KUMAR, SURENDER, M.S. eChirp: Measuring Available Bandwidth for the Internet Using Multiple Chirp Packet Trains. (2008)
Directed by Dr. Shanmugathasan (Shan) Suthaharan. 76 pp.

Measuring available bandwidth over a network path in the Internet is a challenging research problem. In this thesis we have studied this problem and developed a new technique called “eChirp”. First, the effectiveness of pathChirp [1] is studied in terms of model performance of chirp packet train structure, actual bandwidth, queuing delay and excursion segmentation. Then we remodeled the chirp train structure. The eChirp can measure the available bandwidth over a network path efficiently and accurately with heavy and light load links. To measure the available bandwidth, the packet probing rate configuration used in pathChirp technique is modified by changing its chirp train structure. The modified structure uses multiple chirp trains (three trains) that provides better probing rate configuration and ultimately gives better bandwidth measurement. Per-packet available bandwidth is calculated using weighted average of per-packet bandwidth of three trains. We also determined the bounds of probing rate parameter which was questionable in pathChirp and affects the available bandwidth measurement accuracy. The eChirp technique has been experimented with numerous network path topologies with low and high link loads with CBR cross-traffic conditions using NS-2 simulated network and results are compared with most recent pathChirp technique. Simulation results show that the proposed eChirp technique is better than pathChirp scheme in terms of estimating available bandwidth.

ECHIRP: MEASURING AVAILABLE BANDWIDTH FOR THE INTERNET USING MULTIPLE CHIRP PACKET TRAINS

by

Surender Kumar

A Thesis Submitted to
the Faculty of The Graduate School at
The University of North Carolina at Greensboro
in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Greensboro
2008

Approved by

____
Committee Chair

To my family for always supporting and encouraging me

APPROVAL PAGE

This thesis has been approved by the following committee of the Faculty of The Graduate School at The University of North Carolina at Greensboro.

Committee Chair ________________________________

Committee Members ________________________________

____

____
Date of Acceptance by Committee

____
Date of Final Oral Examination

ACKNOWLEDGEMENTS

First and foremost I want to express my gratitude to my professor, Dr. Shanmugathasan (Shan) Suthaharan, for providing valuable guidance, encouragement, many inspiring discussions and debates and always being there when I needed him. It has been a pleasure to work with a person who is enthusiastic, has provocative ideas and great attitude.

Finally, I would like to thank my family for supporting me all the way, especially my wife, Bhagyashree. It is her love, encouragement, and time sacrifice that enabled me to focus on my thesis.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

The knowledge of available bandwidth over a network path in the internet is required by many applications including server selection, end-to-end admission control, peer-to-peer and Internet Service Providers (ISP) network engineering. This knowledge improves the Quality of Service (QoS) of these applications. In order to measure the available bandwidth, the end hosts should acquire network information, at intermediate systems. End hosts are usually not aware of such information. Over the last decade numerous techniques have been proposed by researchers to measure the available bandwidth over a network path. Most of these techniques assume either negligible or fluid cross-traffic in the network for analysis and measurement. Internet is very volatile in nature with so many characteristics and these assumptions are not always correct resulting none of these techniques proved to be accurate to date.

In this thesis, we developed a new probing technique called "eChirp" that is based on most recent tool pathChirp [1]. The eChirp can measure the available bandwidth over a network path efficiently and accurately with heavy and light load links. Unique to eChirp is its packet train methodology; we modify pathChirp's chirp train structure. In the modified structure, the rate of the odd inter-chirp packet will be the same as the rate of previous even inter-chip packet. Additionally, rate of inter-chirp packets will be increased exponentially with even power rather than both even and odd power as done in pathChirp method. We also use two sub eChirp packet trains for getting more granule information of the network path that gives more accurate measurement. The term "Available Bandwidth" has many definitions based on the context that it used, such as link available bandwidth, path available bandwidth, bottleneck; we start this chapter by defining the available bandwidth that we used in this thesis followed by the importance of accurate available bandwidth measurement in above stated applications. The last part of this chapter explains the organization of this thesis.

I. Available Bandwidth

Available bandwidth can be defined as the minimum remaining bandwidth along the best path between the source and destination that can be used by a new flow without disturbing the transmission of other flows on the path. That is, available bandwidth can be calculated as path capacity minus path load. Figure 1(a) and 1(b) shows the common network model used in today's internet and graphical definition of available bandwidth respectively.

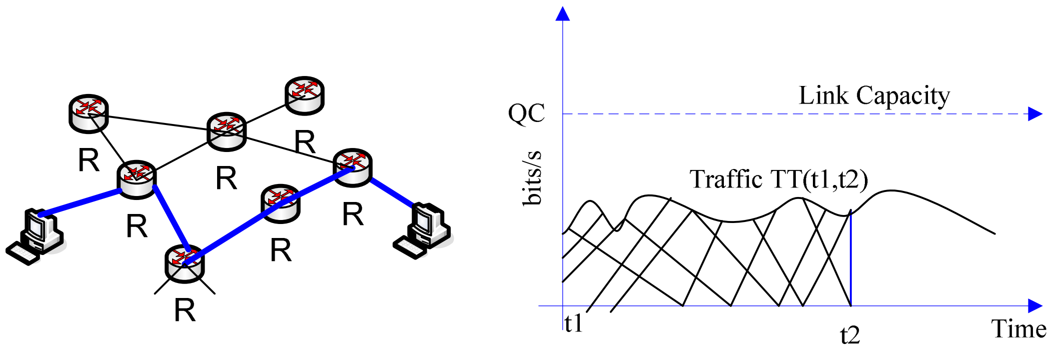


Figure 1: (a) Common Network Model, (b) Available Bandwidth.

In technical terms the actual available bandwidth for every router node with respect to router's output queue capacity, total traffic, propagation delay and packet service time by referring Figure 1(b) as follows:

$$AB[t_1, t_2] = \min_i \left(QC_i - \frac{TT_i[t_1 + td_i, t_2 + td_i]}{t_2 - t_1} \right)$$

where $AB[t1,t2]$ is the available bandwidth of the path in time interval $[t1,t2]$, QC_i is the capacity of the router link i that is determined by node interface, TT_i is the total traffic (other than measurement probe traffic) entering between any time interval at link i and td_i is the processing delay at link i.

Since the term "Available bandwidth" used differently by different author, Figure 2 explains the simplified definition of available bandwidth without cross-traffic conditions, where a host (client) machine connected to the server through three intermediate routers R1, R2 and R3 with 10Mbps, 256Kbps, 512Kbps, 100Mbps respectively. Packets usually flow through the best path from source to destination. The maximum bandwidth of that path is equal to the bandwidth of the link with the smallest bandwidth. On this path, the maximum available bandwidth is 256Kbps because that is the bandwidth of the link with the smallest bandwidth on that path.

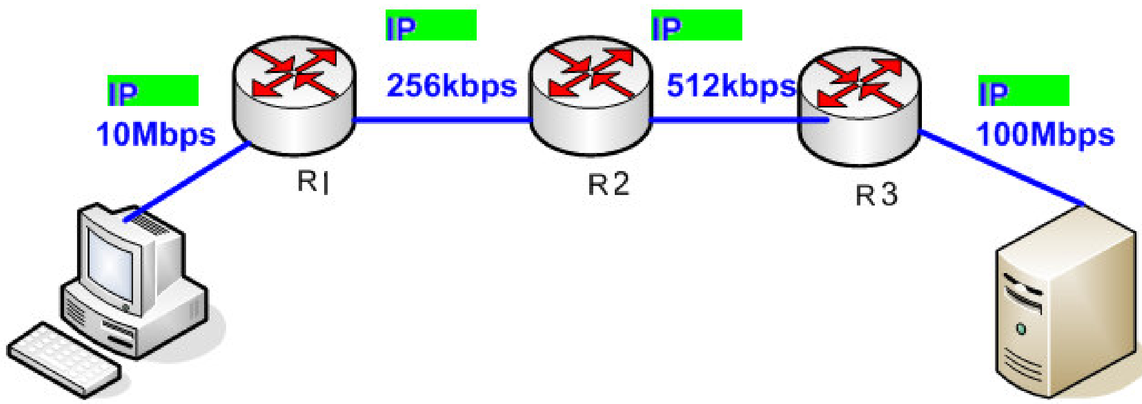


Figure 2: Available bandwidth defined

$$B.W._{max} = \min(10Mb,256kb,512kb,100Mb) = 256kbps$$

$$B.W._{avail} = B.W._{max}/flows$$

Maximum available bandwidth equals the bandwidth of the weakest link.

Multiple flows are competing for the same bandwidth, resulting in much less bandwidth being available to one single application.

II. Importance in Applications

There are many applications that require the estimation of available bandwidth. For example server selection (where network clients or distributed applications, which require service from replicated servers, can use this information to choose the best server or proxy. A server with the highest available bandwidth path might have the shortest response time.), peer-to-peer applications and ISP network engineering are some of the important applications. Figure 3 shows the typical server selection application where an end host user wants to download a file from an internet website. Nowadays on the internet it is possible that there are multiple websites that provide the same file and the end user needs to select one of them. Obviously, the server that has the highest downloading speed is the best choice, and to know that, the end user must have the end-to-end available bandwidth information from each of these websites.

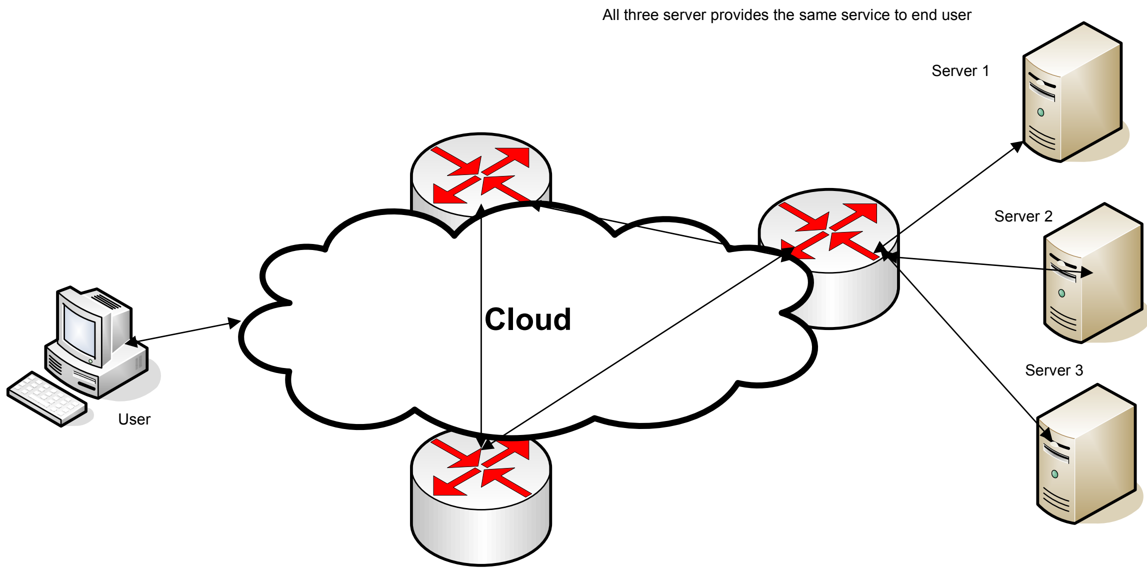


Figure 3 : Application of Available Bandwidth estimation: Server Selection

Another important use of available bandwidth measurement information is in congestion control mechanism that is used by many internet applications and protocols. For example, TCP-based congestion control mechanism. This control mechanism adjust the packet transmission rate at the end-hosts by the congestion window according to the current traffic situation, in which the flow transmission rate increase step-by-step (during slow start phase it increase exponentially and during congestion avoidance phase increases linearly), until higher rate is reached, at that point the network is congested and all the incoming packets are dropped. Then the control-mechanism decreases the transmission flow-rate by half and again starts probing for a higher rate. The knowledge of available bandwidth can be very useful in this congestion control mechanism. The control-mechanism can use this available bandwidth information to set upper bound on the transmission rate that can avoid packet drop during the congestion avoidance phase.

