12-9-2006

A Bandwidth Estimation Method for IP Version 6 Networks

Marshall Crocker

A BANDWIDTH ESTIMATION METHOD FOR

IP VERSION 6 NETWORKS

By

Marshall Crocker

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Computer Engineering
in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

December 2006

A BANDWIDTH ESTIMATION METHOD FOR

IP VERSION 6 NETWORKS

By

Marshall Crocker

Approved:

Georgios Lazarou Assistant Professor of Electrical and Computer Engineering (Major Advisor and Director of Thesis)	Joseph Picone Professor of Electrical and Computer Engineering (Committee Member)
Julie Baca Professor at Center for Advanced Vehicular Systems (Committee Member)	Nicholas H. Younan Professor of Electrical and Computer Engineering (Graduate Coordinator)
Roger L. King Associate Dean for Research and Graduate Studies	

Name: Marshall Crocker

Date of Degree: December 8, 2006

Institution: Mississippi State University

Major Field: Computer Engineering

Major Professor: Dr. Georgios Lazarou

Pages in Study: 66

Title of Study: A BANDWIDTH ESTIMATION METHOD FOR
IP VERSION 6 NETWORKS

Candidate for Degree of Master of Science

Efficiently and accurately estimating bandwidths in packet networks is an intriguing problem. There is no simple method for estimating bandwidths in IPv4 networks that is accurate, efficient, flexible, and multi-applicable. Many techniques suffer from flaws such as inaccuracy due to simple assumptions about the network or an overall high complexity that makes it inappropriate in all but a few specific situations.

The next generation Internet Protocol, IPv6, has the framework necessary to implement feedback mechanisms to assist in bandwidth estimations. This thesis proposes a timestamp hop-by-hop option for IPv6 and then applies this option to create a new bandwidth estimation technique. Instead of passive observations, the network infrastructure actively assists in bandwidth measurements resulting in a bandwidth estimation technique that is accurate, efficient, flexible, and suitable for many different

applications and scenarios. Both analytical and simulation analysis show that the IPv6 bandwidth estimation technique outperforms a comparable IPv4 estimation method.

DEDICATION

In memory of my father who always believed in me.

ACKNOWLEDGEMENTS

I wish to thank my major advisor Dr. Georgios Lazarou who has been an inspiration to me throughout my graduate career. Without his constant encouragement and support I would never have reached my full potential. I am forever grateful for all that he has done. I would like to thank Dr. Julie Baca for her invaluable guidance and support during my thesis composition. There is no doubt this thesis would not have been completed without her constructive criticism and direction. I also wish to express my gratitude to Dr. Joseph Picone. I have learned so much from him and have thoroughly enjoyed our discussions.

I am also grateful to everyone in the IES group who made my graduate career a memorable one. I have learned so much from this group and I know I will never forget the impact they all have had on my life. Finally, I want to thank my wife Kimberly. She has always been an unending source of encouragement, confidence, and love.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Current computer communication systems have expanded far beyond the simple networks created just over twenty years ago. No longer are networks restricted to a small subset of possible physical links with only a few distinguishable properties. Instead, the networks of today are complex, varied, and exhibit a wide range of physical characteristics.

One physical characteristic of importance is bandwidth. Bandwidth is not a property that can be measured directly and so it must be estimated. Accurately estimating the bottleneck and/or available bandwidths in network paths is essential for the correct and efficient operation of many Internet applications and protocols as well as network management applications. The following are examples where bandwidth estimations are valuable:

- End-to-end flow control: Hosts use end-to-end bandwidth estimations to determine the rate at which to send data. Increasing estimations indicate ample available bandwidth that the host can move to capture, while decreasing estimations indicate the host must reduce sending rate to avoid congestion.
- Server selection for downloads and streaming media: By estimating the bandwidth to several different servers, a host can determine which server has the highest

potential bandwidth and whether the network has enough available bandwidth to meet the desired rate of the host.

- Peer-to-peer host selection and content delivery: In peer-to-peer networks, bandwidth estimations provide the host with a representation of the network. With this information, the host can select peers that will result in the most timely and efficient transfer of content.
- Multicast configuration protocols**:** In order to perform multicast transmissions, multicast operations must be configured so that the transmissions do not exceed the network capacity or negatively affect competing transmissions. Bandwidth estimations provide multicast configuration protocols with the information necessary to determine a safe and optimal transmission rate.
- Network Management: Bandwidth estimations provide administrators and network engineers with network utilization information in order to troubleshoot networks, redistribute network capacity, reroute network traffic, and plan for future network expansion.

Knowing the capacity of a network path is indeed valuable in a number of situations, but the best methods for measuring bandwidth are essentially limited due to the nature of the network. The disadvantage of current IPv4 measurement techniques is that they attempt to infer characteristics about a network that is not designed to reveal any information to data that traverses it; this is especially true of the Internet. The network simply transports data to its destination as best it can, which is why the Internet is considered a best effort network [1]. There are no guarantees for data traveling through the Internet and there is no way to accurately predict how data will be handled. The only way to determine link capacities and utilization of the links is by examining how the network delivers a single packet or sequence of packets to the destination or to specific hops along the path to the destination.

Despite these limitations, researchers have been able to devise methods that can measure bandwidths in a network, even the Internet, with some degree of accuracy. Each method has its own limitations and tradeoffs as argued in the Chapter 2, but none of these methods provide a simple, accurate, efficient, and flexible solution. Such a solution is simply not possible with the current Internet Protocol. The next generation Internet Protocol, IPv6, has the potential to offer an improved solution through the use of the *hop-by-hop* option and timestamps.

1.1 Problem Statement and Motivation

Each current IPv4 bandwidth estimation technique suffers from one or more of the following drawbacks:

- Increased complexity
- Decreased efficiency
- Decreased accuracy
- Application specificity
- Susceptibility to various factors

All of the IPv4 estimation methods are susceptible to various factors including network load, cross-traffic, packet-size variability, probing packet size, train length, and cross-traffic routing. Some methods sacrifice accuracy for simplicity and efficiency while others gain accuracy at the expense of increased complexity, decreased efficiency, and are generally limited to specific uses. In addition, network technologies such as wireless local area networks (Wi-Fi), wireless broadband (WiMax), cellular broadband (EvDO, HSDPA), mobile networks, asymmetric rate links, and high capacity links contribute to the various factors that negatively affect bandwidth estimation accuracies.

As network technologies continue to evolve, current techniques and methodologies will need to be improved or abandoned in favor of newly developed approaches. This cycle will continue due to the inability of the network to provide explicit feedback about network conditions. Internet Protocol Version 4 (IPv4) lacks the framework necessary to implement feedback mechanisms, while proprietary or nonstandard feedback mechanisms are impractical and undesirable.

Internet Protocol Version 6 (IPv6) is the future designated successor to IPv4. IPv6 addresses the shortcomings of IPv4 by extending the available address space, improving efficiency, reducing complexity, greatly extending the functionality, in addition to other factors discussed in further depth in Chapter 3. Included with the extended functionality is the *hop-by-hop* header which provides a framework suitable for implementing network feedback. The *hop-by-hop* header is suitable for requesting and holding capacity information for each node in a network path. Ideally, the *hop-by-hop* header would be used to request and receive the exact capacity information from every desired node. In reality, the IETF has rejected a similar proposal due to complexity and potential for divulgence of proprietary information. The other possibility is a timestamp option for the *hop-by-hop* header.

A timestamp option was included for IPv4 but has never been useful due to limitations of IPv4, but with IPv6, the full potential of a timestamp option is possible. Used in conjunction with probing techniques developed for IPv4, the IPv6 timestamp option can be used to estimate the network capacity as well as the network utilization for each hop in a network path. The result is a solution that is stable throughout network advancements, is extremely accurate, only minimally affected by various factors, and can be used for a number of applications and scenarios.

The main goals of this research include the following:

1) Propose a timestamp option for IPv6 routers that can be used for estimating network capacity and utilization.
2) Develop a new bandwidth estimation method using the IPv6 timestamp option which is:
   - simple with respect to current IPv4 methods
   - applicable for a variety of network characteristics and scenarios
   - minimal in network resource requirements for correct operation
   - accurate within 10% of the actual capacities
   - minimally impacted by external factors
3) Create simulation models to evaluate the performance of the IPv6 bandwidth estimation method against methods used in IPv4.

1.2 Summary of Main Contributions

The main contributions of this work are as follows:

1) Proposal for IPv6 *hop-by-hop* timestamp option that includes the following capabilities:
    - Record timestamps at incoming, outgoing, or both interfaces
    - Record timestamps from all or specified network hops
    - Record timestamps in high or low granular resolutions
    - Specify timestamp resolution and format
    - Record counter value in timestamp record
2) Development of bandwidth estimation method using a probe packet pair approach and IPv6 timestamp option.
3) Construction of simulation models for evaluation and comparison.
4) Results that show
    - Increased accuracy for IPv6 bandwidth estimations compared to most accurate IPv4 methods
    - Reduced complexity compared to IPv4 bandwidth estimation methods
    - IPv6 bandwidth estimation technique is applicable for a variety of scenarios and network conditions
    - Tolerance to various factors that affect IPv4 measurement techniques

1.3 Organization of Thesis

The organization of this thesis is as follows:

Chapter 2 presents background information and previous work concerning bandwidth estimation theory and techniques. In this chapter, bandwidth terms such as available bandwidth and bottleneck bandwidth are defined and differentiated. Following bandwidth definitions is a discussion of previous work including introduction and categorization of a number of bandwidth estimation techniques developed for IPv4 networks. The Cartouche IPv4 estimation technique is examined in detail along with an overview of some of the fundamental concepts the Cartouche method employs. Chapter 2 concludes with an analysis of IPv4 techniques including a discussion of what constitutes a “good” estimation technique and why IPv4 techniques all suffer from a primary flaw.

Chapter 3 begins with an introduction of IPv6 and a discussion of the major differences between IPv6 and IPv4 that may affect current IPv4 estimation techniques. Next, timestamps are examined including an explanation of why the IPv4 timestamp option was not useful. The IPv6 extension header is examined next followed by the proposal for a timestamp option in IPv6. Finally, the scheme for applying timestamps in IPv6 to estimate bandwidths of network paths is presented.

Chapter 4 presents the experimental portion of this research. The experimental portion is compromised of a description of the simulation scenarios used to evaluate and compare the IPv6 estimation method with the Cartouche method. Following the simulation descriptions is the comparison of the data from both scenarios. The comparisons examine the differences in accuracy, efficiency, and speed. Chapter 5 concludes this work and discusses areas of future research.

CHAPTER II

BACKGROUND AND RELATED WORK

2.1 Bandwidth Terms

Before discussing related work in bandwidth estimation techniques, it is necessary to first clarify the terms *available bandwidth* and *bottleneck bandwidth*. These two terms are often used incorrectly and interchangeably, but are in fact describing two different network attributes. It is important to clarify these terms since most IPv4 estimation techniques can provide estimates for only one metric. The figure below will be used to aid in the explanation of the two bandwidth terms.

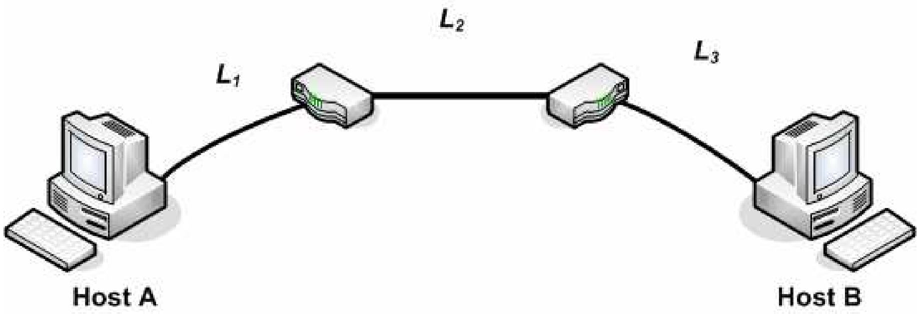


Figure 1 Example network scenario used in illustrating bandwidth terms

2.1.1 Bottleneck Bandwidth

Bottleneck bandwidth is a measure of physical capacity, and is determined by the link with the smallest capacity in a network path. Transfers are limited by the capacity of the bottleneck link for a given network path. The bottleneck bandwidth represents the maximum bandwidth that can be captured between a sender and receiver through that path in the absence of competing traffic. Therefore, a host should never attempt to transmit at a rate that exceeds the bottleneck bandwidth. Since it is a measure of physical capacity, this value does not change in time with the load and dynamics of the network, but remains the same for as long as the communication links in the path do not change.

