This is to certify that the
thesis entitled

INTEGRATION OF A STATISTICAL METHOD IN ONE WAY
DELAY TREND DETECTION FOR AVAILABLE BANDWIDTH
ESTIMATION

presented by

MAHNAZ SHAFII

has been accepted towards fulfillment
of the requirements for the

M.S. degree in COMPUTER SCIENCE

Major Professor's Signature

5/2/2008

Date

MSU is an affirmative-action, equal-opportunity employer

INTEGRATION OF A STATISTICAL METHOD IN ONE WAY DELAY TREND DETECTION FOR AVAILABLE BANDWIDTH ESTIMATION

By

Mahnaz Shafii

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Computer Science

2008

ABSTRACT

INTEGRATION OF A STATISTICAL METHOD IN ONE WAY DELAY TREND DETECTION FOR AVAILABLE BANDWIDTH ESTIMATION

By

Mahnaz Shafii

In this thesis, available bandwidth estimation tools including Delphi, Spruce, Pathchirp, and Pathload are evaluated. Each method has disadvantages in satisfying the requirements of fast, accurate and non-intrusive. Among all, Pathload is the most reliable tool that considers the IC/CS phenomenon. Pathload sends multiple trains called fleets. Each train consists of 1200 same-size packets, with the same inter-packet time gaps. After each fleet is sent, statistical evaluation of the inter-packet time gaps at the destination compared to the inter-packet time gaps at the sender determines whether the trend of the inter-packet time gaps in that fleet has been increasing, non-increasing, or undeterminable. The transmission rate for the next fleet is adjusted based on the trend of the current fleet until the results converge. One challenge of Pathload is that the underlying one way delay detection (*OWD*) and the method of handling IC/CS effect is far from perfect. In this thesis, the previously proposed algorithm for efficient *OWD* trend detection based on a statistical method is implemented into the Pathload. Then, the enhanced Pathload is compared to Pathload. Experiments in our laboratory network and across several Internet paths clearly show that our enhanced Pathload significantly outperforms tests using Pathload with its original *OWD* trend detection algorithm.

TO MY Family

ACKNOWLEDGMENTS

Completing this graduate degree has been truly a marathon event for me, in which I experienced the great joy of giving birth to my son, Benjamin. I would have not been able to complete this journey without the aid and support of countless people over the past three years. Therefore, I am taking this opportunity to thank all those who have assisted me in one way or the other with my Masters thesis.

I would like to thank, first and foremost, my adviser Professor Matt Mutka for giving me the opportunity to work in his group. The friendly and supportive attitude of him contributed essentially to the final outcome of my studies. In this context, I would like to thank particularly my Masters committee professors Li Xiao , and Ning Xi , for their input and interest in this research.

I am thankful to my fiends in ELANS lab and CSE department. Especially, I must acknowledge Dr. Zhiwei Cen and Dr. Alexander Chobanyan for their guidance in my understanding of the network bandwidth measurement. Their friendly and prompt response to my questions is greatly appreciated.

My parents deserve special mention for their inseparable support and prayers. My Father, Dr. Ahmad Shafii, in the first place is the person who showed me the joy of intellectual pursuit ever since I was a child. My Mother, Behnoosh, is the one who sincerely raised me with her caring and gently love. Behshad, Behnaz, and Negah, thanks for being supportive and caring siblings.

Words fail me to express my appreciation to my husband whose dedication, love and persistent confidence in me, has taken the load off my shoulder. I owe

him for being unselfishly let his intelligence, passions, and ambitions collide with mine.

Finally, I would like to thank everybody who was important to the successful realization of this thesis, as well as expressing my apology to those I could not mention personally one by one.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1. Importance of Network Performance Measurements

As the Internet evolves, it is essential to have a clear understanding on the performance of the network. There are a number of terms that are commonly used to refer to various aspects of network performance. Depending on the application, the manner in which data is sent across the network may be more important than the raw speed at which it is transported. Performance measurements involve the introduction of traffic into the network for the purpose of monitoring delay between specific end-points. The most important performance metrics that are monitored include connectivity, delay, packet loss rate, and available bandwidth. Available bandwidth estimation is the most problematic measurement where accuracy is difficult to achieve, particularly in high-speed networks. Due to various technical problems and privacy issues, obtaining an accurate estimate of available bandwidth from routers is not possible. Therefore, estimation of available bandwidth requires an end-to-end measurement.

Bandwidth characteristics are important in several applications such as

- Peer-to-peer applications, where knowledge of available bandwidth is required before allowing peers to join the network.
- Overlay networks, where knowledge of available bandwidth of overlay links is required for configuration of the routing tables and optimal route selection.
- Network providers' charge, where the price is based on the available bandwidth provided to the costumers.
- Adaptive applications; in which applications need to know current network conditions in order to modify their output and maximize network use.
- Proxy selection: After predicting the available bandwidth, the proxy can select the appropriate upper bound of transmission rate.
- Congestion avoidance algorithms and intelligent routing systems: Implementation of connection-oriented protocols, such as the widely-used TCP protocol, and intelligent routing systems generally require the knowledge of available bandwidth in order to adjust the transmit speed.

Due to existing limitations on available bandwidth, better utilization of available network resources is crucial. For real time multimedia applications and congestion control transports, end-to-end QoS is the most critical issue [1-4]. Such applications are very sensitive to the availability of bandwidth, and an accurate available bandwidth measurement method improves their performance dramatically.

Measurements of available bandwidth are very difficult as they require time accuracy for the measurement. Unlike the capacity measurement, linear regression on the dataset of a number of measurements during a given time frame is impossible [5]. Because the network traffic changes frequently, convergence over a longer period of time is not a good estimate for available bandwidth. Inaccurate estimate of available bandwidth lead to buffer overflows, queuing delays, network clogging and other related problems that add up considerably over a period of time.

1.2. Notions for Measuring Traffic in Network Links

In this section definitions of some important terms that describe network performance are provided. Table 1 shows some selected network notions in measuring traffic in a network.

Capacity and the available bandwidth are the two important metrics that are commonly associted with a path. Consider path *P* of store-and-forwarding links, between two network hosts *SND* and *RCV*. Considering the route between *SND* and *RCV* is fixed, and there is no multi-path forwarding during the measurement, available bandwidth *A* and capacity *C* are defined as:

Capacity of a link i is the maximum transmission rate of a link when there is no cross traffic. Capacity of a path p with H number of hops is $C = \min_{i=1,\dots,H} C_i$.

Available bandwidth of a link i is maximum transmission rate of the link when cross traffic exists. Available bandwidth of a path p is the minimum available bandwidth among all links in the path $A = \min_{i=1,\dots,H} A_i$. Available bandwidth measurements ideally should not influence the throughput of the cross traffic of the path.

Term	**Definition**
Latency	A measure of how long it takes for a single packet to get from one machine to another machine
Transmission Delay D_T	The amount of time required to push all of the packet's bits into a link $D_T = L / C_i$ (*L*: packet size; C_i: capacity of the link)
Available bandwidth	The amount of bandwidth left-over after the cross traffic
Narrow link	Link with the lowest capacity along a path
Tight link	Link with the least available bandwidth in a path
Stream	A stream consists of *K* packets of size *L*, sent at a constant rate *R*
Fleet	N streams sent from sender to receiver
SLoPS (Self-Loading Periodic Streams)	A periodic stream sent to a path at a constant rate *R*
Packet Train	A sequence of manually configured data packets

Table 1: Selected Network Notions

What makes the measurement of available bandwidth hard is, first, that there is no consensus on how to precisely define it. Secondly, available bandwidth has a dynamic nature; meaning, it varies with time and exhibits high variability in a wide range of timescales.

Because unused bandwidth of a path changes with time, available bandwidth measurements average unused bandwidth over some time interval T [6]:

$$A_i(t,T) = \frac{1}{T} \int_{t}^{T+t} (C_i - \lambda_i(t))dt \qquad (1)$$

where $A_i(t,T)$ is the available bandwidth at link *i* at time *t*, C_i is the link's capacity, and λ_i is the link cross traffic.

To estimate the available bandwidth many methods have been developed [7-10]; from which, many use One-Way Delay *OWD* trend detection techniques.

One-Way Delay *OWD* is the time for a packet or periodic stream of k packets to travel from sender to receiver and is defined as:

$$OWD_i^k = d_i + D\tau + T \qquad (2)$$

where d_i^k is the queuing delay of packet k at link i ($d_i^k = \frac{q_i^k}{C_i}$; q_i^k: queue size at link i) and T is the propagation delay of the link . Therefore, end-to-end one-way delay of packet k for path p that consists of H links is [11]:

$$OWD_p^k = \sum_{i=1}^{H} d_i + D\tau + T = \sum_{i=1}^{H} \frac{q_i^k}{C_i} + \frac{L}{C_i} + T \qquad (3)$$

Based on the above definition, *OWD* variation is:

$$\Delta OWD = OWD_p^{k+1} - OWD_p^k = \sum_{i=1}^{H} d_i^{k+1} - d_i^k = \sum_{i=1}^{H} \Delta d_i^k \qquad (4)$$

Because the transmission delay term is canceled out, ΔOWD is independent of clock offset.

