Abstract

Neginhal, Mradula S. Efficient Estimation of Available Bandwidth Along Network Paths. (Under the direction of Assistant Professor Khaled Harfoush and Professor Harry Perros).

Measuring the available bandwidth of a network path is important to support congestion control protocols, Internet traffic characterization, overlay network routing, capacity provisioning, and traffic engineering, among other applications. Previous techniques to estimate the available bandwidth are mainly based on (1) simplifying assumptions and theoretical queueing models, (2) estimating the exact queueing delays incurred by probe packets, and/or on (3) saturating the network path with probe packets.

In this thesis we introduce a novel technique to estimate the available bandwidth of a network path and the speed of the most utilized link – the tight link – along this path. Our technique assumes no a priori knowledge about the measured path, does not stress the network with a significant load, and does not require complicated measurements of accurate packet dispersions or queueing delays. It merely relies on a simple binary test to estimate whether each probe packet has queued in the network or not. Theoretical analysis and experimental results validate the ability of the proposed technique to provide accurate bandwidth estimates.

Efficient Estimation of Available Bandwidth Along Network Paths

by

Mradula S Neginhal

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Computer Science

Raleigh

2006

Approved By:

Dr. Michael Devetsikiotis

Dr. Khaled Harfoush
Chair of Advisory Committee

Dr. Harry Perros
Co-Chair of Advisory Committee

To Appa, Amma and Ashok.

Biography

Mradula Neginhal was born on the 5th of December, 1980 in Dharwad, India. She was awarded the Bachelor's degree in Engineering (Computer Science) in 2002 by Visweswaraiah Technological University, India. She is currently a Master's student in the Department of Computer Science at North Carolina State University, Raleigh. She has worked for LVL7 Systems Inc., Morrisville, NC as part of her Master's degree for a period of 9 months.

Acknowledgment

First and foremost, I thank my parents, Subhash and Rohini Neginhal for their unconditional love and support throughout the years. Words cannot begin to describe the ways in which they have encouraged and inspired me. I would never have been able to accomplish this work had it not been for their unwavering faith in my abilities.

I thank my advisor, Dr. Khaled Harfoush for guiding me in this work. This thesis would not have been possible without his help. I thank my co-advisor, Dr. Harry Perros for taking the time to guide me in this work and for teaching me queueing theory - CSC579 was the most enjoyable course of my Master's degree. I thank my committee member, Dr. Michael Devetsikiotis for his valuable suggestions.

I thank John Bass and Jamie Zabala for taking the time and effort to help me out with my work.

I thank my sister, Smita and brother-in-law, Rajesh for always being there for me and for egging me on during stressful times. I thank my uncle, Arvind and aunt, Raksha for giving me a home away from home.

I thank Amit and Vishwas for all the stress-busting conversations. They never failed to entertain. I thank Savera for our coffee sessions and for giving me a place to stay when I had none.

Lastly, but most importantly, I thank my husband Ashok for being my pillar of strength. He has been my greatest source of encouragement throughout this degree. He has had more sleepless nights than me in helping me with this work and for that, I shall always be grateful.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The ubiquity of computer networks and our dependence on them in today's society calls for a deeper understanding of their operation, since efficient utilization of computer networks enables us to leverage their advantages while deriving economic value from them. In other words, a network whose characteristics and behavior are well understood is a network that can be managed to run efficiently, thus increasing its value. To this end, we study in this thesis the measurement of a key characteristic of networks, their *bandwidth*. In data networks, the term *bandwidth* quantifies the data rate that a network link or network path can transfer.

To lay the groundwork for our development of a bandwidth-measurement technique, the present chapter introduces the concepts and theory underlying the rich field of network bandwidth estimation. Section 1.1 introduces the background concepts related to measurement of network bandwidth, Section 1.3 discusses some bandwidth estimation techniques that use concepts similar to this work, Section 1.4 gives the motivation for this line of research, and Section 1.5 summarizes our contributions to the field.

1.1 Background

The bandwidth of a network path is an important network performance metric and has received considerable attention. A network path is a sequence of links in a network. *Bandwidth* corresponds to how fast a network path can transport data from one end to another. The bandwidth of all the successive links that comprise a network path measured from the source of traffic on that path to the destination is called *end-to-end* bandwidth

of that network path, in contrast to the bandwidth of a particular segment of the network path such as a *link* or a *hop*. There are two measures of end-to-end network bandwidth, *Capacity Bandwidth* and *Available Bandwidth*.

Capacity Bandwidth Consider a path P comprised of H store and forward links. Let C_i be the capacity of the i^{th} link, where $1 \leq i \leq H$. The capacity C of the path P is the speed of the slowest link i.e.,

$$C = \min_{i=1,...H} C_i \tag{1.1}$$

The link in the network path with the minimum capacity is called the *narrow link* of the path.

Available Bandwidth At any instant of time, a link is either transmitting a packet at full capacity or is idle. Hence, the instantaneous utilization of a link is either 1 (when it is transmitting) or 0 (when it is idle). The definition of available bandwidth of a link therefore requires time-averaging the instantaneous utilization of that link over the time interval of interest. The average utilization $u(t-\tau, t)$ for a time period $(t-\tau, t)$ is given by:

$$u(t-\tau, t) = \frac{1}{\tau} \int_{t-\tau}^{t} u(x)dx \tag{1.2}$$

where $u(x)$ is the instantaneous available bandwidth of the link at time x. We refer to the time length τ as the *averaging timescale* of the available bandwidth. If C_i is the capacity of a link i of the network path, and u_i is the average utilization of that link in a given time interval, then the average available bandwidth A_i of link i in that time interval is given by the fraction of the capacity bandwidth which is not utilized.

$$A_i = (1 - u_i)C_i \tag{1.3}$$

Extending the above definition to a path composed of H links, the end-to-end available bandwidth A is the maximum transmission rate of P, in the presence of competing cross-traffic i.e.,

$$A = \min_{i=1,...H} A_i \tag{1.4}$$

The link with the minimum available bandwidth is called the *tight link* of the end-to-end path.

1.2 Applications of Bandwidth Estimation

Measurement of network bandwidth is essential for many Internet applications involving high volume data transfer and applications that require QoS such as streaming media, since the bandwidth available to these applications directly affects their performance.

Overlay networks are a prime example of computer networks that can benefit from a knowledge of network path bandwidth. Overlay networks can bypass broken network paths by re-routing traffic through intermediate nodes known as *overlay nodes*. Bandwidth measurements provide information about how much additional traffic rate a network path can carry before it gets saturated. Overlay networks can make use of this information to select efficient paths between adjacent overlay nodes [4, 14] and to determine how much maximum throughput a flow can have on a particular network path [31]. In addition to these applications, bandwidth estimation is beneficial to Internet Service Providers (ISPs) for capacity provisioning, and traffic engineering. The end-users can make use of bandwidth estimates to ensure QoS and bandwidth guarantees. Bandwidth estimation techniques can be integrated with rate-adaptive congestion control mechanisms.

1.3 Bandwidth Estimation Techniques

Several techniques have been employed to measure the end-to-end capacity and available bandwidth of a network path.

Pathchar [11] is a tool that can measure the hop-by-hop bandwidth of a network path by sending ICMP packets limited by the Time-to-Live (TTL) field. This tool statistically infers characteristics of the network path such as available bandwidth, delay, and loss by measuring the round-trip time (RTT) of the ICMP probe packets for different probe packet sizes. The key insight used in pathchar is that as the size of a probe packet increases, the one-way delay of the packet increases since the one-way delay is equal to the ratio of the probe packet size and the link capacity. The drawback of the tool is that it introduces significant amount of traffic on the network path while probing [28]. Also, the accuracy of Pathchar is acceptable for links with a low bandwidth (less than 10Mbps), but it is significantly inaccurate for high bandwidth links as shown in [8].

Clink [7] and *pchar* [19] are extensions of pathchar. *clink* differs from pathchar in that it obtains capacity estimates by dividing a large sample into smaller even and odd

samples. However, it does not significantly improve upon pathchar in terms of accuracy. Pchar essentially uses the same algorithm as pathchar. Only, it uses *Libpcap* to obtain kernel-level timestamps. Pchar suffers from poor accuracy on long network path and high-bandwidth links [18]. Pathchar, clink and pchar all measure the hop-by-hop bandwidth and from these estimates, they get the bottleneck bandwidth of the network. This can be time consuming if all that is needed is the bottleneck bandwidth of the network.

1.4 Motivation

Current bandwidth estimation techniques suffer from a number of drawbacks as listed below.

- Techniques such as Spruce [30] require knowledge of the capacity of the targeted path. This is not known in advance and is difficult to estimate using existing tools.
- Some tools estimate available bandwidth by saturating the network path with probe traffic. For instance, tools such as Iperf and BBFTP [2, 1] use bulk TCP connections, thus filling the path with traffic and affecting its cross-traffic. These techniques are akin to a blind search for available bandwidth and can exacerbate congestion on an already congested network. This in general makes these tools unable to measure the available bandwidth of links behind the tight link.
- Tools/techniques such as pathchirp [25] and packet-pair [29] require accurate measurement of packet dispersions, which is a difficult metric to measure because of variations in cross-traffic on the network path.
- Tools such as Multi-Router Traffic Grapher (MRTG) [23] need access to routers since they are based on the router management software and can only be used by network administrators.

Our goal is to develop a technique that overcomes the above limitations.