For example, consider the network in Figure 1. Assume the network path between Host A and Host B is composed of three network links with physical capacities L_1 = 100 Mbps, L_2 = 200 Mbps, and L_3 = 10 Mbps. The bottleneck link is L_3 with the smallest physical capacity of 10 Mbps. The bottleneck bandwidth then is 10 Mbps. Transfers between Host A and Host B can never exceed this bottleneck regardless of network conditions. In [2], the authors use the following equation to define the capacity of a network path P composed of a sequence of first-come first-served store-and-forward links. For a network with a fixed path of H links and the capacity at each link i is C_i bits per second, then the maximum capacity of the path between two hosts S and R is defined as

$$C \equiv \min_{i=1\cdots H} C_i \tag{1}$$

2.1.2 Available Bandwidth

Available bandwidth is a measure of utilization or more sp111ecifically, it is the maximum unused capacity available to a sender at a single point in time along the same path in the presence of competing traffic. The link with the smallest available bandwidth is usually referred to as the *tight link*. A host may transmit at a rate greater than the available bandwidth, but doing so will result in packet loss and thereby cause competing flows to reduce their sending rates. Since it is a measure of utilization, not physical capacity, the available bandwidth of network path will change in time as the available capacity of the network is utilized by traffic between other hosts.

Consider again the example in Figure 1. The bottleneck link for the network path between Host A and B is L_3 = 10 Mbps. If L_3 is utilized 50% per unit time by network traffic, its available bandwidth is 5 Mbps. Now suppose that the link L_2 = 200 Mbps is experiencing large amounts of cross traffic to the point that the link is transmitting data 99% of the time. The available bandwidth then is 2 Mbps. Link L_2 has a smaller available bandwidth than link L_3 but L_3 is still considered the bottleneck link with bottleneck bandwidth of 10 Mbps. L_2 is considered the tight link since the transmission rate is limited by this link at this particular point in time. Often in a scenario such as this, L_2 is incorrectly called the bottleneck link even though the utilization of L_2 may change dramatically in the next instant. The available bandwidth can also be characterized with the following equation:

The available bandwidth for a path with *H* links during a time interval (t_0, $t_0 + \tau$) and u_i average percent utilization at each link *i* is described as321

$$A^{\mathrm{I}}(t_0) \equiv \min_{i \equiv 1 \cdots H} \{C_i[1 - u_i^{\mathrm{I}}(t_0)]\} \tag{2}$$

2.1.3 Applications

Available bandwidth measurements are most useful in applications that must respond to network conditions in order to provide a level of control and optimization. These applications adjust based on the load of the network to achieve the best performance as well as to be fair to competing traffic. Available bandwidth measurements are also useful for network analysis by allowing an individual to measure the load of various network paths. Bottleneck bandwidth measures are most useful in network management situations, such as capacity provisioning and traffic engineering. Bottleneck bandwidth measures are also useful for establishing the maximum rate that a host can transmit for a particular network path to a given host. Both measures are useful in applications needing to establish connections for long term communications. In long term communication situations a host uses bandwidth measurements to choose a server with adequate bottleneck bandwidth and controls the flow of data based on available bandwidth.

2.2 IPv4 Estimation Techniques

2.2.1 Introduction

In the past two decades substantial research has been conducted concerning bandwidth estimation and the network infrastructure and characteristics that attribute to the handling of data packets. Researchers have studied packet delay, packet dispersion, loss behavior, connection scheduling, network dynamics, link characteristics, TCP throughput, flow control, routing behavior, Internet topology, effects of layer 2 devices, and many additional topics that directly relate to or affect estimating network bandwidths. By examining these various aspects of network behavior, researchers have proposed numerous tools and techniques to estimate link capacity and link utilization.

Many techniques have been proven incorrect by later discoveries that contradicted assumptions or hypotheses on which the related technique relied. As a result, new techniques are introduced each year as more knowledge is gained about the network and the data that traverses it. The study of network behavior is an on-going process due to the difficulty of fully understanding complex networks, especially networks such as the Internet. New and emerging technologies add to the complexity as well as increasingly varied hosts and patterns in data communications. This increasing complexity and incomplete knowledge of network behavior has contributed to the large number of estimation techniques in existence. The following section presents an overview of IPv4 estimation techniques, followed by an in-depth examination of a technique applicable to this research.

2.2.2 Overview of Estimation Techniques

Bandwidth estimation techniques can be classified by several different characteristics. The following is a list of the different categories of estimation techniques, and the techniques that fall in the corresponding category:

- Measured attribute
    - Available bandwidth [2] - [7]
    - Bottleneck bandwidth [8] - [15]
    - Both [16], [17]
- Measurement type
    - Active probing [2]-[10], [13]-[17]
    - Passive probing [12]
- Measurement destination
    - End-to-end [2]-[8], [15]-[17]
    - Hop-by-hop [10]-[14]
- Measurement method
    - Single packet [2], [3], [10], [13], [14],
    - Uniform packet pair [8],[15], [16]
    - Non-uniform packet pair [11], [12]
    - Packet train [9], [16]
    - Packet stream [4], [5]
- Measurement analysis
    - Packet spacing [8], [9], [11], [12], [15], [16],

- Packet delay [4], [5]
- Packet dispersion [1], [17]
- Statistical [3], [10], [13], [14]

The first category by which to classify estimation techniques is the measured attribute. Estimation techniques measure either bottleneck bandwidth, available bandwidth, or both. Measurement type indicates whether a technique actively sends probes with the primary objective to measure bandwidth or whether the technique uses normal data packets to passively measure bandwidth. Measurement destination specifies that the technique measures bandwidth at each hop or between two end points.

The measurement method of an estimation technique describes how the technique implements the metrics necessary to gather data for estimating bandwidth. The earliest techniques sent a single probe packet and examined how the network delivered this packet of data. Subsequent methods increased the number of packets to uniform and non-uniform sized packet pair probes, while others used trains of packets or a constant stream of packets. Generally, the measurement method is critical since it dictates many constraints for the corresponding technique. In addition, the analysis of the gathered data is limited by the metrics used to obtain the data.

Many different measurement analysis methods have been investigated. Simple analysis observes the spacing of packets to calculate bandwidth. Researchers assumed that any cross traffic and/or limits in link capacity would introduce spacing between two packets that is representative of the respective available and/or bottleneck bandwidths. Packet delay is an analysis method by which an increase in delay indicates the host is sending at a rate higher than the available bandwidth in the network path. Packet dispersion is another analysis method that examines trends in packet dispersion and then uses these trends to draw a conclusion about the link capacities in the network path. Finally, statistical analysis is performed on measurement data in order to measure link capacities.

2.2.3 Relevant Bandwidth Estimation Techniques

The *Cartouche* method [9] combines the tailgating technique with packet trains in order to overcome some of the problems associated with packet tailgating. The technique works by sending a series of packet trains called cartouches back to back. These cartouches are formed based on the targeted path to measure. The target path can be a path prefix or path suffix. If the measured path is a prefix, that is the first part of a path is measured, the packet train is sent in the form [$pm\{pq)^{r-1}pm$]. The packet *p* is a magnifer packet and is the leading packet of packet-pair. The *m* and *q* packets are tailgating packets and are called the marker packet. *r* is the length of the packet train and is determined experimentally. For a prefix measurement, all of the packets in the cartouches except *m* are set with a time-to-live (TTL) value for the last hop in the prefix. When the cartouches reach the targeted hop, all but the marker packets are dropped by the router. The remaining marker packets continue to a host at the end of the network path. At the end of the path, the separation of the marker packets is measured to calculate the capacity.

When measuring a suffix of a network path, the cartouche is in the form [$p_{i-1}mp_imp_{i+1}m\ldots p_nm$] where L_i is the first link in the targeted sub path and *Ln* is the final hop in the targeted sub path. Magnifier packets *p* are dropped by successive links starting at L_{i-1} and each marker packet *m* is sent to the end hop L_n. At L_n, bandwidth is calculated by examining the spacing between each set of marker packets. The reasoning behind the Cartouche method is that by sending a train of packets, the spacing between the marker packets is better preserved than just two tailgating packets. In addition, the complex forms of the packets trains give the Cartouche method its ability to measure specific parts of a network path.

The *pathrate* method [17] takes an entirely different approach compared the *Cartouche* method. Instead of trying to directly measure the network capacity, *pathrate* examines trends in the dispersion of probe packets at the end host. The creators of *pathrate* demonstrated in [17] that packet-pair bandwidth measurements generally follow a multimodal distribution. They state that the path capacity is a local mode and is different from the global mode of the distribution. They also state that the mean of a packet-train dispersion corresponds to the average dispersion rate (ADR) and the ADR is the lower bound of capacity and upper bound of the available bandwidth of a path. *pathrate* uses this information about ADR to generate sequences of probes and then measures the ADR at the end host to determine the capacity and available bandwidths.

2.3 Analysis of IPv4 Techniques

2.3.1 Properties of a “Good” Technique

As described in the previous section, a number of techniques can be used to estimate network capacities and utilization; however, what techniques constitute a “good” estimation method? Ten fallacies and pitfalls of available bandwidth estimation are presented in [19] as important misconceptions in available bandwidth estimation techniques. In this paper, the authors identify ten misconceptions in available bandwidth estimation techniques that if not considered, can result in erroneous bandwidth estimations. The converse of the 10 fallacies and pitfalls are given below:

- Number of estimation samples and average time scale must be consistent.
- The estimation stream duration controls the averaging time scale.
- Faster estimation is not better.
- Packet pairs are not as good as packet trains.
- Direct probing requires knowledge of tight link capacity.
- Cross traffic burstiness cannot be ignored.
- Multiple bottlenecks must be considered.
- Increasing one way delay does not indicate that the output rate is less than the input rate.
- Iterative probing converges to an available bandwidth range, not a single point.
- Bulk TCP throughput is not acceptable for verifying or evaluating available bandwidth estimation techniques.

A “good” estimation technique considers each of the previous points for the probing and estimating methodology. The previous points are directed toward available bandwidth estimation techniques but the points concerning the use of packet trains, cross traffic, and the consideration of multiple bottlenecks apply to bottleneck estimation techniques as well.

An estimation method that strictly follows these points would not be efficient or flexible but has potential for very high accuracy. Of course, some points can be ignored or modified to increase efficiency or flexibility but the result will be reduced accuracy. As it stands, an estimation technique must be custom tailored to fit the needs of the application to obtain the best combination of efficiency, flexibility, and accuracy.

The available bandwidth estimation presented in [2] has been shown to produce fairly accurate results (although several other techniques are considered comparable in accuracy). This method does not consider the effects of bursty cross traffic which does indeed produce a non-negligible effect on the accuracy of the estimation [20]. This method also uses one way delay as the measurement technique and does not consider multiple tight links in the path. As a result, estimations using this technique are subject to error inducing effects and therefore are not appropriate for many applications.

2.3.2 Drawbacks of IPv4 Techniques

Every IPv4 technique has drawbacks that limit its usefulness. As a result, incorporating bandwidth estimation into an application or management tool requires careful evaluation of the various existing techniques and tools until a suitable technique is found. If no technique can be found to meet the requirements of the application, a new approach must be taken or the requirements must be changed.

The Cartouche method [9] discussed in section 2.2.3 is a relatively accurate technique and useful for its ability to measure targeted segments of the network. It is efficient since it only requires a few tens of packets to make a single measurement. Unfortunately, the technique is limited in that accuracy is inconsistent in various situations. The accuracy of this method is demonstrated in [9] by real-world experiments in which the difference from the actual value ranged as close as 10% inaccurate up to as large as 200% deviation from the actual value. In addition, some hops could not be measured at all. In reality, this method is not useful since there is no way to know which measurements are accurate and which are not.