Based on the SLoPS methodology, when stream's rate is greater than available bandwidth, one-way delays of a periodic packet show an increasing trend. On the other hand, when the stream's rate is less than the available bandwidth, the stream packets will encounter equal *OWD*s.

To estimate available bandwidth of a path, many of the measurement tools use the trend of change in OWD. In the next section, we will look into some tools that use this method to measure available bandwidth.

Although knowledge of available bandwidth is the most important network path QoS characterization, other practical metrics have been reported in literature. For example, Chobanyan et al. in [12] and [13] use the metric of "crossing probability" to create random time-series statistical techniques for available bandwidth behavior. They claim that a single estimate of expected value of available bandwidth is not representative for describing the complex picture of available bandwidth dynamics. Therefore, they build a probabilistic framework for network path QoS characterization and use the notion of "crossing probability", defined as chances of a real-time task execution to fail because of not having enough available bandwidth in a path. Knowing crossing probability helps network applications response proactively to a changing network environment and improves resource planning. In their attempts, they provide parametric and non-parametric models for computation of cross probability.

In their parametric model, they use a statistical parametric model to describe available bandwidth and use that model for computation of cross probability. On the other hand, for their non-parametric method, they use a moving block bootstrap [14]; which considers dependence between observations of indicators and provides high confidence intervals around the non-parametric estimator of a crossing probability.

1.3. Available Bandwidth measurement Techniques and Methodologies

In measurement of available bandwidth, several methods and techniques have been proposed and various measurement tools have been developed and evaluated. One common technique to estimate available bandwidth is to use end-to-end probing. Some of the requirements for a reliable probing tool are:

- Fast estimation of available bandwidth within few RTTs
- Accurate measurements
- Non intrusive; low network overhead
- Being robust to multiple congested links
- Not require path topology such as link speed or capacity

Among available bandwidth measurement methods, probe gap and probe rate are the most commonly used methods. Therefore, in the following sections, these methods are described and a comparative analysis in terms of accuracy, intrusiveness, and response time of active probing tools are performed.

1.3.1. Probe Gap Model in Available Bandwidth Measurement

Probe Gap Model uses a single probing rate and it infers the available bandwidth from the time gap between the arrivals of two successive probes at the receiver [6]. In this method, two successive packets are sent with a certain time gap Δ_s. This packet pair gets to the receiver with time gap Δ_r. The difference between the two time gaps $\Delta_r - \Delta_s$is time required to transfer the cross traffic at the bottleneck. If no cross traffic exists, $\Delta_r = \Delta_s$ and available bandwidth during sending time is C. When cross traffic exists, a portion of the available bandwidth is utilized with the rate of $\frac{\Delta_r - \Delta_s}{\Delta_s}C$. Therefore, available bandwidth *AB* in presence of cross traffic is:

$$AB = C - \frac{\Delta_r - \Delta_s}{\Delta_s}C \qquad (5)$$

The weakness of the PGM method is its assumption of a known capacity *C* for the path and that the bottleneck has both the smallest capacity (narrow link) and the smallest available bandwidth (tight link). Moreover, in many cases Interrupt Coalescence[1] distorts PGM measurements as PGM fails to detect the

[1] Interrupt coalescence is a technique in which network interface card groups multiple packets received in a short time interval in a single interrupt and the sending host CPU learns about the departure of several packets through a single interrupt.

IC [15]. Delphi, IGI, and Spruce are examples of measurement tools based on PGM:

Delphi is based on direct probing technique [16]; therefore, every probing stream provides an estimation of the available bandwidth. This measurement tool uses a Multifractal Wavelet Model (MWM) to estimate the cross traffic rate [17]. MWM is based on the idea that internet traffic is a self-similar process which shows a fractal structure; meaning, short time patterns are similar to long term patterns. Delphi sends a stream of exponentially spaced packets, called chirp, and computes the queuing delay of each packet. In this tool, using the MWM, either the cross-traffic intensity is directly estimated or its parameters on a larger timescale are reconstructed based on the sampled traffic in small time intervals.

The main advantage of this model is its real-time adaptation to the current traffic. However, it requires some knowledge of the tight-link beforehand [18].

Initial Gap Increase (IGI) algorithm uses a sequence of packet trains to identify the average input probing gap for which the average output gap is equal to the average input probing gap [19]. At that point, called the turning point, the interference introduced by bottleneck cross-traffic is zero mean and the probing rate is the available bandwidth. This method completely neglects the interference of cross-traffic with the probe-gap at the nodes other than the bottleneck node over the entire path. For this reason, IGI fails to estimate available bandwidth accurately where many links are heavily congested.

Spruce is also based on direct probing technique [6]. Spruce sends probing pairs with a known intra-pair gap set to the bottleneck link transmission delay of the packet and inter-pair delay set to an exponentially distributed random variable to maintain the average probing rate below 5 percent of capacity (Figure 1).

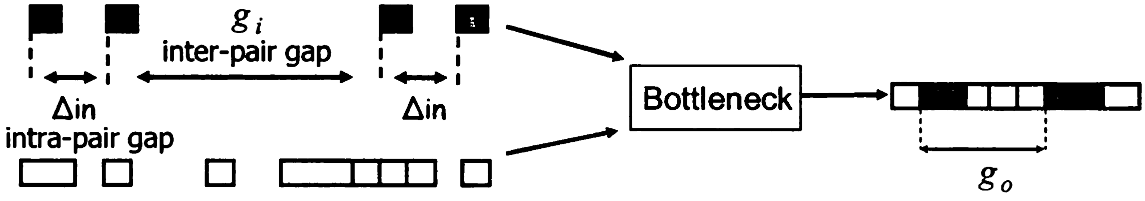

Figure 1: The Spruce Model for Estimation of Available Bandwidth

Each probing rate generates an estimate for available bandwidth using the following formula:

$$A_i = C \times \left(1 - \frac{g_0 - g_i}{g_i}\right) \qquad (6)$$

where g_i is the input inter-pair gap and g_o is the output inter-pair gap of the link. Finally, Spruce averages the last 100 samples of packet-pair available bandwidths to estimate the overall available bandwidth. Again, the main disadvantage of Spruce is its dependency on the knowledge or assumption of the capacity of the bottleneck link.

1.3.2. Probe Rate Model in Available Bandwidth Measurement

In the Probe Rate Model (PRM), sender transmits a periodic probing stream and the receiver observes the dispersion of packets.

Probing packets delay increases when the probing rate is higher than the available bandwidth in the path. In contrast, if sender sends probes to a receiver at a rate less than available bandwidth, probes will experience similar delays. PRM is based on the observation that the delays of successive probing packets increase when the probing rate exceeds the available bandwidth in the path. Therefore, available bandwidth can be measured by searching for the turning point at which the probe sending and receiving rates start matching.

PRM assumes FIFO queuing at all routers along the path, cross traffic follows a fluid model and average rates of cross traffic change slowly.

TOPP, PathChirp, and Pathload implement this technique for available bandwidth measurements [20].

Train of Packet Pairs (TOPP) [21], implemented in a tool called DietTopp [22], sends streams of packet pairs, uniformly increasing their input rates in each iteration. The rate is changed by modifying the input gap of each pair. The available bandwidth is estimated as the maximum input rate gets larger than the measured rate at the destination. Because TOPP uses packet pairs of different spacing well-separated in time, it does not make use of delay correlation information obtainable from packet trains with closely spaced packets. Therefore, TOPP is found to be very sensitive to the cross-traffic packet size and being very slow [23].

PathChirp uses probing trains with exponentially decreasing inter-packet spacing and calculates available bandwidth from the queuing delay signature of the arriving chirp [24]. Pathchirp is based on iterative probing. Instead of sending periodic packet streams, Pathchirp sends streams of exponentially spaced packets called chirps, so by rapidly increasing the probing rate within each chirp, it estimates the available bandwidth dynamically. The main advantage of this approach is to minimize the probing traffic load. Indeed, a single chirp is able to probe the network at different rates. Based on several comparative evaluation of available bandwidth measurement tools, although Pathchirp reacts properly to the cross-traffic variation but widely overestimates available bandwidth [20, 25].