1.5 Contribution

The technique we present in this thesis yields the end-to-end available bandwidth of a network path and capacity of the tight link. Our chief contribution is the development

of a novel technique for the estimation of end-to-end bandwidth of a network path. The technique we propose has the following advantages over existing methodologies and tools for bandwidth estimation. In this thesis, we develop a technique for bandwidth estimation that

1. Does not need *a priori* knowledge of the network path capacity
2. Does not saturate the network with probe traffic
3. Does not need accurate measurements of packet dispersion or queueing delay
4. Does not require access to nodes other than the end nodes of the network path

The key difference between delay-based tools and our technique is that whereas they need accurate estimates of delay to measure the bandwidth, our technique merely uses the delays as an indicator of whether the probe packets queued or did not queue which makes it less susceptible to errors due to variations in cross-traffic. The central concept of our technique is a simple test to find if a probe packet is queued in a targeted queue i.e., the probe sequence. This binary test yields the available bandwidth of the network path and the capacity bandwidth of the tight link.

We demonstrate using simulations the high accuracy of our technique in measuring the available bandwidth of the network path.

Our goal is not to obtain an accurate and complex estimate of the capacity of the tight link, since accuracy targets in bandwidth estimation experiments are, in our view, unrealistic, considering the large size of timescales over which probing is required for accurate capacity bandwidth estimation. Instead, we propose a simple but efficient test to classify the estimated capacity bandwidth. Comparing the estimated capacity of the tight link to one of the standard Internet links provides a more realistic approach.

We show the effectiveness of our methodology via extensive simulations performed using the *network simulator* $ns-2$ [3]. We also chart out future work for this line of research involving multi-link experiments in a controlled lab setting and Internet experiments.

1.6 Organization

The rest of this thesis is organized as follows: Chapter 2 surveys the state-of-the-research in the area of bandwidth estimation, comparing important research efforts with

ours to place our work in perspective; Chapter 3 describes in detail the theory behind our estimation techniques. Chapter 4 describes our experimental setup and the reasons for choices of various network parameters chosen to conduct our experiments. Chapter 5 lists our findings and explains how they support our theory of bandwidth estimation. Chapter 6 concludes the thesis with a summary of our findings and lists the pitfalls of our approach as well as directions for future research.

Chapter 2

Related Work

To place our proposal in perspective, we sketch a broad outline of the field of bandwidth estimation by listing key features of other research efforts in this area. We first discuss tools and techniques that measure the capacity bandwidth of the network followed by those that measure the available bandwidth of the network.

2.1 Capacity Bandwidth

Capacity bandwidth estimation techniques based on packet-pair dispersion measurements [29, 5, 26, 24, 18] suffer from inaccuracy in the presence of heavy cross-traffic due to the cross-traffic interfering with the packet-pairs/trains. These techniques are discussed below.

Packet-pair technique The *packet-pair technique* [29] is one of the earliest and most widely-known mechanisms for estimating capacity bandwidth of a network path. When a packet is transmitted over a link, it encounters some *transmission delay*. If the capacity of the link is C_i and the packet size is S, then the transmission delay T is given by $T = S/C_i$. In the packet-pair technique, two packets both of size S are sent back-to-back from the source to the destination. The *dispersion* δ between the packets which is equal to S/C_i is measured at the destination. From this, the capacity is estimated as $C_i = S/\delta$. This technique requires accurate measurement of the dispersion which can be unreliable due to the dynamic nature of the cross-traffic on the network path.

Bprobe In [26], a tool – *Bprobe* – is implemented to estimate the end-to-end capacity of a network path. Bprobe obtains capacity estimates using packet-pair dispersion and filters the estimates by computing an error interval around each estimate, and subjecting the error intervals to "union and intersection" filtering.

Packet Bunch Method Paxson in [24] proposes a variation to the packet-pair model to estimate the capacity of a network called the Packet Bunch Method (PBM). This method estimates the end-to-end capacity based on the intensity of the local modes in the packet-pair bandwidth distribution.

Nettimer Nettimer [18] is a packet-pair capacity estimation tool that uses a technique called *kernel density estimation* to process packet-pair measurements. This kind of estimation tries to offset the limitations of histogram-based techniques which identify the capacity mode by not assuming a point of origin for the packet-pair bandwidth distribution.

Pathrate In [5] Dovrolis *et al.* show that the packet-pair bandwidth distribution is multimodal with multiple local modes. A tool called Pathrate is implemented which measures the end-to-end capacity of a network path. Pathrate sends variable size packet pairs to uncover the local modes in the bandwidth distribution. It then sends packet trains composed of more than 2 packets and measures the Average Dispersion Rate (ADR) which is the mean dispersion of the packet trains. The ADR is a lower bound of the capacity and is used to find the capacity mode among the local modes in the packet pair bandwidth distribution.

CapProbe Kapoor *et al.* have developed CapProbe [15], a technique that uses both delay and dispersion measurements of packet pairs to measure the capacity of the link. CapProbe estimates the delay sum which is the sum of the delays of the packets in the packet pair. It then uses the dispersion of the sample with the minimum delay to estimate the capacity of the narrow link. CapProbe is based on the idea that the minimum delay sum is the delay sum that is not distorted by cross traffic. Once a packet pair goes through the network without experiencing any delay, CapProbe estimates the bandwidth. Convergence tests and speed up techniques are introduced by varying probing parameters.

2.2 Available Bandwidth

Cprobe Cprobe was one of the earliest tools implemented to measure the end-to-end available bandwidth of a network path. Cprobe measures the dispersion of a train of packets and estimates the available bandwidth based on this measurement. However, cprobe suffers from inaccuracies because [6] shows that measuring the dispersion of a packet train gives the Average Dispersion Rate which is related to the utilization of all the links on the network path while the desired available bandwidth is related only to the tight link of the network path.

IGI and PTR Hu *et al.* [10] have developed two measurement methodologies to estimate the available bandwidth - Initial Gap Increasing (IGI) and Packet Transmission Rate (PTR). The IGI and the PTR algorithms send several packet trains with increasing gap sequences. They obtain the difference between the gap at the source and at the destination. The capacity of the path is then estimated using a tool such as pathrate [5]. The available bandwidth is obtained by subtracting the estimated competing traffic throughput from the capacity.

Many available bandwidth estimation tools and techniques [25, 12, 21, 22] are intrusive in that they contribute to a significant amount of traffic on the network path. In most cases, the estimation tools probe at a rate that is equal to higher than the available bandwidth of the network path that is being probed. These tools are discussed below.

SLoPS and Pathload Jain *et al.* [12] develop a methodology called Self-Loading Periodic Streams (SLoPS), based on the idea that one-way delays of a stream of probes increases if the probing rate is higher than the available bandwidth. SLoPS has been implemented using a tool called Pathload. Pathload sends several fleets of streams of packets, so that the one-way delay trends can be observed successive times. Each stream is sent only after the previous stream has been acknowledged. If a large fraction of each stream shows an increasing trend, it is concluded that the rate of the fleet is higher than the available bandwidth and the same if the fleets show a decreasing trend. The rate of the next stream is adjusted based on this. The tool reports the range of the minimum and the maximum rate of the stream. In other words, it reports the upper and lower bounds of the available bandwidth.

pathChirp Ribeiro *et al.* [25] introduce an active probing technique called pathChirp. pathChirp uses Trains of Packet Pairs (TOPPs) to estimate the end-to-end available bandwidth. Like SLoPS, pathChirp also uses the idea that the spacing between the packets increases if the rate of the probing stream is higher than the available bandwidth. However, instead of using Constant Bit Rate streams as in SLoPS, pathChirp uses varying probing rates within a single train. Also, the spacing between the packets is exponentially decreasing within a train. This technique is hence more efficient than using CBR streams.

Some tools [30] require prior knowledge of the capacity bandwidth to estimate the available bandwidth of the link.

Spruce Strauss *et al.* [30] have implemented a tool called *Spruce* to measure the available bandwidth. Spruce assumes that the capacity of the bottleneck link is known and sends packet pairs in order to measure the available bandwidth. It sets the intra-pair time gap to the transmission rate of a 1500 byte data packet on the bottleneck link. This ensures that the bottleneck link does not empty between the two packets in a packet pair. It uses Poisson arrivals to make the probing non-intrusive and to see the average cross-traffic rate because of the PASTA property [27]. Spruce has been evaluated against IGI and Pathload using Multi-Router Traffic Graph (MRTG) [23], which collects measurements from the Management Information Base (MIB) of a router every 5 minutes. Spruce was found to be more accurate than both IGI and Pathload.

Pathneck Hu *et al.* [9] determine the location of the bottleneck link using a tool called Pathneck which uses a technique called the Recursive Packet Train (RPT). RPT sends a stream of standard traceroute packets, called *measurement packets*, and packets that represent the traffic load, called *load packets*, which are 500 bytes each. The load packets are sent in between the measurement packets. At every hop, the first and the last packets are dropped as their TTL values expire and the router sends two ICMP packets back to the source. The packet train length is estimated by measuring the time gap between the two ICMP packets that are received at the source. The gap sequences thus obtained using the RPTs are used to detect the location of the bottleneck link in three steps. The first step is to label the gap sequences. Pathneck labels the routers where the gap sequences increase significantly as candidate choke points. The next step is to identify the actual choke as those routers that are labeled as candidate choke points with high confidence in at least

half of the probings in the set. The final step is to rank the choke points based on the average gap values in the probing set. The choke point with the largest gap value is the bottleneck of the path. Pathneck cannot cover the last hop of a network path. If the last hop is the bottleneck, Pathneck fails to detect it.