The *pathrate* tool also discussed in section 2.2.3 is one of the most accurate techniques for measuring bandwidth. Since it does not rely on a single measurement but rather examines packet dispersion, it is able to better estimate network resources. The drawbacks to *pathrate* include increased complexity and decreased efficiency. *pathrate* uses several phases of probing in order to generate enough traffic to analyze for network characteristics. For networks with large capacities, *pathrate* probes have a negligible impact, but for networks with limited capacity, the probes can affect competing traffic flows. In addition, it takes time to gather enough samples before an accurate estimate can be calculated; which is not suitable for applications such as peer to peer networks where connections are constantly being opened and closed. Finally, the accuracy of *pathrate* suffers when the network is loaded to 80% or greater capacity.

2.3.3 Critical Flaw

The previous section discussed the necessary properties of a “good” bandwidth estimation technique but these properties do not address the primary flaw with *all* estimation techniques in IPv4. All estimations techniques in IPv4 attempt to determine the characteristics of the network by examining how the network handles the data that traverses it. This may be by measuring how the data is finally delivered at the end of a network path, or how it is sent back at various hops in the path. The problem arises for all methods by the fact that network conditions and properties affect how data is handled as it is transported.

A simple example that illustrates the problem can be seen in Figure 2. In this example, the sender transmits a train of packets to the receiver. At the first router, the packets are received as they are transmitted but cross traffic or other conditions introduce inflation of packet spacing in the train. At the second router, the packets are received but the packet spacing is deflated back to that of the original. The receiver then incorrectly calculates that the sending rate of the sender is less than the bottleneck and available bandwidths in the path.

This simple example may not occur often, but even one occurrence could significantly affect bandwidth estimations. It also clearly conveys the critical flaw in current estimations of network bandwidths. Until network feedback is implemented, there will most likely never be a single estimation technique that can accurately, efficiently, and easily estimate network bandwidths.

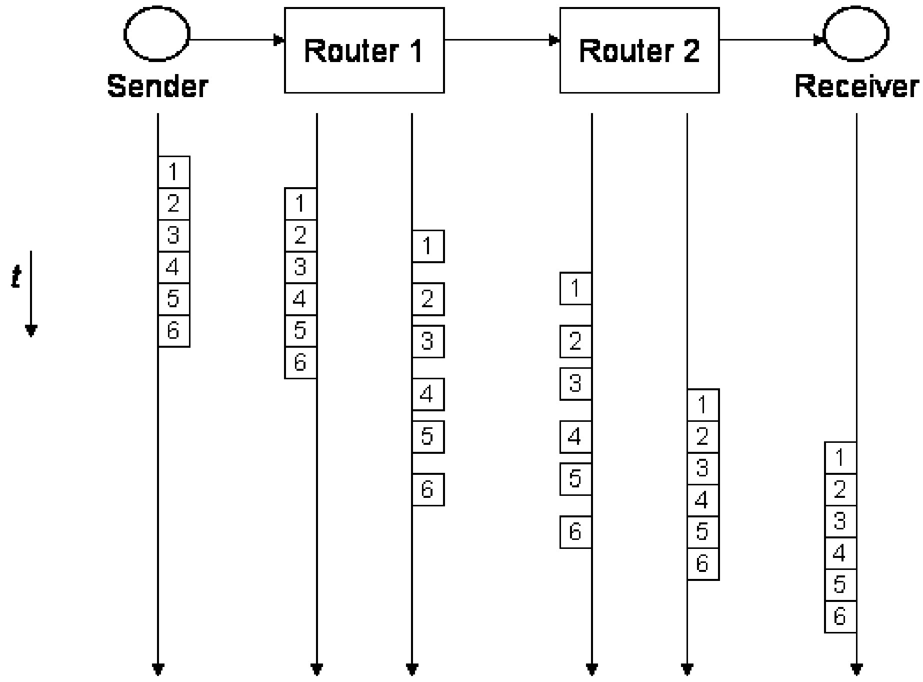


Figure 2 Illustration where packet spacing is not preserved in a network path resulting in incorrect bandwidth estimation

CHAPTER III

IPV6 BANDWIDTH ESTIMATION TECHNIQUE

3.1 IPv6

3.1.1 Overview

Internet Protocol version 6 (IPv6) [21] is the next generation Internet Protocol designed to be the successor to version 4 (IPv4), although IPv4 may never completely go away [22]. Full deployment of IPv6 can be expected to complete within the next 10 years or sooner, depending on the push from other countries and government agencies, not to mention the demand due to the increase in Internet capable devices, and the constant threat of router meltdown due to unmanageable routing tables [22]. Many countries in Europe and Asia have already begun deployment of IPv6 to meet the demand for unique IP addresses. In addition, the US government has mandated a move to IPv6 by 2008 [23] [24], while initial deployment of IPv6 has already begun with the establishment of 6bone [25], the IPv6 Internet backbone.

When IPv6 is fully deployed, many of the current tools and techniques for measuring bandwidths should still be applicable. However, IPv6 differs in enough respects from IPv4 that certain key assumptions may not hold true, requiring some techniques to be reevaluated and adjusted. More specifically, IPv6 primarily differs with

IPv4 in that it offers expanded addressing, simplified header format, and improved extension and option support [22].

These differences should not present any immediate problems for current IPv4 measurement techniques. The differences that will necessitate reevaluation include the increased header size, MTU, and fragmentation properties of IPv6. The header size in IPv6 is fixed at 40 octets whereas the IPv4 header is variable between 20 and 40 octets. The minimum MTU for IPv6 is set to 1280 octets, more than double the 576 octets of IPv4. The tailgating technique described in [11] uses a smaller packet followed by a larger packet in order to give a higher probability that the two packets will be queued adjacently at each router. The increase in the IPv6 header size will reduce the minimum packet size, and therefore reduce the large packet to small packet ratio. This ratio is critical in maintaining back-to-back queuing in network paths where adjacent link capacities increase by a factor greater than the lead/tailgate packet ratio. One would expect the larger MTU of IPv6 to increase the ratio, but most researchers assume that packets of 1500 octets can be sent without fragmentation since this is true of most networks today. Therefore, the commonly used ratio is 1500 octets to 40 octets for IPv4, but this ratio will be reduced to 1500 octets to 60 octets in IPv6.

One benefit of IPv6 is that it eliminates fragmentation at the router level. Hosts must not send packets larger than the MTU of the receiver or an ICMP error message will be returned to the host. The elimination of fragmentation at the router level was decided in order to reduce the complexity and improve efficiency for all routers in the Internet. This decision should prove to benefit bandwidth estimations as the overall network traffic will have a more consistent composition, and may result in more accurate assumptions about the network.

3.1.2 Timestamps

A timestamp option was initially defined for IPv4 in RFC 760 [26]. This original definition was incomplete and only introduced the timestamp specification as part of the DoD standard Internet Protocol. Not until RFC 781 [27] was the timestamp option completely specified. The developers of IPv4 understood the unpredictable behavior and variable delays that are a characteristic of packet switched networks. The timestamp option was meant to provide a solution for critical performance measurement in these networks. Unfortunately, the timestamp option was never very practical and lacked sufficient resources to be useful.

The primary reason the timestamp option failed is because IPv4 only allows up to 40 octets for options. This gives enough room for nine four byte timestamps after accounting for the flag fields. Timestamps without an associated router are useless unless the network path never changes. To address this, the timestamp option defines a flag to record the timestamp and IP of the router, resulting in a total of only 4 router/timestamp pairs. Room for only four hops is not practical in Internet paths that commonly have a dozen or more hops. Also, no control or indication of the accuracy of timestamps at a given router [28] can be specified, further adding to the problem. In addition to each of these shortcomings, IPv4 routers tend to service packets with options more slowly than normal traffic since they are optimized to handle standard packets [22]. In theory, the

IPv4 timestamp option promised to be a valuable feature, but the lack of resources and hardware support negated its utility.

3.1.3 IPv6 Extension Header

IPv6 is much better suited to handle a timestamp option through the use of the hop-by-hop extension header. Extension headers in IPv6 are additional headers added after the main IPv6 header. There is no limit to the number or size (up to the MTU) of these headers. Some of the extension headers that are currently defined include the hop-by-hop header, routing header, fragment header, destination options header, authentication header, and encapsulating security payload header. The IPv6 specification allows for additional headers to be defined in the future as need arises. In addition, IPv6 routers only inspect packets that contain a hop-by-hop extension header, unlike IPv4 where every packet with an option must be inspected. As a result, IPv6 routers should better service packets with options since the IPv6 header is streamlined and routers must only examine packets with the Hop-by-Hop option. The CPU cycles saved by the streamlined header will be more than enough to handle Hop-by-Hop packets in a timely manner.

The hop-by-hop extension header offers a better structure than IPv4 options and has sufficient resources to be a viable option for gathering information about the network. When a router encounters a datagram with a hop-by-hop header, it must inspect the header and act according to the IETF specification for the option. The IETF currently has two options defined for the hop-by-hop header: jumbo payload option and router alert option. There have been a few other proposals for hop-by-hop options but they have been denied by the IPv6 working group within the IETF.

Two of the proposed options include a traceroute [29] option and an option for Connection/Link Status Investigation (CSI) [30]. The former option is based on the IPv4 record route option that provides a mechanism to record the forward path to a host. The IPv6 traceroute option included this same functionality along with the ability to record the return path using an ICMP reply. This draft was not accepted in its present form due to concerns about a possible Denial of Service attack [32]. The working group agreed that work should continue but there were never any subsequent drafts.

The CSI draft proposed a mechanism to gather information about nodes along a communications path. The information that could be gathered included interface attributes and statistics such as IP address, speed, type, number of transmitted and received octets, number of transmitted and received packets, and number of discarded and erroneous packets. A packet passing through a router could use the CSI option to gather any of the available information for each interface with which the packet came in contact. CSI also defined a timestamp option that could be inserted in a datagram for each interface as well.

The CSI option would have been an invaluable tool for investigating and analyzing links in communication paths, but it was not accepted for several reasons. The IPv6 working group viewed the CSI option as a potential security and denial of service problem [33]. They also commented that the majority of this information is considered proprietary by ISPs and they would not be willing to reveal that amount of detail about their network interfaces. Lastly, the working group felt that such functionality would be too complicated to implement for most routers, if it was possible at all.

At the time of writing this thesis, no further drafts have been proposed to the IETF regarding timestamps in IPv6. This thesis contends that although timestamps were not practical due to the limitations of IPv4, IPv6 has the ability to fully support timestamps, thus the benefits of a timestamp option are well justified. The next section introduces an IPv6 timestamp option that expands on the original IPv4 specification by incorporating some of the improved features of the CSI mechanism without presenting security, complexity, or proprietary technology issues.

3.2 IPv6 Timestamp Option

3.2.1 Overview

The following timestamp option takes attributes from RFC 781 and the CSI mechanism. In order for a timestamp option to be practical it will need the following capabilities:

- Timestamp with millisecond or better resolution
- Specification of timestamp resolution and format
- Ability to record interface ID
- Adequate space to include all hops in a path

Timestamps should have a millisecond or better resolution in order to guarantee that each packet can be uniquely time stamped. That is, minimum size packets that are sent or received back-to-back must have unique timestamps that are representative of the instance in time that they are handled. No two packets should ever have the same timestamp. Clocks need not be synchronized at each router since multiple packets can be sent and the relationship of the timestamps between packets used to calculate network delays. The IPv6 estimation method does not force any specific timestamp format due to the differences in hardware; therefore, it requires an ability to indicate the format and resolution. The ability to record interface IDs is necessary to match timestamps with the appropriate hop, although with IPv6 it is possible to specify a predetermined route using the routing extension header.

In addition to a timestamp option, ICMP messages may need to be defined to request and report timestamps. The ICMP reply message could be modified to copy the timestamp header from the request message into the reply message. In this manner, any host that is reachable with a ping message has the capability to return timestamp information about the incoming and return path to the sender. Integration into an ICMP message will eliminate the need for special software running on the destination host as well. The CSI mechanism defines ICMP messages to be used for link investigation but for the sake of brevity, it is assumed that a modified ICMP echo message exists that can carry the timestamp hop-by-hop extension to the destination host and back.

