods.

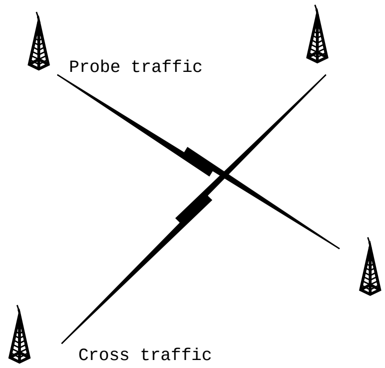

Figure 6.2: A single hop wireless ad-hoc topology. Cross traffic and probe packets use the same shared medium. The cross traffic will be affecting the link capacity.

The diagrams in Figure 6.3 show a comparison of the offered probe rate and the measured probe rate. The offered / measured ratio is shown on the y-axis. The x-axis is the probe rate, starting at 1 Mbps. The measured probe rate is based on the mean of the probe packet dispersion values obtained in the data collection phase. The offered probe rate is predefined by the probe packet sender and hence known in advance.

The figure shows 4 diagrams corresponding to 4 different scenarios. Each scenario uses the same topology, but the cross traffic vary. As long as the offered probe rate and the measured probe rate quotient remains around 1, the mean dispersion between successive probe packets are the same on the sender side and the receiver side. That is, we have an underload situation. When the curve starts to rise the mean dispersion on the receiver side (i.e. the measured value) is larger than on the sender side. This change in dispersion indicates where the available bandwidth is. The slope of the curve after this point is inversely proportional to the link capacity, according to TOPP [3].

In Figure 6.3 the cross traffic is, starting from the upper left diagram, 0

Mbps, 0.5 Mbps, 1 Mbps and 2 Mbps, respectively. The cross traffic is uniformly distributed. The measured available bandwidth is 4.7 Mbps, 4.2 Mbps, 3.2 Mbps and 2.8 Mbps, respectively. In this case the perceived link capacity is equal to the available bandwidth for each scenario. According to TOPP, the slope of the curves is inversely proportional to the minimum link capacity of the network path. The slopes of the curves are approximately 0.22, 0.28, 0.33 and 0.38, respectively. Hence, for wireless networks the perceived link capacity will change due to cross traffic.

The ns-2 wireless simulation topology was configured to run at 11 Mbps. The probe packet size was 1500 bytes.

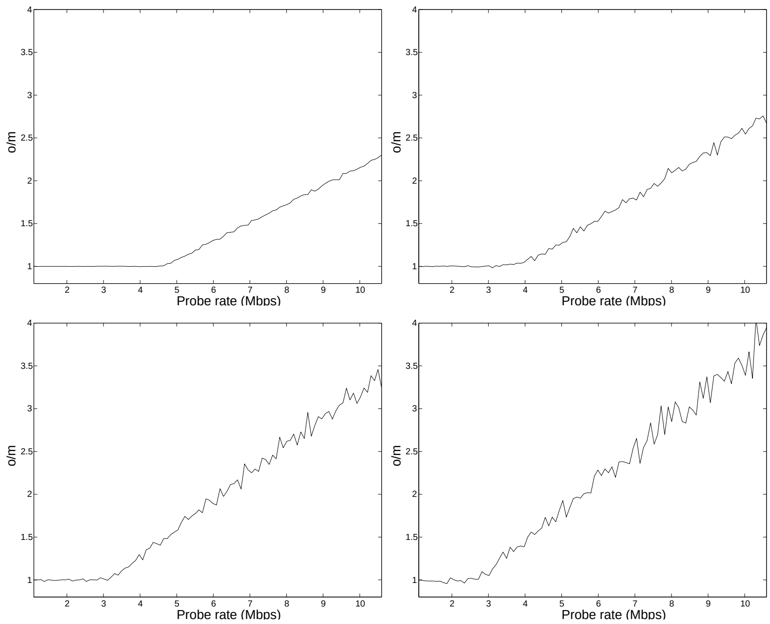

Figure 6.3: The offered probe rate is compared to the measured probe rate. The slope of each curve is inversely proportional to the link capacity. The cross traffic rate is 0 Mbps in the upper left diagram, 0.5 Mbps, 1 Mbps and 2 Mbps in the bottom right diagram.

Since active end-to-end measurement techniques measure the link characteristics during a time interval, the cross traffic load may change during this interval. If that happens, the link capacity might change during the measurement session. Measurement methods today do not address this problem. Since available bandwidth measurement methods rely on an estimate of the link capacity, a changing link capacity will introduce errors in the available bandwidth estimation.

6.2.2 Moving nodes

In ad-hoc wireless networks the distance between mobile nodes may change due to mobility. This creates problems that measurement methods must handle.

When two mobile nodes move away from each other the radio signal strength will go down. The link capacity will decrease in discrete steps when the radio signal strength is reduced to certain threshold levels. That is, we have the same problem as in Section 6.2.1, where variable link capacity was discussed.

As described in the introduction, all routing in ad-hoc networks is done by the nodes themselves. When nodes move around, the route between two nodes may change since the signal strength may become zero between two nodes in the route. Hence, two or more different end-to-end routes may be measured during one single dispersion-based measurement session.

If a static network topology is assumed (which is what dispersion-based measurement methods today typically do), this will cause problems. This topic is subject to further study. We will look into how to detect and handle route changes when using dispersion-based measurement methods.

6.2.3 Loss rate

Packet losses are not very common in single-access networks, especially not in wired networks. When they arise it is typically due to queue overflow at a router somewhere in the path. However, in 802.11b wireless networks the loss rate is much higher. This is due to collisions on the shared medium, corrupt packets, et cetera. Assume the following scenario. A dispersion-based measurement injects probe packets into the wireless network path. Its estimate of the available bandwidth becomes A. Using this rate A when sending packets on the wireless link may cause packet losses due to the transmission error properties of the wireless medium. This means that the available bandwidth is limited not only by the packet dispersion (i.e. the packet rate) but also by the loss rate.

Because of this fact we propose a definition of the available bandwidth that addresses this problem.

The link capacity is by definition the bit rate of the link (i.e. the number of bits that can be transfered during time τ). Further, the path capacity is defined as the link capacity with the least bit rate on a given path consisting of H links. It is defined as $C = \min_{i=1..H}(C_i)$, where i is the link index.

The utilization of a link is by definition the number of bits transfered during time τ divided by the link capacity. The utilization is within the interval $[0, 1]$.

The definition of available bandwidth on one single link is (link capacity) * (1 - utilization) during time τ. The available bandwidth between two end hosts then becomes $A = \min_{i=1..H}(C_i * (1 - u_i))$, where u_i is the utilization of link i.

To compensate for the loss rate, a term describing the loss rate must be added to the equation describing the available bandwidth above. We get

$$A = [\min_{i=1..H}(C_i * (1 - u_i))] * (1 - L_{path}), \qquad (6.1)$$

where L_{path} describes the total probe packet loss rate due to contention seen on the path when sending packets at a rate corresponding to the available bandwidth. Equation 6.1 assumes that all losses occur on or downstream the link with least available bandwidth. Should losses occur elsewhere, our definition is conservative.

This definition also holds in ordinary single-access networks, since the loss rate is close to 0.

6.2.4 Time control

This subsection discusses the difference between single-access and 802.11 networks at the MAC-layer. In 802.11 networks the sender at the application layer does not know when a packet is transmitted on the link. Packets that are sent from a node traverse the network protocol stack from the top layer down to the lower layers. The top layers (application, transport and network layer) are the same regardless if a single-access or a 802.11 network is used by the node sending the packet. However, the lower layers (MAC and physical) differ.

When sending a packet using single-access networks we can approximate the time between sending the packet from the application layer and transmitting the packet onto the link. The time spent in the protocol stack is more or less constant for each packet, assuming that the send rate is not higher than the capacity of the network itself. That is, we can calculate the offered dispersion between two successive probe packets.