Pathvar Dovrolis *et al.* [13] argue that measuring the variation of the available bandwidth is more helpful than its time average, since the available bandwidth can vary widely around the time average when the timescales are small, thus making it a poor estimator of the available bandwidth. Two percentiles of the available bandwidth distribution are estimated, typically 10% and 90%, and these give the range of variation of the available bandwidth. Dovrolis and *et al.*send several probing streams, each of which can estimate if the available bandwidth is less than the probing rate. The binary samples (indicating if the available bandwidth is less than the probing rate) obtained from the probing streams are used to estimate the percentiles of the distribution. Dovrolis *et al.*have introduces two techniques to estimate the percentile . The first one is an iterative technique useful in measuring the percentiles for non-Gaussian distributions and timescales less than 100ms, and the second technique is a parametric technique used to estimate the percentiles when the available bandwidth distribution is Gaussian and timescales are between 100ms and 200ms.

Dovrolis and colleagues have implemented a tool called *Pathvar* to validate the two percentile sampling techniques. Their paper also identifies four factors that affect the variation of the available bandwidth - traffic load, number of competing flows, and rate of competing flows at the tight link and the measurement timescale. As the utilization of the tight link increases, so does the variation, up to a certain point after which the variation decreases because the link approaches saturation. Increasing the capacity proportional to the number of competing flows (capacity scaling) results in the variation of the available bandwidth decreasing but the width of the variation range itself increases. Also, increasing the number of flows while decreasing the rate of the flows (flow scaling) decreases width of the variation range. As the measurement timescale increases the variation decreases.

PathMon PathMon [17] is a delay based tool to measure the available bandwidth. PathMon sends a stream of packets over a network and measures the delay at the receiving end to see if the packets experienced congestion. PathMon first measures the jitter by sending equal sized, equal spaced packets over the network. The inter-arrival time gaps between the

packets are used to calculate the mean and standard deviation of the jitter. Next PathMon sends a stream of same size packets with decreasing time intervals, which means increasing instantaneous bandwidth requirements. PathMon detects the point of divergence of the inter-packet spacing at the source and at the receiver to determine the range of the available bandwidth.

Chapter 3

Estimation Methodology

The present chapter describes the methodology that we propose for calculating the end-to-end available bandwidth of a network path and the capacity of the tight link. The key element of our methodology is the detection of queuing of probe packets at intermediate nodes along the network path on which they are sent. The basic procedure is as follows: A sequence of ICMP Echo Request packets are sent as probe packets from a source at one end of the network path under consideration to a destination at the other. The round-trip time (RTT) of the probe packets is measured. Assuming that probe packets with the minimum RTT are packets that did not get queued, we detect probe packets that did get queued by comparing their RTTs with the minimum RTT. We send two sequences of probe packets, one sequence at a higher rate than the other. A greater number of packets will experience queuing delays in the sequence that was sent at a higher rate. We use this property of probe sequence rates to estimate the utilization of the tight link on a network path.

Section 3.1 describes the theory behind the estimation of bandwidth in a one-link network path and Section 3.2 describes the same for a multi-link network path.

3.1 One-Link Path

In this section, we explain the estimation of utilization, the end-to-end available bandwidth and the capacity of the tight link in a network path composed of a single link.

3.1.1 Estimating Utilization

We make use of concepts from basic queueing theory to estimate the utilization of a network path. In a standard queueing system, the probability that there are no packets in the queue, denoted by π_0, is given by the well-known equation:

$$\pi_0 = 1 - \rho \tag{3.1}$$

where ρ is the utilization of the queue.

Consider a network path made up of a single store and forward link. A sequence of probe packets is sent across the network path at a certain rate and the RTT of each of the probe packets is measured. The measured RTT of a probe packet consists of two components: the *processing delay* and the *queueing delay* of the packet. In the ideal case we assume that the variation in the RTT that is not due to queueing delay is spread across the entire spectrum of measured RTTs. Let RTT_0 be the minimum RTT among the measured RTTs. The probe packets that have a RTT equal to RTT_0 are the ones that did not experience any queuing delays. Therefore, the ratio of the number of probe packets with minimum RTTs to the total number of probe packets gives π_0. From Equation 3.1, we can calculate the utilization of the path, ρ using this value of π_0 as:

$$\rho = 1 - \pi_0 \tag{3.2}$$

3.1.2 Estimating End-to-End Available Bandwidth and Capacity of the Tight Link

Consider a sequence of probe packets that are sent with an arrival rate that has a Poisson distribution which means that their inter-arrival times are exponentially distributed. Since the probe packets have exponential inter-arrival times, the well-known Poisson Arrival See Time Average (PASTA) property holds for the probes [27]. According to PASTA property, the packets arriving in a queue find, on an average, the same situation in the queueing system as an outside observer looking at the system at an arbitrary point in time will find. Therefore, if r is the rate at which the probe packets are sent, the *effective utilization* $\rho'(r)$ at probing rate r will be:

$$\rho'(r) = \rho + \frac{r}{C} \tag{3.3}$$

where ρ is the utilization of the network path due to the cross-traffic on the path (not including the probe traffic) and C is the capacity of the link on the network path. The above equation is linear and can be written as:

$$\rho'(r) = c_0 + rc_1 \tag{3.4}$$

where $c_0 = \rho$ and $c_1 = 1/C$ are constants over the averaging timescale τ.

If A is the available bandwidth of the link on the network path, then $\rho'(r) = 1$ when $r = A$, and Equation 3.4 reduces to $1 = c_0 + Ac_1$ i.e.,

$$A = \frac{(1 - c_0)}{c_1} \tag{3.5}$$

which is equivalent to Equation 1.3.

In order to estimate A without having to increase the rate r to A, we need to find the values of c_0 and c_1. We achieve this by sending two probing sequences at two different rates r_1 and r_2. We measure the values of the effective utilization $\rho'(r_1)$ and $\rho'(r_2)$ corresponding to the two probing rates using the concept of probe packets that are queued against those that are not as explained above. From Equation 3.4, we have:

$$\rho'(r_1) = c_0 + r_1c_1 \tag{3.6}$$

$$\rho'(r_2) = c_0 + r_2c_1 \tag{3.7}$$

Solving Equations 3.6 and 3.7, we calculate c_0 and c_1. We note that $1/c_1$ is exactly C, the capacity bandwidth of the link. Also, substituting the values of c_0 and c_1 in Equation 3.5, we obtain the value of A which is the available bandwidth of the link on the network path.

3.2 Multiple-Link Path

In this section, we present the methodology to estimate the utilization, the end-to-end available bandwidth and the capacity bandwidth of the tight link of a network path made up of multiple links. The multi-link case is an extension of the one-link case presented above.

3.2.1 Estimating Utilization

Consider a network path P with H links, $1, \ldots H$. Approximating the path queues of all links on the network path to a single queue, the probability that there are no packets

in any of the queues in the network path is:

$$\prod_{1 \leq i \leq H} \pi_0^i \tag{3.8}$$

where π_0^i is the probability of having no packets queued in link i. Hence the utilization ρ of the network path can be expressed as:

$$\rho = \prod_{1 \leq i \leq H} 1 - \pi_0^i \tag{3.9}$$

Also, by Equation 3.3, we can express the effective end-to-end utilization $\rho'(r)$ where r is the probing rate as:

$$\rho'(r) = 1 - \prod_{1 \leq i \leq H} (1 - (\rho_i + r/C_i)) \tag{3.10}$$

where ρ_i is the utilization due to cross-traffic excluding the probe traffic on link i.

3.2.2 Estimating End-to-End Available Bandwidth and Capacity of the Tight Link

Equation 3.10 is of degree H and can be written as:

$$\rho'(r) = \sum_{i=0}^{H} c_i r^i \tag{3.11}$$

where

$$c_0 = 1 - \prod_{1 \leq i \leq H} (1 - \rho_i)$$

,

$$c_k = (-1)^{k+1} \sum_{\substack{1 \leq i_1, i_2, \ldots i_k \leq H \\ i_1 \neq i_2 \ldots \neq i_k}} \frac{\prod_{\substack{1 \leq j \leq H \\ j \neq i_1 \neq i_2 \ldots \neq i_k}} (1 - \rho_j)}{C_{i_1} C_{i_2} \ldots C_{i_k}}$$

and

$$c_H = (-1)^{H+1} / \prod_{1 \leq i \leq H} C_i$$

The above equation clearly shows that $|c_k| \gg |c_{k+1}|$ since the numerator of each constant term, c_k, is relatively small and the denominator dramatically increases as k increases. To visualize the differences in the magnitude of the c_k values, consider a path comprised of 5 links l_i with per-link capacities C_i and utilization u_i where $i = 1 \ldots 5$. Let

the links' capacities be 100Mbps, 1Gbps, 100Mbps, 10Mbps, and 1Mbps in order. Also, let the links' utilizations be 0.4, 0.2, 0.3, 0.3 and 0.1 in order. Substituting these values in Equation 3.11, $\rho'(r) = 0.78 + 2.7X10^{-7}r - 4.2X10^{-14}r^2 + 1.15X10^{-21}r^3 - 9.3X10^{-30}r^4 + 10^{-38}r^5$. Thus, $c_0 = 0.78, c_1 = 2.7X10^{-7}, c_2 = 4.2X10^{-14}, c_3 = 1.15X10^{-21}, c_4 = 9.3X10^{-30}$ and $c_5 = 10^{-38}$. Notice that the first co-efficient, 0.78, is the end-to-end utilization of the path (without any induced probe traffic).