Also as mentioned in [2] the performance of several applications can be improved with the knowledge of available bandwidth of an end-to-end network path. For example, initial bit-rate selection for video applications. The knowledge of available bandwidth can be very useful here and allow these applications to select the appropriate encoding scheme (G.721, 726, etc standard) and utilize the current network conditions optimally. ISP network engineering, service level agreement (SLA) verification, end-to-end admission controls are some more examples that can also benefit from available bandwidth estimation.

In general a large number of measurement packets are required to measure available bandwidth [3]. This requirement is common to all techniques because they fill the network path with probe packets to obtain the network information at the intermediate systems. One way of measuring available bandwidth would be to deploy specialized software on every router (intermediate systems) in the network so that the router’s load can be reported to the end hosts continuously [3]. However, this is impractical because (i) it is very expensive to upgrade all the existing intermediate routers and (ii) the overload situation resulted from the large number of reporting traffic can cause congestion as well as security threat. The alternative and preferred approach would be to use end-to-end software that runs on the end-hosts.

III. Thesis Structure

This thesis is organized as follows. There are total 7 chapters, starting chapter 1 with Introduction of this thesis. The second chapter briefly describes the related work done on available bandwidth estimation tools, this chapter also explains some concepts and principles, and those are common to all these tools. The third chapter presents the most popular available bandwidth measurement techniques such as pathload, pathChirp, Train of Packet-Pair (TOPP), Initial Gap Increasing/Packet transmission rate (IGI/PTR) in detail. We also discuss network performance metrics common to these techniques. Fourth chapter that is evolution of pathChirp (underlying tool of this thesis) briefly describes the mathematical model and concepts used. In the fifth chapter, eChirp, the new proposed available bandwidth measurement technique is introduced; its structure, goals of this structure and how we achieve these goals, eChirp packet train methodology etc are explained in this chapter. This chapter also determines the bounds of probing parameters which affects the available bandwidth measurement accuracy. Chapter six presents the simulation results of eChirp conducted on NS-2 network simulator with different topologies and parameters, simulation results for pathChirp technique is also presented here and compared with eChirp in this chapter. Conclusion and future work interest are described in chapter 7. Thesis texts are ended with all references.

CHAPTER II

BACKGROUND

Early work on bandwidth estimation and idea of using packet train (initially with packet-pair) started back in 1988 when Jacobson [5] designed the packet conservation principle of TCP to allow the senders indirectly to hint the available bandwidth based on the spacing between the ACK packets. In 1989 Keshav [6] followed Jacobson's footstep and worked on packet-pair method for congestion control and by assuming the fair queue support from all intermediate systems in the network path. After some years (1996) Carter Crovella developed a tool called Cprobe [7] to measure the available bandwidth. Cprobe uses a train of ICMP echo packets of the target host and recorded the spacing between the first and the last returning packet. The rate of the arriving echo packets was used as an estimate of the available bandwidth. Later it was discovered by Dovrolis [8] that Cprobe actually measure the metric called asymptotic dispersion rate (ADR) [8], which does not generally equal to the available bandwidth. In recent years several available-bandwidth estimation-software tools have been proposed and they are grouped into two categories [4]: direct probing (DP) tools and active probing (AP) tools, and can be distinguished according to the two main approaches [11] underlying the available bandwidth measurement techniques. The probe gap model (PGM) and the probe rate model (PRM). In this chapter we describe the overview of these techniques. Since our proposed eChirp fall into second category that is active probing, and will be discussed in details in next chapter. The principle and concepts behind these techniques such as packet pair probing, self-induced congestion, terms direct probe, active probe, PGM, PRM and fluid-cross traffic model are also discussed.

I. Packet-Pair Probing

Packet pair probing method has been used in several tools that measures available bandwidth [2], [9], [10], [11], [13]. Originally this method was introduced by Jacobson [5] and Keshav [6]. In this method sender sends two equal-sized packets back-to-back and packet-pair dispersion is measured at the output queue of the router. The dispersion between the packets is introduced due to many types of delays. But two types of delay are more common on the networks, first is the transmission delay of the packets over the link. Second is the queuing delay (the amount of time that a packet spends in the output queue of a router interface) introduced because of the packets from other flows (e.g. cross traffic) queuing between the two packets. The dispersion of a packet-pair at the receiver is measured as the amount of time between the last bit of each packet.

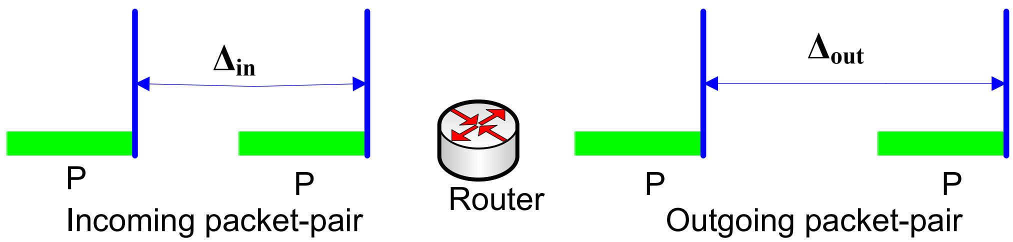


Figure 4: Packet-Pair probing

The figure 4 shows the packet-pair probing method in which,

$$\Delta_{out} = \max(\Delta_{in}, P/C_i)$$

Where P is the size of packet, Δ_{out} and Δ_{in} are the inter-spacing between packet-pair at input interface and at output interface of the router respectively, P/C_i is the packet transmission time at link $_i$, C_i is the rate at link that is the capacity of router's output queue and Packet-pair dispersion that is the time interval between last bit of two packets. If the inter-arrival spacing between two packets is greater than the transmission delay at any router, then $\Delta_{out} = \Delta_{in}$.

If we assume there is no cross-traffic and probe traffic is only the flow over the network path, the maximum dispersion between the packet pair would be in narrow link with minimum capacity. Therefore, the maximum dispersion at the receiver in the absence of cross-traffic gives the bottleneck bandwidth C and can be estimated as:

$$C = P/\Delta_{out}.$$

However, in real scenarios there is always some traffic other than probe and the dispersion between the packet-pair is not restricted to transmission delays.

The basic principle of this method is that the source host sends packet probes to the destination over the network path and measures the inter-packet delay between the packets as they arrive at the destination. All the measurement tools that use packet-pair probing method is trying to find, how the probing packet should be sent, what are the optimal parameter for changing the rate of packet with minimum congestion and fast convergence, basically trying to find better ways to create improved configurations of probing packets to gain better information for the analysis.

II. Self-Induced Congestion

Self-Induced congestion principle is used in most of above mentioned measurement tools. The basic principle of self induced congestion is that, the network path is temporarily filled by probe packets and the path is congested only if the probing rate larger than the available bandwidth resulting queuing delay. The minimum probing rate that cause congestion, onset of this probing rate that cause congestion reflects the available bandwidth of the path. All measurement techniques that use self-induced congestion principle relied on FIFO queuing (Hardware queue of router is always FIFO) at all the routers in the path. This principle can be explained mathematically as follows: If an end user host sends probe packets to destination host with probing rate (PR) less than available bandwidth (ABW), probe packets will not experience any delay or experience similar delays. On the other hand if the PR is exceeded than ABW probe packets will be queued at routers in the network path and experiences increasing queuing delays.

$$\textbf{PR} < \textbf{ABW} \rightarrow \textbf{no Queue delay increase}$$

$$\textbf{PR} > \textbf{ABW} \rightarrow \textbf{Queue delay increase}$$

Where PR = Probing rate and ABW = Available bandwidth.

This principle is based on notation that the delays of successive probing packet will increase when probing rate exceeds the available bandwidth in the network path. This is shown in figure 5.

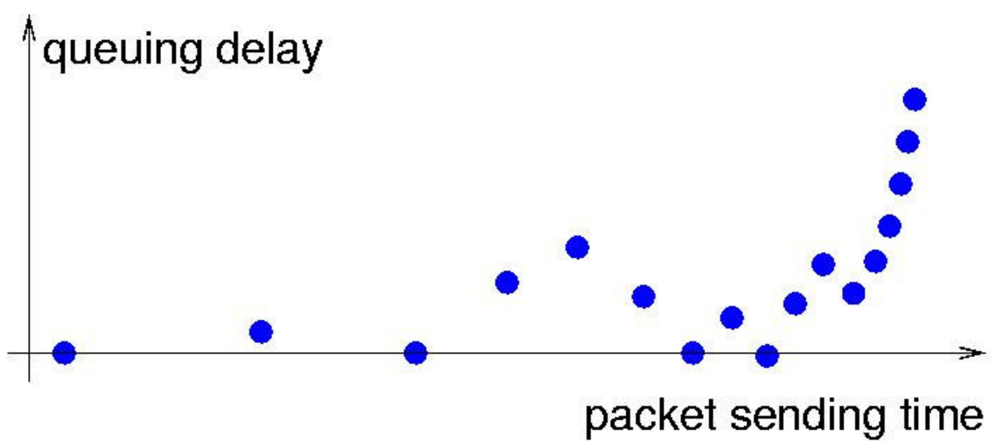


Figure 5: Self Induced congestion Heuristic [1]

$$Q_J > Q_{J+1} \rightarrow ABW_J > PR_J$$

$$Q_J < Q_{J+1} \rightarrow ABW_J < PR_J$$

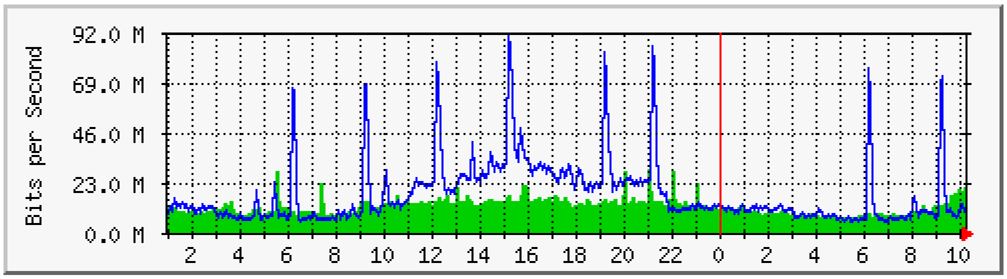
Where J and J+1 are successive packets, QJ and QJ are their corresponding delays.

PR_J = Packet Size/Δt_J is an instantaneous probing rate for packet J.

III. Direct or Passive Probing

Direct or Passive probing [17] measurement tools use the background history of existing data transfers, while potentially very efficient and accurate, their scope is limited to network paths that has recently carried user traffic. The cross-traffic rate is the major player in DP [4] mechanism and it is used to estimate the available bandwidth by sampling at each packet train. One of the examples of this technique is *Delphi* [16], and the other one is Initial gap increase (IGI) [17]. The main advantage of this approach is that it can adapt to the current traffic condition in real-time. However the main problem with this technique is that it needs the capacity of the tight-link (i.e. the link with minimum bandwidth) *a'priori*. Additionally DP mechanism is only suitable for single-hop scenarios hence it is not suitable for multi-hop networks (note that most of the networks are multi-hop network today). Passive probing is currently used in many monitoring tools including well known Multi Router Traffic Grapher (MRTG). Figure 6 shows the MRTG graph on one of our link to the North Carolina Research Education Network (NCREN).