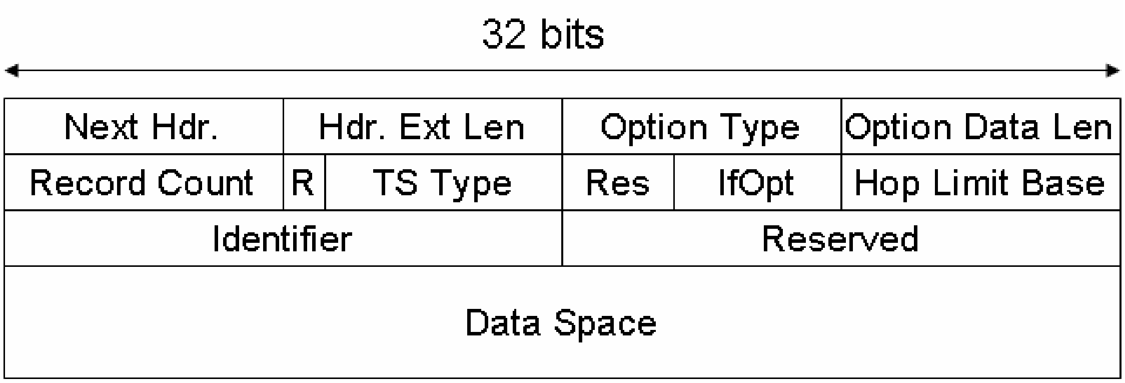


Figure 3 Proposed IPv6 hop-by-hop timestamp option header

3.2.2 Timestamp Hop-by-hop Format

The proposed timestamp option format is shown in Figure 3. This figure represents fields in the hop-by-hop extension header to specify desired handling of the timestamp option and it also includes fields for holding information about this packet.

Each of the fields is described as follows:

- Option Type: 8-bit integer identifying this option as the timestamp option. The IPv4 timestamp option type value is 68 but this cannot be used for IPv6. The IPv6 option must start with 001 indicating that routers should skip this option if they do not support it and that the data in this option may change en route to the destination.
- Opt Data Len: 8-bit length of the timestamp option in number of octets. Option length starts from record count to the end of data space.

- Record Count: 8-bit unsigned integer indicating the number of records contained in data space. Each router must increment this field after inserting a record in the data space. The position for the next record can be calculated using [Record Count] * [Record Length] + 12.
- TS Type: The TS Type field indicates the desired timestamp behavior. This is an 8-bit field with the upper bit R specifying a high resolution timestamp (finer than 1 millisecond) or a normal resolution timestamp (1 millisecond or less). The TS Type currently has 3 values specified below that follow the flags field in RFC 791. The unused values for this field are reserved for future use:

  0 – insert timestamp record only.

  1 – insert Internet address of the registering entity before the timestamp record.

  3 – the Internet address fields are pre-specified. An IP module only registers a timestamp if it matches its own address with the next specified Internet address.

- R = 0 indicates that the registering entity should use a normal resolution (1 millisecond or less) while R = 1 indicates a higher resolution finer than 1 millisecond is desired if available. Entities that are not capable of greater than 1 millisecond resolution should insert a normal timestamp and indicate the resolution in the timestamp record.
- IfOpt: 6 bit integer indicating the interface where the timestamp should be recorded:

  0 – reserved.

  1 – timestamp at incoming interface.

2 – timestamp at outgoing interface.

3 – timestamp at both interfaces.

If the registering entity does not have the ability to timestamp immediately after receiving or before sending, this must be indicated in the timestamp record. The upper 4 bits of this option are reserved for future use.

- Hop Limit Base: this field is a copy of the initial hop limit field in the IPv6 header. This value is not decremented at each hop, but is used to calculate the Hop Number in the timestamp record.
- Identifier: 16-bit unsigned integer used to distinguish this packet from other probes.
- Data Space: This space contains the timestamp records as described in the next subsection.

3.2.3 Timestamp Record

Each timestamp record contains not only the timestamp, but also information about the timestamp including timestamp resolution, format, interface, link type, hop number, internet address and count. The timestamp record format can be seen in Figure 4. The first two fields of the timestamp record consist of the upper and lower portions of the IPv6 address. These fields are not included as part of the timestamp record when a TS Type of 0 is specified in the options. When a TS Type of 3 is specified, then the address fields will be pre-filled by the sending host and entities that match the address field will insert the timestamp after the address.

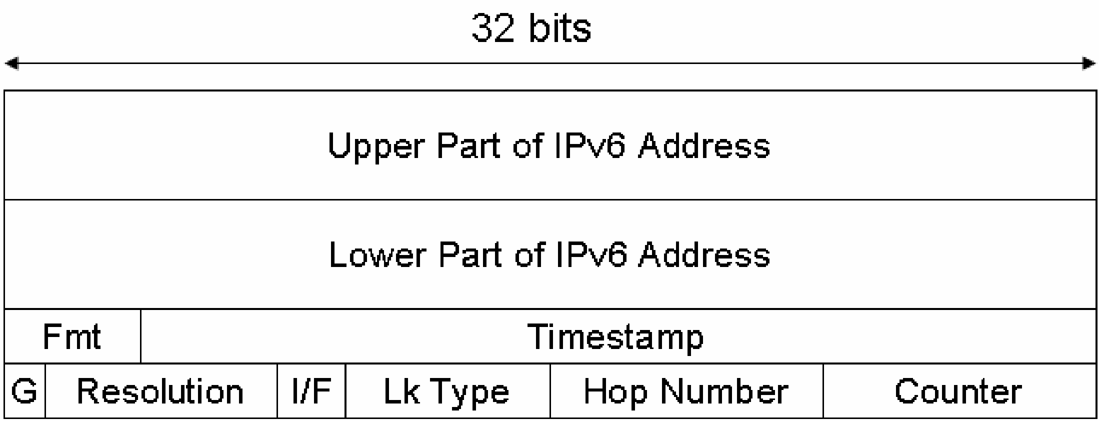


Figure 4 Proposed IPv6 timestamp record format.

Each of the other fields is described as follows:

- Fmt: 4-bit field that specifies the format of the recorded timestamp. Since all hops cannot be guaranteed to be using the same time format, this field will reveal the formatting of the timestamp at this hop. The only format currently defined is the number of milliseconds that have elapsed since midnight UTC. This format is specified with a value of 0001.
- Timestamp: 28-bit unsigned integer. The format of the timestamp is determined by the capabilities of the recording entity. If the R bit is set in the timestamp options then the recording entity must use a timestamp with a resolution finer than 1 millisecond if possible.
- Resolution: 7-bit unsigned integer. The resolution of the timestamp is specified as a multiple or divisor of 1 millisecond. For a low resolution where the timestamp is

in units greater than 1 millisecond, then G=0 and the timestamp is multiplied by [Resolution] to get the timestamp in units of milliseconds. When the timestamp is a high resolution timestamp finer than 1 millisecond, G=1 and the timestamp is divided by [Resolution].

- I/F: 2-bit unsigned integer. Designates the interface where the timestamp was recorded:

  =00 Neutral (independent of interface).

  =01 Timestamp at incoming interface.

  =10 Timestamp at outgoing interface.

  =11 Reserved.

- Link: 6-bit unsigned integer. Specifies the link layer type used for framing packets at the specified interface. This value is used for calculating bandwidths since the link layer frame must be transmitted over the physical layer as well.
- Hop Number: 8-bit unsigned integer calculated using the formula [Hop Number] = [Hop Limit Base] – [Hop Limit].
- Counter: 8-bit unsigned integer indicating the packet count at this hop. Each hop increments an internal counter for every packet sent. The lower 8 bits of this counter is placed into the counter field of the timestamp record. The counter field of two back-to-back probe packets can then be used to determine if the packets were indeed sent back-to-back at each hop.

3.3 Bandwidth Measurements

Now that a timestamp option has been defined, the next step is to apply the timestamp option to measure both available and bottleneck bandwidths in a network path. With the aid of timestamps, available bandwidth can be measured trivially, but bottleneck bandwidth measurements require a little more work including techniques from methods used in IPv4.

3.3.1 Bandwidth

Before discussing the IPv6 timestamp technique for measuring, first bandwidth must be defined in respect to how it relates to the transmission of packets. The capacity or bandwidth of a link can be defined using the following equation:

$$C = \frac{b}{t} \tag{3}$$

where *b* is some number of bits and *t* is some amount of time. Simply stated, the bandwidth is equal to the number of bits that can be transmitted per unit time. This equation can be further expanded to the following:

$$C = \frac{b}{t_2 - t_1} \tag{4}$$

where t_2 is the final transmit time and t_1 is the beginning transmit time of some number bits *b*. Based on this equation, if the beginning and ending times for transmitting a sequence of bits is known, the capacity of the link in bits per unit time can be easily calculated.

3.3.2 Available Bandwidth

As stated earlier, available bandwidth is the unused capacity in a network path between two hosts. Available bandwidth can also be described as the maximum throughput a sender can achieve to a destination host without affecting competing traffic. Available bandwidth measurements are quite useful for network applications that want to send data at the highest rate possible without burdening the network by causing competing flows to lose and retransmit packets. In fact, applications using available bandwidth measurements to control their sending rate can be more efficient than TCP flows since TCP control mechanisms continually force the network beyond its limit before backing off.

To calculate available bandwidth, equation (4) is employed by using packet pairs to obtain the time interval for transmitting a packet at each hop. For a path of *n* physical links $L_1, L_2, \ldots, L_n$, two back-to-back probes are sent denoted by [fs], where f is the first packet and s is the second packet in the packet pair. The size of the probe packets should be representative of the data being sent by the application desiring the measurement. The available bandwidth for the entire path as seen by the sender is

$$A = \min_{i \le i \le n} \left(\frac{s(f) + k}{d(s)_i - d(f)_i} \right) \tag{5}$$

where *s(f)* is the size of the first packet, *k* is the size of the link layer frame overhead and *d(s)*, *d(f)* are the departure times of the second and first packets respectively.

Figure 5 illustrates an example where the packet pair is sent through a path made up of five links. All links are of equal capacity except for link *L3*. This link has a higher capacity and can therefore transmit packets faster than all the other links. Due to congestion, however, other packets are queued behind packet *f* and as a result, packet *s* must wait in the queue for some time before being transmitted to the next hop. The departure times at *L3* then determines the maximum rate a sender can achieve through this path. Effectively, it takes a time of *d(s) – d(f)* to transmit packet *f* across *L3* since the probe packets were separated at *L3* even though the packet pair was initially sent with no separation. If the traffic dynamics were to remain constant and the router had an infinite queue, any attempt to send packets at a faster rate than the rate observed at *L3* would only further congest the router and the throughput seen by the sender would drop. In a real world situation, the router would drop packets from the queue. Subsequently, all TCP and TCP-friendly flows would reduce sending rates, and the abusive sender would eventually capture more of the available bandwidth.

