DISSERTATION TITLE:

ACTIVE LOSS PAIR MEASUREMENT

Author: Fok, Wai Ting

MSc in Information Technology

THE HONG KONG POLYTECHNIC UNIVERSITY

JULY 2008

Abstract

Packet Pair measurement has been used to discover end-to-end network characteristics. One of such measurement is loss-pair analysis [11], which has been shown to be able to passively measure the queue size in the bottleneck router along a path in ns-2 simulator. However, the methodology only models routers for which the queue is managed in the unit of byte. The application of the method is thus limited.

In this paper, we have proposed an active measurement methodology that can apply to all routers. It makes use of our newly proposed model that relates router queue size in unit of packet, bandwidth, packet size and queuing delay. Testbed experiment is performed to validate our model. We have illustrated that our methodology can estimate queue size of bottleneck router which manage buffer in unit of packet. Finally, we tried our methodology in a more complex baseline traffic. The estimation is distorted by effect of relatively small-size packets in baseline traffic on packet pair. We suggested some methods to lower such distortion.

The major contributions of our research are:

1. Propose an active loss-pair measurement methodology to support routers of different implementations of buffer management, such that the application can be extended to all routers in real world environment;

2. Perform testbed experiment to verify the relationship between queue size of bottleneck router and queuing delay observed by loss-pair;

3. Illustrate the use of linear regression and different statistical measures of central tendency to correct the error from data collected under different network traffic characteristics.

Acknowledgement

I would like to thank my supervisor Dr. Rocky K. C. Chang for his valuable comments and advice. I would also like to thank Dr. Daniel X. P. Luo and Edmond W. W. Chan for their useful idea and advice. Lastly, I would like to express my appreciation to my families and fiancee for their support in my dissertation work.

Contents

List of Figures

List of Tables

1 Introduction

Internet is a network of networks that are working based on common rules and protocols, and without central authority to control the setting of all networks in the Internet. The traffic passes through each network may be shaped by the configuration and policy of each network, including the AQM, together with the variation of network loading and link quality.

Under the rapid development of Internet and it's application, nowadays traffics are running between client and server, server and server, and client and client. The end points may be as close as situated in the same city, or as far as in different continentals. The last mile may be an 802.11 wireless network, and the link in between probably consists of optical fiber, ATM, etc.

Major characteristics of end-to-end network quality includes the bandwidth, loss and queuing of packets. A number of Internet applications, e.g. Internet phone, Internet video conferencing, and streaming of data, requires the information of end-to-end network characteristics to optimize the network utilization and application performance. No one can provide such information, and that is why Internet end-to-end measurement remains a hot topic in recent years.

Packet pair measurement, a technique of active measurement by sending probes consist of back-to-back packet-pair from one end to the other, can be used in the measurement of end-to-end bottleneck bandwidth and buffer size. The dispersion and the round-trip delay of the packets captured would provide necessary information to estimate these two characteristics.

loss-pair is introduced by Liu et al [11] to address one of the problem: to estimate the buffer size of the bottleneck router. Having review the paper, we found that the tool only partially address the issue: the estimation of buffer size in unit of byte has been introduced in detail. However, it can only apply to one of the two methods on buffer management. Actual implementation and academic research has divide the router buffer management in two categories: in unit of byte, and in number of packet. The actual implementation may have more complicated algorithms on how to handle the packets in different size and class to optimize the performance, but these are out of scope of our research.

In this paper, we will investigate the end-to-end measurement of queue size of bottleneck router managed in the unit of packet. The rest of the paper is structured as follows. Section II summaries background works that this paper based on. Section III explains challenges faced by each measurement tool and the problems addressed. Section IV describes the . Section V presents the original model in [11] and an our amended model of queue size of bottleneck router. Section VI illustrates the testbed experiment results that validate our model. We conclude in Section VII.

2 Motivation

Loss-Pair [11] is a novel methodology aimed at measuring the buffer size of bottleneck router along an end-to-end path. When we conduct experiment on testbed consisting of Click [8] routers, we found that the buffer sizing is different from that used in [11]. The unit of queue size of Click is in unit of packet rather than byte. Therefore, a limit is placed on the number of packets that can be buffered before incoming traffic packets are discarded. This contradicts the baseline router design in Loss Pair, that number of packets stored can be buffered is limited to the total buffer space available.

The deviation from the model causes error in estimation of bottleneck router. We demonstrate this with data from our experiment, which is summarized in Table 2. The layer-2 packet size of the UDP cross-traffic is configured to be uniformly distributed from 500 to 1000 bytes. To eliminate the effect of random error in experiment, we illustrate the calculation with the values obtained from linear regression. For a 404-byte (encapsulated header inclusive) probe, from line regression of results obtained, the queuing delay observed by the probe, which is the difference of the loss-pair round trip delay and the minimum round trip delay, is 0.231304 - 0.107990 = 0.123314. With a correct measured bandwidth of 3Mbit/s, the buffer size is estimated as:

$$n = w \times r = 0.123314 \times 3000000 = 369943\,[bit] = 46243\,[byte]$$

Compared with the actual buffer size of 60 packets, or converted to size in byte,