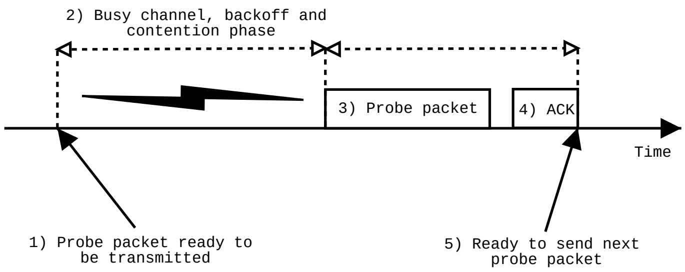

Figure 6.4: The MAC-layer scheme when sending a packet using a 802.11 network.

However, in a 802.11 wireless network the lower layers will induce time jitter between successive probe packets. This jitter depends on, among other things, cross traffic using the same radio frequency channel. In Figure 8.11 the jitter effect is illustrated. A probe packet is sent in the application layer and traverses down through the protocol stack to the MAC layer. The probe packet is ready to be transmitted on the shared medium, indicated by (1) in the figure. Because of other traffic the probe packet has to wait until the shared medium is idle. The time a probe packet is in the contention phase (shown by (2) in the figure) depends on the cross traffic. When the probe packet is actually successfully transmitted (3) the node has to wait until an acknowledgment is received (4). The time to receive an acknowledgment depends on other competing packets and the network latency. That is, the dispersion between two successive probe packets is not deterministic.

When the analysis phase is performed, the offered dispersion values obtained from the data collection phase is one important component to produce a good estimate of the measured network path characteristic. Probe-packet dispersion jitter will make the estimate of the network characteristic less accurate. To get more accurate estimates, the analysis requires more dispersion samples which means that more probe packets have to be sent. However, this should be avoided since the bandwidth in wireless networks is usually the bottleneck.

Exactly how the dispersion jitter will affect current end-to-end dispersion-based measurement methods is a subject of further research.

6.3 Conclusions

In this paper we have analyzed properties in IEEE 802.11 ad-hoc networks that differs from single-access networks, from the view of dispersion-based measurement methods.

We have performed ns-2 simulations to show that the link capacity vary depending on the cross traffic load. Further, we have extended the term available bandwidth to cope with packet losses in wireless networks. In the article we raised the thought that variable link capacity and loss rate may cause problems to dispersion-based measurement methods.

We have also discussed the problems of moving nodes, route changes and dispersion time jitter at the probe sender.

In future research, we intend to study how the link capacity vary in larger non-simulated topologies. Further we will study how to detect and handle route changes in an ad-hoc network when using dispersion-based methods. Our extended definition of available bandwidth is also subject of further study. We will investigate to what extent our extended definition is overly pessimistic. Finally, we will explore how time jitter will affect contemporary available bandwidth measurement methods.

Bibliography

[1] Van Jacobson. Pathchar - a tool to infer characteristics of Internet paths. Presented at the Mathematical Sciences Research Institute (MSRI), April 1997. Slides available from ftp://ftp.ee.lbl.gov/pathchar/.

[2] Bruce Mah. pchar: A tool for measuring Internet path characteristics, June 2001. http://www.employees.org/˜bmah/Software/pchar/.

[3] Bob Melander, Mats Björkman, and Per Gunningberg. Regression-based available bandwidth measurements. In *Proceedings of the 2002 International Symposium on Performance Evaluation of Computer and Telecommunications Systems*, San Diego, CA, USA, July 2002.

[4] Manish Jain and Constantinos Dovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. In *Proceedings of ACM SIGCOMM*, Pittsburg, PA, USA, August 2002.

[5] Vinay Ribeiro, Rudolf Riedi, Richard Baraniuk, Jiri Navratil, and Les Cottrell. pathchirp: Efficient available bandwidth estimation for network paths. In *Passive and Active Measurement Workshop*, 2003.

[6] Charles Perkins. Ad hoc networking. Addison-Wesley, 2001.

[7] ISO/IEC. Ieee 802.11 standard. ieee standard for information technology. ISO/IEC 8802-11:1999(E), 1999.

[8] ISO/IEC. Corrigendum to ieee 802.11b standard 1999. ieee standard for information technology. ISO/IEC 8802-11:1999/Cor 1-2001(E), 2001.

[9] Robert Carter and Mark Crovella. Measuring bottleneck link speed in packet-switched networks. Technical Report 1996-006, Boston University Computer Science Department, Boston, MA, USA, March 1996.

[10] Constantinos Dovrolis, Parameswaran Ramanathan, and David Moore. What do packet dispersion techniques measure? In *Proceedings of IEEE INFOCOM*, pages 905–914, Anchorage, AK, USA, April 2001.

[11] Ningning Hu and Peter Steenkiste. Evaluation and characterization of available bandwidth probing techniques. *IEEE Journal on Selected Areas in Communication*, 2003.

[12] The network simulator - ns-2. http://www.isi.edu/nsman/ns/.

Chapter 7

Paper C: DietTopp: A first implementation and evaluation of a simplified bandwidth measurement method

Andreas Johnsson, Bob Melander and Mats Björkman
In proceedings to the Second Swedish National Computer Networking Workshop (SNCNW), Karlstad, 2004

Abstract

This paper describes the active available bandwidth measurement tool DietTopp. It measures the available bandwidth and the link capacity on an end-to-end path having one bottleneck link. DietTopp is based on a simplified TOPP method. This paper describe and motivate the simplifications and assumptions made to TOPP. Further, the paper describes some of the DietTopp implementation issues.

A first evaluation of DietTopp in a testbed scenario is made. Within this evaluation the performance and measurement accuracy of DietTopp is compared to the state-of-the-art tools Pathload and Pathrate.

We show that DietTopp gives fast and accurate estimations of both the available bandwidth and the link capacity of the bottleneck link.

7.1 Introduction

Measurements in best-effort networks are important for network error diagnosis and performance tuning but also as a part of the adaptive machinery of user applications such as streaming video. Within our research we have focused on actively measuring available bandwidth between two network end-points. Such active measurements are done by injecting probe packets (with a predefined separation) into the network. The probe packets are time stamped at the receiver end. These time stamps are then used to form an estimate of the available bandwidth. This is discussed in more detail in Section 7.2.

State-of-the-art bandwidth measurement tools and methods are for example TOPP [1], Pathload [2], Pathchirp [3], Delphi [4] and Spruce [5]. An overview of methods and tools in this area can be found in [6].

Within the scope of this paper we have developed, implemented and evaluated a new available bandwidth measurement tool called DietTopp. This tool relies on a simplification of the TOPP bandwidth measurement method [1]. We show that DietTopp gives fast and accurate estimates of both the available bandwidth and the measured link capacity when there is one congested link in the end-to-end path.

This paper is organized as follows. The TOPP measurement method is briefly described in Section 7.2. The simplifications and assumptions made by DietTopp are discussed and motivated in Section 8.2.1. DietTopp implementation issues are described in Section 7.4. Section 8.2.2 describes the testbed that has been used to evaluate our DietTopp implementation, while Section 8.3 shows and discusses the obtained results. Section 8.3 also compare DietTopp to other state-of-the-art measurement tools.

7.2 TOPP: the original method

This section briefly describes the original TOPP measurement method that estimates the available bandwidth and link capacity on an end-to-end path. More information on definitions and theory can be obtained from [1].

The TOPP measurement method is divided into two phases, the probing phase and the analysis phase. These two phases are separately described in the following two subsections.

7.2.1 Probe phase

Starting at some offered probe rate o_{min}, the TOPP method injects m probe packet trains, where each train contains k equally sized probe packets, into the network path. When all probe trains corresponding to a probe rate o_{min} have been received on the other end, TOPP increases the offered rate $o = o_{min} + \Delta o$. Another set of probe packet trains are sent through the network at the new probe rate. This is repeated until the offered probe rate reaches some specified probe rate o_{max} (i.e. $o > o_{max}$).