Self-Loading Periodic Streams (SLoPS) [11], implemented in a tool called Pathload, sends streams of equally spaced packets. Instead of changing the input rate linearly or exponentially, it performs a binary search for the maximum feasible transmission rate which is the available bandwidth of the path.

One advantage of Pathload over other measurement tools is that it considers Interrupt Coalescence and Context Switch phenomena, which to the best knowledge of the author is the only working tool with this consideration. Therefore, the next chapter is fully devoted to description of Pathload, its advantages, and challenges in estimation of available bandwidth. Also, additional algorithms implemented in Pathload's successor Pathvar [27], which aims to measure the variability of the available bandwidth, are described.

CHAPTER 2

Available Bandwidth Estimation Using Pathload

Pathload is an active available bandwidth measurement tool developed by Jain and Dovrolis [11, 15, 26].

For this study our foucs is on Pathload because it does not considerably increase the network delays, losses and utilization. Moreover, Pathload was one of the first tools to consider the variability of the available bandwidth process. This is why it estimates a variation range of available bandwidth rather than a single estimate.

Pathload uses two types of connection between end-points: UDP and TCP. UDP is used for sending and receiving of periodic packet streams and TCP is used as a control channel to transfer messages regarding the characteristics of each stream such as end of measurement or abortion. In Pathload, sender sends several fleets to the receiver. Each fleet consists of 12 streams. Each stream consists of 100 packets. To avoid congestion, sender waits for the previous stream acknowledgment, before sending the next stream. After receiving each stream, receiver will eliminate the effect of IC/CS, as will be described. Then, OWD of remaining packets is determined. After receiving 12 streams, receiver determines the OWD of the fleet and send the information to sender for next fleet rate adjustment.

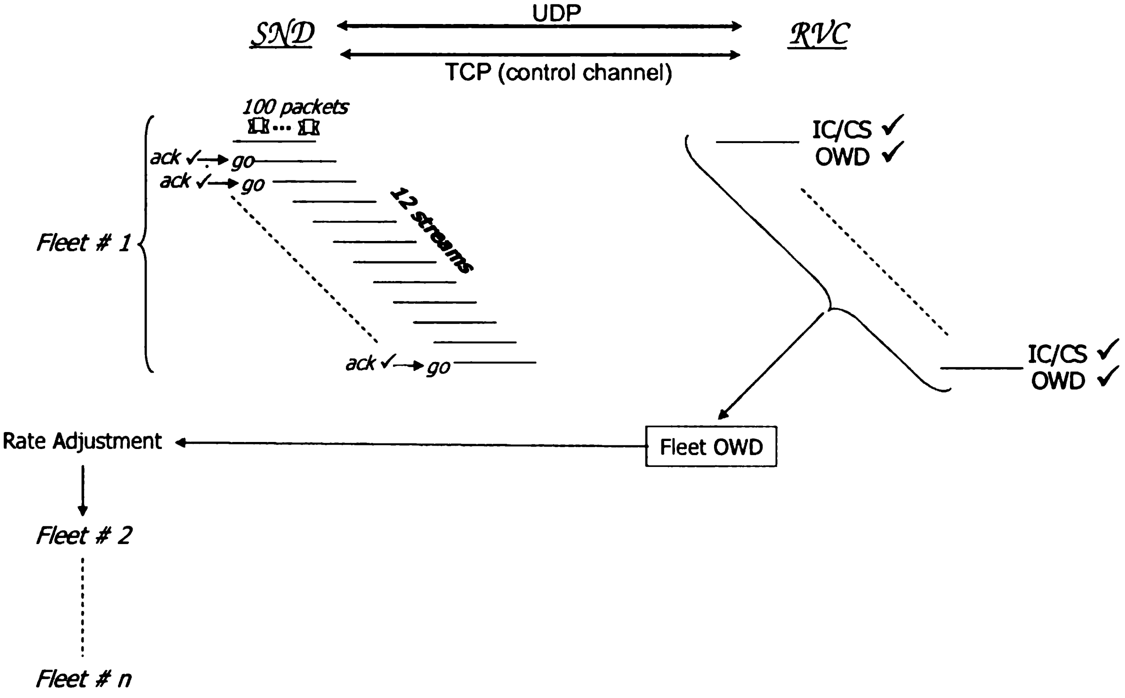

Figure 2: Pathload connection between sender and receiver

Pathload introduces a measurement technique based on Self-Loading Periodic Streams (SLoPS). Figure 3 illustrated the idea of SLoPS. Suppose sender *SND* transmits a periodic packet stream to receiver *RCV*. If the stream rate *R* is larger than available bandwidth *A*, an overload happens in the tight link of the path; and therefore, the queue of the tight link gradually increases. As a result, queing delay for packet *k+1* would be more than the previuos packet *k*. Based on equatiuon 4, ΔOWDs of the stream packets are expected to have an increasing trend ($d_i^k < d_i^{k+1}$). On the otherhand, if $R \le A$, with every new stream that is introduced to the link, the backlog of tightlink will not increase. Threfore ΔOWDs do not show an increasing trend.

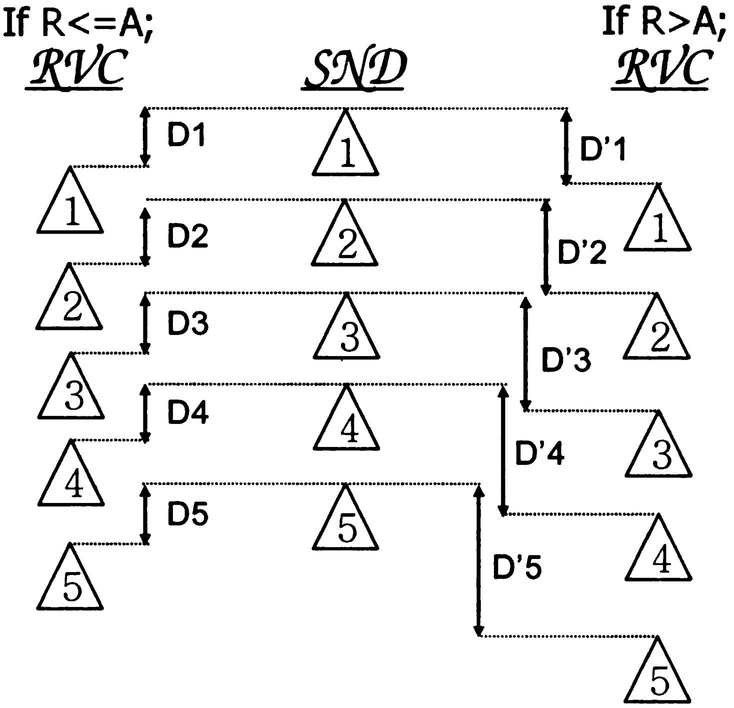

Figure 3: Illustration of Self-Loading Periodic Streams (SLoPS)

Because of significant variability of available bandwidth in different time scales, developers of Pathload improved their measurement tool to track the available bandwidth variation with relative errors up to 10-20%. For that, they developed and evaluated two estimation algorithms in an available measurement tool called Pathvar [27]. For very short time scales or in bottlenecks with limited flow multiplexing, where the available bandwidth distribution may not be Gaussian, the algorithm is iterative and non-parametric. For situations where the available bandwidth follows the Gaussian distribution, the algorithm is parametric; and therefore, much faster. Depending on the time scale, Pathvar invokes either the non-parametric or the parametric algorithm. When time scales is larger than 100 msec, Pathvar automatically switches from parametric to non-parametric algorithm.

In both algorithms, the sender generates N_1 probing streams of rate R_1 and N_2 probing streams of R_2 to track two low percentiles for available bandwidth variation. The receiver, on the other hand, calculates the fraction f_1 and f_2 of streams that are greater than the realization of the available bandwidth random variable in the corresponding time interval.

In the non-parametric approach, the information about each iteration f_n is sent back to the sender, which then sets the probing rate R_{n+1} accordingly. Adjustment of probing rate in non-parametric algorithm of Pathvar is very similar to Pathload, as it is described in 2.3.1.

In the parametric approach, using Cumulative Distribution Function, the expected value of an indicator variable *I(R)* is calculated. Finally, using an inverse mapping method, the corresponding probing rate is estimated.

Detection of OWD trend is the most important part of any algorithm based on SLoPS. OWD trend detection in Pathload is based on a statiscal analysis which is described next.

2.1. *OWD* Trend Detection in Pathload

OWD trend detection in Pathload consists of trend decision in a stream, and an overall decision in a fleet. Based on the trend decision in a fleet, Pathload uses a rate adjustment algorithm to adjust the rate of each successive fleet based on the trend determined by the previous fleet. In the following sections, these algorithms are described.