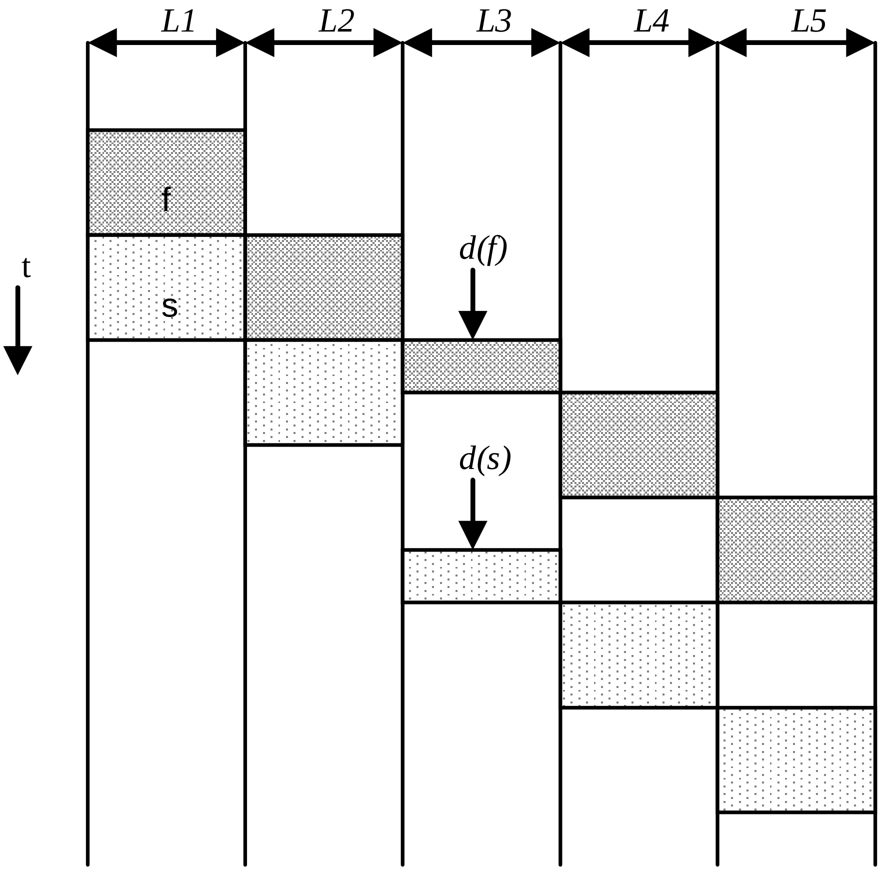


Figure 5 Packet-pair probe separated by congestion at *L3* produces incorrect bandwidth calculation when probes reach *L5*

In this illustration, the separation *d(s) – d(f)* is preserved all the way to the destination. Many IPv4 measurement techniques rely on the preservation of packet spacing from the host to the destination; however, this is not practical due to the unpredictable behavior of the network. Time stamping packets at each hop will allow observing the packet spacing at each link and the preservation of packet spacing will not be necessary.

3.3.3 Bottleneck Bandwidth

In the available bandwidth measurements, the approach investigated in this thesis relies on competing traffic to introduce separation between probe packets to indicate the amount of bandwidth available through a network path. For measuring bottleneck bandwidths, the opposite is required to prevent competing traffic from separating probes. If two probe packets are queued back-to-back at each hop through a network path then the bottleneck bandwidth for this path can be calculated using (5). The departure time of the second packet is the completion time for sending the first packet and therefore the differences in these times is the interval it takes to send the first packet as given in (4).

In order to increase the probability of back-to-back queuing, the packet tailgating technique described in [11] is used. The tailgating technique sends a large pacer packet immediately followed by a much smaller tailgating packet. The tailgating packet has a much higher probability of being queued behind the pacer packet since the amount of time for other packets to be received and queued while the tailgating packet is being transmitted is small. Obviously, back-to-back queuing cannot be guaranteed through a network path but measurements can be repeated until two packets are sent back-to-back as indicated by the *Count* field in the timestamp record.

While timestamps will make it possible to measure transmission time intervals and ultimately calculate link capacity, they will also increase the minimum size of tailgating packet and greatly reduce the probability of back-to-back queuing. In IPv4, a common scenario for a tailgating packet pair is a 1500 octet pacer packet and a 40 octet tailgating packet. This gives a ratio of 37.5 and means that a link can only be 37.5 times faster than the previous link to ensure the two packets are queued and sent back-to-back [11]. This ratio is substantially reduced for an IPv6 packet with the timestamp option. The minimum size for an IPv6 timestamp packet is 40 octets for the IPv6 header, 12 octets for the hop-by-hop extension header with the timestamp option and fields, and 8 octets for each timestamp record without recording IP addresses. For a 20 hop path, a timestamp probe would have a minimum size of 212 octets reducing the ratio to 7. This ratio reduces even more if interface addresses for each hop are recorded.

The most accurate but least efficient way to measure capacities is to measure each hop individually. Each hop can be measured by setting TS Type = 3 in the timestamp options and insert the IP address of the desired hop to investigate. The resulting packet is a minimum of 76 octets and the ratio is around 20. This will limit the ability to accurately measure capacities in a network path where two consecutive link capacities increase by a factor greater than 20.

A final option is to use packet trains and use the beginning and ending packets to calculate the bottleneck bandwidth. The size of the first packet then is equal to the size of the first packet plus all of the following packets before the final packet. The time difference between the first and final packet is the total time it took to transmit all the packets before the final packet.

CHAPTER IV

EXPERIMENTAL ANALYSIS

4.1 Simulations

In this section, the details of simulation experiments are given for simulation scenarios using the proposed IPv6 timestamp option and the Cartouche IPv4 estimation method. The two bandwidth estimation methods are simulated in identical scenarios and the results are compared. The Cartouche method was chosen since it is most comparable to the IPv6 timestamp method in that it performs active, end-to-end probing, uses a form of the packet pair technique, and is more accurate than any other comparable IPv4 technique.

4.1.1 Experimental Setup

Simulations were performed using the MLDesigner [31] simulation platform. A network model was created consisting of a single 20-hop path connecting two hosts *A* and *B* as seen in Figure 6. Network bandwidths consisted of 100 Mbps for each hop with the bottleneck hop ranging in bandwidth from 10 Mbps to 100 Mbps. At each hop, cross traffic flows were modeled using a Poisson process with packet sizes determined by an exponential random distribution with a mean packet size of 200 octets. The mean rate of the cross traffic flows ranged from 100 Kbps to 1 Mbps. Table 1 below gives a summary of the parameters used in each simulation scenario.

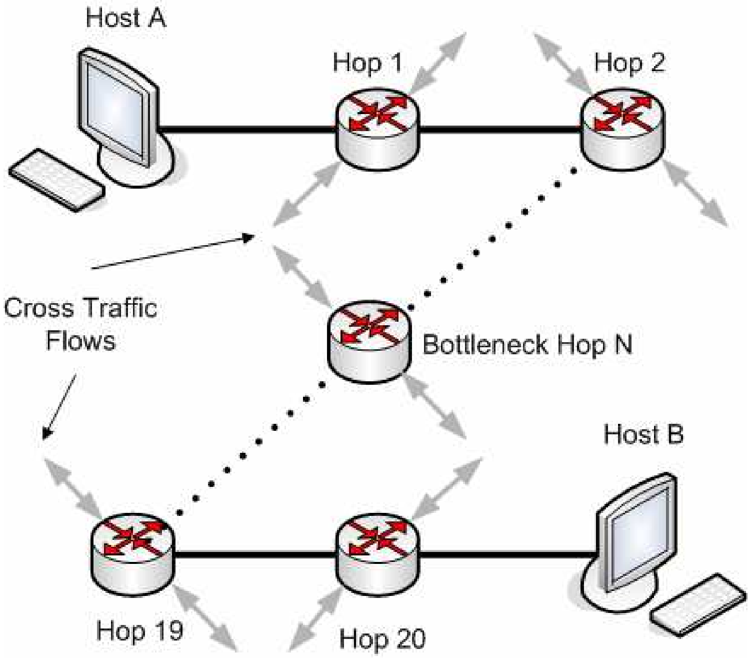


Figure 6 Network configuration for bandwidth estimation simulation scenarios

Table 1 Parameters for IPv6 and Cartouche simulation scenarios

Parameter	Values
Number of Cross Traffic Flows	8, 16, 24, 32, 40, 48
Cross Traffic Rates	100 Kbps, 1 Mbps
Average Cross Traffic Size	200 Bytes
Leading Probe Packet Size	1500 Bytes
Bottleneck Rates (Mbps)	10, 20, 30, 40, 50, 60, 70, 80, 90, 100
Hop Rates	100 Mbps

Cross traffic originating at a hop passes through the originating hop and continues to the next hop where a binomial random distribution determines if the packet continues through the network. The probability that a packet continues to the next hop is 0.1 thereby allowing 1/10 of all cross traffic from previous hops to pass the current hop. The cross traffic characteristics can best be described using the equations below:

$$R = \lambda\beta \tag{6}$$

$$\lambda = \frac{1}{\mathrm{E[X]}} \tag{7}$$

$$H_{i \cdots N} = (.1)H_{i-1} + R \tag{8}$$

In equation (6), *R* is the desired cross traffic rate in bits per second (bps). *R* is calculated by multiplying the average number of packets generated per second (λ) with the average packet size in bits (β). λ is calculated using 1/E[X] where E[X] is an exponential random variable that describes the inter-arrival rate between packets. The total cross traffic at any hop *i* is given in (8) whereby the cross traffic rate *R* at the current hop is added to the rate of the previous hop multiplied by 10%.

With a mean cross traffic rate of 100 Kbps, the aggregate cross traffic rate at the first hop in the network topology shown in Figure 6 results in 800 Kbps, 1.6 Mbps, 2.4 Mbps, 3.2 Mbps, 4.0 Mbps, and 4.8 Mbps for the respective number of cross traffic flows given in Table 1. A cross traffic rate of 800 Kbps for a network with a bottleneck bandwidth of 10 Mbps represents a lightly loaded network with the load increasing as the number of cross traffic flows increases. A network with 8-1 Mbps cross traffic flows represents a fairly heavily loaded network while 48-1 Mbps cross traffic flows is enough to cause the network queues to overflow and drop a large amount of data. These scenarios provide various network conditions for evaluating bandwidth estimation methods.

4.1.2 IPv6 Simulation Scenarios

Simulations for the IPv6 timestamp method were performed with the parameters and values as shown in Table 1. Various scenarios were executed with the tailgating packet size as given in Table 2. Each hop in the network model recorded the departure time for each probe packet and also inserted the value of an internal counter into the *Count* field. Bandwidth estimations were only made for a hop if the probe packet pair was sent back-to-back from that particular hop as indicated by the *Count* field in the timestamp record. If the probe packet pair was sent back-to-back, equation (3) was used to calculate the bottleneck bandwidth.

Table 2 IPv6 simulation scenarios

Scenario	CT Rate	Tailgating Packet Size (Bytes)
1	100 Kbps	212
2	100 Kbps	76
3	1 Mbps	212
4	1 Mbps	76

4.1.3 Cartouche Simulation Scenarios

Identical scenarios were used for the Cartouche probing experiments with the parameters the same as given in Table 1 and scenario values described in Table 3. The additional parameter *r* is the length of the Cartouche train. According to the creators of the Cartouche method, a higher value of *r* yields more accurate results. Values chosen for *r* were 2 and 3. *r*=1 would more closely resemble the probe packet pair of the IPv6 method but the creators for the Cartouche method discovered that *r=1* produces inaccurate results. *r*=2 increases accuracy while *r=3* produces the most accurate results. The bandwidth is estimated from the average inter-arrival time of the probe marker packets at host *B*.

Table 3 IPv4 Cartouche simulation scenarios

Scenario	CT Rate	Tailgating Packet Size (Bytes)	Cartouche Length (*r*)
1	100 Kbps	40	2
2	100 Kbps	40	3
3	1 Mbps	40	2
4	1 Mbps	40	3

4.2 Results

4.2.1 Accuracy

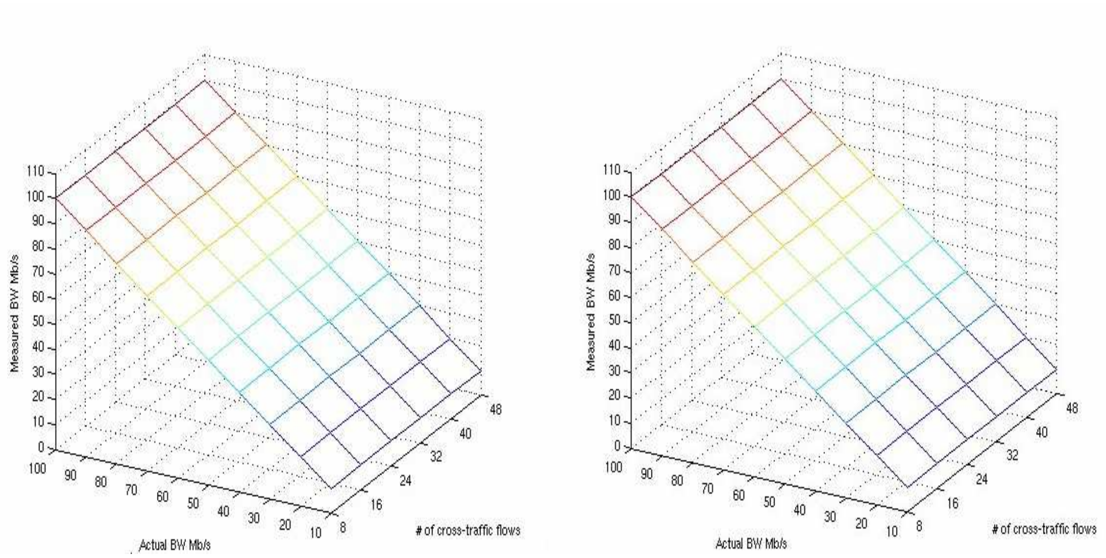
The first metric analyzed for the IPv6 and Cartouche methods is accuracy. The IPv6 estimation method’s ability to discern and ignore separated probe packets guarantees that only valid back-to-back probes are used for bandwidth calculations. In addition, the timestamp records provide data that allows for exact network capacity estimates. Table 4 presents a summary of the results of the IPv6 and Cartouche simulations. For each scenario, 100 estimations were performed and the average bandwidth was calculated for each cross traffic flow value and actual bandwidth combination. This table shows the average bandwidth estimates for the minimum and maximum cross traffic flows, i.e., 8 and 48, with the bottleneck link in the network varied from 10 Mbps to 100 Mbps. Results are reported for two scenarios, 1 and 4, for both the IPv6 (described in Table 2) and Cartouche method (described in Table 3). These scenarios were chosen because they represent the worst performance under a light 100 Kbps cross traffic load, i.e., Scenario 1, and the best performance under a heavy 1 Mbps cross traffic load, i.e., Scenario 4.