Given that $|c_k| \gg |c_{k+1}|$, Equation 3.11 can be approximated by first-order or second-order equations. A **first-order approximation** of $\rho'(r)$ could be shown in Equation 3.4 as in the one-link case. A **second-order approximation** of $\rho'(r)$ could be shown as:

$$\rho'(r) = c_0 + rc_1 + r^2c_2 \tag{3.12}$$

Following the example above, by the first-order approximation, $\rho'(r) = 0.78 + 2.7X10^{-7}r$. By second-order approximation, $\rho'(r) = 0.78 + 2.7X10^{-7}r - 4.2X10^{-14}r^2$. When $r = A$, $\rho'(r) = 1$. For the first-order approximation case,

$$A = \frac{1 - c_0}{c_1} \tag{3.13}$$

as in the one-link case. For the second-order approximation case, Equation 3.12 becomes $1 = c_0 + Ac_1 + A^2c_2$, and thus

$$A = \frac{-c_1 - \sqrt{c_1^2 - 4(c_0 - 1)c_2}}{2(c_0 - 1)} \tag{3.14}$$

Note that while the second-order approximation yields two solutions for A, we only solve it to get the lowest value solution because the available bandwidth along a path is the minimum bandwidth value of all the links along the path. In fact, Equation 3.11 has H solutions representing the A values of each link along the path. The minimum such root is the path available bandwidth. Notice that while this equation is non-linear, in practice, $\rho'(r)$ will not have H different values of 1; instead, our measured $\rho'(r)$ will approach 1 only once and then saturates independent of how large the probing rate is.

In first-order approximation case, two probing rates r_1 and r_2 are used and the resulting $\rho'(r_1)$ and $\rho'(r_2)$ are used to infer c_0 and c_1. In the second-order approximation case, three probing rates r_1, r_2 and r_3 are used to infer c_0, c_1 and c_2. Following our example, A should be 0.9Mbps. Using first-order approximation, $A \approx 0.81$ and using second-order approximation, $A \approx 0.9$. Though second-order approximation yields a more accurate

estimate of A, it assumes that the path being probed is stable over the probing period. Since it requires three probing sequences as opposed to two in the first-order approximation case, we resort to first-order approximation in our experiments, which should be reasonably accurate.

Thus, estimation of end-to-end available bandwidth and capacity of the tight link in the case where there is more than one link in the network path is essentially the same as in the case where there is a single link in the network path, as described in Section 3.1 using the first-order approximation. We send two probing sequences at two different rates r_1 and r_2 as in the one-link case. We measure the effective utilizations $\rho'(r_1)$ and $\rho'(r_2)$ corresponding to the two probing rates. By Equations 3.6 and 3.7, we derive the expression for c_1 as:

$$c_1 = \frac{\rho'(r_2) - \rho'(r_1)}{(r_2 - r_1)} \tag{3.15}$$

Substituting c_1 in Equation 3.6, we derive the expression for c_0 as:

$$c_0 = \rho'(r_1) - r_1 c_1 \tag{3.16}$$

We substitute the values of c_0 and c_1 in Equation 3.5 to obtain an estimate of the available bandwidth of the network path.

The reciprocal of c_1 i.e., $1/c_1$ is the capacity bandwidth of the tight link. The algorithm for first-order approximation is given in Algorithm 1

input : $\rho'(r_1)$ and $\rho'(r_2)$ resulting from two probe sequences, at rates of r_1 and r_2, respectively
output: A, the end-to-end available bandwidth and C, the capacity of the tight link

1 $c_1 = (\rho'(r_2) - \rho'(r_1))/(r_2 - r_1)$;
2 $C = 1/c_1$;
3 $c_0 = \rho'(r_1) - r_1 c_1$;
4 $A = (1 - c_0)/c_1$;
5 **return** A, C ;

Algorithm 1: Estimating the end-to-end available bandwidth and capacity of the tight link of the network path

Chapter 4

Experimental Setup

In this chapter, we describe the experimental setup used to evaluate the performance of our technique using $ns-2$ simulations. Section 4.1 is a detailed description of the network setup we have employed for simulations using the network simulator, $ns-2$. Section 4.2 lists our choices of probing rates, probe packet sizes, and the cross traffic model used and outlines the impact of these parameters on the results of our experiments.

4.1 Network Simulation Setup

We simulate a 3-link and a 4-link network path using the network simulator $ns-2$ [3]. The topologies of the 3- and the 4-link networks that we simulate are shown in Figures 4.1 and 4.2, respectively.

The network with 3 links has 4 nodes connected by links $L1$, $L2$, and $L3$, whose capacities vary with the scenario being tested. Cross-traffic on the network exists on a per-link basis, i.e., cross-traffic introduced into a link exits at the end of that link rather than exiting at the final destination of the network path. Cross-traffic sources are marked *cts1*, *cts2*, *cts3* and *cts4*. Each source is representative of a number of TCP and UDP flows, as explained in detail in Section 4.2. The sinks for the cross-traffic at each link are marked *sink1*, *sink2*, *sink3* and *sink4*. Probe packet-sequences are sent from the node marked s, which is the source, to the node marked d, which is the destination. The 4-link network path, shown in Figure 4.2, is a simple extension of the 3-link case.

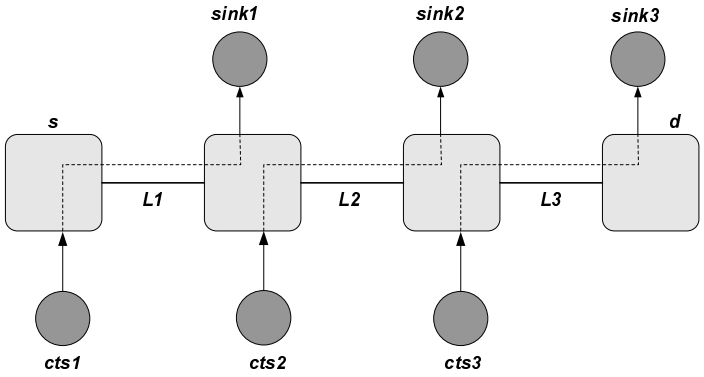

Figure 4.1: The simulation setup for a 3-link network with per-link cross-traffic. *L*s denote links, *cts*s denote cross-traffic sources, *sink*s denote cross-traffic sinks, *s* denotes the source, and *d* the destination of probe packets.

Table 4.1: Common Internet links and their capacities

Link Type	Mbps	Link Type	Mbps	Link Type	Mbps
T3	44.736	10BaseT	10.000	OC1	51.840
T4	274.176	100BaseT	100.000	OC12	622.080
		1000BaseT	1000.000	OC96	4976.000
				OC192	9953.280

4.1.1 Scenarios

We gather data to validate our technique by simulating the above network topologies under various *scenarios*, as described in Sections 4.1.2 and 4.1.3. Each scenario is unique in the capacities of the various links in the network. All scenarios are composed of links whose capacities are those of the standard links shown in Table 4.1 since these are the most commonly encountered links on the Internet. In all scenarios we set up links in the interior of the network path to have higher capacities than links at the ends of the path since that is generally the case in the real world. However, we test the performance of our technique when the links in the middle are the tight links.

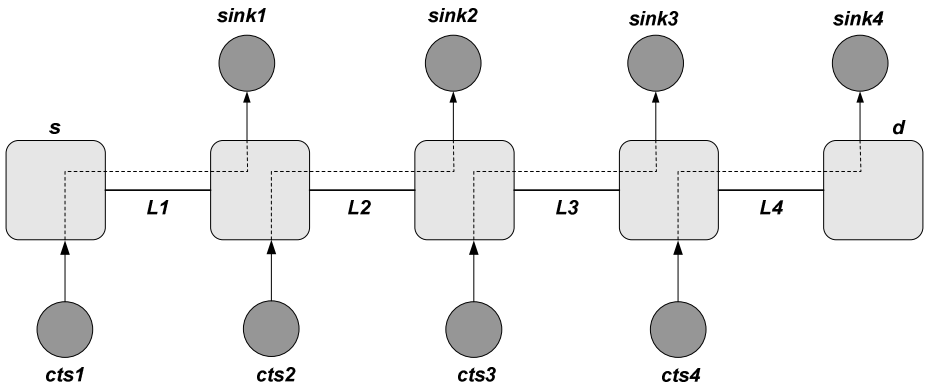

Figure 4.2: The simulation setup for a 4-link network with per-link cross-traffic. Ls denote links, *cts*s denote cross-traffic sources, *sink*s denote cross-traffic sinks, s denotes the source, and d the destination of probe packets.

4.1.2 Network setup with 3 links

This section lists scenarios that we have simulated for the setup composed of 3 links shown in Figure 4.1.

Scenario I

In Scenario I, $L1$ is a 100BaseT link with a capacity of 100Mbps, $L2$ is an OC12 link with a capacity of 622.08Mbps, and $L3$ is a 10BaseT link with a capacity of 10Mbps.

Scenario II

In this scenario, $L1$ is a 1000BaseT (1000Mbps), $L2$ is an OC192 (9953Mbps) and $L3$ is a 100BaseT (100Mbps) link. This scenario demonstrates the performance of our technique in the presence of a very high speed link in the middle of the network path and a relatively low speed link at the end.

Table 4.2: Scenarios for the 3-link network setup shown in Figure 4.1

	L1	L2	L3
Scenario I	100BaseT	OC12	10BaseT
Scenario II	1000BaseT	OC192	100BaseT
Scenario III	OC1	OC96	10BaseT

Table 4.3: Scenarios for the 4-link network setup shown in Figure 4.2

	L1	L2	L3	L4
Scenario I	1000BaseT	OC96	OC12	100BaseT

Scenario III

The links used in Scenario III are OC1 (51.84Mbps) for $L1$, OC96 (4976Mbps) for $L2$, and 10BaseT (10Mbps) for $L3$.

All the scenarios for the 3-link network setup are summarized in Table 4.2.

4.1.3 Network Setup with 4 links

This section describes the scenario used to simulate the network setup composed of 4 links.

Scenario I

The links used in this scenario are 1000BaseT (1000Mbps) for $L1$, OC96(4976Mbps) for $L2$, OC12 (622.08Mbps) for $L3$ and 100BaseT (100Mbps) for $L4$. This scenario is listed in Table 4.3.