2.1.1. *OWD* Trend Detection in a stream

Pathload uses two statistical criteria to determine a streamΔOWD trend: Pa*irwise Comparison Test* (PCT) and Pa*irwise Difference Test* (PDT). For these criteria, k number ofOWD measurements in a stream is partitioned to $\Gamma = \sqrt{k}$ groups of consecutive measurements. Then, in each group the median of OWD measurements is computed asD_k. Metrics that *PCT* and *PDT* use are:

$$S_{PCT} = \frac{\sum_{k=2}^{\Gamma}(D_k > D_{k-1})}{\Gamma - 1} \qquad (7)$$

$$S_{PDT} = \frac{D_\Gamma - D_1}{\sum_{k=2}^{\Gamma}|D_k - D_{k-1}|} \qquad (8)$$

In the current release of Pathload, the *PCT* metric shows an increasing trend if$S_{PCT} > 0.66$, while the *PDT* shows an increasing trend if$S_{PDT} > 0.55$. Similarly, the *PCT* metric shows no trend, if$S_{PCT} < 0.54$; while the *PDT* shows a no trend, if$S_{PDT} < 0.45$.

The decision on ΔOWD trend is made based on both criteria as:

If $(S_{PCT} > 0.66$ and $S_{PDT} \geq 0.45)$ or $(S_{PDT} > 0.55$ and $S_{PCT} \geq 0.54)$ → I (increasing trend)

If $(S_{PCT} < 0.54$ and $S_{PDT} \leq 0.55)$ or $(S_{PDT} < 0.45$ and $S_{PCT} \leq 0.66)$ → N (no trend)

Otherwise → U (undetermined) → stream is discarded

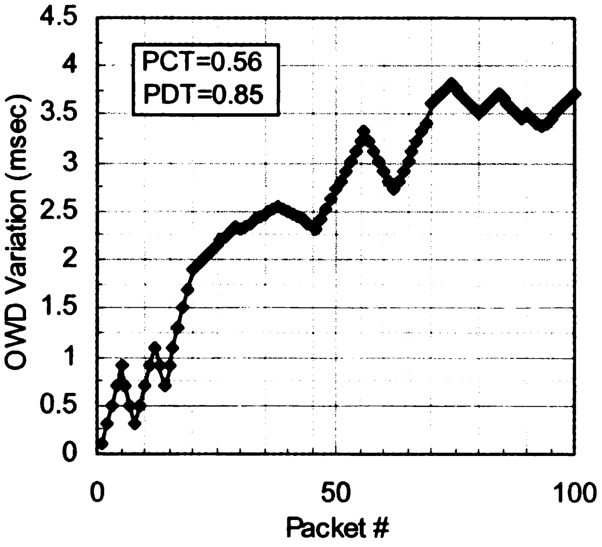

Figure 4: A Hypothetical *ΔOWD* vs. Packet number

Figure 4 shows a hypothetical case of ΔOWD vs. packet number. This graph obviously shows an overall increasing trend forΔOWD. Although *PCT* fails to recognize the increasing trend, *PDT* is greater than 0.54. Therefore, based on the above decision criteria, the overall trend is "I".

2.1.2. OWD Trend Detection in a fleet

Overall decision on ΔOWD trend of a fleet is made by the receiver. If a large fraction of the streams ($f > 0.7$) in a fleet show "I", the entire fleets is determined as an increasing fleet type. If a large fraction of the streams ($f > 0.7$) in a fleet show "N", the entire fleet is determined as no trend type. If a fleet does not have a majority of "I" or "N" ($f \leq 0.7$), the fleet is determined as grey-region type.

2.1.3. Rate Adjustment Algorithm

Pathload adjusts the rate of each successive fleet based on the trend determined by the previous fleet. The rate adjustment algorithm in Pathload results in three different orderings between *R* (rate) and *A* (available bandwidth): Increasing Trend I that is where $R > A$; Not Increasing Trend N, where $R < A$; and grey-region G; where there is no strict ordering between *R* and *A*.

The key state variables used in rate adjustment algorithm are:

- R^{max}: Lowest rate that has been shown to be higher than the available bandwidth up to the measured time interval.
- R^{min}: Highest rate that has been shown to be less than the available bandwidth up to the measured time interval.
- G^{min}: Lowest rate that has been shown to be in grey region.
- G^{max}: Highest rate that has been shown to be in grey region.

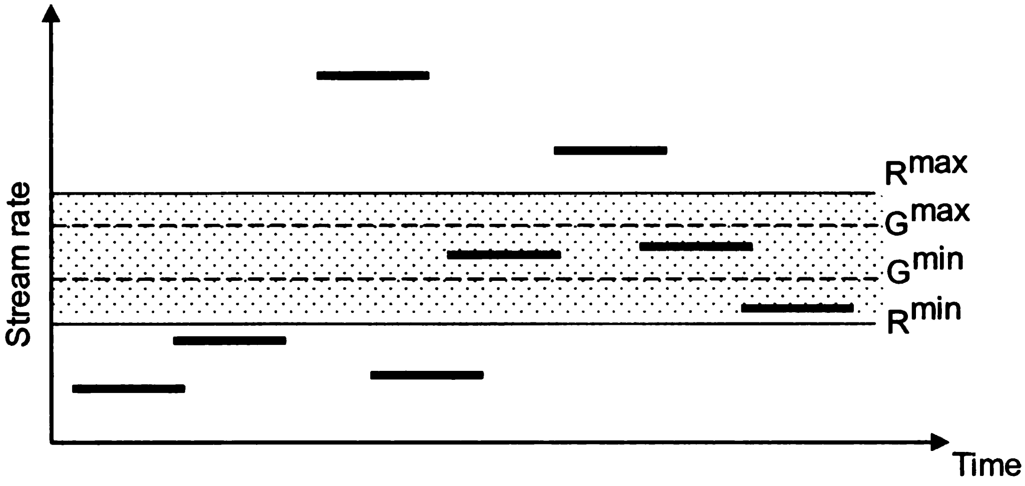

Figure 5: Rate Adjustment, Grey Region, and Available Bandwidth variability in Pathload

The iterative estimation algorithm of *Pathload* terminates when the rate of two successive fleets is less than a user-specified resolution; or, the available bandwidth varies in a grey-region, which is larger than the user-specified resolution.

2.2. Trend Detection Challenges in Pathload

Pathload has been validated by several research studies and, in comparison with other available bandwidth estimation tools; it has been shown to be the most accurate [7]. However, literature shows several challenges and pitfalls in available bandwidth estimation using Pathload:

1- Pathload is the most intrusive tool and, in some cases, can be very slow [20].

2- In the presence of IC/CS, Pathload throws away significant part of measurement information [28].

3- When the slow link is not in the middle of the path, trend detection in Pathload faces difficulties and measurements are more vulnerable to IC/CS effect [29-31].

4- PCT/PDT test in Pathload is not as accurate as what their authors claim. When the experimental setup is not as friendly as shown in [11], the PCT/PDT test shows significant inaccuracy [29, 30].

Although Pathload is not the ideal tool for available bandwidth estimation, it remains as one of the most powerful tools today. Therefore, addressing the challenges described above, can lead to a powerful upgrade to Pathload for available bandwidth estimation. In the next chapter, the author will describe a novel statistical integration into Pathload to address the above challenges.

CHAPTER 3

A Novel Approach for Efficient OWD Trend Detection

In this chapter, the proposed *OWD* trend detection in [29, 30] is reviewed. This method uses a simple algorithm of dealing with the IC/CS effect together with a trend detection algorithm based on a simple linear regression model and p-value reported by t-test. First, we review some related statistical notions for *OWD* trend detection; and then, we describe how this algorithm works and how it deals with IC effect. Finally, in the following chapter, we present the results of our integration of this algorithm into Pathload.

3.1. Statistical Notions for OWD Trend Detection

- Null hypothesis: Denoted by H_0, represents a theory that has been put forward, either because it is believed to be true or because it is to be used as a basis for argument, but has not been proved. In our trend detection approach, the null hypothesis is the assumption that a train of packets does not contain any *OWD* trend.

- Alternative hypothesis: Denoted by H_1, is the hypothesis that is contrary to the null hypothesis. Usually the alternative hypothesis is the hypothesis that supports your prediction. In our trend detection approach, the alternative hypothesis is the assumption that a train of packets have an increasing *OWD* trend.

- Hypothesis testing: Use of statistics to determine the probability that a given hypothesis is true. In our trend detection approach, we use hypothesis testing to determine if packets of a train contain any *OWD* trend.

- t-distribution: a special probability distribution in estimating the mean of a normally distributed population when the sample size is small or the population standard deviation is not known. In our *OWD* trend detection technique t-distribution is used because in some cases the number of sample points is as low as 4 packets. Please refer to appendix 1 for detailed discussion of t-distribution.