The probe packet separation changes between the probe sender and the probe receiver. This is due to the *bottleneck spacing effect* [7] which basically says that the time separation increases in a predictable manner when a link is congested.

The receiver time stamps each packet arrival. Hence, the change in probe packet separation can be measured. The time stamps are then used to calculate the measured probe rate m. When the measured probe rate and its corresponding offered probe rate is known the analysis phase of TOPP can be executed. The analysis is described in the next subsection.

7.2.2 Analysis phase

The analysis builds on comparing the offered probe rate o_i to the measured probe rate m_i on each probe rate level i. If plotting the ratio o_i/m_i on the y-axis and o_i on the x-axis for all i, we get a plot like the theoretical one in Figure 8.1. When the network is underloaded the o_i/m_i ratio will be close to $y = 1$. When TOPP increases the offered probe rate some link on the network path will eventually get saturated. Hence, the measured probe rate will decrease since the probe packet separation increases (due to the bottleneck spacing effect). This causes the curve to rise. Segment b_1 is linear and the slope corresponds to the link bandwidth of the first congested link. The available bandwidth of the end-to-end path is defined as the intersection of $y = 1$ and the linear segment b_1 (t_1).

If there are more than one congested link on the end-to-end path the curve will be divided into several linear segments, when increasing the offered probe rate. Each new segment, b_2 and b_3, corresponds to the fact that an additional link has been congested. That is, the number of linear segments depends on the number of saturated links on the end-to-end path. The saturation point for each link can be calculated using t_1, t_2 and t_3 in the figure.

The original TOPP method can, in many cases, determine not only the end-

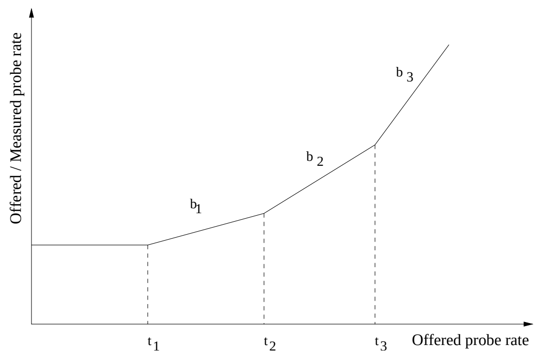

Figure 7.1: Plot of offered prob rate / measured probe rate as a function of the offered probe rate. The three segments b_1 - b_3 corresponds to three congested links while t_1 - t_3 are breakpoints between congested links.

to-end available bandwidth (corresponding to the intersection of $y = 1$ and the linear segment b_1) but also the link available bandwidth of each congested link. This is done by extracting additional information from the linear segments shown in Figure 8.1. Exactly how this is done is described in [1].

7.2.3 TOPP complications

The problem with the original TOPP method is its complex iterative algorithm it uses to estimate the segment intersection points. This makes the TOPP analysis phase computationally expensive. It is feasible, as shown in [1] but takes a lot of computation power.

7.3 DietTopp

This section describes DietTopp. That is, the simplifications we have made to the TOPP analysis.

DietTopp assumes that only one link between the sender and the receiver is congested. That is, there will only be one linear segment in Figure 8.1 (i.e. b_1). Hence, the end-to-end link capacity is proportional to the slope of segment b_1 and the end-to-end available bandwidth is defined as the intersection of b_1 and $y = 1$ (t_1). That is, the iterative part of TOPP can be omitted.

Since we want to identify b_1, the first step is to ensure that DietTopp sends probe packets at a rate above the available bandwidth (corresponding to t_1 in Figure 8.1). This is done by calculating the proportional share, ps_{max}, of the end-to-end path at probe rate o_{max}. Hence, $ps_{max} = o_{max}/(x + o_{max}) * l$ where o_{max} is the maximum offered probe rate, x the cross traffic and l the link capacity of the congested link. The ps_{max} is estimated by injecting a set of probe packets back-to-back into the network and then measure their separation at the receiver. The ps_{max} is an overestimation of the available bandwidth, as discussed in [1] (and referred to as the asymptotic dispersion rate in [8]).

When DietTopp has obtained a value of ps_{max} it continues the probing phase by injecting probe packets at rates in the interval $[ps_{max}, z * ps_{max}]$, where z is a constant. This will generate o_i/m_i values corresponding to the linear segment b_1 in Figure 8.1. That is, DietTopp can calculate the link capacity by finding the slope of b_1 and the available bandwidth by calculating the intersection of b_1 and $y = 1$. This is done using linear regression.

We argue that the assumption that only one link is congested in a path is not too far fetched. Usually the bottlenecks are found in the access networks, close to the user, for example when the user is using a wireless connection. Usually wireless links provides a much lower bandwidth than the rest of the links in the path.

Other probing tools that assume one single bottleneck link are Spruce [5] and Delphi [4]. In addition, these tools require prior knowledge of the bottleneck link capacity. This is not the case when using DietTopp. On the contrary, DietTopp will estimate that capacity as part of its estimation procedure.

7.4 Implementation of DietTopp

This section gives an overview of the DietTopp implementation. More information about the implementation ca be obtained by downloading the tool along with its source code [9].

DietTopp is designed for Unix system and is implemented in C++. It has a sender and a receiver part. The sender *probes* the network path by injecting a set of packet trains at different rates similarly to TOPP (in the original method TOPP used pairs of probe packets instead of trains of probe packets). The receiver records the time for each probe packet arrival. These values are sent back to the probe sender for analysis. The analysis is done using our simplified TOPP method described in Sections 7.2 and 8.2.1.

The probe packets used by DietTopp are UDP/IP packets with a size of

1500 bytes. The packets are divided in trains where each train consists of 16 packets. On each probe rate level DietTopp sends 5 trains. DietTopp uses 15 prob rate levels in the interval $[ps_{max}, z * ps_{max}]$ by default.

When DietTopp is measuring the proportional share (described in Section 8.2.1), 15 trains with 48 packets in each are sent at the maximum probe rate (i.e. the sender's link speed).

The DietTopp implementation can be summarized as follows:

1. Send a set of probe trains at maximum rate
2. Record the probe packet arrival times at receiver
3. Send time stamps back to sender
4. The proportional share (ps) is estimated using the arrival times
5. The sender initiates and transmits a set of probe trains with rates in the interval $[1 * ps, 1.5 * ps]$
6. The receiver records the probe packet arrival times
7. The arrival times are sent back to the sender
8. The arrival times are analyzed using our simplified TOPP method
9. If the correlation is large enough and if the standard deviation small enough DietTopp presents its results and terminates
10. Else, repeat from step 1, but increase the number of probe packets in each train

7.5 The testbed

The testbed we have used in this work consists of 7 ordinary PC machines. The PCs acting as routers (R1 - R3) are connected by 10 Mbps links while the rest of the links can communicate at 100 Mbps. The testbed is shown in Figure 8.2. The probe tool sender is running on *S* while the receiver is running on the destination machine *D*.

The cross traffic can either take the route $X1 \rightarrow R1 \rightarrow R2 \rightarrow X2$ or the route $X1 \rightarrow R2 \rightarrow R3 \rightarrow X2$. The cross traffic itself is exponentially distributed and consists of 60, 148, 500 and 1500 byte packets (this corresponds

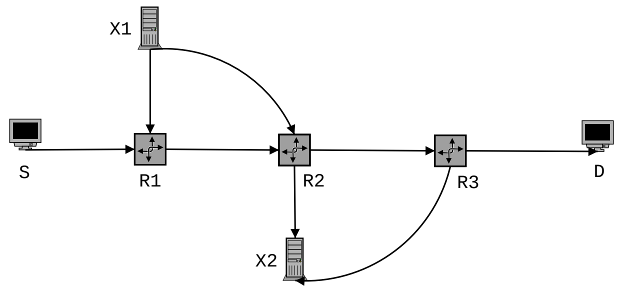

Figure 7.2: The testbed used for a first evaluation of DietTopp.

to the packet size at the *link layer*). This distribution of packet sizes originates from findings in [10]. The cross traffic is generated by *tg* [11].