4.1.4 Test Cases

For each of the scenarios of the 3-link and 4-link network topologies described in Sections 4.1.2 and 4.1.3, we use 4 test cases to validate our technique. These 4 test cases differ from one another in the percentage of cross-traffic (utilization) induced on every link. Consequently, the available bandwidth and the location of the tight link are different for each test case. The test cases have been formulated as follows:

- Case 0: This is the test case where there is no competing cross-traffic on any of the links

Table 4.4: Percentage utilization of links for all test cases of Scenario I of the 3-link network topology

	L1	L2	L3
Case 0	0%	0%	0%
Case 1	30%	40%	20%
Case 2	95%	20%	.01%
Case 3	20%	99%	.01%

Table 4.5: Percentage utilization of links for all test cases of Scenario II of the 3-link network topology

	L1	L2	L3
Case 0	0%	0%	0%
Case 1	20%	10%	50%
Case 2	92%	10%	.01%
Case 3	10%	99.1%	0%

- Case 1: In this test case, the narrow link is the tight link.
- Case 2: In this test case, the narrow link is not the tight link.
- Case 3: In this test case, the high speed link in the interior of the network setup is the tight link which means that the utilization on this link is at least 90% for all the scenarios.

The test cases for Scenarios I, II and III of the 3-link network setup have been shown in Tables 4.4, 4.5 and 4.6 in terms of the percentage of link capacity that is utilized by the cross-traffic. The test cases for the 4-link network setup in terms of the cross-traffic percentage on the links are shown in Table 4.7.

Table 4.6: Percentage utilization of links for all test cases of Scenario III of the 3-link network topology

	L1	L2	L3
Case 0	0%	0%	0%
Case 1	50%	10%	20%
Case 2	10%	20%	70%
Case 3	0%	98.99%	10%

Table 4.7: Percentage utilization of links for all test cases of Scenario I of the 4-link network topology

	L1	L2	L3	L4
Case 0	0%	0%	0%	0%
Case 1	20%	10%	10%	30%
Case 2	95%	10%	10%	.01%
Case 3	20%	99%	20%	.01%

4.2 Network traffic Parameters

This section describes our choices of probing rate, probe size, and cross-traffic characteristics.

Probing Rate

To estimate the available bandwidth of a network path and the capacity of the tight link, we probe the network path with two probe sequences with rates r_1 and r_2. The premise is that the utilizations $\rho'(r_1)$ and $\rho'(r_2)$ corresponding to the two probing rates r_1 and r_2 will be different owing to Equation 3.3 and the difference between the two values of utilization can be used in calculating c_1, which is the slope of the curve of the estimated capacity.

In order to obtain a reasonable estimate of available bandwidth and capacity of the tight link, it is important that the utilizations $\rho'(r_1)$ and $\rho'(r_2)$ differ at least by 10 percent. To achieve this difference, the difference between the probing rates should be at least 10 percent of the capacity bandwidth of the narrow link. In real world experiments, since the capacity bandwidth of the narrow link is not known in advance, we adjust the probing rates such that the difference between $\rho'(r_1)$ and $\rho'(r_2)$ is at least 10 percent.

Probe packet Size

The probe packet size used in our experiments is 1500 bytes.

Number of probes

We use a sequence of 1000 probes to probe the network path. The probing rate to be used varies inversely with the number of probes sent, i.e., fewer probes require a higher probing rate. This is because our technique depends on there being a measurably significant

difference in the utilizations as the probing rate increases, due to queueing induced by the probe packets themselves. The most accurate estimates of available bandwidth are obtained using a sequence of 1000 probes.

Cross-Traffic

We have created a cross-traffic model based on the CAIDA studies explained in [20] and [16]. According to [16], the percentage of TCP traffic on the Internet ranges between 60% to 90% and the percentage of UDP traffic ranges between 40% to 10%. The CAIDA study also showed that the packet sizes have a tri-modal distribution, with the bulk of the packets on the Internet falling among one of the following three modes:

1. 40-44 bytes: TCP ACKs, Control Segments etc.
2. 556-576 bytes: Maximum Segment Size (MSS) when Maximum Transfer Unit (MTU) discovery is not used
3. 1500 bytes: Ethernet Maximum Transfer Unit (MTU)

We model cross-traffic based on this study. 60% of the cross-traffic is TCP traffic and the remaining 40% is UDP traffic. The cross-traffic packet sizes are distributed between 40 bytes, 576 bytes and 1500 bytes.

Chapter 5

Results

In this chapter, we present the results of $ns-2$ simulations to demonstrate the accuracy of the proposed technique. In all of the results, we use two performance metrics namely ϵ^A which is the error in the estimation of the available bandwidth and ϵ^C which is the error in the estimation of capacity of the tight link. The error ϵ^A is defined as:

$$\epsilon^A = \frac{A - \overline{A}}{A} \tag{5.1}$$

where A is the actual end-to-end available bandwidth of the network path and $\overline{A}$ is the estimated end-to-end available bandwidth of the network path. Similarly, the error ϵ^C is defined as:

$$\epsilon^C = \frac{C - \overline{C}}{C} \tag{5.2}$$

where C is the actual capacity bandwidth of the tight link on the network path and $\overline{C}$ is the estimated capacity bandwidth of the tight link on the network path. The scenarios and test cases which are used in the following sections have been described in detail in Chapter 4.

5.1 Network setup with 3 links

This section presents the results of the estimation of available bandwidth of the network path and the capacity of the tight link for the scenarios of the 3-link network setup.

Table 5.1: Available Bandwidth Estimates for Scenario I of the 3-link network setup. A is the actual value of the capacity of the tight link, $\overline{A}$ is the estimated value and ϵ^A is the error in estimation.

	100BaseT (100Mbps)	OC12 (622.08Mbps)	10BaseT (10Mbps)
	A	$\overline{A}$	ϵ^A
Case 0	10	10.14	-0.014
Case 1	8	8.14	-0.017
Case 2	5	5.09	-0.018
Case 3	6.22	6.12	+0.016

Table 5.2: Available Bandwidth Estimates for Scenario II of the 3-link network setup. A is the actual value of the capacity of the tight link, $\overline{A}$ is the estimated value and ϵ^A is the error in estimation

	1000BaseT (1000Mbps)	OC192 (9953Mbps)	100BaseT (100Mbps)
	A	$\overline{A}$	ϵ^A
Case 0	100	100.37	-0.004
Case 1	50	52.72	-0.049
Case 2	80	72.1	+0.098
Case 3	89.6	93.05	-0.038

5.1.1 Available Bandwidth

In this section we present the estimates of end-to-end available bandwidth obtained using our technique.

Scenario I

Table 5.1 shows the actual value of the available bandwidth A, the estimated value $\overline{A}$ and the error in estimation ϵ^A. Note that for all the cases in this scenario, the error ϵ^A is less than 2%. This is true even in the case where the OC12 link in the middle has a high utilization of 99% (Case 3).

Scenario II

The actual values, estimated values and the errors in estimation of the available bandwidth of the network path for Scenario II are presented in Table 5.2. The error in estimation is less than 5% in all the cases except in Case 2.

Table 5.3: Available Bandwidth Estimates for Scenario III of the 3-link network setup. A is the actual value of the capacity of the tight link, $\overline{A}$ is the estimated value and ϵ^A is the error in estimation

	OC1 (51.84Mbps)	OC96 (4976Mbps)	10BaseT (10Mbps)
	A	$\overline{A}$	ϵ^A
Case 0	51.84	52.37	-0.010
Case 1	25.92	25.54	+0.015
Case 2	30	29.14	-0.028
Case 3	50.2	39.92	+0.204

Table 5.4: Capacity Bandwidth Estimates of the tight link for Scenario I of the 3-link network setup. C is the actual value of the capacity of the tight link, $\overline{C}$ is the estimated value and ϵ^C is the error in estimation

	100BaseT (100Mbps)	OC12 (622.08Mbps)	10BaseT (10Mbps)
	C	$\overline{C}$	ϵ^C
Case 0	10	10.16	-0.016
Case 1	10	24.3	-1.43
Case 2	100	127.15	-0.271
Case 3	622.08	722.03	+0.16

Scenario III

The actual values, estimated values and error in estimation of the end-to-end available bandwidth of the network path for Scenario III are shown in Table 5.3. The high error in Case 3 is because the OC96 link is 98.99% utilized.

5.1.2 Capacity Bandwidth

In this section, we present the results of the estimation of the capacity bandwidth of the tight link for the scenarios of the 3-link network setup.

Scenario I

The values for the actual capacity, the estimated capacity and the error in estimation are shown in Table 5.4. We see that the error in estimation of the capacity bandwidth of the tight link is quite high in the presence of competing cross-traffic. To correct this error we use a classification technique where we classify the estimated capacity bandwidth of the tight link as one of the standard links shown in Table 4.1.