`Daily' Graph (5 Minute Average)



	Max	Average	Current
In	28.7 Mb/s (2.9%)	11.5 Mb/s (1.2%)	20.9 Mb/s (2.1%)
Out	90.4 Mb/s (9.0%)	17.0 Mb/s (1.7%)	8640.8 kb/s (0.9%)

Figure 6: MRTG graph, a passive measurement.

Direct probing techniques can be thought of probe gap model (PGM) [11]. The probe gap model exploits the information in the time gap between the arrivals of two successive probes at the receiver. A probe pair is sent with a time gap Δ_{in}, and reaches the receiver with a time gap Δ_{out}, Assuming a single bottleneck and that the queue does not become empty between the departure of the first probe in the pair and the arrival of the second probe, then Δ_{out} is the time taken by the bottleneck to transmit the second probe in the pair and the cross traffic that arrived during Δ_{in}, as shown Figure 7.

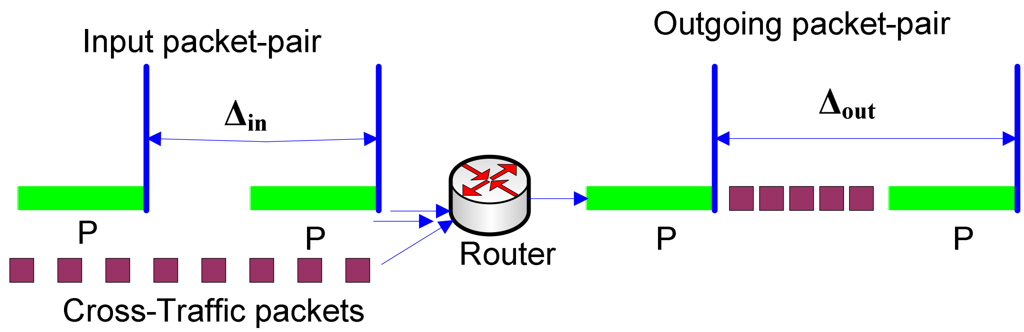


Figure 7: Probe gap model for estimating available bandwidth

Thus, the time to transmit the cross traffic is $\Delta_{out} - \Delta_{in}$ and the rate of the cross-traffic is:

$$\text{Rate of cross traffic} = C \times (\Delta_{out} - \Delta_{in}) / \Delta_{in}.$$

Where C, is the capacity of the bottleneck. The available bandwidth is:

$$Abw = C \times (1 - (\Delta_{out} - \Delta_{in}) / \Delta_{in}).$$

The Figure 7 looks same as Figure 4, but the major difference between the two is, Figure 4 is used to explain packet-pair model on single-hop. Figure 7 explains the Probe gap model where rate of cross-traffic is being measured and capacity is known.

IV. Active Probing

The *self-induced congestion* (explained in section 2 of this chapter) is the main concept used in AP probing tools in which stream of packets are induced with exponentially increasing rate to measure the traffic conditions. Subsequently it takes the lowest input rate that overloads the network as available bandwidth of that network path. This is adopted in *TOPP* [2], *pathLoad* [12], [13], *pathChirp* [1] with some modifications. This technique is also known as iterative probing [4] which consists of sending streams of packets whose input rate iteratively increases.

Active probing techniques can be thought of probe rate model (PRM) [11]. This model is based on the concept of self-induced congestion that we already explained, in which if one sends probe traffic at a rate lower than the available bandwidth along the path, then the arrival rate of probe traffic at the receiver will match their rate at the sender. In contrast, if the probe traffic is sent at a rate higher than the available bandwidth, then queues will build up inside the network and the probe traffic will be delayed. As a result, the probes' rate at the receiver will be less than their sending rate. Thus, one can measure the available bandwidth by searching for the turning point at which the probe sending and receiving rates start matching.

V. Fluid Cross-traffic model

Fluid cross-traffic model that is a major player in many of available bandwidth measurement technique those are categories as passive probing measurement. In this model non-probe packets have an infinitely small packet size and the average rates of cross traffic change slowly and is constant for the duration of a single measurement. Melander *at al.* (2002) [9] studied the relationship between the input and output rates R_I and R_O of probing train in a single-hop path and presented the following FIFO fluid cross-traffic model.

$$R_O = R_I \text{ when } R_I < C - \lambda$$

And

$$R_O = C\,(R_I / R_I + \lambda) \text{ when } R_I \geq C - \lambda$$

Where C is the capacity of the hop and λ is the cross traffic rate. Many of measurement techniques are based on a deterministic analytical model that assumes a single-hop path and constant rate fluid cross traffic arrival as shown in Figure 8. Such a model leads to deterministic relationship between the input and output signals (inter-packet spacing or probing rates) carried by packet train.

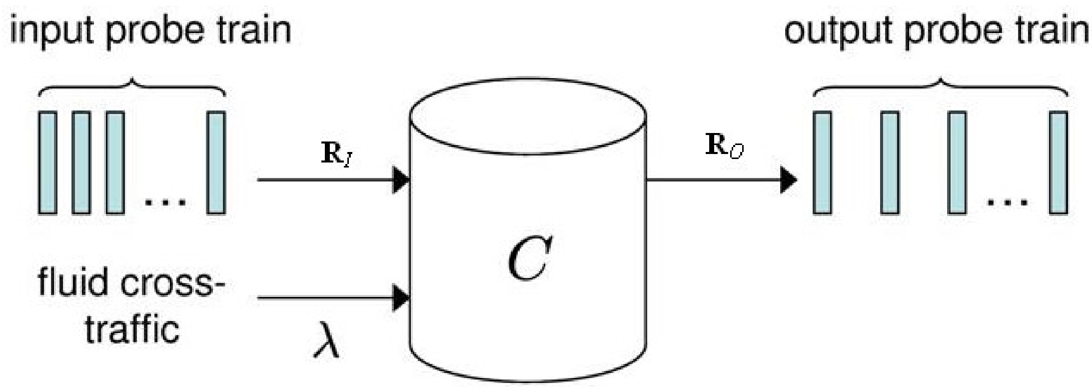


Figure 8: Single-hop fluid model, cross-traffic has very small packet size and arrives at hop with constant rate [20]

Figure 9 shows the single-hop fluid response curve modeling the mathematical dependency between the ratio of input and output probing rates and the input rate. This response cover is important theoretical foundation, based on a number of existing techniques designed to measure tight-link bandwidth characteristics over a multi-hop path. These techniques assume that the single-hop fluid curve is a valid first-order approximation of the real situation, largely neglecting cross traffic burstiness and the effect of non-tight links.

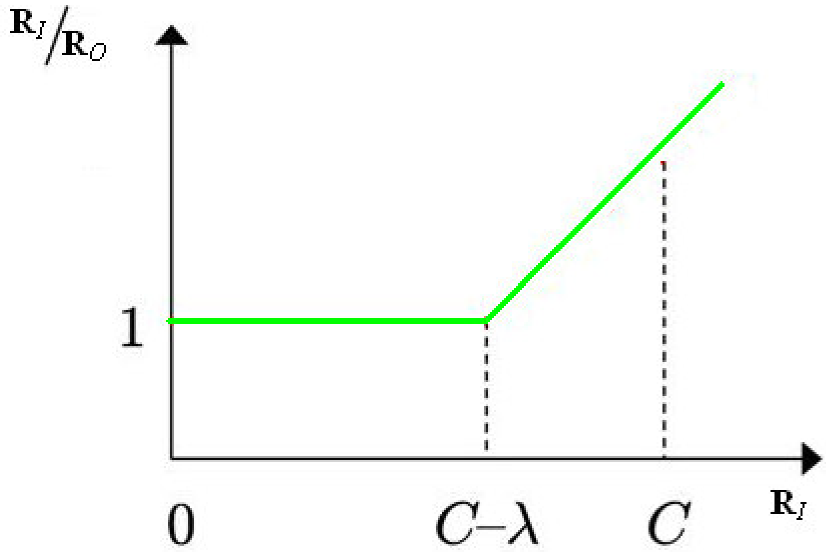


Figure 9: Single-hop fluid model response curve [20]

CHAPTER III

AVAILABLE BANDWIDTH MEASUREMENT TECHNIQUES

As we briefly discussed in previous chapter there are several available-bandwidth estimation-software tools have been proposed and they are broadly grouped into two categories [4]: direct probing (DP) tools and active probing (AP) tools. In this chapter we theoretically evaluate some of Active probing tools and one of the direct probing tool those are recently proposed, since our proposed eChirp fall into second category that is Active probing, and will be discussed in later chapter. The last part of this chapter discusses some network performance metrics which are common to all techniques. Some of these are delay metrics such as queuing delay, serialization delay, propagation delay, forwarding delay, delay variation, round trip time (RTT), packet size maximum transmission unit (MTU) and congestion are briefly discussed.

I. PathLoad

Pathload estimate the available bandwidth of an end-to-end network path between a source host S to a receiving host R. Pathload defines the available bandwidth as the maximum IP-layer throughput that a flow can get in the network path from source to receiver, without reducing the rate of the rest of the traffic in the path.

The Pathload algorithm is based on the concept called Self-Loading Periodic Streams (SLoPS), which is described in [2] [13]. Figure 10 shows this concept, the key idea that pathload is based on is : When a source host process at S sends a periodic stream of UDP packet at a rate higher than the available bandwidth in the path, At the receiver R the relative one-way packet delays show an increasing trend. When the stream rate is lower than the available bandwidth, the relative one-way packet delays show no consistent trend.

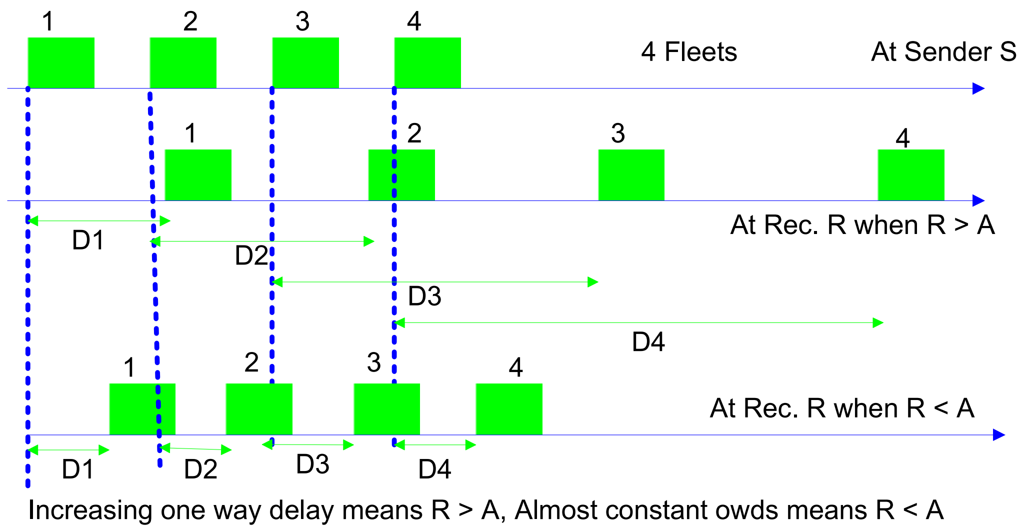


Figure 10: Self-Loading Periodic streams (SLOP)

Increasing one way delay means the rate of the fleet is greater than the available bandwidth and when one way delay is almost constant means rate is less than available bandwidth.