An important observation is that the ethernet cards used in the testbed can not send packets back-to-back. The cards add a gap corresponding to 25 bytes between two successive packets. This will be manifested as if all packets will have an increased effective packet size. That is, a 500 byte packet will have an effective packet size of 525 byte when the ethernet card continuously has to forward packets.

7.6 DietTopp Evaluation

This section presents a first evaluation of DietTopp. We have performed measurements in the testbed described in Section 8.2.2. The cross traffic flows through the route $X1 \rightarrow R1 \rightarrow R2 \rightarrow X2$. That is, the probe packets injected by DietTopp are only affected by cross traffic on one hop (i.e. between one pair or routers). The cross traffic composition and distribution is described in Section 8.2.2.

We have used the state-of-the-art tools Pathload and Pathrate [2, 12] to compare the accuracy and performance of the measured available bandwidth and link capacity. We show that DietTopp gives accurate and fast estimates. The results are presented and discussed in the following subsection.

7.6.1 Measurement results

The measurements in Figure 8.3 originate from DietTopp measurements under four different cross traffic intensities - 0, 3.75, 6.26 and 8.76 Mbps (shown on the x-axis). The y-axis shows the measured link capacity (thick solid line) and measured available bandwidth (thin solid line). The link capacity has a decreasing trend when increasing the cross traffic intensity. Exactly why this happens is subject of further research.

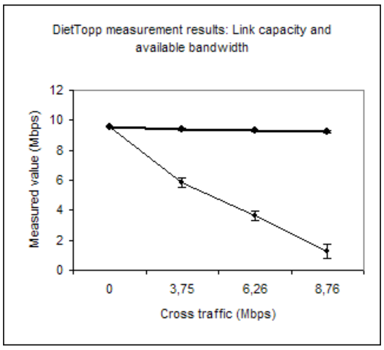

Figure 7.3: Link capacity (thick solid line) and available bandwidth (thin solid line) measured by DietTopp.

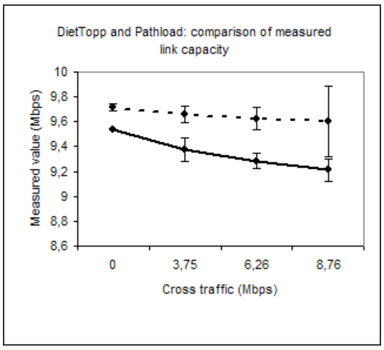

Figure 7.4: Link capacity measured by DietTopp (solid line) and Pathrate (dashed line).

The diagram in Figure 8.4 compares the measured link capacity when using DietTopp (solid line) and Pathrate (dashed line). It is clear that DietTopp underestimates the link capacity in comparison to Pathrates estimation. However, Pathrates estimation is in turn an underestimation compared to the theoretical link capacity of 10 Mbps.

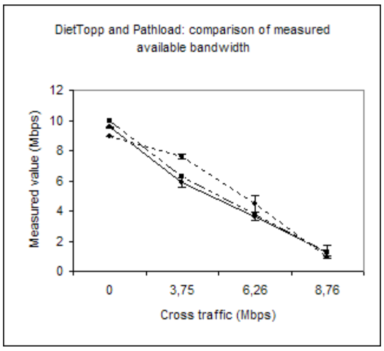

Figure 7.5: Available bandwidth measured by DietTopp (solid line) and Pathload (dashed line). The theoretical available bandwidth is shown as the double dot dashed line.

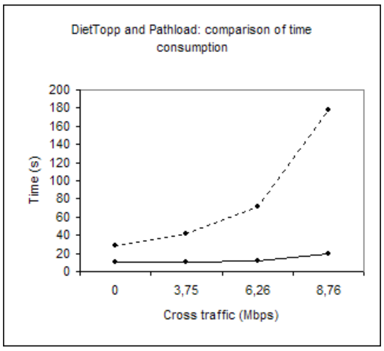

Figure 7.6: Time consumption at different cross traffic rates. The solid line corresponds to DietTopp while the dashed line corresponds to Pathload.

The diagram in Figure 8.6 is a comparison of the measured available bandwidth. The solid line corresponds to DietTopp while the dashed line corresponds to Pathload. The double dot dashed line is the theoretical available bandwidth under different cross traffic rates. Here we see that both methods estimate the available bandwidth quite well, even if Pathload tends to overestimate in some cases.

The diagram in Figure 8.5 is a performance comparison of DietTopp and Pathload. The diagram compares the time consumption of DietTopp (solid line) and Pathload (dashed line). As can be seen, the time to measure the available bandwidth using DietTopp is almost constant while for Pathload the time grows exponentially.

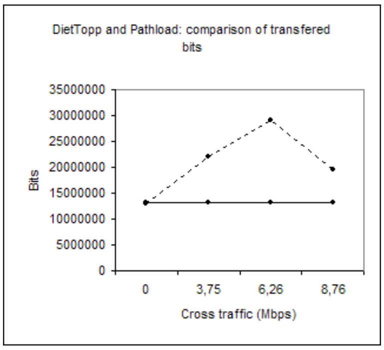

Figure 7.7: Number of transfered bits by DietTopp (solid line) and Pathload (dashed line).

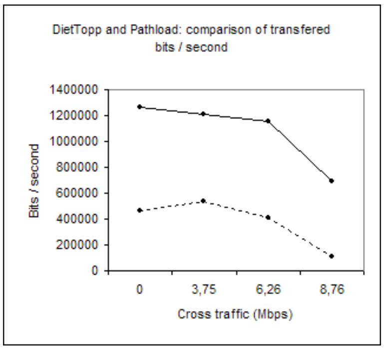

Figure 7.8: Number of transferred bits / second by DietTopp (solid line) and Pathload (dashed line).

It should also be noted that DietTopp does not only measure the available bandwidth during its execution time, but also the link capacity of the congested link.

The diagrams in Figure 8.7 and 8.8 compare the number of bits transferred totally during a measurement session and the number of bits transferred per second when using DietTopp (solid line) and Pathload (dashed line). Pathload uses more bits, but since DietTopp runs over a shorter period of time DietTopp is more aggressive as can be seen in the diagram in Figure 8.8.

To summarize, our first evaluation of DietTopp shows that DietTopp estimates both the link capacity and the available bandwidth with comparable and in some cases better accuracy than Pathload and Pathrate. Further, our tool runs over a shorter period of time. The drawback is that DietTopp is more aggressive than both Pathload and Pathrate.

In our continued work we will investigate how the aggressiveness of DietTopp affects TCP flows and other communication protocols. We will also try to make estimations of the available bandwidth and the link capacity with equal accuracy but with fewer probe packets.

7.7 Conclusions

We have simplified the TOPP measurement method and implemented the new method in a tool that we call DietTopp. We have described and motivated the simplifications and assumptions made to TOPP. Further, we have done an initial evaluation of the accuracy and performance of DietTopp. We have also compared DietTopp to the measurement tools Pathload and Pathrate.

We have shown that our tool gives accurate estimate, in the one hop case, of the available bandwidth and an acceptable estimate of the link capacity. Compared to Pathload our solution is quicker, but with the drawback of a higher network aggressiveness.

We will continue our research by investigate how DietTopp reacts when cross traffic is present on multiple links. We will try to find a way to keep the estimation accuracy but decrease the number of probe packets sent. Finally, we will investigate how the accuracy and speed is affected by wireless networks.

Bibliography

[1] Bob Melander, Mats Björkman, and Per Gunningberg. Regression-based available bandwidth measurements. In *Proceedings of the 2002 International Symposium on Performance Evaluation of Computer and Telecommunications Systems*, San Diego, CA, USA, July 2002.