- Linear regression: An attempt to model the relationship between two variables by fitting a linear equation to observed data. One variable is called "response variable" (dependant) and the other one is called "predictor" (explanatory). In our *OWD* trend detection, one way delay of a packet is "response variable" and the packet index is the "predictor". Using linear regression method, best-fit linear line across data points of one-way packet delays and the packet index is found. (see appendix 2)

- T-test: Any hypothesis testing in which with the null hypothesis being true, the test statistics has a t-distribution. In our *OWD* trend detection, t-test is a test of whether the slope of a linear regression line differs significantly from 0.

- p-value: In hypothesis testing under the null hypothesis, p-value is the probability of obtaining a result at least as extreme as a given data point. The p-value provides an objective measure of the strength of the evidence which the data supplies in favor of the null hypothesis. A small p-value provides evidence against the null hypothesis. Therefore, in our *OWD* trend detection p-values smaller than 0.01 reject the null hypothesis and votes on existence of a trend for *OWD*. See Appendix 3 for calculation of p-value for a t-distributed dataset.

3.2. IC/CS Elimination Procedure

To decrease per-packet interrupt and Context Switch overhead, Network Interface Cards use Interrupt Coalescence (IC); which is generating a single interrupt for multiple packets sent or received in a short interval. IC; however, introduces queuing delays and alters interval time spacing of trains. In our way of handling IC/CS, we use the following idea:

At the presence of IC, successive OWDs in the same burst decreases [15]

Therefore, When *OWDs* of back-to-back receiving packets with similar packet pair delays is strictly linear with a negative slope; we put them in one group and discard all such packet but the very last packet.

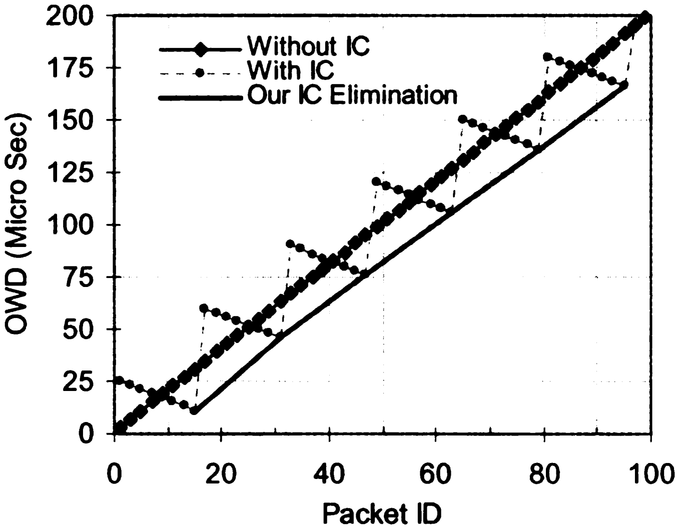

Figure 6: Our Interrupt Coalescence Elimination Technique

Figure 6 shows *OWD*'s in 100-packet train with and without IC. The red solid line shows how we eliminate the IC effect. After elimination, clearly the overall increasing trend has been preserved.

3.3. OWD Trend Detection using t-test

Here is a simple way of describing our incorporated t-test method in Pathload:

After TCP/UDP established the connection and the settings between the sender and receiver, sender starts sending 100 back-to-back packets in a train. Depending on network cross traffic, receiver receives a portion of these 100 packets. After receiving each packet, the receiver immediately checks for IC/CS, as described in the previous sub-chapter.

In our method, one needs to keep track of packet indices. If there is a sudden dramatic change in *OWD* of two back-to-back receiving packets or a gap more than 4 packet indexes, the packets so far received are put in a sub-train. Then, we plot the best fit least square line across data points. As explained in appendix 2, using simple linear regression we can find the slope coefficient of the line. It is know that the slope of linear regression follows a t-distribution. Here, the null hypothesis is that a train does not have any trend. For each sub-train of packets, we obtain the p-value of the t-test for judgment in trend presence.

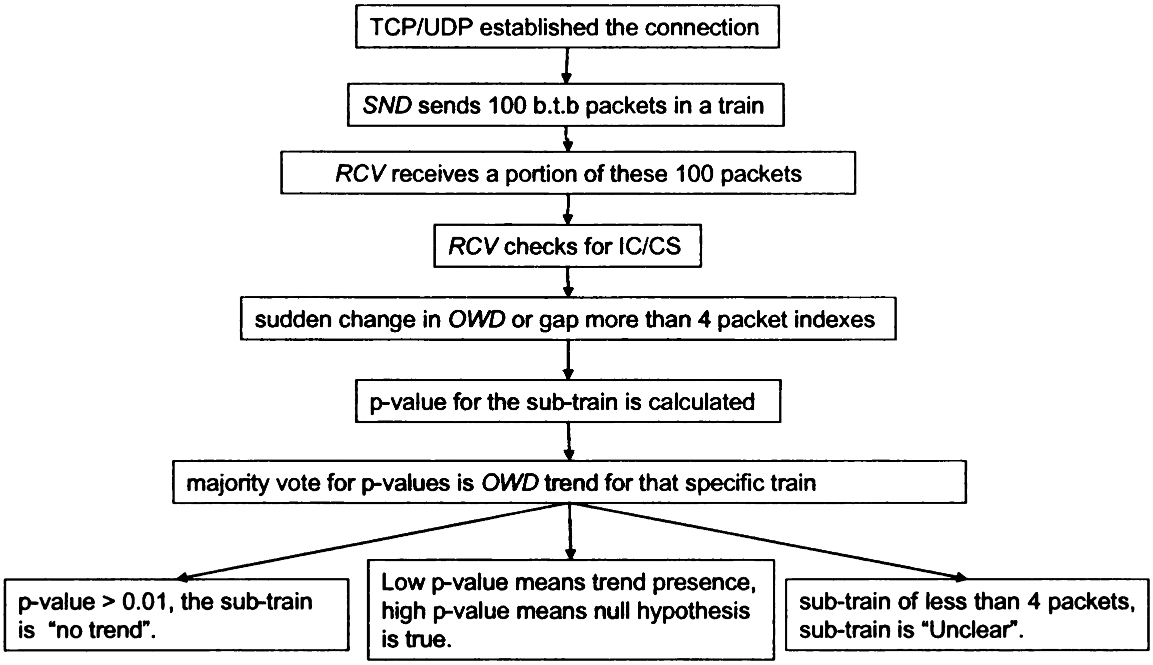
TCP/UDP established the connection
SND sends 100 b.t.b packets in a train
RCV receives a portion of these 100 packets
RCV checks for IC/CS
sudden change in OWD or gap more than 4 packet indexes
p-value for the sub-train is calculated
majority vote for p-values is OWD trend for that specific train
p-value > 0.01, the sub-train is "no trend".
Low p-value means trend presence, high p-value means null hypothesis is true.
sub-train of less than 4 packets, sub-train is "Unclear".

Figure 7: OWD Trend Detection Using T-test

At the end, the majority vote for the p-values of a train determines the *OWD* trend for that specific train. In our code, we consider the threshold of 0.01 for p-value; meaning, if the p-value is more than 0.01, the decision for that sub-train would be "no trend". If splitting of packets lead to a sub-train of less than 4 packets, the decision for that sub-train will be "Unclear".

3.3. Packet loss

To recover from instantaneous overflows, routers discard packets. When there are packet losses, Pathload throws away lots of information. Because our algorithm splits trains into sub-trains, it handles packet loss efficiently. Therefore, there we will much less reported unclear trains. This leads to much less number of required fleets, which ultimately results in less measurement time.

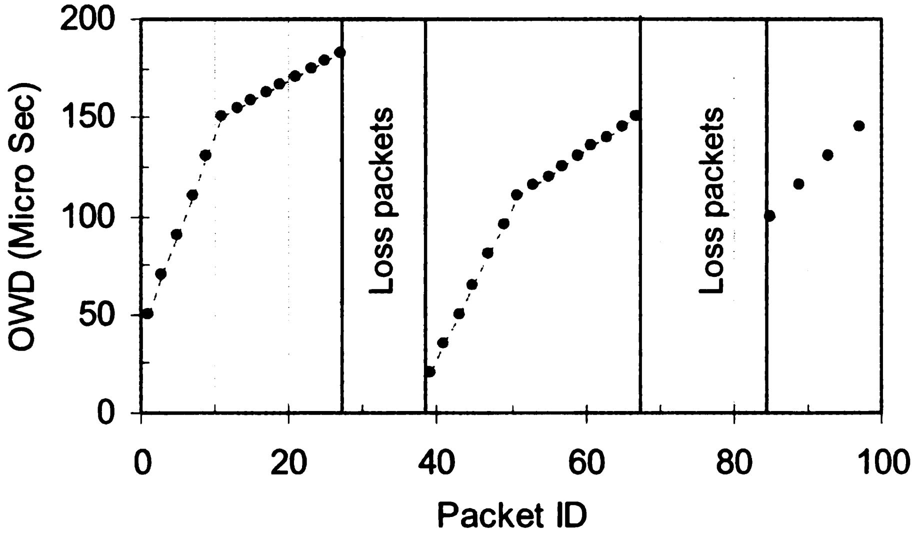

Figure 8: Packet loss in a fleet

CHAPTER 4

Experimental Evaluation

In this chapter, the efficiency of our *OWD* detection algorithm integrated in Pathload (we call it enhanced-Pathload) and Pathload's PCT/PDT test are compared.