Figure 11 shows the structure of packet train used in pathload, where packet probing rate vary only with successive trains, but data rate is constant per train that is big shorting of this structure by means of efficiency.

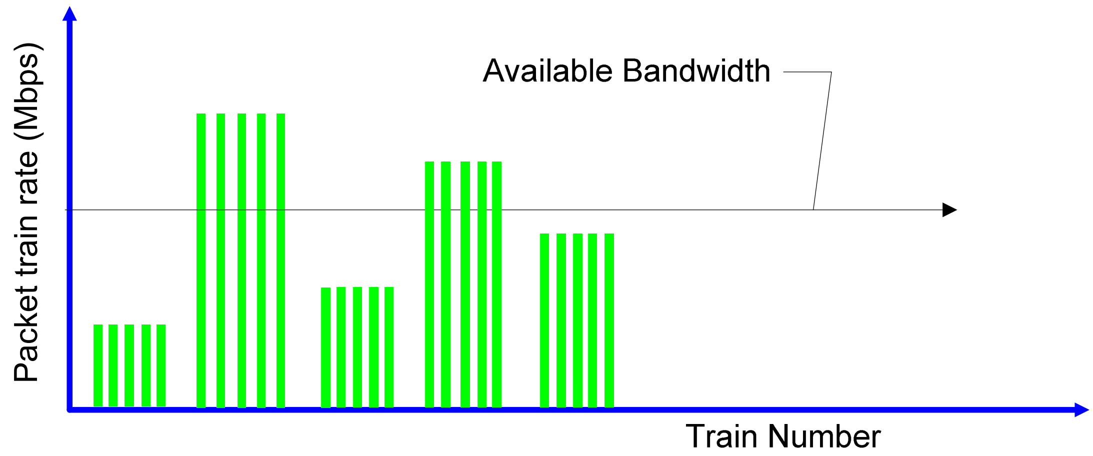


Figure 11: PathLoad Packet Train Structure, vary rate of successive trains only

There are two main components that Pathload consists, a process running at source S and a process running at receiver R. A process running at S sends periodic streams of UDP packets from S to R at a certain rate. Pathload does not determine whether a particular rate (Pr) is greater than available bandwidth (Abw) based on just one stream. Instead, it sends "a fleet of N streams", so that it has N samples to decide whether Pr > Abw, or not. Upon the receipt of a complete fleet, receiver R checks if there is an increasing trend in the relative one-way packet delays in each stream. If a large fraction f of the N streams in a fleet show increasing trend, then the entire fleet is said to have increasing trend and the next fleet rate is lower than Pr. If a large fraction f of streams in a fleet show no trend, then the entire fleet is said to have no trend and the next fleet rate is higher than Pr.

Pathload uses the following probing methodology:

- Sender transmits periodic packet stream of rate Pr.
- Let's say there are k packets in a stream and the size of the each packet is P.
- Receiver measures One Way Delay (OWD) for each packet.
- Let's say D(k) is the delay, $D(k) = t_{rec}(k) - t_{snd}(k)$.
- And one way delay variation: $\Delta(k) = D(k+1) - D(k)$.
- With stationary and fluid-modeled cross traffic:
- If Pr > Abw, then $\Delta(k) > 0$ for all k. Else, $\Delta(k) = 0$ for all k.
- Estimate upper and lower bound for available bandwidth variation range.

II. PathChirp

PathChirp is the most recent active probing tool for available bandwidth measurement. And the author of pathChirp has shown through the comparative studies with pathload and TOPP that it is most accurate and efficient as compared with two till date. For that reason we have selected pathChirp as our comparative tool and in this thesis we have dedicated separate chapter 'chapter 4' to evaluate it. Here in this section we will briefly describe this scheme and rest of its features discussed in chapter 4.

PathChirp is based on the concept of self-induced congestion (that is explained in chapter 2). The pathChirp algorithm induces the congestion using probing packets called "Chirp" and estimates the available bandwidth based on the delay information obtained using the chirp packets. Unique to pathChirp is an exponentially spaced chirp probe train that features an exponential flight pattern of probs. PathChirp also uses the shape of the queuing delay signature that is shown in Figure 16 (chapter 4) to make and estimate per packet available bandwidth.

In pathChirp, probe packets travel one way from sender to receiver, and the receiver performs the estimation of available bandwidth.

III. TOPP

Train of packet pairing (TOPP) [2] probing is an extension to the packet probing technique for measuring available bandwidth of bottlenecks in the network paths. TOPP sends out many packet pairs with equally spaced in time, i.e. the streams of packet pairs are sent with uniformly increasing input rate. It uses a simple mechanism to increase the input by reducing the time gap between each pair. This finds the maximum input rate that is less than the measured rate at the destination and assigns it to the available bandwidth as an estimate. Figure 12 shows the structure of train of packet pair in which pair of packet is equally spaced.

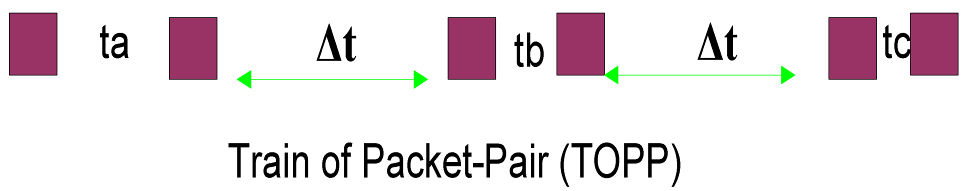


Figure 12: Train of Packet-Pair (TOPP), ta > tb > tc

The TOPP measurement technique has two phases. The first phase is active probing phase where pair of probe packets are injected into the network and the second phase is analysis phase where it analyze the first phase and the available bandwidth measurement is calculated based on the reception times of probe packets. The probe packet traffic is generated in the following way.

Starting at some rate R_{min}, and N equal sepearted pairs of equally sized probe packets are sent to the destination host. After these n packets have been sent the offered rate R is increased by Δ_R and new set of N probe packets are sent. R is then increased again (by the same amount by Δ_R) and another set of N probe packets are sent. This goes on until the offered rate R reaches some rate R_{max} which marks the end of probe phase. Therefore, there will be $N_k = (R_{max} - R_{min})/ \Delta_R$ offered probing rates levels. Figure 13 shows this probing phase behavior.

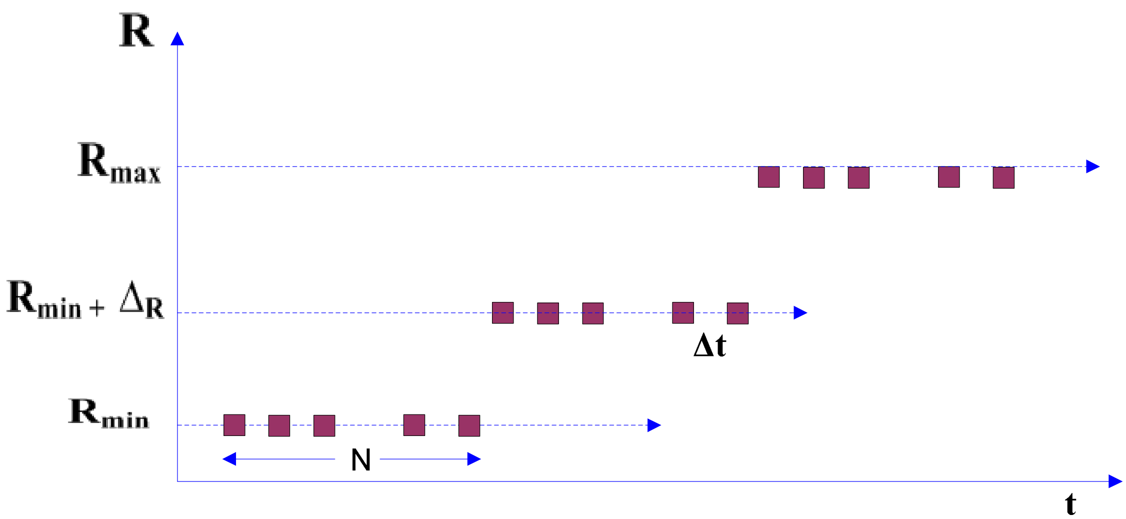


Figure 13: Train of Packet-pair (TOPP) Probe sequence

Where N = set of equally spaced probe packets,

R_{min} = Minimum starting probing rate

R_{max} = Maximum probing rate.

Δ_R = Probing rate increment in each of N set.

At the receiver end, probe packets are time stamped upon reception. Once all packets have been received, the time stamps are sent back to the probe sending host. Hence all measurements are done one way.

The second phase of TOPP is analysis phase and that is relies on the principle of bottleneck spacing effect. That is, when two packets with the time separation Δ_t are transmitted over a link with a service time $T_d > \Delta_t$, then as the packet leave the link they will separed by $\Delta_{T =}$ $T_{d.}$ Using the size of the packets, P, and the time separation $\Delta_{T,}$ the experienced bandwidth across that link can be estimated as $AB_i = P/\Delta_{T.}$

IV. IGI/PTR

IGI (Initial Gap Increasing) [17] is based on probe gap model that we briefly discussed in previous chapter, a model that construct the relationship between competing flows on a bottleneck link and the change in gap between a input packet-pair and output packet-pair for a single hop network. In this model [18] the amount of cross traffic is measured that is a function of traffic inserted between the gaps of packet pair. IGI searches the initial gap value so that a probing packet train interacts with the cross traffic in a non empty narrow link queue which is called by the author " Joint Queuing Region". In this region, there is a proportional relation between the gap when probing packet leave the queue and the cross traffic. When the initial gap is increased and equal to the output gap, the available bandwidth on the narrow link is equal to the average rate of the packet train. After that point, called the turning point, the narrow link will be overflowed by the probing packets. It is shown by the authors that in the case of multiple hops and significant cross traffic, the accuracy of the IGI suffer.

The basic idea in this technique is that, probing host

- Start by sending out packet train with minimum gap (g_{min}).
- At the receiver gap is compared, if gap at receiver not equal to the gap was at

sender, probing host sends another train with increased gap.

- On the other hand if gap at receiver is equal to the gap at sender, the available bandwidth is calculated.

Figure 14 shows the probing behavior of this technique.

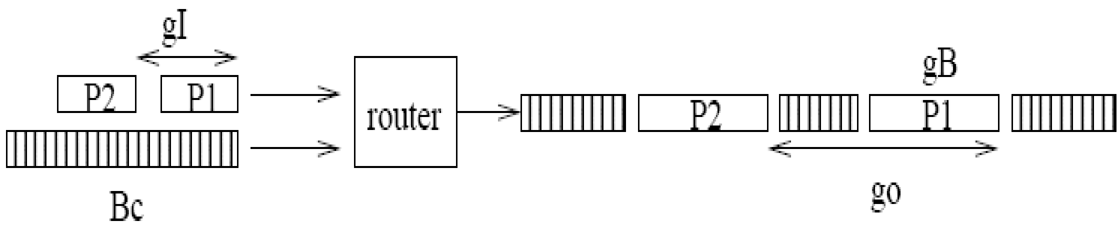


Figure 14: Probing and competing flows on a single hope network [17]

In the above figure the probing host sends a pair of packets in quick succession and to measure how the gap between the two packets changes as a result of traversing the network. As the packets travel through the network, packets belonging to competing traffic will be inserted between them, thus increase the gap. As a result, the gap value at the destination host will be a function of the competing traffic rate, making it possible to estimate the amount of competing traffic. Unfortunately, how competing traffic affects the gap of a packet pair is much more complex than this description suggests.

There are many shortcoming of this technique. First, this technique use single hope model, in that:

- IGI need to know the capacity of the tight link.