[2] Manish Jain and Constantinos Dovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. In *Proceedings of ACM SIGCOMM*, Pittsburg, PA, USA, August 2002.

[3] Vinay Ribeiro, Rudolf Riedi, Richard Baraniuk, Jiri Navratil, and Les Cottrell. pathchirp: Efficient available bandwidth estimation for network paths. In *Passive and Active Measurement Workshop*, 2003.

[4] Vinay Ribeiro, Mark Coates, Rudolf Riedi, Shriram Sarvotham, Brent Hendricks, and Richard Baraniuk. Multifractal cross-traffic estimation. In *Proceedings of ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, Monterey, CA, USA, September 2000.

[5] Strauss, Katabi, and Kaashoek. A measurement study of available bandwidth estimation tools. In *ACM SIGCOMM Internet Measurement Workshop*, 2003.

[6] R.S. Prasad, M. Murray, C. Dovrolis, and K. Claffy. Bandwidth estimation: metrics, measurement techniques, and tools. IEEE Network Magazine, 2003.

[7] Van Jacobson. Congestion avoidance and control. In *Proceedings of ACM SIGCOMM*, pages 314–329, Stanford, CA, USA, August 1988.

[8] Constantinos Dovrolis, Parameswaran Ramanathan, and David Moore. What do packet dispersion techniques measure? In *Proceedings of IEEE INFOCOM*, pages 905–914, Anchorage, AK, USA, April 2001.

[9] Andreas Johnsson. Diettopp implementation, beta version. http://www.idt.mdh.se/ãjn12/.

[10] Yongmin Choi, Heung-No Lee, and Anurag Garg. Measurement and analysis of wide area network (wan) traffic. In *SCS Symposium on Performance Evaluation of Computer and Telecommunication Systems*, July 2000.

[11] Paul E. McKenney, Dan Y. Lee, and Barbara A. Denny. Traffic generator software release notes. SRI International and USC/ISI Postel Center for Experimental Networking, January 2002.

[12] Manish Jain and Constantinos Dovrolis. Pathload: a measurement tool for end-to-end available bandwidth. In *Passive and Active Measurement (PAM) Workshop 2002 Proceedings*, pages 14–25, Ft Collins, Co, USA, March 2002.

Chapter 8

Paper D: Bandwidth Measurement in Wireless Networks

Andreas Johnsson, Bob Melander and Mats Björkman
Submitted for publication

Abstract

For active, probing-based bandwidth measurements performed on top of the unifying IP layer, it may seem reasonable to expect the measurement problem in wireless networks, such as ad-hoc networks, to be no different than the one in wired networks. However, in networks with 802.11 wireless links we show that this is not the case. We also discuss the underlying reasons for the observed differences.

Our experiments show that the measured available bandwidth is dependent on the probe packet size (contrary to what is observed in wired networks). Another equally important finding is that the measured link capacity is dependent on the probe packet size *and* on the cross-traffic intensity.

The study we present has been performed using a bandwidth measurement tool, DietTopp, that is based on the previously not implemented TOPP method. DietTopp measures the end-to-end available bandwidth of a network path along with the capacity of the congested link.

8.1 Introduction

Wireless networks, used when connecting to the Internet or when several nodes want to communicate in an ad-hoc manner, are becoming more and more popular. Because of the increased dependence on wireless network technology, it is important to ensure that methods and tools for network performance measurement also perform well in wireless environments. In this paper, we focus on performance measurements in terms of network bandwidth, both link bandwidth and the unused portion thereof; the available bandwidth.

Measurement of network properties such as available bandwidth in for example ad-hoc networks are important for network error diagnosis and performance tuning but also as a part of the adaptive machinery of network applications such as streaming audio and video. Since the exact route between two nodes in an ad-hoc network is usually unknown and may change without notification to the application layer the end-to-end measurement of the available bandwidth should not require any infrastructure or pre-installed components on each node. To achieve that, intermediate end-to-end bandwidth measurement methods can be applied.

State-of-the-art bandwidth measurement methods are for example Pathchirp [1], Pathload [2], Spruce [3] and TOPP [4]. The basic principle is to inject a set of measurement packets, so called *probe packets*, into the network. The probe packets traverse the network path to a receiver node, which time stamps each incoming probe packet. By analyzing these time stamps estimates of the link capacity and/or the available bandwidth can be made. For many end-to-end available bandwidth measurement methods no previous knowledge of the underlying network topology is needed. Therefore, bandwidth estimation methods are well suited for end-to-end performance measurements in ad-hoc networks.

The existing methods differ in how probe packet are sent (the flight patterns) and in the estimation algorithms used. An overview of methods and tools in this area can be found in [5].

In the following sections, we describe and measure bandwidth estimation characteristics when probing in 802.11 wireless networks. We show that both the measured available bandwidth and the measured link capacity are dependent on the probe packet size. Furthermore, our measurements indicate that the measured link capacity is also dependent on the cross-traffic rate. We discuss the origins of some of the observed behavior.

The measurements have been performed in a testbed containing both wireless and wired hops. Our testbed topology only consist of one wireless hop, but we believe that our results illustrate the measurement problem for larger ad-hoc networks, consisting of several wireless hops, as well. To produce measurement results we have used DietTopp, a tool that measures the available bandwidth and link capacity of an end-to-end path. For comparisons and to illustrate that our observations are not tied to a certain measurement tool, we have also used the tool Pathload, that measure the available bandwidth of an end-to-end path, in our experiments.

Earlier work has touched upon the problem of active measurements of bandwidth in wireless networks. In [6] we discuss the main problem areas when deploying existing bandwidth measurement methods in ad-hoc networks. For example, we observed using ns-2 simulations, that the measured link capacity show dependence on the cross-traffic rate.

Measurement results presented in [7] indicate that the available bandwidth is dependent of the probe packet size. Our study extends that study by showing that *both* the available bandwidth and the measured link capacity depend on both the probe packet size and the cross-traffic rate. Further, we use a more complex measurement topology to verify their findings.

The rest of this paper is organized as follows. Section 8.2.1 describes the original TOPP measurement method. DietTopp, which is our implementation of a modified TOPP method, is also presented. Section 8.2.2 is a description of the testbed we have used for the investigation of the bandwidth measurement problem in wireless networks. Section 8.3 shows measurement results from using DietTopp in wired as well as in wireless networks. We discuss the results and compare them to results obtained by Pathload. In section 8.4 some important observations are made. The paper ends with conclusions in section 8.5.

8.2 Experimental setup

This section describes our experimental setup. That is, the measurement tool (DietTopp), our testbed and what kind of measurements we have performed and their relevance to ad-hoc networks.

8.2.1 DietTopp

DietTopp has its origins in the previously not implemented TOPP [4] method and uses the measured dispersion of probe packet trains to calculate bandwidth estimates.

In short summary DietTopp works as follows. Starting at some offered probe rate o_{min}, DietTopp injects m probe packet trains, where each train contains k equally sized probe packets, into the network path. When all probe trains corresponding to a probe rate o_{min} have been transmitted, DietTopp increases the offered rate o by Δo. Another set of probe packet trains are sent into the network with the new probe rate. This is repeated i times until the offered probe rate reaches some specified probe rate o_{max}.

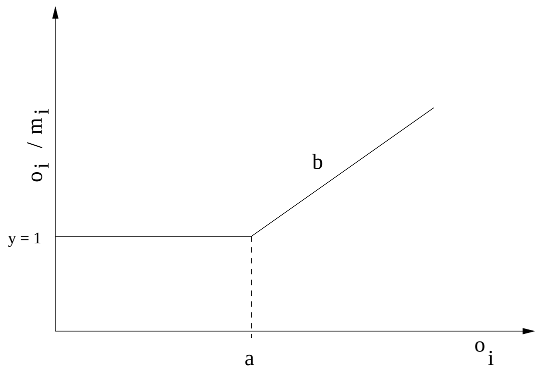

Figure 8.1: Plot of the ratio o_i/m_i as a function of o_i.