Table 4 Bandwidth calculation results

Number of CT Flows	Actual Bandwidth	IPv6 Calculated BW (scenario 1)	Cartouche Calculated BW (scenario 1)	IPv6 Calculated BW (scenario 4)	Cartouche Calculated BW (scenario 4)
8	10	10.00	9.90	10.00	9.01
	20	20.00	19.85	20.00	18.16
	30	30.00	29.50	30.00	27.13
	40	40.00	39.92	40.00	36.03
	50	50.00	49.68	50.00	44.96
	60	60.00	59.26	60.00	54.41
	70	70.00	69.45	70.00	64.83
	80	80.00	78.05	80.00	71.55
	90	90.00	85.84	90.00	75.75
	100	99.98	90.40	100.00	75.68
48	10	10.00	9.44	10.00	3.96
	20	20.00	18.90	20.00	8.48
	30	30.00	27.95	30.00	16.58
	40	40.00	38.33	40.00	21.20
	50	50.00	47.12	50.00	23.26
	60	60.00	56.34	60.00	25.32
	70	70.00	65.41	70.00	27.86
	80	80.00	71.13	80.00	33.02
	90	90.00	73.34	90.00	36.01
	100	100.00	75.34	100.00	38.06

The IPv6 and Cartouche methods are close in accuracy in scenario 1 when the bottleneck bandwidth is less than 80 Mbps. When the bottleneck bandwidth increases to 80 Mbps, the network bandwidths begin to equalize and the Cartouche probes experience larger separation at hops other than the bottleneck. The IPv6 method is immune to the errors caused by packet separation and therefore is still accurate within 1% of the actual values. As the cross traffic rate is increased to 1 Mbps, the Cartouche method is significantly affected since the Cartouche probes experience decompression caused by the higher cross traffic rates. Again, the IPv6 method is still accurate despite the increase in cross traffic.

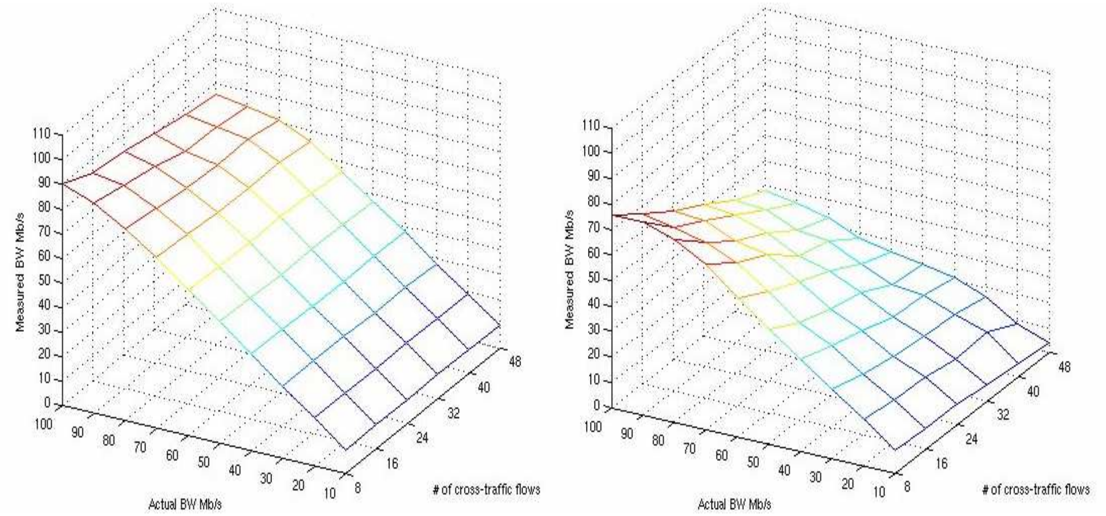
Figures 7 and 8 below present the accuracy information in graphical form. The X and Y axes represent the independent variables: the X axis represents the number of cross traffic flows while the Y axis represents the actual bandwidth. Finally, the dependent variable, measured bandwidth, is represented on the Z axis. In Figure 7 the graphs are a flat plane since the measured and actual bandwidths are equal and therefore produce a straight line. Figure 8 shows the results from the Cartouche measurements. At lower bandwidths, the Cartouche method is most accurate and the graphs resemble that of the IPv6 methods. As the bottleneck bandwidth and number of cross traffic flows increase, accuracy is severely affected and the plane becomes distorted.



a) Scenario 1

b) Scenario 4

Figure 7 Measurement results from IPv6 scenario 1 (a) and scenario 4 (b)

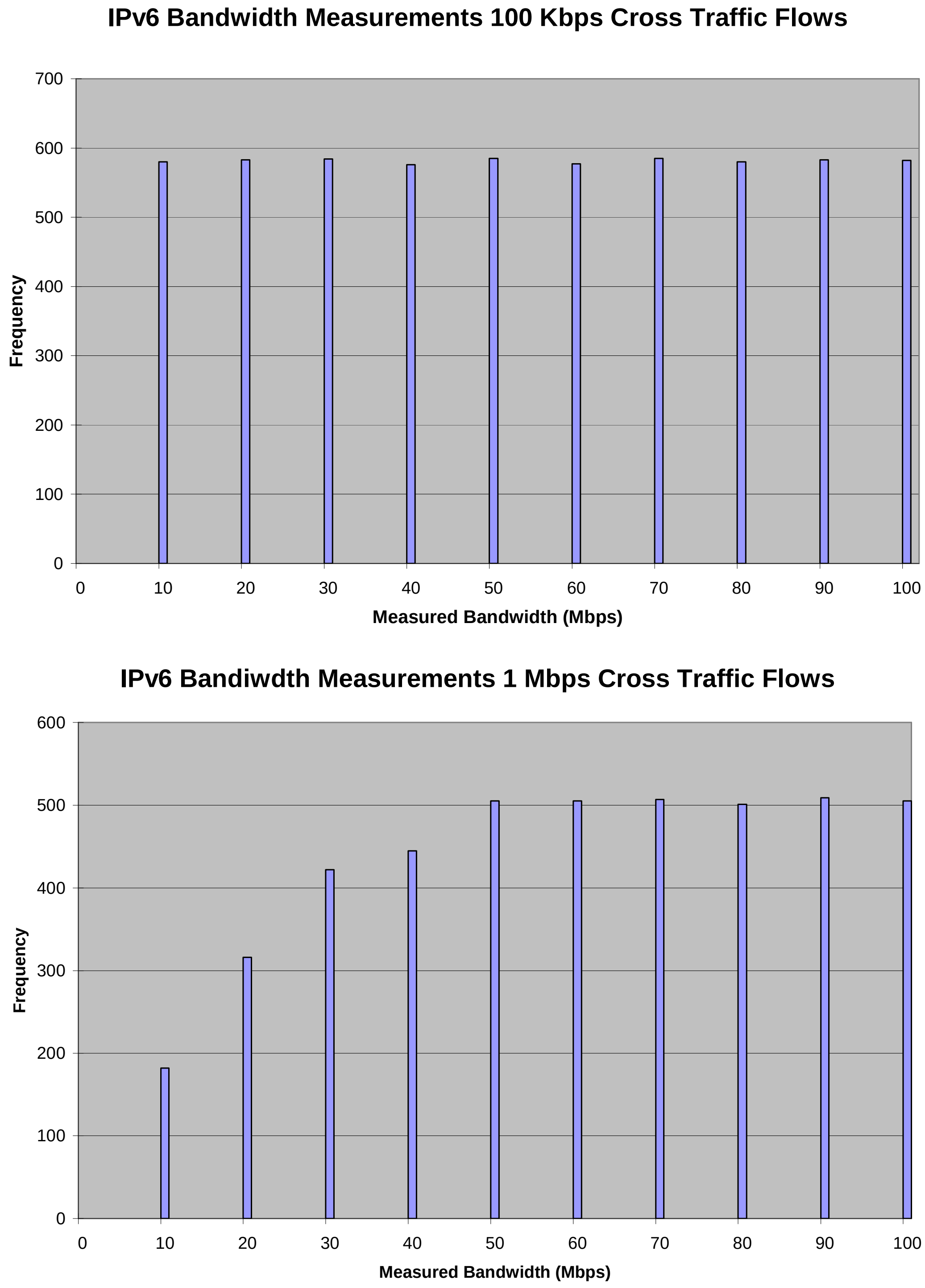


a) Scenario 1

b) Scenario 4

Figure 8 Measurement results from Cartouche scenario 1 (a) and scenario 4 (b)

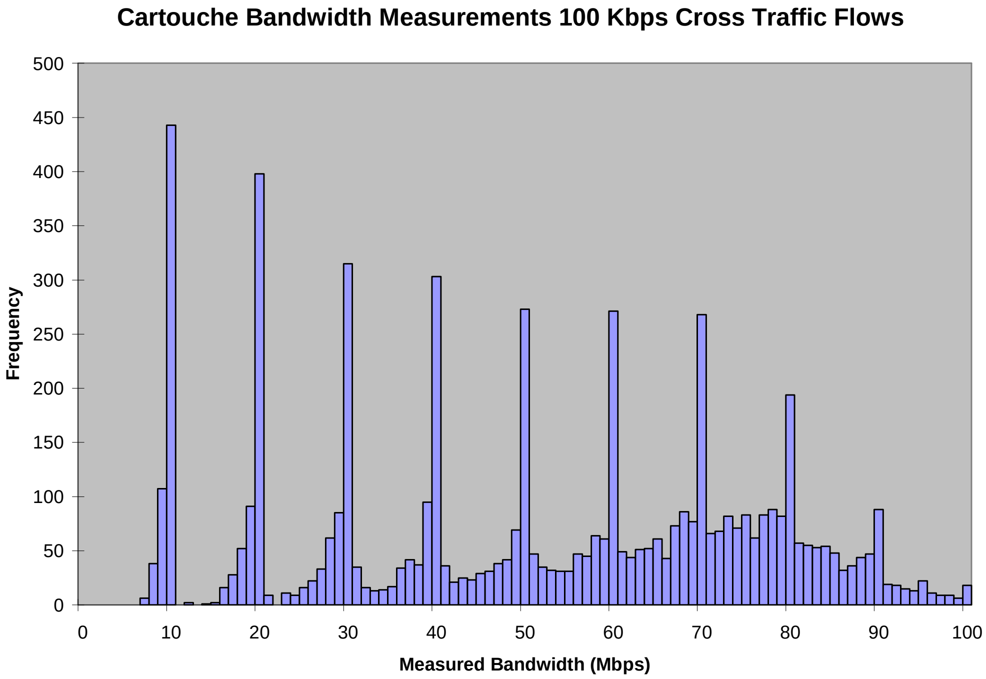
As stated earlier in this section, the previous data is the average measured bandwidth of 100 measurements for each parameter/scenario combination. In general, averages are not the chosen analysis for estimating the bandwidth of a network path. Instead, most IPv4 methods use the frequency of an estimation in order to filter erroneous measurements and identify the true capacity of the network. This frequency analysis method is based on the assumption that the majority of probe packets will not have their spacing affected by cross traffic or other network factors; therefore, the most frequent measurement is the correct measurement. In Figures 9 and 10 below, the frequency of all bandwidth measurements from scenarios 1 and 4 for the IPv6 and Cartouche techniques is presented.