- IGI assume that tight link is same as narrow link.
- The relationship of amount of cross-traffic and gap does not hold in multi-hop path.

V. Network performance metrics [19]

There are many metrics that are commonly used to characterize the performance of networks and parts of network. Here in this thesis we present some of these metrics briefly, only definitions because each of this itself is a large subject. We defines the metrics those are closely related with available bandwidth measurement such as all types of delays (queuing delay, serialization delay, propagation delay, forwarding delay, delay variation also known as jitter), Round-Trip Time (RTT), packet loss, Maximum transmission unit (MTU) affects the measurement of available bandwidth, infect Available bandwidth itself is an performance metric in networks.

Available Bandwidth

Available bandwidth is as we already defined in Chapter 1 and can be re-elucidate here that is common to all measurement technique, as the minimum remaining bandwidth along the best path between the source and destination that can be used by a new flow without disturbing the transmission of other flows on the path. That is, available bandwidth can be calculated as path capacity minus path load.

Queuing Delay

Queuing delay is defined as the time a packet has to spend inside a node such as a router while waiting for availability of the output link. It depends on the amount of traffic competing to send packets towards the output link and on the priorities of the packet.

Serialization Delay

It's the time taken to separate a packet into sequential link transmission units (bits). It is obtained by dividing the packet size (in bits) by the capacity of the link (in bits per second). Nowadays, as links increasingly have a higher bit rate, serialization delay is less relevant.

Propagation Delay

Propagation delay is the duration of time for signals to move from the transmitting to the receiving end of a link. On simple links, this is the product of: the link's physical length and the characteristic propagation speed of media. On high-speed wide-area network (WAN) paths, delay is usually dominated by propagation times.

Forwarding Delay

It is due to processing at the node reading forwarding-relevant information e.g. destination address plus other headers. Another factor is forwarding decision which is based on the routing table and other information, and to actually forward the packet towards the destination, which involves copying the packet to different interface inside the node, rewriting parts of it (such as IP TTL and any media specific headers) and possibly other processing such as fragmentation, accounting or checking access control lists.

One-Way Delay (OWD)

The time it takes for a packet to reach its end-to-end destination is called OWD. It is considered a property of network links or paths. It can be broken down into: per-hop one-way delays, and these in turn into: per-link and per-node delay components. The per-link component of one-way delay consists of two sub-components: propagation delay and serialization delay. The per-node component of one-way delay also consists of two sub-components: forwarding delay and queuing delay.

Round-Trip Time (RTT)

It is the sum of the one-way delays from source to destination plus time it takes to formulate the response. Large RTT values can cause problems for TCP and other window-based transport protocols. The round-trip time influences the achievable throughput, as there can only be a window's worth of unacknowledged data in the network.

Delay Variation

OWD is not constant on a real network because of competing traffic and contention for processing resources. The difference between a given packet's actual and average OWD is termed 'delay variation' or jitter. It only compares the delays experienced by packets of equal size, as OWD is dependent on packet size because of serialization delay.

Packet Loss

Packet loss is determined as the probability of a packet being lost in transit from a source A to a destination B. Applications requiring reliable transmission e.g. Bulk data transfers, use retransmission, which reduces performance. In addition, congestion-sensitive protocols such as standard TCP assume that packet loss is due to congestion, and respond by reducing their transmission rate accordingly. Congestion and errors are the two main reasons for packet loss.

Congestion

When the offered load exceeds the capacity of a part of the network, packets are buffered in queues. Since these buffers are also of limited capacity, congestion can lead to queue overflows, which leads to packet losses. Congestion can be caused by moderate overload condition maintained for an extended amount of time or by the sudden arrival of a very large amount of traffic (traffic burst).

Errors

Another reason for loss of packets is corruption, where parts of the packet are modified in-transit. When such corruptions happen on a link (due to noisy lines etc.), this is usually detected by a link-layer checksum at the receiving end, which then discards the packet.

Maximum Transmission Unit (MTU)

It describes the maximum size of an IP packet that can be transferred over the link without fragmentation. Common MTU sizes are: 1500 bytes (Ethernet, 802.11 WLAN), 4470 bytes (FDDI, common default for POS and serial links), 9000 bytes (Internet2 and GÉANT convention, limit of some Gigabit Ethernet adapters) and 9180 bytes (ATM, SMDS).

Bandwidth Delay Product (BDP)

The Bandwidth Delay Product (BDP) of an end-to-end path is the product of the bottleneck bandwidth and the delay of the path. It is often useful to think of BDP as the "memory capacity" of a path, i.e. the amount of data that fits entirely into the path between two end-systems. This relates to throughput, which is the rate at which data is sent and received. Network paths with a large BDP are called Long Fat Networks or LFNs. BDP is an important parameter for the performance of window-based protocols such as TCP.

CHAPTER IV

EVOLUTION OF PATHCHIRP

For different performance metrics, the probing packets can be structured differently. It is evident from the recent research that the structure of the probing packet is a major player in estimating the available bandwidth. Hence, several probing-packet structures have been proposed in the recent approaches. For example single packet concept is used by simple protocol like *ping* and *traceroute*; packet train structure is used by Pathload [9] [13] [14] and cprobe [7]; packet chirp, another type of packet train, is used by pathChirp [1]; and packet tailgating is used by STAB [10]); packet pair is used by TOPP [2] and Spruce [11]; packet triplet is used by Tulip [12]. The packet probing structures of these methods differ based on the number of packets and the gap (or spacing) between the packets. Packet gap values can be either fixed size (as in packet train in Pathload [9] [13] [14]) or follow exponential distribution (as in packet chirp in pathChirp [1]). In this chapter we briefly discuss the mathematical model and the concept of the model used in the pathChirp scheme. PathChirp consists of (i) a unique structure for the chirp probe train, (ii) an actual bandwidth definition, (iii) a model for queuing delay, (iv) an excursion signature segmentation and (v) bandwidth estimator.

I. PathChirp structure

Fig. 15 shows the structure of pathChirp packet train in which a source host sends *N* number of packets with constant packet size of P_{size}, starting with time gap of $T\alpha^{N-2}$ (where T is lowest specified probing rate), and probing rate that is increased exponentially by decreasing inter-packet gap of successive packet using a parameter α. The following equation shows this probing rate (PR) increase.

$$PR = T\alpha^{N-2}, T\alpha^{N-3}, T\alpha^{N-4}, \ldots\ldots, T\alpha^{2}, T\alpha, T. \qquad (1)$$

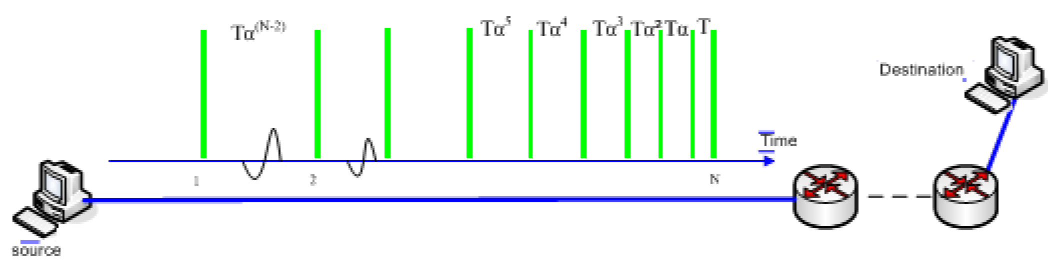


Figure 15: Structure of pathChirp packet train.

The basic idea of this structure is to induce congestion in the path by increasing the probing rate of successive packets until the point where a packet suffers from queue delay. Suppose packet *j* experiences queue delay q_j and the gap between the two successive packets *j* and *j*+1 is Δ_j then the probing rate of packet *j* is calculated as (for m^{th} chirp train):

$$PR_j^m = P_{size} / \Delta_j^m \quad (2)$$

At some point the packets can experience queue delay due to burst traffic and this delay is called excursion signature delay in pathChirp. This delay is used to calculate the available bandwidth. It is possible that there will be more than one excursion points in the path. Each delay is compared with the next delay until the maximum and the resulting delay is used for calculating the available bandwidth.

II. Actual Bandwidth

PathChirp defines the actual available bandwidth for every router node with respect to router's output queue capacity, total traffic, and propagation delay and packet service time as follows:

$$AB[t_1, t_2] = \min_i \left(QC_i - \frac{TT_i[t_1 + td_i, t_2 + td_i]}{t_2 - t_1} \right) \quad (3)$$

where $AB[t1,t2]$ is the available bandwidth of the path in time interval $[t1,t2]$, QC_i is the capacity of the router link i that is determined by node interface, TT_i is the total traffic (other than measurement probe traffic) entering between any time interval at link i and td_i is the processing delay at link i.

In reality probe packets suffer queuing delays in addition to the minimum delay td_i. Thus probes transmitted during [$t1,t2$] can arrive at router i outside time interval [t_1+ td_i, t_2+ td_i.] and do not exactly measure $AB[t1,t2]$.

III. Queue Delay

In real scenario there are many factors that add delay in the network path that includes propagation delay, transmission delay, processing delay and queuing delay. Queue delay is the amount of time that a packet spends in the output queue of a router interface. These delays are contributed in end-to-end delay. In general propagation delay and transmission delay are considered as constant where as processing delay and queuing delay are unpredictable. However, in pathChirp, processing delays are considered as constant because of the use of constant packet size. The pathChirp uses cross-traffic model for bandwidth estimation.

In the pathChirp model the probing packet j will not experience any queue delay as long as it satisfies the following condition:

$$if\ (PR_j^m \leq AB[t_1, t_2])\ then\ q_j = 0\ else\ q_{j-1} < q_j; \qquad (4)$$

IV. Excursion Segmentation

The purpose of finding excursion segmentation is to obtain the time duration where the delay is maximum. However the condition is that this maximum delay should not be created by the burst traffic. Sometimes packet can experience queue delay because of sudden burst of traffic. These queuing delays do not follow the same increments therefore it should not be included in the calculation of available bandwidth. To find an accurate excursion segmentation (where excursion starts and ends) for a particular chirp is very important to estimate the correct bandwidth. PathChirp detects the excursion using relative packet delay within a chirp.

Figure 16 shows the shape of the queuing delay signature, pathChirp uses this to estimate per packet available bandwidth.

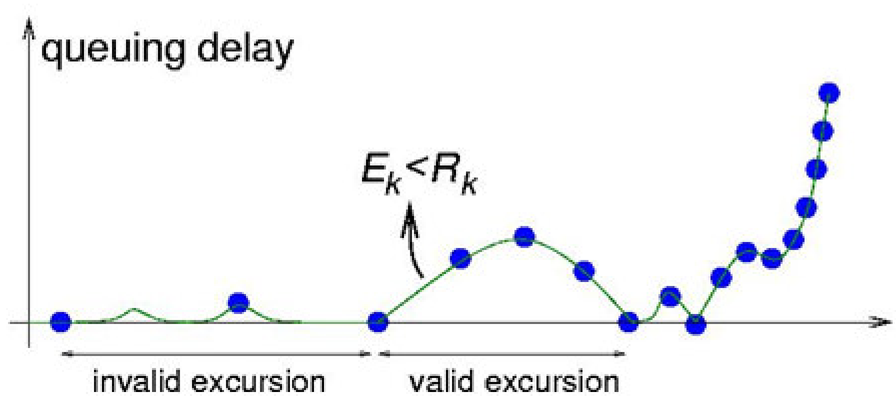


Figure 16: Excursion, queue delay signature plot. [1]

This queuing delay signature plot shows the excursion from the zero axis ($Q_k > 0$ for several consecutive packets, where Q_k is the queue delay of packet k in one of particular chirp) caused by burst of cross-traffic. In particular, due to bursty traffic queuing delays do not follow the same increment within a chirp or any probing train for that matter. The first few excursions end with the queuing delay returning to zero. We can see in above figure where the valid excursion is and where invalid excursion is. To find an accurate excursion (where excursion starts and ends) for a particular chirp is very important to estimate the correct bandwidth. PathChirp detects the excursion using packet delay within a chirp.