The probe packet dispersion may change as the probe packets traverse the network path between the probe sender and the probe receiver. This is due to the *bottleneck spacing effect* [8] and/or interactions with competing traffic.

The receiver time stamps each probe packet arrival. Hence, any change in probe packet separation can be measured. The time stamps are used to calculate the measured probe rate m_i.

When all measurements are collected, DietTopp computes the ratio o_i/m_i for all i. If plotting the ratio o_i/m_i on the y-axis and o_i on the x-axis for all i, we get a plot like the theoretical one in Figure 8.1. If the dispersion of the probe packets would remain unchanged after traversal of the network path, the measured rates, m_i, on the receiver side would be the same as the offered rates o_i. Expressed differently, the ratio o_i/m_i would equal 1. The link that limits the available bandwidth of the path will eventually get congested when increasing the offered probe rate. This causes the curve to rise since the rate m does not increase as much as the rate o. If the link capacity is l and the available bandwidth is a the relation between o_i and m_i is given by $o/m = (1 - a/l) + o/l$ (when one link is congested) [4].

Segment b in the figure is linear and the slope corresponds to the link capacity of the congested link. The available bandwidth of the end-to-end path is defined as the intersection of $y = 1$ and b (i.e. a in the figure) [4].

To speed up the probing phase of DietTopp we want to avoid measurements below a. That is, we want to ensure that $o_{min} > a$. This is done by estimating m_{max} which is done by injecting a set of probe packets at rate o_{max} and then measure their separation at the receiver. According to [4] m_{max} is greater than the available bandwidth (m_{max} is referred to as the asymptotic dispersion rate in [9]).

Having a value of $o_{min} > a$, the procedure described above is executed to find the link capacity and available bandwidth.

DietTopp is implemented in C++ on Unix platforms and can be downloaded from [10].

8.2.2 The testbed

The testbed used consists of 9 computers running Linux, shown in Figure 8.2. The link speed for each link is shown in the figure. The links between $Xw1$, $Xw2$ and $R1$ are 802.11b wireless links while the link between S and $R1$ can either be a 802.11b wireless link or a 100 Mbps wired link.

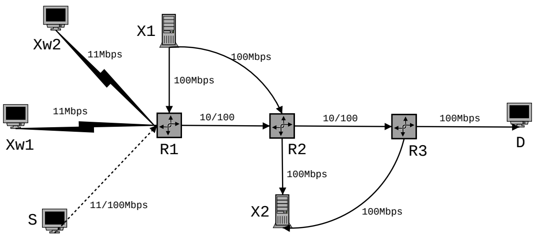

Figure 8.2: The testbed is constructed by one wireless link, three routers and several cross-traffic generators (on both the wireless and the wired links)

The cross traffic, generated by a modified version of *tg* [11], can either take the route $X1 \rightarrow R1 \rightarrow R2 \rightarrow X2$ or the route $X1 \rightarrow R2 \rightarrow R3 \rightarrow X2$. Cross traffic can also be generated by *Xw1* and *Xw2* on the wireless hop. The cross traffic is either constant bit rate (CBR), exponential or pareto distributed (shape = 1.5). Further, the cross traffic consists of 60 (46%), 148 (11%), 500 (11%) and 1500 (32%) byte packets. This distribution of packet sizes originates from findings in [12].

8.2.3 Experiments

In this paper we want to identify possible problems associated with bandwidth measurements in wireless networks, such as ad-hoc networks. First we show two measurements using DietTopp in a wired scenario. This is to validate that our tool is sound in the simple wired case before turning attention to the more complex case of estimating end-to-end bandwidth in wireless networks. We compare DietTopp results to theoretical values as well as to values obtained from Pathload.

The measurements in the wireless scenario is done using DietTopp. We elaborate on the impact of probe packet size, the cross-traffic distribution, the number of probe packets sent and on the number of cross-traffic generators in the wireless network. We compare our results to results obtained from Pathload.

This work is related to the work presented in [7]. We extend and complement that work in the following way: We use our newly developed tool DietTopp, that measured both the link capacity and the available bandwidth of the bottleneck link. Previous work has only focused on the available bandwidth on wireless links. Further, we use a more complex testbed topology.

8.3 Experimental results

This section presents the results obtained using DietTopp in wired and wireless scenarios. We have used Pathload [2] to compare and discuss the obtained measurement results. In the diagrams all measurement results are shown with a 95% confidence interval.

8.3.1 Measurement results in wired networks

This section presents measurements done with both DietTopp and Pathload in an all wired scenario. This section is to show by example that our tool, DietTopp, measures both the link capacity and the available bandwidth in a sound way.

The diagram in Figure 8.3 illustrates results from DietTopp measurements using four different cross traffic intensities on link R1 - R2 (10 Mbps link capacity in this case), shown on the x-axis. The cross traffic at link R2 - R3 (100 Mbps link capacity) is a 8.76 Mbps stream. Both cross-traffic streams are exponentially distributed. The y-axis shows the measured link capacity (thin solid line), the measured available bandwidth (thin dashed line), the theoretical link capacity (thick solid line) and the theoretical available bandwidth (thick dashed line). As can be seen the correlation between measurement results and the theoretical values is good.

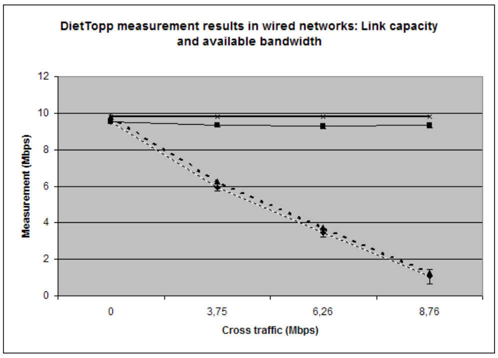

Figure 8.3: Link capacity (solid lines) and available bandwidth (dashed lines). Thick lines corresponds to theoretical values while thin lines are values obtained from DietTopp.

The diagram in Figure 8.4 is a comparison of the measured available bandwidth using DietTopp (dashed line) and Pathload (solid line). The same testbed and cross traffic setup is used as in Figure 8.3. We see that both tools report similar estimates of the available bandwidth.

We have now given an indication that DietTopp estimates both the link capacity as well as the available bandwidth in wired network with good accuracy, both compared to theoretical values and compared to one state-of-the-art bandwidth measurement tool, Pathload. In the next subsection we investigate the impact of wireless bottlenecks on the measurement results.

8.3.2 Measurement results in wireless networks

This subsection presents our results from measurements using DietTopp where the bottleneck is a wireless link (the link between S and R1 in the testbed as described in subsection 8.2.2) which is the case in ad-hoc wireless networks. Cross traffic is present on both of the wired links R1 - R2 and R2 - R3, but the rate is limited to approximately 9% of the corresponding link capacity (100 Mbps in this case). That is, the wireless link is the link that limits both the link capacity and the available bandwidth. The cross traffic at the 100 Mbps links

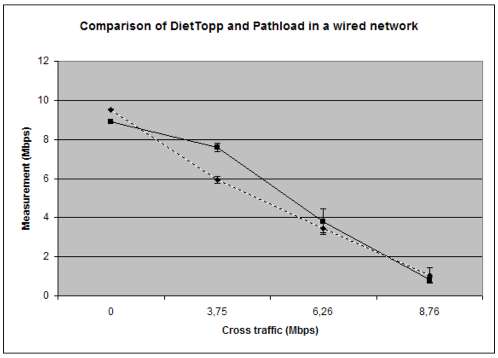

Figure 8.4: Available bandwidth measured by DietTopp (dashed line) and Pathload (solid line).

between R1, R2 and R3 is pareto distributed and consists of 4 different packet sizes. The cross-traffic configuration on the wired links is the same for each experiment presented in this section.