IPv6 Bandwidth Measurements 100 Kbps Cross Traffic Flows

IPv6 Bandiwdth Measurements 1 Mbps Cross Traffic Flows

Figure 9 Frequency of IPv6 bandwidth measurements for scenario 1 (top) and scenario 2 (bottom)



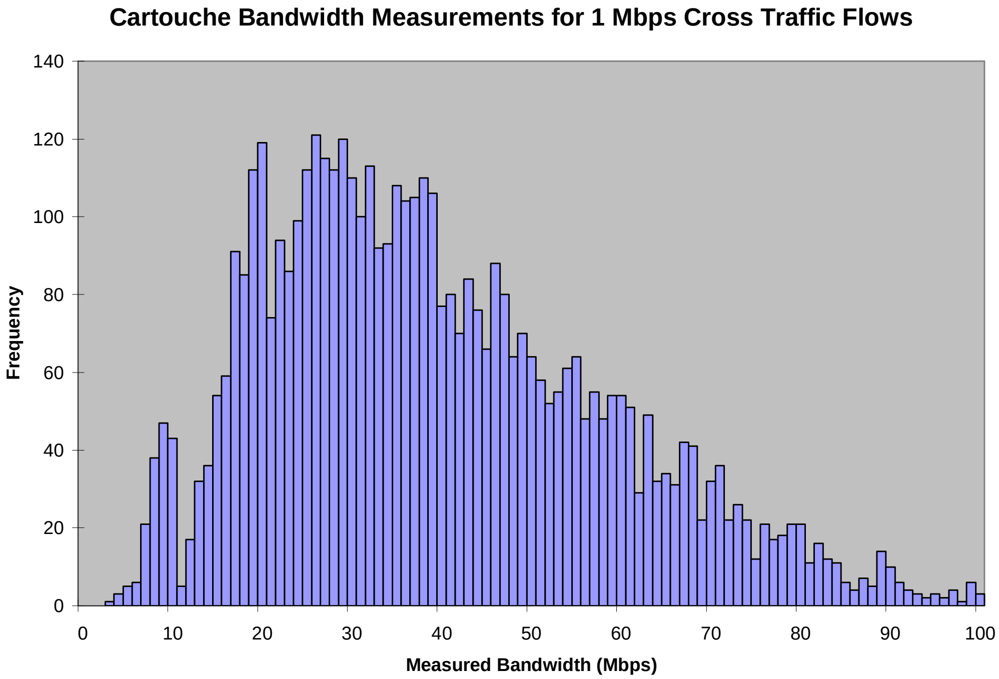


Figure 10 Frequency of Cartouche bandwidth measurements for scenario 1 (top) and scenario 2 (bottom)

As shown in Figure 9, all of the bandwidth estimations for the IPv6 estimation technique occur for the exact values of the bottleneck. The top histogram in Figure 9 has approximately 600 measurements for each bottleneck value. A few of the probes are lost due to the routers dropping packets and so the full 600 measurements (i.e. 100 probes for each 8, 16, 24, 32, 40, and 48 cross traffic values) are not seen. The bottom histogram in Figure 9 has considerably less than 600 measurements for the lower bottleneck bandwidths due to router queues being overloaded especially for cross traffic rates that exceed the bottleneck. Despite the load of the network and the number of probes lost, the IPv6 probes all measure the bottlenecks exactly.

However, the Cartouche technique does not produce exact measurement but instead; measurements vary across the entire bottleneck range. In Figure 10, the top histogram shows frequency of measurements when the cross traffic rate is 100 Kbps. For this scenario, the bottlenecks of 10 – 80 can be clearly seen since around 200 or more measurements were received for each of these bottlenecks. The bottleneck of 90 can also be seen but significantly less than 200 measurements were taken for this value. If the histogram is limited to measurements taken when the bottleneck is 90 Mbps as seen in Figure 11, the bottleneck is easily distinguished. The 100 bottleneck is indistinguishable due to the entire network bandwidths reaching equilibrium which results in any packet probe separation being interpreted as the bottleneck. Limiting the histogram to measurements for 100 only does not improve the ability to distinguish the bottleneck.

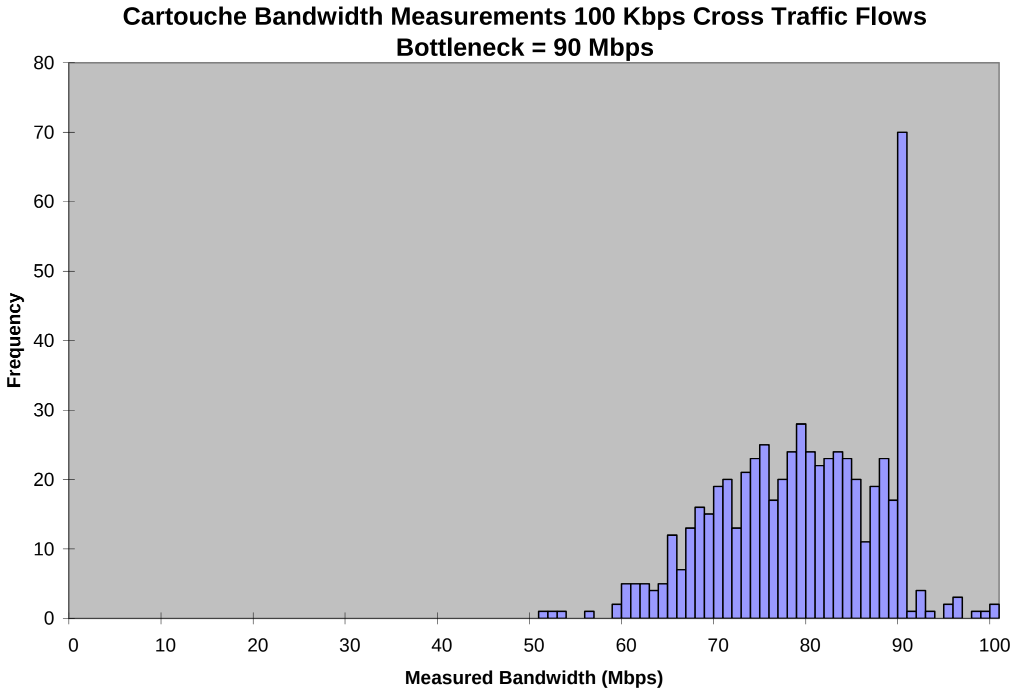


Figure 11 Frequency of Cartouche bandwidth measurements for 90 Mbps bottleneck and 100 Kbps cross traffic flows

The bottom histogram in Figure 10 shows the frequency of measurements when the cross traffic rate is 1 Mbps. In this figure, there are no distinguishable bottlenecks. Bandwidth measurements range throughout the possible network capacities due to probe packet spacing being affected by the high cross traffic rate. For this scenario, the assumption that the measurement with the highest frequency is the correct value does not apply. Limiting the histogram to measurements for a single bottleneck as seen in Figure 13 does not help in determining the bottleneck.

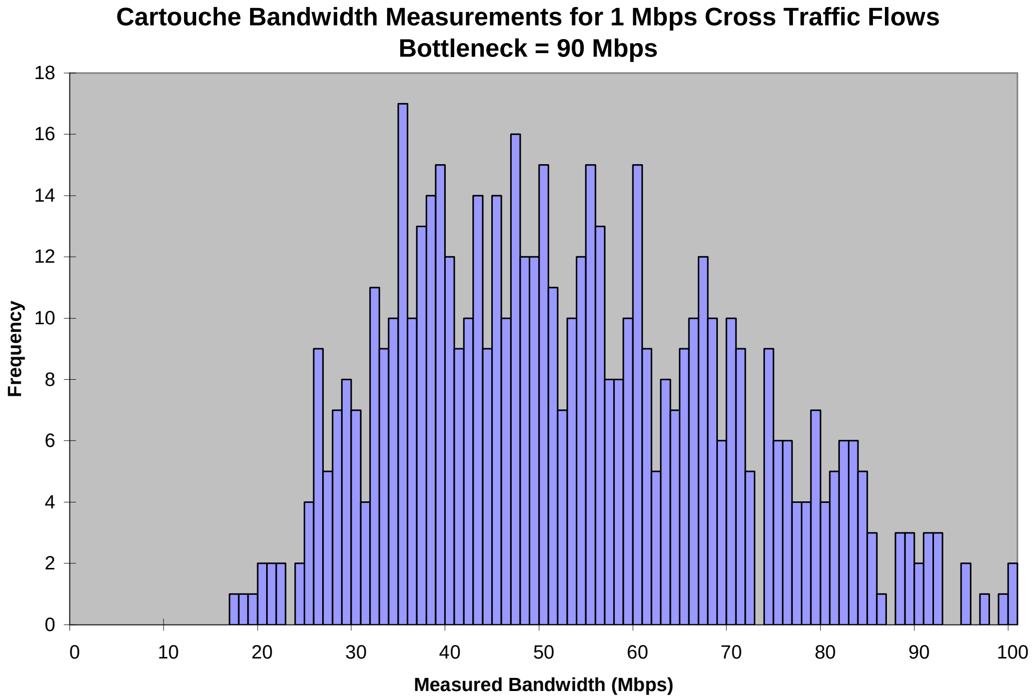


Figure 12 Frequency of Cartouche bandwidth measurements for 90 Mbps bottleneck and 1 Mbps cross traffic flows

4.2.2 Efficiency

The IPv6 method is efficient in measuring bandwidth since only two probe packets are required as shown in Table 5. The lead packet size is always 1500 octets, while the tail packet size is dependent on the timestamp option. In order to record 20 hops, a size of 212 octets is needed while only 76 octets is needed to record a single targeted hop. The Cartouche method is also efficient in that it requires a few probe packets as shown in Table 6. The Cartouche method uses a train of packets in order to increase accuracy but this reduces the efficiency compared to the IPv6 method since the train is always larger than two packets. In large bandwidth networks, both methods have a negligible impact on network resources but for small bandwidth networks, the Cartouche method has a greater impact compared to that of the IPv6 method.

Table 5 Size of IPv6 probes

	Lead Packet Size (octets)	Tail Packet Size (octets)	Total Size (octets)
1 hop record	1500	76	1576
20 hop record	1500	212	1632

Table 6 Size of Cartouche probes

	p packet size (octets)	*m*,*q* packet size (octets)	Total Size (octets)
Cartouche r=1	1500	40	3080
Cartouche r=2	1500	40	4620
Cartouche r=3	1500	40	6160
Cartouche r=*l*	1500	40	1540 * *l*

The efficiency of the two methods is also determined based on the properties of the network. If the network hops vary significantly in capacity, the IPv6 method will be forced to use a 1500/76 octet packet-pair combination. With a 76 octet packet, only a single hop can be measured at a time. If 20 hops must be measured, then 20 packet-pair probes must be sent. The total bytes sent then is 1576 * 20 = 31520 or approximately 32 Kilobytes (KB). 32 KB is still negligible for a network with bandwidths of several Mbps or greater, but network links such as 56K modems and ADSL have limited upstream bandwidths that would be significantly affected by 20 probe-packet pairs.

The efficiency of the cartouche method would also be affected based on the targeted path to measure and how many measurements must be taken. The Cartouche method has the ability to measure specific sections of the network path. If several sections of the path must be measured, then multiple packets trains must be sent. If the entire path is measured, the length of the train is based on the length of the network path *l*. For an *l* of 20, the train must be 31 KB long. As explained in the previous section, a single measurement is not reliable and as a result, multiple measurements must be taken. The efficiency of the Cartouche method continues to decrease as more measurements are performed.