In order to accurately measure per packet available bandwidth, pathChirp segments each delay signature into regions belonging to excursion and regions not belonging to excursions. The basic idea behind pathChirp excursion segmentation algorithm is quite simple. Intuitively if Q_k increases and remains larger than zero for several consecutive packets, then it is likely that these packets are all part of same busy period (Time interval during which the queue is never ideal) at a congested queue along the path.

V. Bandwidth Estimator

PathChirp estimates the bandwidth using delay excursion. It calculates per-packet available bandwidth first between two successive packets j and j+1 and it is denoted by E_j^m. It then calculates per-chirp available bandwidth by taking weighted average of all the packets of a chirp train that packet *j* belongs to as follows:

$$D^{m} = \frac{\sum_{j=1}^{N-1} E_{j}^{m} \Delta_{j}}{\sum_{j=1}^{N-1} \Delta_{j}} \qquad (5)$$

Finally, it estimates total available bandwidth *AB*[*t1,t2*] by averaging D^{m}s of selected chirps over time [*t1,t2*].

CHAPTER V

RE-MODELED – ECHIRP TECHNIQUE

eChirp" that is based on most recent tool pathChirp[1]. eChirp can measure the available bandwidth over the network path efficiently and accurately with heavy and light load links. Unique to eChirp is its packet train structure and methodology. In the modified structure, the rate of the odd inter-chirp packet will be the same as the rate of previous even inter-chip packet. Additionally, rate of inter-chirp packets will be increased exponentially with even power rather than both even and odd power as done in pathChirp method. We also use two sub eChirp packet trains for getting more granule information of the network path that gives more accurate measurement that is contrast to pathChirp's chirp train structure that uses chirp probes of single train with an exponentially increasing probing rate. eChirp train structure captures more network information with added advantage of 50% reduction in chirp packet generation, and subsequently finds a better estimate for the available bandwidth.

In this chapter we discuss the eChirp structure in details; goals of this structure and how we achieve these goals, eChirp packet train methodology are also discussed. We also evaluate the bounds of probing parameters which affects the available bandwidth measurement accuracy.

I. eChirp structure

Figure 17 (b) shows the structure of eChirp packet train scheme along with pathChirp structure figure 17 (a) for comparison purpose. The main goals of this scheme are:

- To reduce the number of chirp packets generated while maintaining the probing rate as close as possible to that of pathChirp.
- To obtain good estimation of burst traffic (is it really a short term burst traffic or indication of congestion?) and excursion segmentation (when does the excursion happen?).
- To allow more data packets between chirp packets.
- To gain better approximation to correlate between two consecutive excursion segmentation.

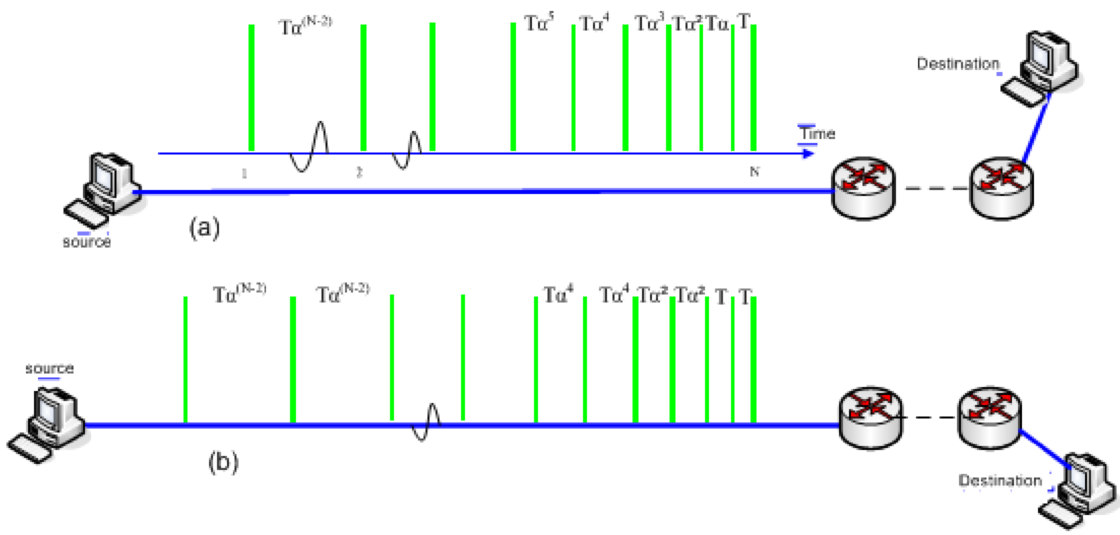


Figure 17: eChirp train structure, probing rate showing repeated rate and exponential increase with even power and factor of α, (a) pathChirp train structure.

To achieve these goals, the probing packet structure of our eChirp scheme follows:

- Every odd packet repeats in the packet probing structure.
- The timing gaps between the repeated consecutive packets are the same.
- Probing rate is increased exponentially with even power.
- Two chirp sub trains with different timing gaps are mixed.

II. eChirp packet train methodology

We first generalize the pathChirp scheme in order to compare both the pathChirp scheme and the proposed eChirp scheme. Suppose N is the number of packets in a chirp train and M is the number of chirp packets generated. Then $M = N$ in pathChirp scheme and the probing rate increase is as follows [1]:

$$T\alpha^{M-2}, T\alpha^{M-3}, T\alpha^{M-4}, T\alpha^{M-5}, \ldots\ldots\ldots, T\alpha^{2}, T\alpha, T. \qquad (6)$$

For example if the chirp train requires even number of packets (N=8) then the source generates even number of chirp packets (M=8) and in this case the probing rate increase is:

$$T\alpha^{6}, T\alpha^{5}, T\alpha^{4}, T\alpha^{3}, T\alpha^{2}, T\alpha, T. \qquad (7)$$

Similarly if the chirp train requires odd number of packets (N=7) then the source generates odd number of chirp packets (M=7) and the probing rate increase is:

$$T\alpha^{5}, T\alpha^{4}, T\alpha^{3}, T\alpha^{2}, T\alpha, T. \qquad (8)$$

In the proposed eChirp scheme only even number of packets are allowed in the chirp train. This is because we have two sub chirp trains within the chirp train. Thus if the chirp train requires *N* (always even) number of packets the source needs to generate *M*=*N*/2 packets, which is half the size of the number of packets that the pathChirp is required to generate. In this case the probing rate increase is:

$$T\alpha^{2M-2}, T\alpha^{2M-4}, T\alpha^{2M-4}, ... T\alpha^{2}, T\alpha^{2}, T, T. \qquad (9)$$

That is, if the chirp train requires *N*=10 number of packets then the source generates *M*=5 number of packets. Then the probing rate increase is:

$$T\alpha^{8}, T\alpha^{6}, T\alpha^{6}, T\alpha^{4}, T\alpha^{4}, T\alpha^{2}, T\alpha^{2}, T, T. \qquad (10)$$

Hence in the proposed structure we have three trains with 50% reduced number of packets compared to pathChirp. The first train is spaced by the probing rate increase of:

$$T\alpha^{2M-2}, T\alpha^{2M-4}, T\alpha^{2M-4}, ... T\alpha^{2}, T\alpha^{2}, T, T. \qquad (11)$$

The second train is spaced by the probing rate increase of:

$$T\alpha^{2M-4}(\alpha^{2}+1), T\alpha^{2M-6}(\alpha^{2}+1), ... T(\alpha^{2}+1). \qquad (12)$$

The third train is spaced by the probing rate increase of:

$$2T\alpha^{2M-4}, 2T\alpha^{2M-6}, \ldots\ldots 2T\alpha^{2}, 2. \qquad (13)$$

Hence we have more data to characterize the delay and excursion segmentation.

III. Composite structure eChirp packet train

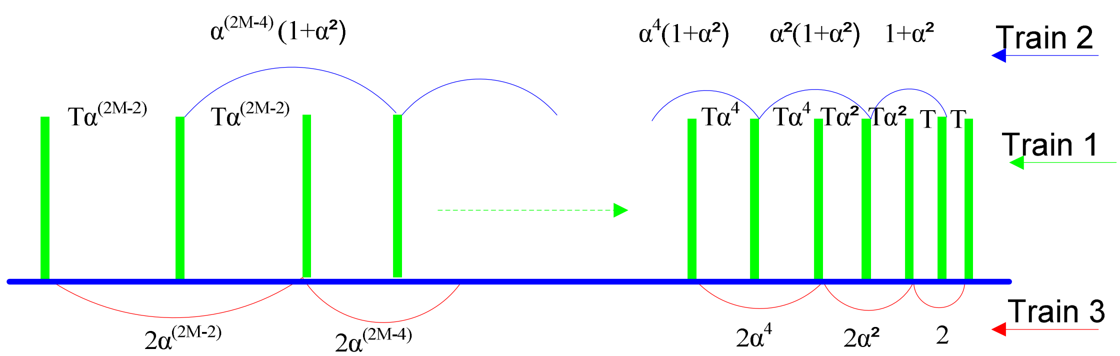


Figure 18: Composite eChirp packet train structure.

Figure 18 shows the composite structure of eChirp packet trains in which three trains with different probing rates are combined to measure the available bandwidth using equation 25. Relationship between each train provides more accurate information for accurate measurement analysis.

IV. eChirp packet Header and parameter choices

- eChirp is a UDP packet (encapsulated in UDP Header) with the following header fields.
- Packet size.
- Packet Number.
- Send time.
- Low-rate.
- High-rate.
- Spread-factor
- Number of packets per Chirp train.

In all our simulation experiments we considered same packet size of 1200 Bytes. Obviously there is an effect of the packet size on estimation performance, the number of bytes transmitted per chirp decreases with packet size.

Low-rate (probing rate) and High-rate are initially specified and later vary based on the spread-factor α.

The value of spread-factor α is considered as 1.2. That is explained in next section. Other fields in the packet header are used for calculations.

V. Bounds of probing packet parameter α

Let us assume *N* (even) number of packets for the pathChirp train and eChirp train. If we denote the duration of chirp trains of size *N* packets for pathChirp and eChirp schemes by D_p^N and D_e^N, then assuming T=1 for simplicity,

$$D_p^N = \alpha^{N-2} + \alpha^{N-3} + \alpha^{N-4} + ... + \alpha^2 + \alpha + 1 \qquad (14)$$

$$D_e^N = \alpha^{N-2} + 2\alpha^{N-4} + 2\alpha^{N-6} + ... + 2\alpha^2 + 2 \qquad (15)$$

We choose the α value such that it satisfies the following boundary conditions:

$$D_e^N < D_p^N < D_e^N + D_e^{N-1}/2 \qquad (16)$$

where $D_e^{N-1} = 2\alpha^{N-4} + 2\alpha^{N-6} + ... + 2\alpha^2 + 2$ which represents half the duration of the (*N*-1) packets in the chirp train of *N* packets. Hence, $D_e^{N-1} = D_e^N - \alpha^{N-2}$ (see equation (15)). If we first consider the lower bound condition in equation (16) then we have

$$D_p^N - D_e^N = (\alpha - 1)(\alpha^{N-4} + \alpha^{N-6} + ... + \alpha^2 + 1) > 0 \quad (17)$$

It is obvious $\alpha > 0$ hence the second expression is greater than 0. Therefore $\alpha > 1$ which matches with the value α=1.2 chosen in pathChirp. Similarly upper bound condition gives

$$D_p^N < D_e^N + D_e^{N-1} / 2 \quad (18)$$

We already have $D_e^{N-1} = D_e^N - \alpha^{N-2}$. Hence

$$D_p^N - D_e^N < (D_e^N - \alpha^{N-2}) / 2 \quad (19)$$

$$2(D_p^N - D_e^N) / (D_e^N - \alpha^{N-2}) < 1 \quad (20)$$

$$\frac{2(\alpha - 1)(\alpha^{N-4} + \alpha^{N-6} + ... + \alpha^2 + 1)}{2\alpha^{N-4} + 2\alpha^{N-6} + ... + 2\alpha^2 + 2} < 1 \quad (21)$$

That is, $\alpha - 1 < 1$. It gives $\alpha < 2$.