The probe packet size affects both the measured link capacity and the available bandwidth estimate when the bottleneck in an end-to-end path is a wireless link. We illustrate and describe this phenomenon in a set of diagrams below.

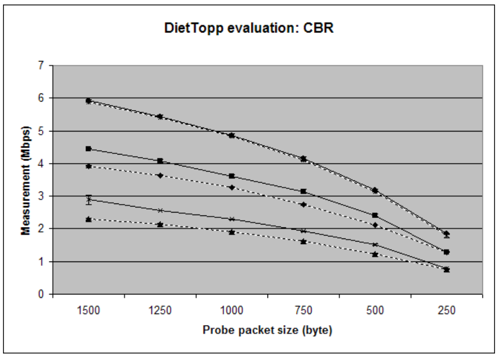

Figure 8.5: Available bandwidth (dashed lines) and measured link capacity (solid lines) measured under 0, 250 Kbps and 500 Kbps cross-traffic rates).

The two upper curves in Figure 8.5 show the measured link capacity (solid line) and the measured available bandwidth (dashed line) when no cross traffic

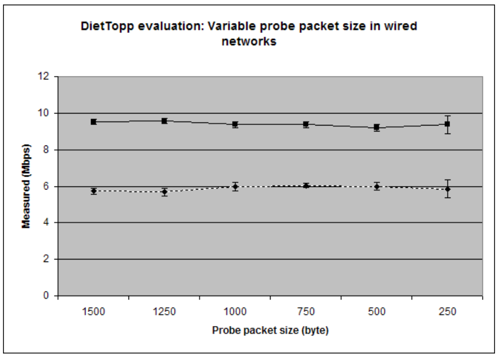

Figure 8.6: Available bandwidth (dashed line) and link capacity (solid line) measured by DietTopp in a wired network using different probe packet sizes. The cross traffic is a 3.26 Mbps pareto distributed stream on a 10 Mbps link.

is present on the wireless link. Varying the probe packet size from 1500 bytes down to 250 bytes gives decreasing values of both the measured link capacity and the measured available bandwidth. It should be observed that the total number of bits remains constant independent of the probe packet size. The total amount of probe data sent by DietTopp in these measurements is 1.2 Mbit. Each probe train consists of 16 probe packets and we send 5 probe trains on each probe rate level. The number of probe rate levels depends on the probe packet size; decreasing the probe packet size increases the number of probe rate levels.

The two middle curves show measurement values when there is a 250 Kbps CBR cross-traffic stream on the wireless link. The two bottom curves correspond to the case when a 500 Kbps CBR stream is present. The same decreasing trend for the measured link capacity and the measured available bandwidth is visible. An interesting phenomenon is that the difference between the measured link capacity and the measured available bandwidth tends to be smaller for small probe packet sizes. Why this is the case is a subject of further research.

For comparison we have varied the probe packet size in an all wired network. The measurement results can be seen in Figure 8.6. Both the measured link capacity and the available bandwidth are quite stabile, that is independent of the probe packet size.

We have also done measurements using Pathload, a tool that estimates the available bandwidth using 300 byte packets. The results obtained from using Pathload in our testbed with different cross-traffic distributions and intensities can be seen in Table 8.1. When comparing results obtained by Pathload (in Figure 8.5) to those of DietTopp we can see that Pathload reports available bandwidth measurement estimations that are in line with estimations made by DietTopp (using interpolation between packet sizes 250 and 500 bytes).

Cross traffic	Measurement (Mbps)
0	2.32 - 2.39
250k cbr	1.67 - 1.67
250k exp	1.73 - 1.73
250k par	1.40 - 1.63
500k cbr	0.96 - 0.99
500k exp	0.87 - 0.95
500k par	1.27 - 1.29

Table 8.1: Measurement results obtained from Pathload under the influence of different cross-traffic distributions.

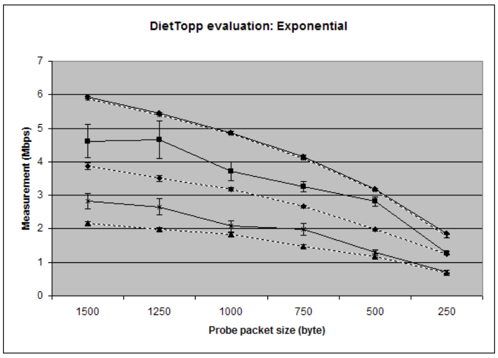

Figure 8.7: Available bandwidth (dashed lines) and measured link capacity (solid lines) measured under 0, 250 Kbps and 500 Kbps exponentially distributed coss-traffic.

Figures 8.7 and 8.8 report results from the same type of measurements as in Figure 8.5. However, in these two scenarios we have used more complex cross-traffic distributions. In Figure 8.7 we have used exponentially distributed

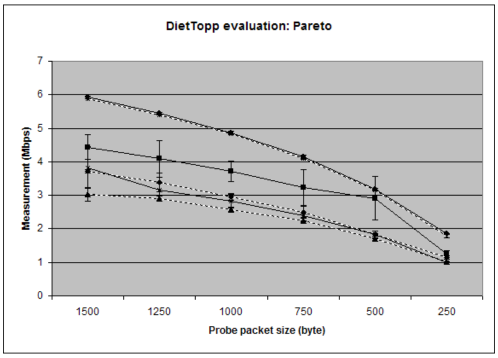

Figure 8.8: Available bandwidth (dashed lines) and measured link capacity (solid lines) measured under 0, 250 Kbps and 500 Kbps pareto distributed cross-traffic.

arrival times for the cross-traffic packets while in Figure 8.8 we have used pareto distributed arrival times. As can be seen in both figures the confidence intervals are larger when the cross traffic is burstier. It is also obvious that the curves are less smooth compared to the CBR case in Figure 8.5. In the pareto case (Figure 8.8) it is hard to distinguish between the 250 Kbps and 500 Kbps measurements of link capacity and available bandwidth. However, we can still see that the measured link capacity and available bandwidth is dependent on both the probe packet size and the cross-traffic rate. Again, comparing the measurement results (at the 300 byte probe packet size level) with results obtained by Pathload (in Table 8.1) we can conclude that the available bandwidth estimate characteristics are compatible.

In Figure 8.9 we vary the probe packet size in the same manner as above. However, instead of keeping the total number of bits transfered constant we keep the number of probe packets sent constant. The cross traffic is pareto distributed. We see that even though the total amount of probe data sent is less at each probe packet size level the confidence intervals remain low.

In Figure 8.10 two cross-traffic generators are generating 250 Kbps of CBR cross traffic each. Comparing Figure 8.10 to the measurement results in Figure 8.5 we see that the confidence intervals are larger when having multiple cross-traffic generators.

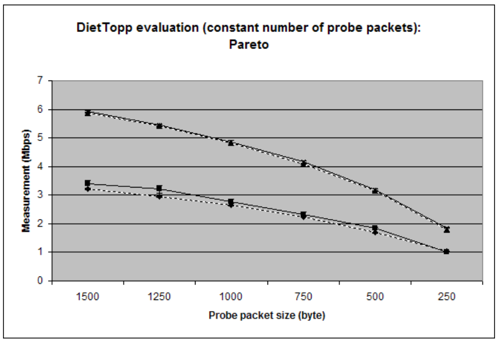

Figure 8.9: Available bandwidth (dashed lines) and measured link capacity (solid lines) measured under 0 and 500 Kbps pareto distributed cross-traffic. The number of probe packets is constant.

8.3.3 Wireless measurement results discussed

In this subsection we will discuss the results obtained in the previous subsection and the reasons for the difference between DietTopp measurements in wired and in wireless networks.