Capacity Classification

The capacity curves of the standard Internet links are the curves of traffic against utilization. These curves for the links of Table 4.1 are shown in Figures 5.1 and 5.2. We

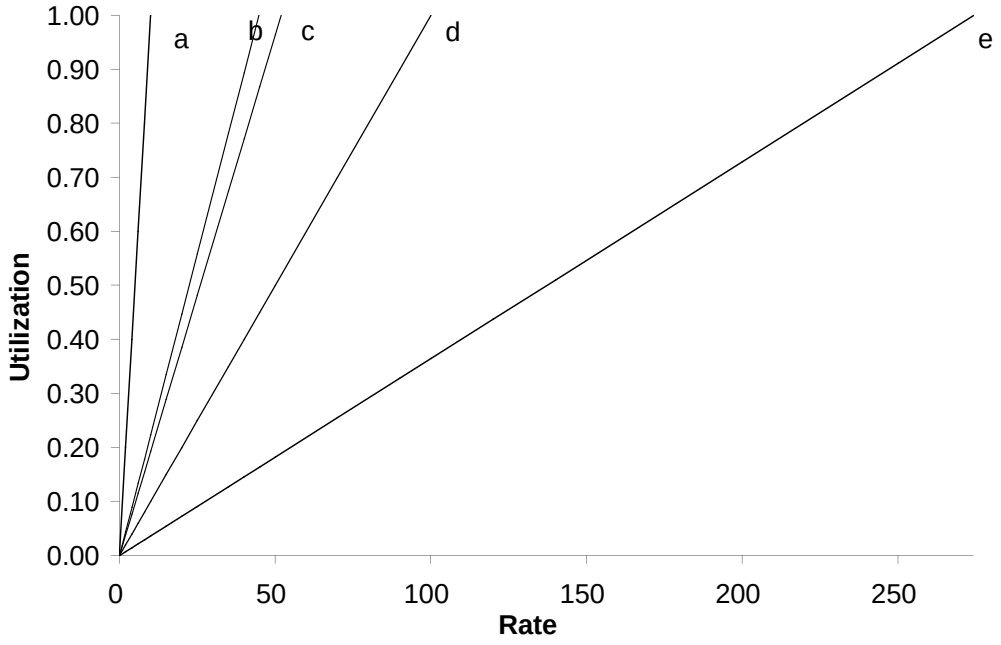

Figure 5.1: Standard link characteristics, with the vertical axis representing utilization corresponding to the cross-traffic shown on the horizontal axis. (a)10BaseT (b)T3 (c)OC1 (d)100BaseT (e)T4

have estimated $\rho'(r_1)$ and $\rho'(r_2)$ for the two probing rates r_1 and r_2. We plot these points on the graph and extrapolate the curve to obtain the capacity curve of the estimated capacity.

For Scenario I of the 3-link network setup, this has been shown in Figure 5.3 for all the test cases. Not all the capacity curves of the standard links have been included in the graph. This is for the sake for clarity. We note that the difference between the estimated capacity of the tight link and the standard link capacities is minimum for the 10BaseT i.e., the 10Mbps link. From this difference and from the capacity curve for Case 0 in Figure 5.3, we classify the tight link in Case 0 of Scenario I as a 10BaseT link which is accurate.

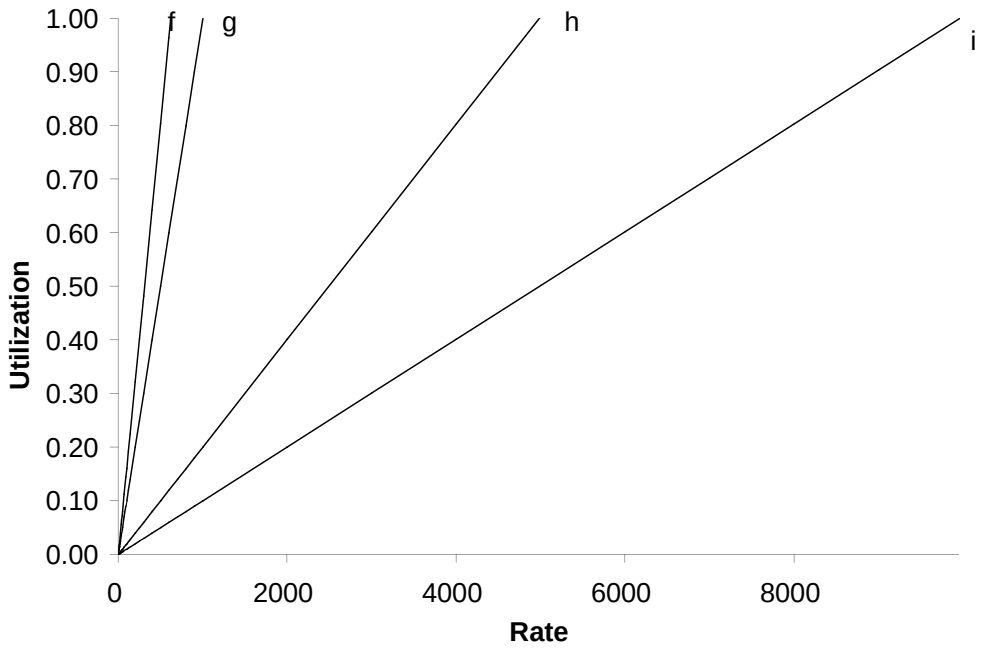

Figure 5.2: Standard link characteristics, with the vertical axis representing utilization corresponding to the cross-traffic shown on the horizontal axis. (f)OC12 (g)1000BaseT (h)OC96 (i)OC192.

Similarly, for Cases 1, 2 and 3, we classify the tight links correctly as 10BaseT, 100BaseT and OC12.

Scenario II

The Table 5.5 shows the actual and estimated values of the capacity bandwidth of the tight link and the error in estimation for this scenario.

Capacity Classification

The capacity curves for this scenario have been shown in Figure 5.4. From the respective capacity curves and from the difference between estimated capacities and the capacities of standard links, we classify the tight link as 100BaseT, 100BaseT, 1000BaseT

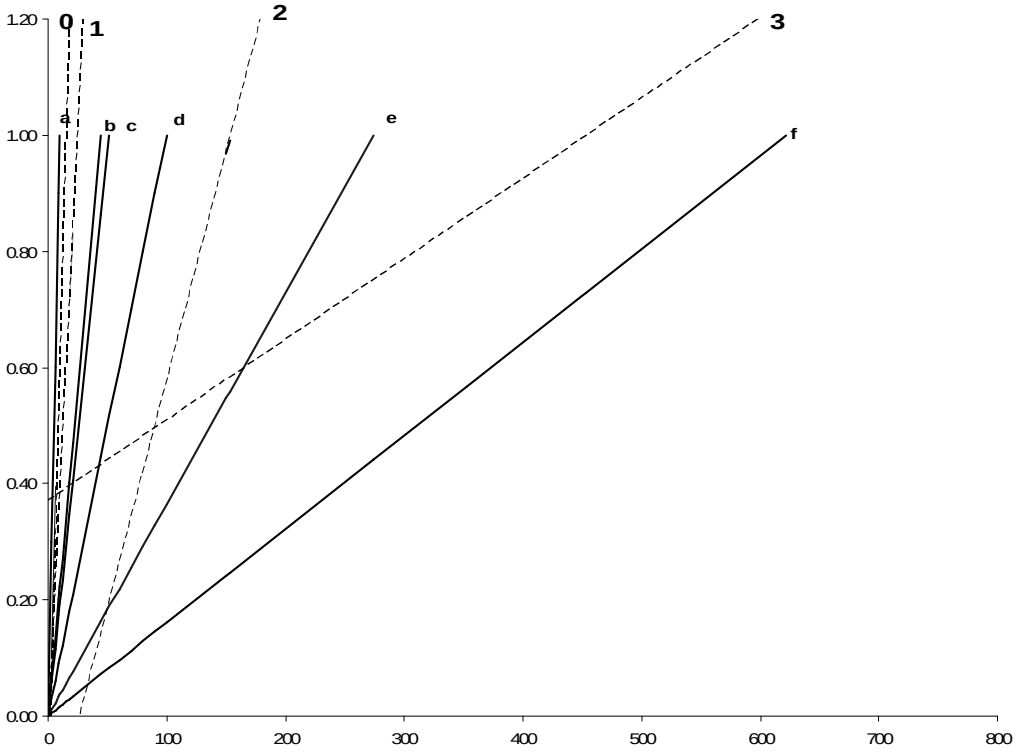

Figure 5.3: Comparison of standard link characteristic and the slopes of curves of the estimated capacity for Scenario I of the 3 link network setup. The dotted lines represent the extrapolated curves of the estimated capacity of the tight link. The solid lines represent the characteristic of standard links. The standard links represented in this graph are (a)10BaseT (b)T3 (c)OC1 (d)100BaseT (e)T4 (f) OC12. The dotted lines correspond to (0) Case 0 (1) Case 1 (2) Case 2 (3) Case 3

and OC96 for Cases 0, 1 2 and 3 respectively.

Scenario III

The results for the estimation of the capacity of the tight link for Scenario III have been shown in Table 5.6.