4.1. Laboratory Network Testing

We experimented within our laboratory network which consists of two 100 Mbps capacity segments. The sender and receiver are placed at two ends of the segments with a switch connecting the segments. We wanted to have both Pathload's PCT/PDT test and our approach to have absolutely equal conditions.

We conducted a series of real measurement experiments in our laboratory network where we were able to evaluate the efficiency of both *OWD* measurement methods in the absence of any cross-traffic; therefore, possessed some verification information about available bandwidth behavior. Moreover, we could investigate the efficiency of both methods when CPU-load changes at the receiver side.

	Pathload PCT/PDT				Enhanced Pathload			
test #	# of Fleets	Latency [s]	Low-BW	High-BW	# of Fleets	Latency [s]	Low-BW	High-BW
1	6	11.87	95.60	97.10	6	12.44	95.40	95.40
2	14	18.69	96.40	96.40	6	11.77	95.60	95.60
3	17	22.39	96.20	96.20	6	11.77	95.60	95.60
4	13	18.54	96.80	96.80	6	11.78	95.70	95.70
5	11	16.61	98.76	109.20	6	11.78	95.60	95.60
6	13	18.53	18.53	96.60	6	11.77	95.60	97.10
7	20	25.27	25.27	96.60	6	11.77	95.60	97.10
8	5	11.00	98.67	109.60	6	11.77	95.60	97.10
9	12	17.56	96.20	96.20	6	11.77	95.60	97.10
10	6	11.77	95.60	97.10	6	11.77	94.00	95.60
11	40	44.62	98.60	99.24	6	11.77	95.60	95.60
12	14	119.50	96.40	96.40	6	11.77	95.60	95.60
13	6	11.77	95.60	97.10	6	11.92	96.00	96.00
14	62	65.99	100.43	109.20	7	12.75	96.20	96.20
15	6	11.95	96.00	97.50	4	9.86	96.90	96.90
Average	**16**	**28.40**	**87.00**	**99.42**	**6**	**11.76**	**95.64**	**96.15**

Table 2 : Pathload PCT/PDT and Enhanced Pathload Performance in Absence of Cross Traffic or a Significant CPU-Load

The result of Pathload PCT/PDT and enhanced Pathload in absence of cross traffic is tabulated in Table 2. For the enhanced Pathload, average number of fleets required to estimate the available bandwidth is significantly less than the Pathload PCT/PDT. Also, lower and higher bounds for bandwidth estimation are much closer in enhanced Pathload; whereas, Pathload PCT/PDT estimates a wider range for available bandwidth. Furthermore, the results of enhanced Pathload are much more consistent for the number of measurements conducted.

Table 3 shows the efficiency of Pathload PCT/PDT and enhanced Pathload when there is 7% or 14% CPU-load on the receiver computer. This configuration clearly shows how well enhanced Pathload perform when Pathload PCT/PDT is unable to estimate the available bandwidth.

7% RCV cpuload	Pathload PCT/PDT				Enhanced Pathload			
test #	# of Fleets	Latency	Low-BW	High-BW	# of Fleets	Latency	Low-BW	High-BW
1	7	11.05	59.4	60.9	7	9.09	96.0	96.0
2	2	7.16	> 93.8		7	7.77	95.6	95.6
3	3	8.90	48.7	97.0	6	6.79	95.6	95.6
4	1	5.66	> 0		6	6.90	95.6	95.6
5	5	10.85	99.4	109.2	15	16.26	95.6	95.6
6	2	6.17	> 0		5	5.83	93.8	93.8
7	13	16.91	59.7	61.2	6	6.94	96.1	96.1
8	16	19.77	24.8	49.6	6	6.96	94.2	95.8
9	Unable	U	U	U	6	6.82	95.2	95.2
10	1	5.53	> 0		6	6.83	96.0	96.0
11	7	12.81	97.1	98.2	6	6.78	91.0	97.1
12	7	12.81	97.1	99.4	6	6.80	95.6	95.6
13	7	11.22	17.0	11.2	6	6.90	95.6	95.6
14	3	6.86	> 0		9	9.93	82.4	97.5
15	Unable	U	U	U	12	13.37	96.0	96.2
2 X 7% RCV cpuload	Pathload PCT/PDT				Enhanced Pathload			
test #	# of Fleets	Latency	Low-BW	High-BW	# of Fleets	Latency	Low-BW	High-BW
1	Unable	U	U	U	13	14.12	80.4	86.5
2	5	10.95	48.80	97.50	16	17.83	94.4	99.8
3	9	14.88	95.60	95.60	6	8.94	94.0	94.0
4	5	11.12	97.40	97.40	15	17.93	95.1	96.4
5	Unable	U	U	U	13	13.26	81.3	81.5
6	Unable	U	U	U	11	11.51	30.9	31.0
7	Unable	U	U	U	5	7.92	94.0	94.0
8	7	12.82	23.80	28.50	6	8.92	96.1	96.1
9	Unable	U	U	U	6	6.94	95.6	95.6
10	2	8.02	97.00	97.00	10	11.29	89.3	89.3
11	9	15.49	14.14	14.14	9	24.20	>7.21	
12	Unable	U	U	U	6	6.97	95.6	95.6
13	3	9.90	97.40	97.40	21	23.59	86.1	96.4
14	8	24.61	16.89	17.21	8	8.82	95.6	95.6
15	9	13.62	17.54	17.71	15	16.15	90.4	90.5

Table 3 : Pathload PCT/PDT and Enhanced Pathload Performance with CPU-Load at the Receiver

According to these results, in the presence of CPU-load, Pathload most of the time either fails or reports a wrong available bandwidth (for our configuration in a 100 Mbs path with no cross traffic we expect an available bandwidth of 90 to 99

Mbs). On the other hand, our enhanced Pathload perform much reliable in the presence of CPU-load.

4.2. Long Distance Testing In Presence of Internet Traffic

For our long distance testing, Pathload PCT/PDT and enhanced Pathload were evaluated between two Linux machines in Southfield, MI and Tullahoma, TN. The tests were conducted in a week day between 9 A.M. and 6 P.M., when the Internet traffic is usually high. Each test was repeated 30 times for each of the Pathload *OWD* detection methods alternatively; meaning, immediately after the first test with Pathload PCT/PDT, the enhanced Pathload was tested. Then, the Pathload PCT/PDT, and so on. The reason for such sequential testing was to have almost similar Internet traffic for the two methods. If any of the methods could not determine the available bandwidth by 50 fleets, that measurement was aborted and categorized as "measurement with no result".

The results of our tests are tabulated in Table 4 and Table 5. The detailed results of the measurements with no CPU-load are tabulated in Appendix 4.

We started the tests with no CPU-load on the receiver machine and evaluated the Pathload *OWD* detection methods. As seen in the tables, the two methods very well predict the available bandwidth with 100% efficiency. Please note that in the presence of Internet traffic, the correct estimate for available bandwidth is not known. Therefore, here the efficiency is defined as number of attempts that the code could report a reasonable available bandwidth over the whole number of attempts.

	Pathload PCT/PDT		
CPU load on the receiver	% of measurements with a result	Avg # of Fleets when applicable	Avg Latency of measurement [s]
0	100%	6.5	39
20-25%	100%	6.8	41
45-50%	93%	7.6	45
70-75%	82%	7.2	44
95-100%	70%	7.8	51
0-100%	15%	34.7	84

Table 4 : Pathload PCT/PDT Performance in Long Distance in the Presence of Internet Traffic with varying CPU-Loads at the Receiver Machine

	Enhanced Pathload		
CPU load on the receiver	% of measurements with a result	Avg # of Fleets when applicable	Avg Latency of measurement [s]
0	100%	6.0	39
20-25%	100%	5.8	39
45-50%	100%	6.2	41
70-75%	100%	5.8	40
95-100%	100%	6.1	42
0-100%	100%	17.4	55

Table 5 : Enhanced Pathload Performance in Long Distance in the Presence of Internet Traffic with varying CPU-Loads at the Receiver Machine

We continued our tests by increasing the CPU-load on the receiver machine. As expected, Pathload PCT/PDT started having difficulties in determining the available bandwidth with more load on the CPU. The efficiency of Pathload PCT/PDT decreased significantly when the receiver machine had more than 95% CPU-load. Also, for those times Pathload PCT/PDT was able to determine the available bandwidth, it required more number of fleets and consequently more measurement time. On the other hand, for the same conditions, the enhanced Pathload performed very well to the point we can claim it was independent of the CPU-load. The latency of the measurements and the number of fleets required for enhanced Pathload stayed almost the same across different CPU-loads on the receiver machine.