This matches with the pathChirp conditions. PathChirp experiments indicated that α > 2 can give very high errors. Now we have shown mathematically that is true.

VI. eChirp Bandwidth estimation

PathChirp, as we stated before, defines the instantaneous chirp rate at packet *j* as

$$PR_j^m = P_{size} / \Delta_j^m \tag{22}$$

In our eChirp scheme we have three sub chirp trains within a chirp train hence we have three chirp rates per packet. Therefore we added a subscript to indicate the chirp train that the packet belongs to as follows:

$$PR_{j,k}^m = P_{size} / \Delta_{j,k}^m \tag{23}$$

Where the subscript k can have a value, either 1, 2 or 3 to indicate if the packet belongs to the first, second or third chirp trains. In addition we modified the per-chirp available bandwidth formula of pathChirp as follows:

$$D^m = \frac{\sum (E_j^m + E_{j+1}^m)\Delta_k}{2\sum \Delta_j} \tag{24}$$

This modification comes from the equal spacing between two consecutive packets used in the proposed eChirp packets. However E_j^m, the available bandwidth per-packet is calculated differently. For our eChirp we define the per-packet available bandwidth as a linear combination of per-packet available bandwidth of three chirp trains as follows:

$$E_j^m = (3E_{j,1}^m + E_{j,2}^m + E_{j,3}^m)/5 \tag{25}$$

The per-packet available bandwidth, $E_{k,l}^m$, of each train is calculated using the same approach used in pathChirp [1]. We can see from Fig. 18, three timing gaps of the first chirp train in the proposed eChirp is the sum of single timing gaps of other two chirp trains. Hence we have chosen the weights 3:1:1 in the above equation. These weights are applied in equation (25) to get a weighted average of per-packet available bandwidths of the three chirp trains.

CHAPTER VI

SIMULATION RESULTS

In this chapter we present simulation results to compare the performance of proposed eChirp technique with pathChirp scheme. To carry out experiments initially we adopt the scenarios used in pathChirp experiments [1] for compare purposes. More than 25 simulations carried out with different parameters and different network topologies

I. Initial Network topology and parameters

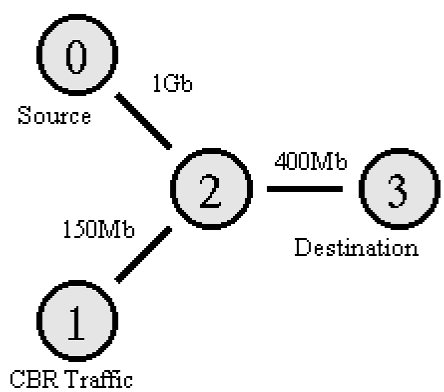


Figure 19: Network topology used in experiments.

Figure 19 shows the topology used in experiments. This topology shows four nodes 0, 1, 2, and 3 where bandwidth between node 0 and node 2 is 1000 Mb, between node 2 and node 3 is 400Mb and node 1 generates 150 Mb CBR traffic load to reduce the available bandwidth on link between node 2 and node 3. We ran this simulation over 160 simulation seconds with these parameters using NS-2 network simulation software [15]. And measure the available bandwidth for all three eChirp packet train as well as for pathChirp packet train, the below sections represents these results.

II. Actual Available Bandwidth without probing: Expected

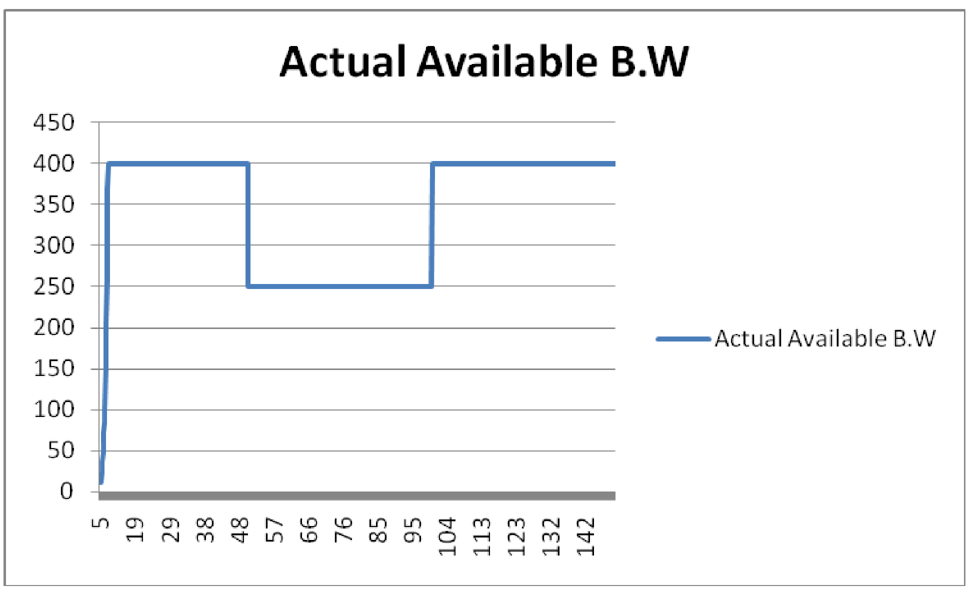


Figure 20: Actual available bandwidth of 250Mb.

Figure 20 shows the Actual available bandwidth of 250Mb that is expected without any measuring tool used. As we see in the figure for first 50 simulation seconds available bandwidth is 400Mb, that is the minimum bandwidth between the link 0-2-3. At 50 second 150Mb of CBR traffic is generated and sent between the link 1-2-3 that reduces the link bandwidth beween 2-3 to 250Mb. The sending rate of CBR is constant and stopped at 100 second at time-scale. After 100 to 160 seconds available bandwidth is again 400Mb. So from this discussion we can conclude that available bandwidth is a time varying metric.

III. Simulations result: pathChirp

We then ran pathChirp algorithm on the same network with same parameters, and the bandwidth fluctuation is plotted in Fig. 21. It shows that the pathChirp scheme measures the available bandwidth as 332Mb which is much higher than the actual available bandwidth of 250Mb (see Fig. 20). This indicates pathChirp needs improvement.

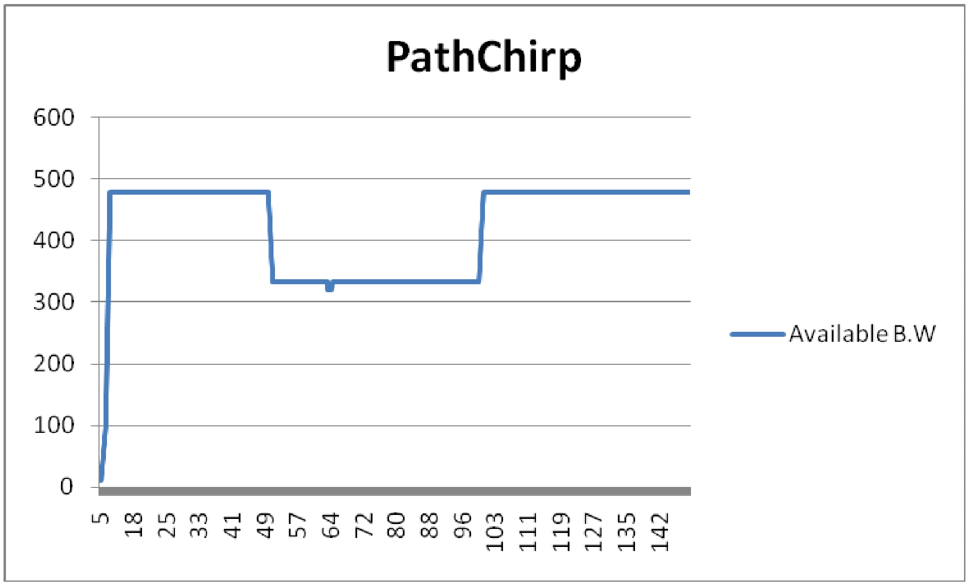


Figure 21: pathChirp measures available bandwidth of 332 Mb, where 250 Mb is expected

IV. Simulation result: eChirp packet train 1

As the next step, we ran the proposed eChirp with three chirp trains Train 1 (*eq*. 11), Train 2 (*eq*. 12) and Train 3 (*eq*. 13). The bandwidth fluctuations measured by these three trains are plotted in Figs. 22, 23 and 24 respectively.

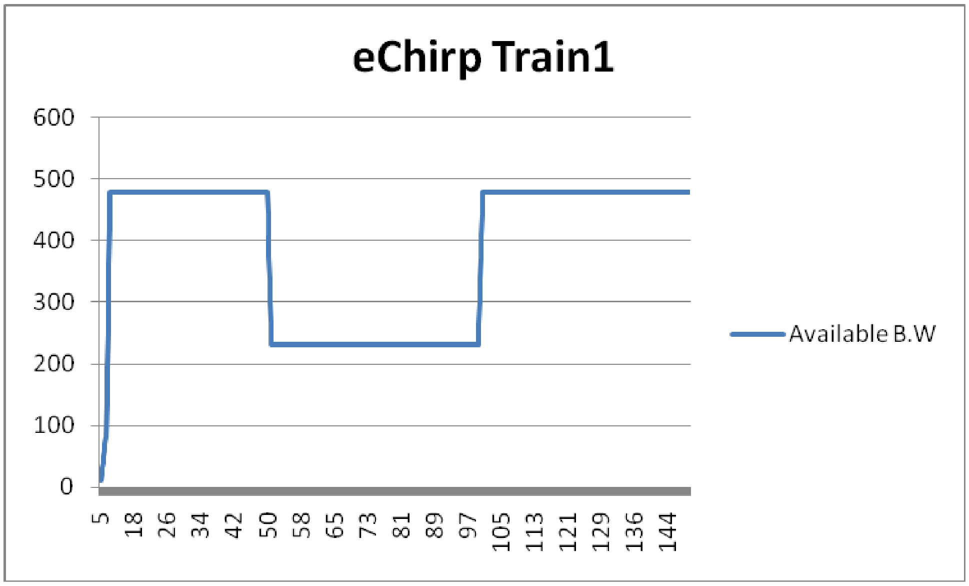


Figure 22: eChirp Train 1 measures available bandwidth of 231 Mb.