Capacity Classification

Figure 5.5 shows the capacity curves for the test cases in Scenario III. The classification of the tight link for the Cases 0, 1, 2 and 3 is OC1, OC1, 100BaseT and OC96

Table 5.5: Capacity Bandwidth Estimates of the tight link for Scenario II of the 3-link network setup. C is the actual value of the capacity of the tight link, $\overline{C}$ is the estimated value and ϵ^C is the error in estimation

	1000BaseT (1000Mbps)	OC192 (9953Mbps)	100BaseT (100Mbps)
	C	$\overline{C}$	ϵ^C
Case 0	100	100.39	-0.004
Case 1	100	148.48	-0.484
Case 2	1000	994.13	+0.006
Case 3	9953	8481.9	+0.148

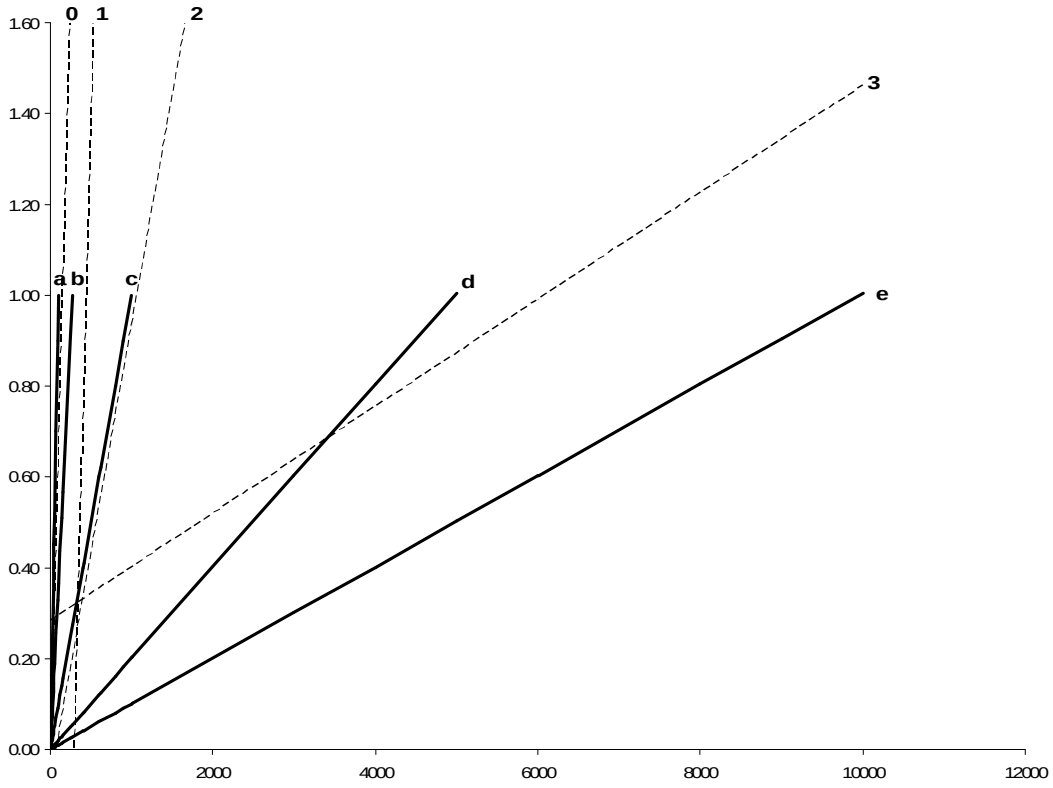

Figure 5.4: Comparison of standard link characteristic and the slopes of curves of the estimated capacity for Scenario II of the 3 link network setup. The dotted lines represent the extrapolated curves of the estimated capacity of the tight link. The solid lines represent the characteristic of standard links. The standard links represented in this graph are (a)100BaseT (b)T4 (c)1000BaseT (d)OC96 (e)OC192 The dotted lines correspond to (0) Case 0 (1) Case 1 (2) Case 2 (3) Case 3

respectively.

Table 5.6: Capacity Bandwidth Estimates of the tight link for Scenario III of the 3-link network setup. C is the actual value of the capacity of the tight link, $\overline{C}$ is the estimated value and ϵ^C is the error in estimation

	OC1 (51.84Mbps)	OC96 (4976Mbps)	10BaseT (10Mbps)
	C	$\overline{C}$	ϵ^C
Case 0	51.84	52.42	-0.011
Case 1	51.84	55.14	-0.064
Case 2	100	102.34	+0.0214
Case 3	4976	3360.14	+0.324

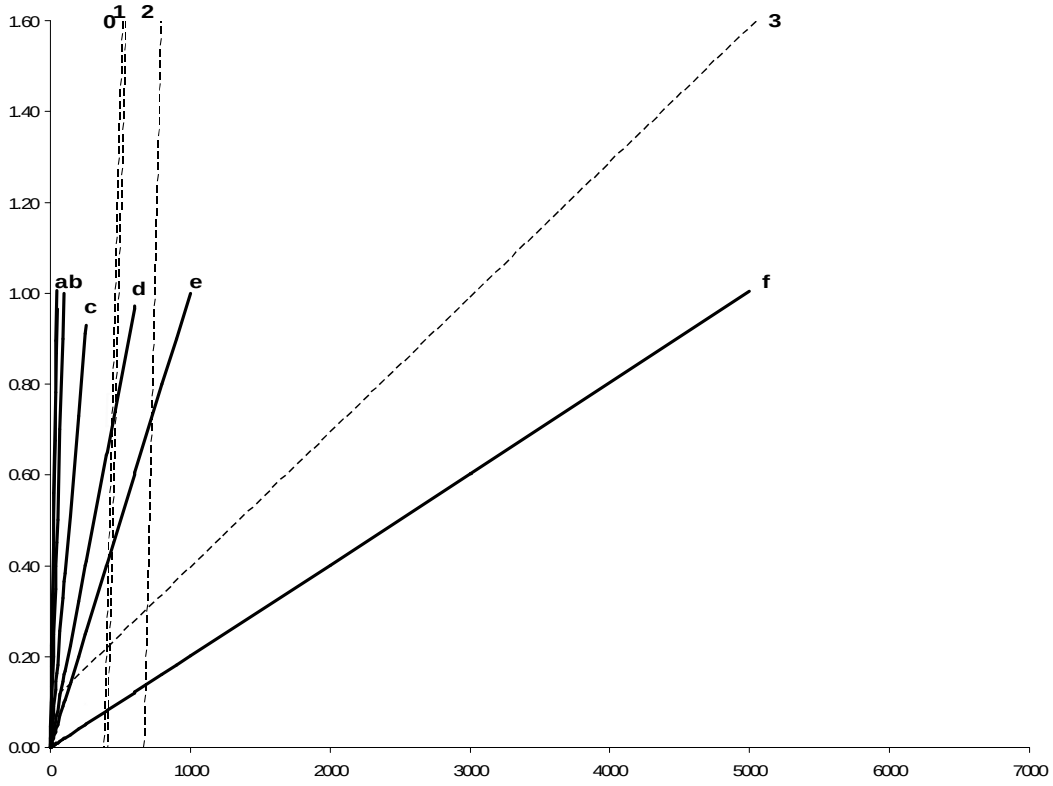

Figure 5.5: Comparison of standard link characteristic and the slopes of curves of the estimated capacity for Scenario III of the 3 link network setup. The dotted lines represent the extrapolated curves of the estimated capacity of the tight link. The solid lines represent the characteristic of standard links. The standard links represented in this graph are (a)T3 (b)100BaseT (c)T4 (d)OC12 (e)1000BaseT (f) OC96. The dotted lines correspond to (0) Case 0 (1) Case 1 (2) Case 2 (3) Case 3

Table 5.7: Available Bandwidth Estimates for Scenario I of the 4-link network setup. A is the actual value of the capacity of the tight link, $\overline{A}$ is the estimated value and ϵ^A is the error in estimation

1000BaseT (1000Mbps)	OC96 (4976Mbps)	OC12 (622.08Mbps)	100BaseT (100Mbps)
	A	$\overline{A}$	ϵ^A
Case 0	100	100.33	-0.003
Case 1	70	71.1	-0.015
Case 2	50	51.73	-0.035
Case 3	49.76	43.15	-0.133

5.2 Network setup with 4 links

In this section, we present the results of estimation of the available bandwidth of the network path and the capacity bandwidth of the tight link for the Scenarios described in Chapter 4.

5.2.1 Available Bandwidth

This section gives the results of the available bandwidth estimation using our technique.

Scenario I

Table 5.7 gives the actual and estimated values of the available bandwidth of the network path composed of 4 links. It also presents the error ϵ^A in estimation. The error in estimation is less than 4% in all the cases except in Case 3 (13%) wherein we have a high bandwidth OC96 link that is almost completely utilized.

5.2.2 Capacity Bandwidth

In this section we show the results of estimation of the capacity bandwidth of the tight link.

Scenario I

The results for the capacity bandwidth estimation have been shown in Table 5.8.

Table 5.8: Capacity Bandwidth Estimates of the tight link for Scenario I of the 4-link network setup. C is the actual value of the capacity of the tight link, $\overline{C}$ is the estimated value and ϵ^C is the error in estimation

1000BaseT (1000Mbps)	OC96 (4976Mbps)	OC12 (622.08Mbps)	100BaseT (100Mbps)
	C	$\overline{C}$	ϵ^C
Case 0	100	100.33	-0.003
Case 1	100	127.22	-0.272
Case 2	1000	1276.65	-0.277
Case 3	4976	5350.6	-0.075

Capacity Classification

Figure 5.6 shows the capacity curves for the standard links and the estimated capacity curve for Cases 0, 1 2, and 3 of Scenario I of the 4-link network setup. From the Figure and by taking the minimum difference between the capacity of the standard links and the estimated capacity, we classify the tight link for Cases 0, 1, 2 and 3 as 100BaseT, 100BaseT, 1000BaseT and OC96 which are the correct values of the capacity. We see from the above results that from the 16 cases simulated, only 3 have an error in the estimation of available bandwidth higher that 5%. The 3 cases are the ones where one of the high bandwidth links is very highly utilized (greater than 95%). In those cases, the highest error in estimation is 20.4%. Hence we conclude that we can estimate the available bandwidth of a network path with reasonably high accuracy using this technique of bandwidth measurement.

In case of the capacity estimation of the tight link, we see a fairly high error in estimation mainly in the presence of competing cross-traffic. To correct this error we use a capacity classification technique which gives the capacity bandwidth of the tight link with 100% accuracy.