At the end, we evaluated the performance of Pathload PCT/PDT and enhanced Pathload when the CPU-load changed frequently during a single measurement. To do that, we increased the CPU-load on the receiver machine from 0 to 100% every 10 seconds. In this condition, performance of Pathload PCT/PDT was very poor and could only determine the available bandwidth for 15% of measurements.

On the other hand, the enhanced Pathload could very well adapt itself with the changing conditions on the CPU-load. Although, the latency of the measurements and number of fleets required increased, it was able to determine the available bandwidth 100% of the times.

CHAPTER 5

Conclusions and Future work

1.1. Conclusions

In thesis we integrate a previously proposed statistical method in One Way Delay (*OWD*) detection into an open source and widely accepted available bandwidth measurement tool, Pathload.

Pathload introduces a technique based on Self Loading Periodic Stream (SLoPS). The algorithm is based on sending a stream of packets to the receiver which causes queuing delays. At some point, congestion happens; and therefore, the OWD increases. The process of estimating the available bandwidth is then transformed into identifying the turning point at which the *OWD* sees an increasing trend.

Based on literature and our investigations, Pathload PCT/PDT method for *OWD* detection has some shortcomings, especially in the presence of CPU-load.

Also, interrupt coalescence (IC) and context switch (CS) in Pathload is far from ideal.

In our enhanced Pathload algorithm, the process of *OWD* is handled by a simple linear regression model and p-value reported by t-test, as proposed in [29, 30]. This algorithm perform better than the algorithm used in Pathload because

1- Pathload throws away lots of packets because it only considers the median of each group of packets. However, the replaced algorithm throws away packets only when there is a decreasing trend.

2- Pathload uses empirical thresholds for PCT/PDT tests with no statistical motivation. Therefore, there is no reasoning to perform well for an arbitrary network path. The replaced algorithm, however, set a threshold in the p-value space based on a statistical motivation. In this algorithm, for each train of packets we obtain p-value of the t-test as our final quantity for judging trend presence.

3- Interrupt coalescence elimination procedure in the replaced algorithm is more robust and is done in a much simpler manner.

We conducted a series of real measurement experiments in our laboratory network and in the presence of Internet traffic with different CPU-loads on the receiver machine. In our laboratory network, we created a route with no cross traffic or CPU load on the measurement units. Our measurement, in this setup, shows how well our enhanced Pathload performs compare to Pathload PCT/PDT. Similarly, In the presence of Internet traffic, our integrated algorithm performs better especially when the CPU-load on the receiver side is heavy or changes frequently.

1.2. Future Work

In the future work of this research, the integrated *OWD* trend detection algorithm can be further refined and verified. Moreover, it is promising to explore the performance of the described *OWD* algorithm into other methods of available bandwidth measurement.

Also, performance of the enhanced Pathload, introduced in this thesis, can be compared among broader range of available methods for available bandwidth measurement to establish it creditability among other methods.

Because we believe Pathload is among the best available measurement tools today, we recommend future researches address other challenges that Pathload faces in available bandwidth measurement. For example, trend detection in Pathload still faces difficulties when the slow link is not in the middle of a path. Also, because Pathvar considers variability of available bandwidth in a greater detail, we recommend implementation of the *OWD* trend detection technique used in this thesis into Pathvar and computation of "crossing probability", as recommended in [12] and [13].

We conclude this dissertation with a hope that our newly developed package improves the estimation of available bandwidth; which consequently, results in better QoS for large class of data-intensive applications.

Also, we hope this thesis motivates future efforts to use/integrate the OWD trend detection used in this thesis.

APPENDIX 1

T-distribution

Suppose $x_1, x_2, \ldots, x_n$ are normally distributed variables with expected value of μ and variance of σ^2. For this dataset sample mean can be defined as:

$$\bar{x} = \left(\frac{x_1 + x_2 + \ldots + x_n}{n} \right)$$

Therefore, sample variance can be defined as:

$$S^2 = \frac{1}{n-1} \sum_{i=1}^{n} \left(x_i - \bar{x} \right)^2$$

Let's note z as:

$$Z = \frac{\bar{x} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

where n is number of samples. It can be shown that z is normally distributed with mean 0 and variance 1.

Gosset introduced a similar quantity as:

$$T = \frac{\bar{x} - \mu}{\frac{S_n}{\sqrt{DF}}}$$

In T exact standard deviation is replaced by the random variable S_n . Gosset showed that T has the probability density function of

$$f(t) = \frac{\Gamma\left(\frac{DF+1}{2}\right)}{\sqrt{DF\pi}\Gamma\left(\frac{n}{2}\right)}\left(1+\frac{t^2}{DF}\right)^{-\frac{DF+1}{2}}$$

where DF is degrees of freedom defined as $DF = n-1$ and Γ is the gamma function defined as:

$$\Gamma(x) = \int_0^\infty x^{h-1}e^{-h}dh$$

The distribution of T is now called the *t*-distribution. The lack of dependence on μ and σ is what makes the t-distribution important.

APPENDIX 2

Slope and Y-intercept of a Linear Regression Line

Suppose that we are given a sequence of observations:

$$(x_1, y_1),\ \ldots,\ (x_n, y_n)$$

To predict variable y as a function of x, we assume that y can be approximated by a linear function of x: $f(x) = \beta_0 + \beta_1 x$

To find the line that fit the data the best, we use the method of least squared. Therefore, the loss function for our approximation can be written as:

$$L = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x))^2$$

To get the best fit to the dataset L needs to be minimized over β_0 and β_1:

$$\frac{\partial L}{\partial \beta_0} = -\sum_{i=1}^{n} 2(y_i - (\beta_0 + \beta_1 x_i)) = 0$$

$$\frac{\partial L}{\partial \beta_1} = -\sum_{i=1}^{n} 2(y_i - (\beta_0 + \beta_1 x_i))x_i = 0$$

Solving for β_0 and β_1, we get:

$$\beta_1 = \frac{n\sum xy - \sum x \sum y}{n\sum x^2 - \left(\sum x\right)^2}$$

$$\beta_0 = \frac{\sum y}{n} - \beta_1 \times \frac{\sum x}{n}$$

APPENDIX 3

P-value Calculation in a T-distributed Dataset

Here are the steps to calculate the P-Value:

1- Using OWDs of packets in a sub-train, calculate standard error of the slope SE using the following equation:

$$SE = \sqrt{\frac{\sum \frac{(y_i - (\beta_1 x_i + \beta_0))^2}{n-2}}{\sum \left(x_i - \overline{x_i}\right)^2}}$$

where y_i is the value of the dependent variable for observation i, β_0 and β_1 are y-intercept and slope of regression line to the data as calculated in appendix 2, x_i is the observed value of the independent variable for observation i, $\overline{x_i}$ is the mean of the independent variable, and n is the number of observations.

3- Calculate degrees of freedom DF using $DF = n - 1$; where n is the number of observations.

4- Calculate t-score t of the test statistics using $t = \frac{\beta_1}{SE}$; where β_1 is the slope of regression line and SE is the standard error of the slope.