4.2.3 Measurement Time

The time it takes to make a measurement is also a concern for bandwidth estimation techniques. The amount of time a technique takes before an accurate measurement is obtained strictly limits the applicability of that technique. If a technique takes too long to obtain an accurate measurement, then it cannot be used in applications such as end-to-end flow control where bandwidth measurements are needed quickly and continually. Many IPv4 techniques reduce accuracy in order to increase speed, while others reduce speed but increase accuracy. No technique currently exists that provides both speed and accuracy; such a technique would be ideal for almost every desired application.

The minimum measurement time for the IPv6 method is one round trip time (RTT). In this situation, a single probe packet pair is sent through the desired network path and immediately returned to the original host where the bandwidth is calculated. Since the IPv6 method is exact, only a single measurement is needed. If more than one probe packet pairs must be sent, the total measurement time increases to RTT + 2 * *tn* where *t* is the time it takes to send a probe pair and *n* is the number of pairs sent.

The Cartouche method can also obtain a bandwidth measurement in approximately one RTT but the measurement is most likely incorrect. Repeated measurements can increase the accuracy but measurement time increases. 100 measurements that were sent one per every RTT would take 100 RTT before a final value could be calculated.

4.3 Analysis

The results clearly show the increased accuracy of the IPv6 probing method compared to Cartouche probing. The IPv6 timestamp method produced estimations that were 100% accurate compared to the actual bandwidth capacity, whereas the Cartouche method produced estimations that were as low as 30% accurate. The Cartouche probes experience compression and decompression as the probes encounter cross traffic at each hop resulting in artificial estimations. The IPv6 timestamp method is immune to skewed data caused by inter-hop, cross traffic effects due to its ability to detect when the probes are not sent back-to-back at a hop.

While these simulations show favorable results, certain factors in real world situations may contribute to skewed estimations. Slower processing path in routers for packets with hop-by-hop options, limited timestamp clock resolution, and significantly varying capacities in a network path are a few factors that could cause skewed bandwidth estimations. Correct router timestamp implementation will eliminate the majority of router specific concerns, but varying link capacities and other network and traffic concerns will have to be studied in more detail. On the other hand, application specific networks such as a military battlefield network can guarantee correct router implementation or take a step further and implement explicit feedback mechanisms that report link capacity and utilization.

CHAPTER V

CONCLUSION AND FUTURE WORK

5.1 Conclusions

Bandwidth measurements are useful for a number of different applications and situations. Network applications can use information about network paths to select the best path and also decide how to efficiently use the available bandwidth in that path. Administrators and network designers can use bandwidth information to perform load balancing, traffic engineering, and network optimization. Current techniques for measuring available and bottleneck bandwidths are acceptable in certain applications, but they suffer from one or more of the following including limited accuracy, increased complexity, decreased efficiency, limited applicability, and susceptible to various factors..

The measurement techniques in IPv4, although not extremely accurate or dynamic, should be applicable in IPv6, but improved options of IPv6 provide the resources necessary to implement a timestamp option that will make these techniques simpler, accurate, and more efficient. The simulations show that timestamps used in conjunction with IPv6 provide an accurate solution and are up to 70% more accurate than similar IPv4 estimation methods.

The drawback to the IPV6 estimation methods is the necessary support by the routing entities in the network path. The routers must be able to insert timestamps immediately upon reception or transmission of the packet and the timestamp resolution must be fine enough for bandwidth calculations. In addition, the processing path for IPv6 packets that contain options must not be severely different from that of non-option packets in order for the timestamp packets to be processed equivalently to that of non-option packets. This property is necessary only for available bandwidth measurements since available bandwidth probe packets must be treated the same as normal traffic in order to give an accurate representation of normal traffic.

Despite these concerns, a timestamp option would benefit not only bandwidth measurements, but other areas such as calculating router load and computation time. Until there is feedback from the network itself, consistent accurate bandwidth measurements will be a difficult challenge to overcome.

5.2 Future Work

5.2.1 Extended Simulation Models

The simulation models and scenarios presented in Chapter 4 represent only a small subset of possible network conditions and characteristics. More complete and exhaustive simulation scenarios are needed in order to examine the timestamp method in more diverse networks and compare it with available bandwidth and additional bottleneck bandwidth estimation techniques.

In addition, the network models should be extended to more accurately represent real world hardware systems. Models are needed that accurately represent routing hardware and how that hardware would implement a timestamp option. The following list describes how models for network routers should be extended:

- Include models for various hardware types and configurations
- Include models for layer 2 switches and other non layer 3 hardware
- Include models that represent wireless links
- Include models that accurately represent the packet processing path
- Limit timestamp resolution to hardware capabilities
- Implement timestamp option in proper location of processing path

Various models that more accurately represent real world hardware systems will provide a better evaluation of the IPv6 timestamp method.

Cross traffic models should also be extended to include cross traffic with different characteristics. A single cross traffic model cannot completely represent the diversity of network traffic, especially traffic in the Internet. Additional cross traffic models that represent a wider range of network hosts including the following:

- Hosts sending and receiving real-time audio/video
- Hosts sending large amounts of data
- Hosts that are intermittently sending bursts of data
- Hosts performing peer-to-peer communications

Finally, additional simulations are needed that expand the network scenarios. Instead of a single 20 hop path that varies the bandwidth from 10 Mbps to 100 Mbps, more diverse networks need to be created with the following properties:

- Multiple bottlenecks
- Paths with layer 2 switches and other non routing hardware
- Hops with extreme variations in bandwidth
- Paths with links of various physical properties such as wireless

5.2.2 Real World Implementation

A real-world implementation is also included for future study. In a real-world implementation, the IPv6 timestamp hop-by-hop option is implemented using routing hardware or standard computers and a modified operating system. Various scenarios can then be created and the IPv6 timestamp estimation method can be tested in order to evaluate its accuracy and performance in real-world settings.

5.2.3 Applications

Future work also includes examining applications for the IPv6 bandwidth estimation method as well as applications for the IPv6 timestamp option. A few applications where the IPv6 estimation method should be evaluated include the following:

- Flow control for a reliable transport protocol such as TCP
- Server selection for content download
- Peer selection for content delivery

- Network management and provisioning

Many more applications are possible for a bandwidth estimation method that is accurate, efficient, flexible, and simple.

REFERENCES

[1] D. Clark. “The Design Philosophy of the DARPA Internet Protocols,” *Proceedings of ACM SIGCOMM*, pp. 106-114, August 1988.

[2] J. C. Bolot, “End-to-end Packet Delay and Loss Behavior in the Internet,” *Proceedings of ACM SIGCOMM*, pp. 289-298, September 1993.

[3] V. J. Ribeiro, R. H. Riedi, R.G. Baraniuk, J. Navratil, and L. Cottrell, “PathChirp: Efficient Available Bandwidth Estimation for Network Paths,” *Proceedings of Passive and Active Measurements Workshop*, April 2003.

[4] M. Jain and C. Dovrolis, “End-to-end Available Bandwidth: Measurement Methodology, Dynamics, and relation with TCP Throughput,” *IEEE/ACM Transactions on Networking*, vol. 11, pp. 537-549, August 2003.

[5] M. Jain, and C. Dovrolis, “Pathload: An Available Bandwidth Estimation Tool,” *Proceedings of Passive and Active Measurements Workshop*, 2002.

[6] J. Navratil, “ABwE: A Practical Approach to Available Bandwidth,” *Proceedings of Passive and Active Measurements*, 2003.

[7] G. Jin, G. Yang, B. Crowley, and D. Agarwal, “Network Characterization Service (NCS),” Technical Report, LBNL report # 47892, 2001.

[8] B. Melander, M. Bjorkman, and P. Gunnigberg, “A New End-to-end Probing and Analysis Method for Estimating Bandwidth Bottlenecks,” *Proceedings of IEEE Global Internet Symposium*, 2000.

[9] K. Harfoush, A. Bestavros, and J. Byers, “Measuring Bottleneck Bandwidth of Target Path Segments,” *Proceedings of IEEE INFOCOM*, pp. 2079-2089, March 2003

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

[11] K. Lai and M. Baker, “Measuring Link Bandwidths Using a Deterministic Model of Packet Delay,” *Proceedings ACM SIGCOMM*, pp. 283-294, September 2000.

[12] K. Lai and M. Baker, “Nettimer: A tool for Measuring Bottleneck Link Bandwidth”, *Proceedings of USITS*, pp. 123-134, Mar. 2001.

[13] B. Mah. pchar. http://www.ca.sandia.gov/bmah/Software/pchar, 2000.

[14] A. Downey, “Using Pathchar to Estimate Internet Link Characteristics.” *Proceedings of ACM SIGCOMM*, pp. 222-223, August 1999.

[15] S. Keshav, “A Control-Theoretic Approach to Flow Control,” *Proceedings of ACM SIGCOMM*, pp. 3 – 15, September 1991.

[16] R. Carter and M. Crovella, “Measuring Bottle-neck Link Speed in Packet-switched Networks,” *Performance Evaluation*, vol. 27/28, pp. 297-318, 1996

[17] C. Dovrolis, P. Ramanathan, and D. More, “Packet-Dispersion Techniques and a Capacity-Estimation Methodology,” *IEEE/ACM Transactions on Networking*, vol. 12, pp. 963-977, 2004

[18] N. Hu, and P. Steenkiste, “Evaluation and Characterization of Available Bandwidth Probing Techniques,” *IEEE Journal on Selected Areas in Communications*, vol. 21, pp. 879- 894, 2003.

[19] M. Jain and C. Dovrolis, “Ten Fallacies and Pitfalls on End-to-End Available Bandwidth Estimation”, *Proceedings of ACM SIGCOMM*, pp. 272-277, Oct. 2004.

[20] X. Liu, K. Ravindran, and D. Loguinov, “Multi-Hop Probing Asymptotics in Available Bandwidth Estimation: Stochastic Analysis”, *Proceedings of Internet Measurement Conference (IMC)*, October 2005.

[21] Internet Engineering Taskforce and Internet Engineering Steering Group: IPv6 Working Group. RFC 2460: Internet Protocol, Version 6 (IPv6) Specification. http://www.rfc-editor.org/rfc/rfc2460

[22] P. Loshin, *IPv6 Theory, Protocol, and Practice*, second ed., Morgan Kaufmann, San Francisco, 2004.

[23] J. Miller, OMB sets deadlines for agencies’ move to IPv6, Washington Technology, August 2005. http://www.washingtontechnology.com/news/1_1/daily_news/26715-1.html

[24] C. Wilson, Network Centric Warfare: Background and Oversight Issues for Congress, CRS Report for Congress, pp. 30, June 2004.

[25] IPv6 Internet Backbone. http://www.6bone.net.

[26] Internet Engineering Taskforce and Internet Engineering Steering Group: Network Working Group, RFC 760: DoD standard Internet Protocol. http://www.rfc-editor.org/rfc/rfc760.

[27] Internet Engineering Taskforce and Internet Engineering Steering Group: Network Working Group, RFC 781: Specification of the Internet Protocol (IP) Timestamp Option. http://www.rfc-editor.org/rfc/rfc781.

[28] W. Stevens, *TCP/IP Illustrated*, first ed., vol. 1, Addison Wesley, Indiana, 1994.

[29] A. Durand, L. Toutain, IPv6 Traceroute Option, IPv6 Working Group Internet Draft, June 1997. http://www.join.uni-muenster.de/Dokumente/drafts/draft-durand-ipv6-traceroute-00.txt.

[30] H. Kitamura, Connection/Link Status Investigation (CSI) for IPv6 Hop-by-Hop option and ICMPv6 messages Extension, IPng Internet-draft, NEC Corporation, Oct. 1999. http://www.watersprings.org/pub/id/draft-ietf-ipngwg-hbh-ext-csi-02.txt.

[31] MLDesigner, MLDesign Technologies. http://www.mldesigner.com.

[32] S. Deering, B. Hinden, "Proceedings of the Fortieth Internet Engineering Task Force", December 1997

[33] S. Deering, B. Hinden, "Proceedings of the Forty-Sixth Internet Engineering Task Force", November 1999