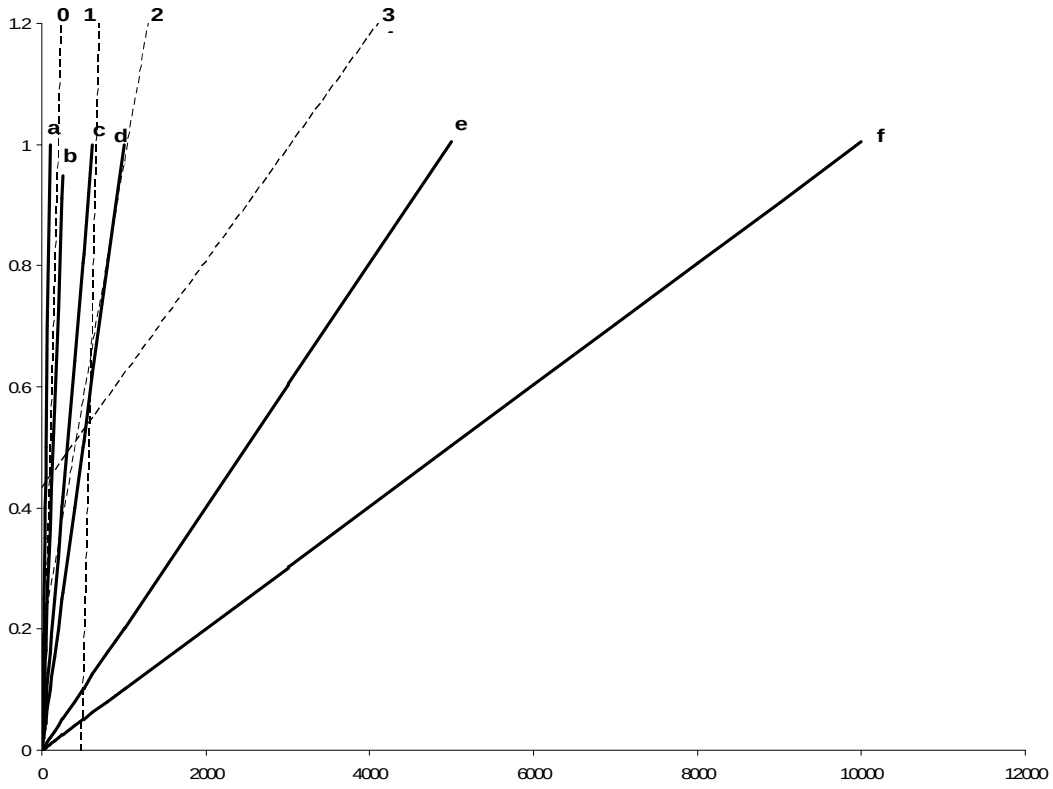

Figure 5.6: Comparison of standard link characteristic and the slopes of curves of the estimated capacity for Scenario I of the 4 link network setup. The dotted lines represent the extrapolated curves of the estimated capacity of the tight link. The solid lines represent the characteristic of standard links. The standard links represented in this graph are (a)100BaseT (b)T4 (c)OC12 (d)1000BaseT (e)OC96 (f) OC192. The dotted lines correspond to (0) Case 0 (1) Case 1 (2) Case 2 (3) Case 3

Chapter 6

Conclusion and Future Work

In this thesis, we introduced a novel technique to estimate the available bandwidth of a network path and the capacity bandwidth of the most utilized link – the tight link – of the the network path using a probing technique and a simple binary test of whether probes are queued over a network path or not. We showed using $ns-2$ simulations that our technique estimates the available bandwidth to within 5% of the actual value in most cases. We also introduced a technique to correct the errors in estimation of the capacity bandwidth of the tight link.

Our technique was demonstrated to have some desirable properties

1. We do not need prior knowledge of the capacity of the link to estimate the available bandwidth using this technique.

2. This technique is non-intrusive in that probe traffic is sent at a rate that is much lower than the available bandwidth of the network path.

3. This technique is efficient and simple in that it does not need accurate measurements of queueing delays or packet dispersions.

We make the assumption that at least one of our probe packets experiences no queueing delay across the network path. In cases where this assumption does not hold, the proposed technique over-estimates the available bandwidth of the network path.

This work has provided a promising start to a novel bandwidth-estimation technique. To further investigate the efficacy of our methodology, controlled lab and Internet experiments are necessary to ensure validity of this estimation methodology in a realistic

setting. We also intend to extend the technique to estimate available bandwidth of arbitrary segments of a network path. Another goal for the future is to integrate it with a rate-adaptive congestion control algorithm.

Bibliography

[1] Bbftp. http://doc.in2p3.fr/bbftp/.

[2] Iperf. http://dast.nlanr.net/Projects/Iperf/.

[3] The network simulator-ns-2. http://www.isi.edu/nsnam/ns/.

[4] David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, and Robert Morris. Resilient Overlay Networks. In *Symposium on Operating Systems Principles*, pages 131–145, 2001.

[5] Constantinos Dovrolis, Parameswaram Ramanathan, and David Moore. Packet-Dispersion Techniques and a Capacity Estimation Methodology. *IEEE/ACM Transactions on Networking (TON)*, 12(6):963–977, 2004.

[6] Constantinos Dovrolis, Parameswaran Ramanathan, and David Moore. What do packet dispersion techniques measure? In *INFOCOM*, pages 905–914, 2001.

[7] Allen B. Downey. clink: A tool for estimating internet link characteristics, 1999. rocky.wellesley.edu/downey/clink/.

[8] Allen B. Downey. Using pathchar to estimate internet link characteristics. In *Measurement and Modeling of Computer Systems*, pages 222–223, 1999.

[9] Ningning Hu, Li(Erran) Li, Zhuoging Morley Mao, Peter Steenkiste, and Jia Wang. Locating Internet Bottlenecks: Algorithms, Measurements and Implications. In *SIGCOMM*, 2004.

[10] Ningning Hu and Peter Steenkiste. Evaluation and characterization of available bandwidth and probing techniques. *IEEE JSAC Special Issue in Internet and WWW Measurement, Mapping, and Modeling*, 21:879–894, August 2003.

[11] V. Jacobson. pathchar - A Tool to Infer Characteristics of Internet Paths, 1997. ftp://ee.lbl.gov/pathchar.

[12] Manish Jain and Constantinos Dovrolis. End-to-End Available Bandwidth : Measurement Methodology, Dynamics, and Relation with TCP Throughput. *IEEE/ACM Transactions on Networking*, 11(4):537–549, August 2003.

[13] Manish Jain and Constantinos Dovrolis. End-to-End Estimation of the Available Bandwidth Variation Range. In *Proceedings of ACM SIGMETRICS*, June 2005.

[14] John Jannotti, David K. Gifford, Kirk L. Johnson, M. Frans Kaashoek, and James W. O'Toole, Jr. Overcast: Reliable Multicasting with an Overlay Network. pages 197–212.

[15] Rohit Kapoor, Ling-Jyh Chen, Li Lao, Mario Gerla, and M.Y. Sanadidi. CapProbe: A Simple and Accurate Capacity Estimation Technique. *ACM SIGCOMM Computer Communication Review*, 34(4):67–78, 2004.

[16] Marina Fomenkov Ken. Longitudinal study of internet traffic in 1998-2003.

[17] Diane Kiwior, James Kingston, and Aaron Spratt. PATHMON, A Methodology for determining Available Bandwidth over an Unknown Network. In *IEEE Sarnoff Symposium on Advances in Wired and Wireless Communications*, 2004.

[18] Kevin Lai and Mary Baker. Nettimer: A tool for measuring bottleneck link bandwidth. pages 123–134.

[19] Bruce A. Mah. pchar: A Tool for Measuring Internet Path Characteristics, 2001.

[20] S. McCreary and K. Claffy. Trends in Wide Area IP Traffic Patterns - A View from Ames Internet Exchange. *Proceedings of the 13th ITC Specialist Seminar on Internet Traffic Measurement and Modelling*, 2000.

[21] B. Melander, M. Bjorkman, and P. Gunningberg. A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks. In *IEEE Global Internet Symposium*, 2000.

[22] B. Melander, M. Bjorkman, and P. Gunningberg. Regression-Based Available Bandwidth Measurements. In *International Symposium on Performance Evaluation of Computer and Telecommunications Systems*, 2002.

[23] Tobias Oetiker and Dave Rand. MRTG. http://people.ee.ethz.ch/ oetiker/webtools/mrtg/.

[24] V. Paxson. End-to-End Internet Packet Dynamics. *IEEE/ACM Transactions on Networking (TON)*, 7(3):277–292, 1999.

[25] Vinay Ribeiro, Rudolf Riedi, Richard Baraniuk, Jiri Navratil, and Les Cottrell. pathChirp: Efficient Available Bandwidth Estimation for Network Paths. In *Proceedings of The Conference on Passive and Active Measurements (PAM)*, April 2003.

[26] R.L. Carter and M.E. Crovella. Measuring Bottleneck Link Speed in Packet-Switched Networks. *Performance Evaluation*, 27(28):297–318, 1996.

[27] R.W.Wolff. Poisson Arrivals See Time Average. *Operations Research*, 30:223–231, 1982.

[28] Srinivasan Seshan, Mark Stemm, and Randy H. Katz. SPAND: Shared passive network performance discovery. In *USENIX Symposium on Internet Technologies and Systems*, 1997.

[29] S.Keshav. A Control-Theoretic Approach to Flow Control. In *ACM SIGCOMM*, 1991.

[30] Jacob Strauss, Dina Katabi, and Frans Kaashoek. A Measurement Study of Available Bandwidth Estimation Tools. In *ACM/USENIX Internet Measurement Conference (IMC)*, 2003.

[31] Yong Zhu, Constantinos Dovrolis, and Mostafa H. Ammar. Dynamic Overlay Routing Based on Available Bandwidth Estimation: A Simulation Study. *Computer Networks*, 50(6):742–762, 2006.