5- Calculate $tcdf(t, n)$ from the following equation:

$$tcdf(t,v) = \int_{-\infty}^{t} \frac{\Gamma\left(\frac{n+1}{2}\right)}{\Gamma\left(\frac{n}{2}\right)} \frac{1}{\sqrt{n\pi}} \frac{1}{\left(1+\frac{x^2}{n}\right)^{\frac{n+1}{2}}} dx$$

where, t is the t-score of the dataset and Γ is the gamma function defined as:

$$\Gamma(x) = \int_{0}^{\infty} x^{h-1} e^{-h} dh$$

6- Calculate p-value using: $p = 1 - tcdf(t, n)$

APPENDIX 4

Measurements Between TN and MI

Following table is sample results of the Pathload PCT/PDT and Enhanced Pathload measurements when there is no load on the receiver machine CPU:

[illegible]	Pathload PCT/PDT				Enhanced Pathload			
test #	# of Fleets	Latency	Low-BW	High-BW	# of Fleets	Latency	Low-BW	High-BW
1	6	34.65	4.81	4.81	6	36.52	4.62	4.70
2	6	32.88	4.88	4.88	6	35.87	4.53	4.62
3	6	35.75	4.80	4.89	6	35.98	4.66	4.75
4	7	40.73	4.76	4.74	6	35.39	4.68	4.76
5	6	25.27	4.32	4.42	6	35.94	4.69	4.77
6	5	30.39	4.85	4.85	6	36.18	4.00	4.08
7	5	30.81	4.97	4.97	6	35.35	4.71	4.80
8	7	39.80	4.88	4.88	6	35.83	4.38	4.47
9	6	35.31	4.89	4.89	6	36.57	4.69	4.69
10	6	34.81	4.86	4.96	6	36.58	4.69	4.77
11	6	35.74	4.80	4.89	6	35.32	4.62	4.71
12	6	39.67	4.89	4.89	6	37.47	4.33	4.42
13	5	29.69	4.93	4.93	6	35.32	4.62	4.71
14	5	30.46	4.84	4.84	6	35.57	4.58	4.68
15	8	47.41	4.29	4.84	5	27.28	3.41	3.13
16	6	23.08	4.13	4.19	6	54.80	2.04	2.04
17	6	60.98	2.63	2.73	6	39.17	3.43	3.50
18	8	45.86	3.17	3.21	6	32.92	4.64	4.64
19	8	48.40	4.19	4.71	6	41.90	3.53	3.53
20	7	38.69	4.53	4.71	6	37.29	4.12	4.20
21	5	30.49	4.84	4.84	6	36.70	4.68	4.68
22	8	42.16	4.31	4.50	6	62.47	2.75	2.84
23	6	35.75	4.80	4.80	5	31.62	4.65	4.65
24	8	48.63	4.18	4.70	7	48.65	3.84	3.93

TN to MI -No cpuload	Pathload PCT/PDT				Enhanced Pathload			
25	7	40.24	4.34	4.86	7	48.93	3.17	3.52
26	8	41.21	4.53	4.62	6	49.90	3.10	3.22
27	6	35.34	4.88	4.88	6	35.34	4.62	4.71
28	6	35.61	4.85	4.85	6	31.67	4.49	4.58
29	8	47.69	4.18	4.88	6	40.03	4.10	4.18
30	7	68.17	2.68	2.74	6	35.66	4.66	4.75
Average	**6.47**	**38.86**	**4.47**	**4.60**	**6.00**	**38.61**	**4.17**	**4.23**

As explained in this thesis, when there is no CPU-load on the receiver machine, both Pathload and enhanced Pathload perform well. However, by increasing the CPU-load on the receiver enhanced Pathload outperforms Pathload (refer to the main text).

REFERENCES

[1] Cen, Z, Goradia, A, Mutka, M, Xi, N., 2005, "Improving the Operation Efficiency of Supermedia Enhanced Internet Based Teleoperation via an Overlay Network", Proceedings of 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain.

[2] Cen, Z, Mutka, M, 2006, "QoS Provision for Remote Sensing and Control in Heterogeneous Environments", Proceedings of the Third IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2006), Vancouver Canada.

[3] Cen, Z, Liu, Y., A, Mutka, M, Goradia, A., Xi, N., 2005, "QoS Management of Supermedia Enhanced Teleoperation via Overlay Networks", Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), Edmonton, Alberta, Canada.

[4] Cen, Z, Mutka, M, Zhu, D., Xi, N., Xi, N., 2005, "Improved Transport Service for Remote Sensing and Control over Wireless Networks", Proceedings of 2nd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2005), Washington, DC, USA.

[5] Guojun J., 2003, "Available Bandwidth Measurement and Sampling ", ISMA 2003 Bandwidth Estimation Workshop.

[6] Strauss J., Katabi D., Kaashoek F., 2003, "A Measurement Study of Available Bandwidth Estimation Tools", IMC'03 Florida USA, ACM 1581137737/ 03/0010.

[7] ThanhMan, C, Hasegawa, G, Murata, M, 2003, "A New Available Bandwidth Measurement Technique for Service Overlay Networks", 6th IFIP/IEEE International Conference on Management of Multimedia Networks and Services.

[8] Melander, M, Björkman, M, Gunningberg, P, 2002, "Regression-based Available Bandwidth Measurements", International Symposium on Performance Evaluation of Computer and Telecommunication Systems.

[9] Navratil, J, Cottrell, L, 2003, "ABwE: A Practical Approach to Available Bandwidth Estimation", Passive and Active Measurements Workshop.

[10] Nishikawa, H, Takuya, A, Takashi, T, 2004, "ABdis: Approach to Estimate Available Bandwidth Distribution Using a Multi-rate Probe", International Conference on Communication and Broadband Networking (ICBN04).

[11] Jain P., Drovolis C., 2003, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput ", ACM/IEEE Transactions on Networking.

[12] Chobanyan, A, Mutka, M., Mandrekar,V., and Xi, N., 2005, "Modeling Available Bandwidth for an Efficient QoS Characterization of a Network Path" 4th IFIP-TC6 Networking Conference, Networking 2005.

[13] Chobanyan, A, Mutka, M., Levental, S., and Xi, N., 2006, "Behavior of available end-to-end bandwidth: Non-parametric approach" International Conference on Quantitative Evaluation of SysTems (QEST).

[14] Liu, R., Singh, K., 1992, "Moving Blocks Jacknife and Bootstrap Capture Weak Dependence" Exploring the Limits of Bootstrap edited by LePage and Billiard. JohnWiley Sons, New York.

[15] Jain P., Drovolis C., 2004, "Effects of Interrupt Coalescence on Network Measurements ", Passive and Active Measurements Workshop.

[16] Ribeiro V., Coates M., et al., 2000, "Multifractal cross-traffic estimation", In Processing of ITC Specialist Seminar on IP Traffic Measurement.

[17] Ribeiro V., Riedi R., et al., 2003, "A Multifractal Wavelet Model with Application to Network Traffic", IEEE Transactions on Information Theory, 45(4).

[18] Jain P., Drovolis C., 2004, "Ten Fallacies and Pitfalls on End-to-End Available Bandwidth Estimation", Internet Measurement Conference, Italy.

[19] Hu, N, Steenkiste, P, 2003, "Evaluation and Characterization of Available Bandwidth Probing Techniques" IEEE JSAC Special Issue in Internet Measurement, Mapping, and Modeling, Vol. 21(6).

[20] Ait Ali A., Michaut F., Lepage F., 2006, "End-to-End Available Bandwidth Measurement Tools: A Comparative Evaluation of Performances ", In 4th International Workshop on Internet Performance, Austria.

[21] Melander, M, Björkman, M, 2004, "A new end-to-end probing and analysis method for estimating bandwidth bottlenecks" Global Internet Symposium, 2000.

[22] Johnsson, A, Melander, M, Björkman, M., 2005, "DietTopp: A First Implementation and Evaluation of a Simplified Bandwidth Measurement Method" Second Swedish National Computer Networking Workshop, pages 5, Karlstad.

[23] Castellanos, C, Villa, D, et al., 2006, "Comparison of Available Bandwidth Estimation Techniques in Packet-Switched Mobile Networks" 2006 IEEE 17th International Symposium, ISBN: 1-4244-0330-8.

[24] Ribeiro V., Riedi R., et al., 2003, "Pathchip: Efficient Available Bandwidth Estimation", PAM 2003.

[25] Montesino-Pouzols F., 2004, "Comparative Analysis of Active Bandwidth Estimation Tools", PAM 2004.

[26] Jain P., Drovolis C., 2002, "Pathload: Measurement Tool for End-to-end Available Bandwidth ", Proceedings of Passive and Active Measurement Workshop, Fort Collins, CO.

[27] Jain P., Drovolis C., 2005, "End-to-end Estimation of the Available Bandwidth Variation Range", Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, Alberta Canada.

[28] Shriram A., Murray Y., et al., 2005, "Comparison of Public End-to-end Bandwidth Measurements", PAM 2005.

[29] Chobanyan, A, 2007, "Estimating Available Bandwidth for Real Time Supermedia Applications", Ph.D. Thesis Michigan State University.

[30] Chobanyan, A, Mutka, M, Zhiwei, C., and Xi, N., 2005, "One Way Delay Trend Detection for Available Bandwidth Measurement" IEEE GlobCom 2005.

[31] Cen, Z, 2007, "Supporting Remote Sensing and Control Over IP Networks", Ph.D. Thesis Michigan State University.

[32] Prasad, R, Jain, M, Drovolis, C, 2004, "Effects of Interrupt Coalescence on Network Measurements", Ph.D. Passive and Active Measurements Workshop.