V. Simulation result: eChirp packet train 2

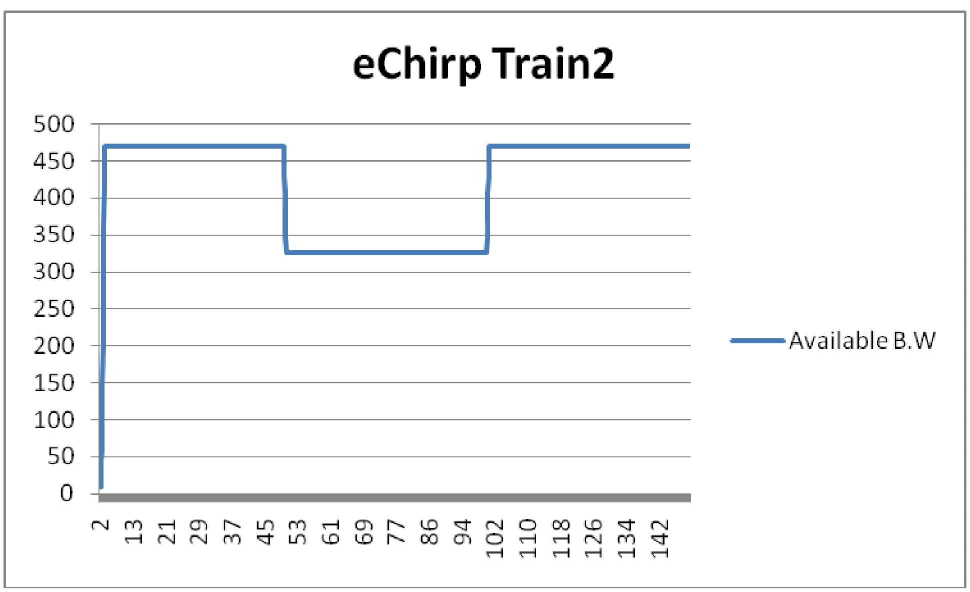


Figure 23: eChirp train 2 measures available bandwidth of 326 Mb.

VI. Simulation result: eChirp packet train 3

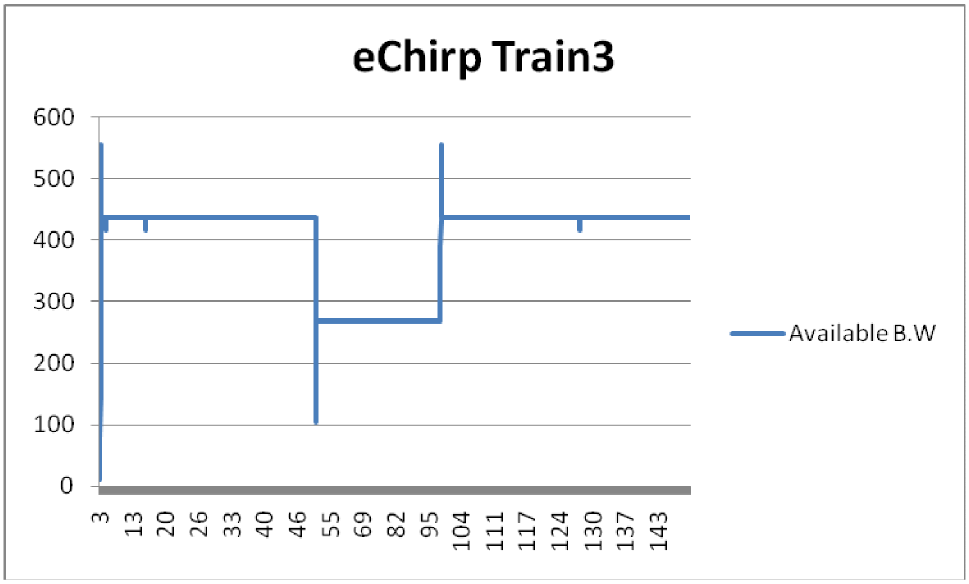


Figure 24: eChirp Train 3 measures available bandwidth of 267, where 250 Mb is expected.

We can see in all of these three cases the available bandwidths are 231Mb, 326Mb and 267Mb which are closer to the actual available bandwidth of 250Mb. With sudden changes in traffic load it is expected to see burst traffic which affects the bandwidth calculation. Although this effect is in all eChirp train but it is visible in the case of Train 3 (see Fig. 24).

We then used our proposed weighted average presented in equation (25) and calculated the overall estimate for the available bandwidth as a combination of three available bandwidths (231Mb, 326Mb and 267Mb) obtained using the three trains. This estimated available bandwidth is 257Mb, which is a good estimate to the actual bandwidth of 250Mb.

VII. Network Topology with 5 Hops

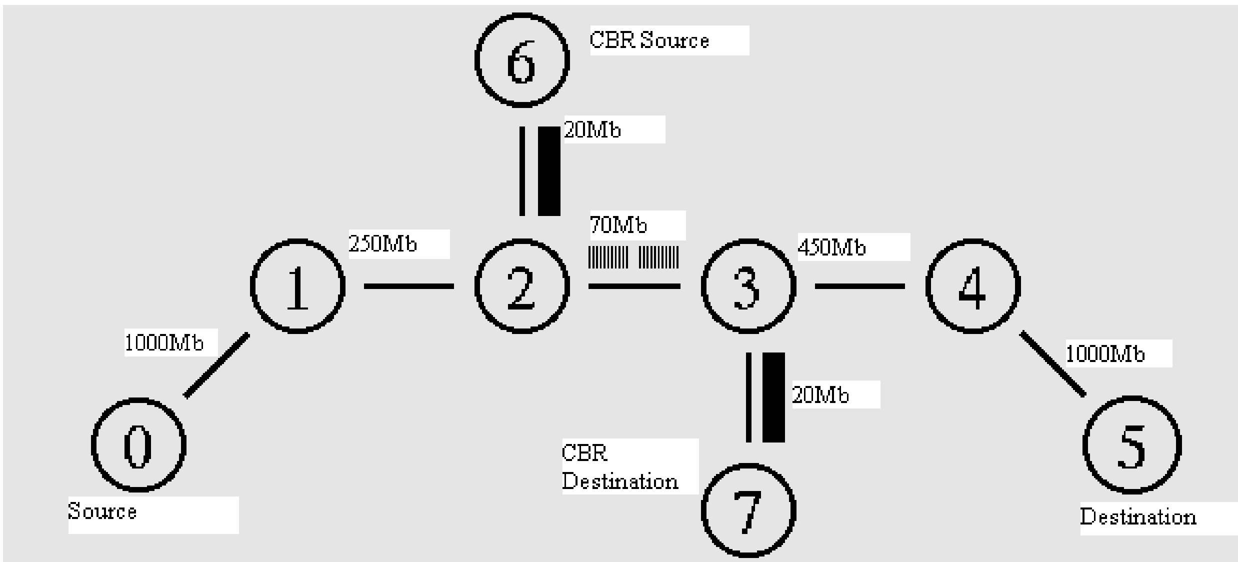


Figure 25: Network topology with 5 hops used with different link parameters.

The above network topology diagram is directly taken from the live simulated network while measuring 50Mb of available bandwidth. The bandwidth between the links is already shown in the figure 25. The noticeable link is the, link between 2 and 3 that has capacity of 70 Mb. We ran the simulation for 160 simulation seconds between source host 0 and receiving host 5. At 50 second, we started 20Mb of CBR traffic between source host 6 and destination host 7. The measured the available Bandwidth of 46.23Mb that is very close to expected bandwidth of 50Mb. We also ran the pathChirp on the same topology with same link bandwidth parameters and observed the bandwidth of 57.61Mb.

We ran many simulations on this topology with different link bandwidth parameters for both eChirp and pathChirp. Results of these simulations are shown below in table 1.

Expected B.W	**eChirp**	**pathChirp**		**eChirp Train1**	**eChirp Train2**	**eChirp Train 3**
50 Mb	**46.23**	**57.61**		36.97	56.31	63.94
70Mb	**76.71**	**82.96**		68.92	81.09	95.72
100Mb	**103.8**	**101.63**		92.67	121.23	119.76
150Mb	**159**	**177.18**		133.94	183.81	209.39
200Mb	**205.6**	**222.43**		180.8	204.7	281.03
250Mb	**254.8**	**306**		216.39	342.86	282.24

Table 1: Measured Available Bandwidth on different links.

CHAPTER VII

CONCLUSION

This thesis presents a new proposed available bandwidth measurement technique, called eChirp, eChirp can measure the available bandwidth over a network path efficiently and accurately with heavy and light load links.

Simulation results presented in the previous chapter shows that the proposed eChirp technique is better than pathChirp scheme in terms of estimating available bandwidth. eChirp captures more network information with its multiple train terminology and gives better estimate than pathChirp with 50% reduction in the number of chirp packets generated. It reduces the load at the source node and estimates the available bandwidth at the sink node.

REFERENCES

[1] Vinay J. Ribeiro, Rudolf H. Riedi, Richard G. Baraniuk, Jiri Navratil,and Les Cottrell. “pathChirp: Efficient available bandwidth estimation for network paths,” In *Passive and Active Measurement Workshop*, April2003.

[2] Manish Jain and Constantinos Dovrolis. “End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput,” *IEEE/ACM Trans. Netw.*, 11(4):537–549, 2003.

[3] J.Curtis, T. McGregor, “Review of Bandwidth Estimation Techniques” http://wand.cs.waikato.ac.nz/pubs.php

[4] C. Ubeda and D. lopez, “Comparision of ABw Techniques in packet switched mobile networks,” in IEEE International Symposium, Indoor and mobile radio communications (PIMRC’06).

[5] V. Jacobson, “Congestion Avoidance and Control," *ACM SIGCOMM*, 1988.

[6] S. Keshav, “A Control-Theoretic Approach to Flow Control," *ACM SIGCOMM*, 1991.

[7] R. Carter and M. Crovella, “Measuring Bottleneck Link Speed in Packet-Switched Networks," *Internation Journal on Performance Evaluation*, 2728, 1996.

[8] C. Dovrolis, P. Ramanathan, and D. Moore, “What Do Packet Dispersion Techniques Measure?," *IEEE INFOCOM*, April 2001.

[9] Bob Melander, Mats Bj¨orkman, and Per Gunningberg. “A new end-toend probing and analysis method for estimating bandwidth bottlenecks,” *IEEEGlobal Internet Symposium*, November 2000.

[10] Vinay Ribeiro. “Spatio-temporal available bandwidth estimation for high-speed networks,”In *Proc. of the First Bandwidth Estimation Workshop (BEst)*, December 2003.

[11] Jacob Strauss, Dina Katabi, and Frans Kaashoek. “A measurement study of available bandwidth estimation tools,” In *Proc. ACM IMC*, October 2003.

[12] R. Mahajan , N. Spring, D. Wetherall, and T. Anderson. “User-level internet path diagnosis,” In *Proc. SOSP*, October 2003.

[13] M Jain, C. Dovrolis, "Pathload: a Measurement Tool for Available Bandwidth Estimation", Proc. PAM’02, 2002.

[14] R.Prasad, M. .Murray, C. Dovrolis, K. Claffy "Bandwidth Estimation: Metrics, Measurement Techniques, and Tools", IEEE Network, November-December 2003 issue.

[15] [online] http://www.isi.edu/nsnam/ns/

[16] Vinay J. Ribeiro, Mark Coates, Rudolf H. Riedi, Shriram Sarvotham, Brent Hendricks, and Richard Baraniuk. Multifractal cross-traffic estimation. In *Proc. of ITC Specialist Seminar on IP Traffic Measurement*, September 2000.

[17] N. and P. Steenkiste. Evaluation and Characterization of Available Bandwidth Techniques. IEEE JSAC Special Issue in Internet and WWW Measurement, Mapping, and Modeling, 2003.

[18] Cesar D and Miguel A. Labrador. “Experimental and Analytical Evaluation of Available Bandwidth Estimation Tools”. 2006

[19] [Online] http://kb.pert.geant2.net/PERTKB/NetworkPerformanceMetrics.

[20] [online] http://irl.cs.tamu.edu/projects/bw