We will derive the differences from Figure 8.11 which illustrates the procedure for sending a packet in a 802.11 wireless network. First, the radio transmitter at the wireless node needs a clear channel to send its packet on. This is illustrated by step 1 and 2 in the figure. If someone else is using the channel the sender does a back-off. It tries again after some time. Eventually the packet is sent, step 3 in the figure. When the receiving node gets the whole packet it responds with a link-layer acknowledgment to the sender (step 4). The sender can now transmit the next packet.

The reason for the decreasing measurement of the measured available bandwidth can be traced to the link-level acknowledgments in step 3 and 4 in the figure. That is, if the probe packet is small, the overhead induced by the link-level acknowledgment is larger than if the probe packet were large. We come to the conclusion that large probe packets will measure a larger available bandwidth than small probe packets.

The contention phase (step 1 and 2 in the figure) is independent of the packet size. The contention phase is instead dependent on the number of sending nodes in the wireless networks. Increasing the number of stations that want to send traffic over the wireless network increases the waiting time for each

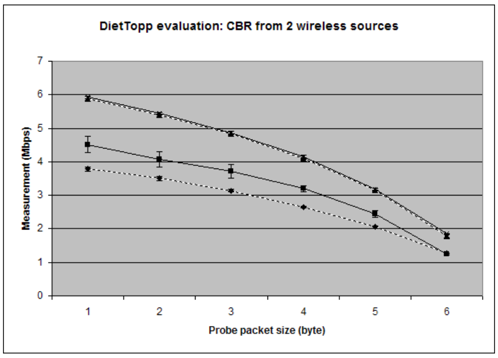

Figure 8.10: Available bandwidth (dashed lines) and measured link capacity (solid lines) measured under 0 and 500 Kbps CBR cross-traffic. The cross traffic is generated by two different sources (250 Kbps each).

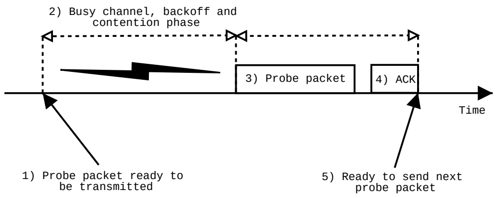

Figure 8.11: A schematic picture of the procedure for sending a packet in a 802.11 wireless network.

node. It also increases the variance of the waiting time.

In Figure 8.10 two cross-traffic generators are generating 250 Kbps of CBR cross traffic each as described above. Since we have two wireless nodes sending traffic, this is likely to affect the contention phase in Figure 8.11 in such a way that we get larger confidence intervals in our measurement results. Comparing Figure 8.10 to the measurement results in Figure 8.5 we indeed see that the confidence intervals are larger when having multiple cross-traffic generators.

The results concerning the available bandwidth are in line with results discussed in [7]. We validate and extend those findings by using more complex testbed scenarios and our own tool DietTopp.

A theoretical description of why the measured link capacity is dependent on both the probe packet size and the cross-traffic intensity is a subject of future research.

A final remark is that in most figures we can see that the confidence intervals decrease with the probe packet size. Hence, we can draw the conclusion that we get values with low standard deviation with small probe packets. However, why this is the case is also a subject of future research.

8.4 Other observations

Due to the fact that the probe packet size affects both the measured link capacity and the measured available bandwidth when using DietTopp, a possible method to identify a wireless bottleneck link in a network path could be: if the available bandwidth (and the measured link capacity) changes when probing the path with different packet sizes, this can be taken as an indication that the path includes a wireless bottleneck. This is important since, as we have discussed, wireless bottlenecks have different characteristics than wired bottlenecks. This is also interesting from an semi-ad-hoc perspective: when one node of an ad-hoc network is connected to an infrastructure, such as the Internet, it is important to determine whether the bottleneck is within the ad-hoc network or within the infrastructure. Is the bottleneck within the ad-hoc network there might be possibilities to route the data differently. Also, ad-hoc router protocols can perform better with an understanding of bottlenecks within the ad-hoc network. However, this subject is left to future research.

An important consequence of the measurements we have presented in this paper is that the available bandwidth will be application dependent in ad-hoc networks and when wireless links are a bottleneck in general. For example, a voice over IP application or a distributed game probably use small packets to send data while a file transfer application may use larger packets. The available bandwidth for the applications will not be the same due to their packet size distribution, as indicated by the figures above that show decreasing measurement values when decreasing the probe packet size. This means that when probing a path containing a wireless bottleneck link the estimation tool must use a probe packet size distribution that corresponds to the specific application.

8.5 Conclusion

In this paper we have shown measurements that illustrate the difference between bandwidth measurements in wired and wireless networks, such as ad-hoc networks. We have discussed some of the underlying reasons for these differences while other reasons are left to further research. We have used our own tool, DietTopp, to produce measurement results throughout the paper. For comparison and validity we have used Pathload. The measurements have been performed in a testbed where we have used different kinds of cross traffic, from simple CBR to bursty pareto distributed cross traffic.

Our conclusions are that measurements in wireless networks are associated with difficulties that can result in misleading bandwidth estimations. We have shown that the packet size is critical to the bandwidth measurement value of both the link capacity and the available bandwidth. Further, we have shown that the measured link capacity on wireless links does not only depend on the packet size, but also on the cross traffic intensity. We have also addressed the problem of application dependent probing.

Future research is to investigate why small packets gives a lower variance when used for active probing in wireless networks. We will also investigate why the measured link capacity vary when the probe packet size vary. It is also important to study what the variable measured link capacity obtained means for wireless network applications.

Bibliography

[1] Vinay Ribeiro, Rudolf Riedi, Richard Baraniuk, Jiri Navratil, and Les Cottrell. pathchirp: Efficient available bandwidth estimation for network paths. In *Passive and Active Measurement Workshop*, 2003.

[2] Manish Jain and Constantinos Dovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. In *Proceedings of ACM SIGCOMM*, Pittsburg, PA, USA, August 2002.

[3] Strauss, Katabi, and Kaashoek. A measurement study of available bandwidth estimation tools. In *ACM SIGCOMM Internet Measurement Workshop*, 2003.

[4] Bob Melander, Mats Björkman, and Per Gunningberg. Regression-based available bandwidth measurements. In *Proceedings of the 2002 International Symposium on Performance Evaluation of Computer and Telecommunications Systems*, San Diego, CA, USA, July 2002.

[5] R.S. Prasad, M. Murray, C. Dovrolis, and K. Claffy. Bandwidth estimation: metrics, measurement techniques, and tools. IEEE Network Magazine, 2003.

[6] Andreas Johnsson, Mats Bjrkman, and Bob Melander. A study of dispersion-based measurement methods in ieee 802.11 ad-hoc networks. In *Proceedings of the International Conference on Communication in Computing*, Las Vegas, 2004.

[7] Karthik Lakshminarayanan, Venkata N. Padmanabhan, and Jitendra Padhye. Bandwidth estimation in broadband access networks. In *In Proceedings to the Internet Measurement Conference*, 2004.

[8] Van Jacobson. Congestion avoidance and control. In *Proceedings of ACM SIGCOMM*, pages 314–329, Stanford, CA, USA, August 1988.

[9] Constantinos Dovrolis, Parameswaran Ramanathan, and David Moore. What do packet dispersion techniques measure? In *Proceedings of IEEE INFOCOM*, pages 905–914, Anchorage, AK, USA, April 2001.

[10] Andreas Johnsson. Diettopp implementation, beta version. http://www.idt.mdh.se/ãjn12/.

[11] Paul E. McKenney, Dan Y. Lee, and Barbara A. Denny. Traffic generator software release notes. SRI International and USC/ISI Postel Center for Experimental Networking, January 2002.

[12] Yongmin Choi, Heung-No Lee, and Anurag Garg. Measurement and analysis of wide area network (wan) traffic. In *SCS Symposium on Performance Evaluation of Computer and Telecommunication Systems*, July 2000.