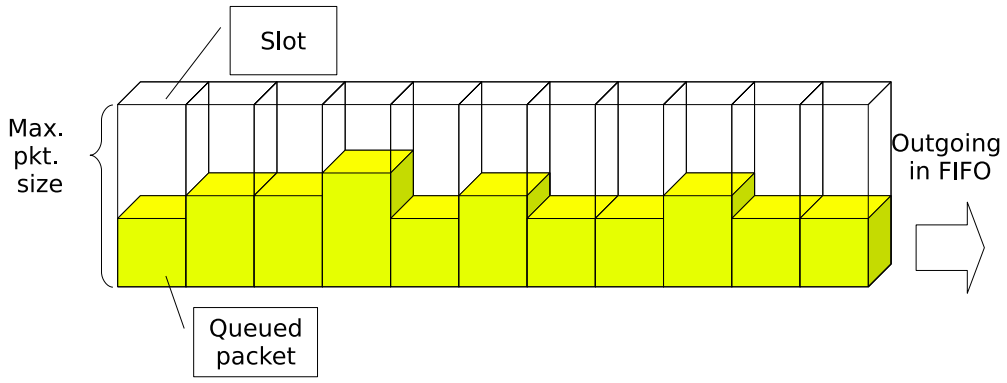

Figure 1: Illustration of error in size estimation of router's buffer in unit of byte

$60 \times 1500\,[byte] = 90000\,[byte]$, this estimated figure is clearly not correct. The algorithm is therefore irrelevant to buffer allocation in unit of packet.

The error in estimation is illustrated in Figure 1. The buffer is partitioned into fixed-size slots to accommodate incoming packets, and the size equals to the maximum frame that the network can transmit. The slots are not fully filled by the buffered packets since not all packets have the maximum size. When loss-pair is found, the queuing delay observed by the loss-pair is a good estimation of the time to drain all the packet queued in the full buffer of bottleneck router. However, as represented by the colored volume in the diagram, the total size of the queued packets does not equal to the buffer size. The measured buffer size in this case is therefore not the buffer size.

Keshav et al, in discussing the issues and trends of router [7], pointed out that the price drop of memory has dramatically increase the attractiveness of using a large fixed-size single piece of memory to hold data for each packet. In view of the increasing throughput requirement, this is one of the solution to overcome the

bottleneck in accessing the output queue.

Spalink et al have commented the packet buffer allocation in deployment of software-based router [16]. They choose to divide the memory into slots of 2KB each to accommodate maximally sized (1518 bytes) Ethernet packet. They consider this simple allocation scheme saves a lot of complexity. Experiments showed that the router built can achieve a sustainable rate of 3.47Mpps.

Given various researches based on routers, in either simulation or testbed, with buffer allocation in unit of packet such as [3], [12], [2], and the performance of buffer allocation implementation mentioned above, the measurement of queue size in unit of packet is considered necessary.

Without a new methodology to address the measurement of bottleneck router's queue size in unit of packet, Internet measurement is not guaranteed to provide correct results for all cases.

3 Background

Several packet pair measurement tools are developed to discover the end-to-end characteristics along a path. Below summaries the methodology of some of the interesting tools.

A. Loss-Pair

Loss-pair [11] is a pair of packets that travels along the same path closely enough such that they experienced the same situation, and that exactly one of them is dropped.

Research showed that round trip time (t_q) of the successfully transmitted packet in the loss-pair and the minimum round trip delay (t_p) is a good estimation of the queue length at the link at which packet loss occurred. The model of the queue size and queuing delay will be explained in details in later section. The buffer size (B) of the link is estimated as $B = C(t_q - t_p)$. In addition, the drop ratio of AQM such as random early detection can be reasonably characterized.

There are generally two kinds of queue capacity management in router: queues with capacity in unit of byte or in unit of packet. For queue with capacity in unit of byte, each packet fills the buffer with its size. For queue with capacity in unit of packet, the outgoing queue is partitioned into a fixed number of slot with the size of maximum packet size, and each packet fills a slot regardless of its size. The loss-pairs [11] only measures the queue size of bottleneck routers that manage the

queue in unit of byte. This left room for further study on router queue in unit of packet.

B. Packet Pair Dispersion

Packet dispersion techniques is one of the methodology used to estimate the end-to-end capacity [4]. It make use of the observation of packet service time at each router [14]. When a back-to-back packet pair of size L travels through a network consisting of a number of store-and-forward link i with bandwidth r_i, the service time at each router, $\tau_i = L/r_i$. Although the service time between the two packets at each router varies, without the interference of other packet, the time of dispersion of the packet pair, which is the difference between the time the pair of packets reach the receiver, should always equals to the longest packet service time, $\max\tau_i$. Such phenomenon is illustrated in Figure 2. Since the bandwidth of the narrow link defines the capacity of an end-to-end path, and the highest packet service time of the packet pair is resulted from the narrow link, the dispersion of the packet pair, t_d, measured at the receiver is a good estimator of the capacity along the path. A back-to-back packet pairs of size L is sent through the network. Therefore, the capacity of the end-to-end path, C, can be estimated as

$$C = \frac{L}{t_d}$$

Nevertheless, a major flaw of the packet pair methodology is the error introduced by cross-traffic. If cross-traffic packet of size L_t reaches the router i just before the first packet in the packet pair arrive, the time taken by the router to first forward the cross-traffic packet at the outgoing interface before it forward the probing packet

is $\tau_t = L_t/r_i$. The first packet in the packet pair is thus queued at i-th router for a maximum delay time τ_t. If the second packet in the packet pair reaches the i-th router before it sends out the first packet, the dispersion of the packet pair in the link after the i-th router is shortened by no more than τ_t. In contrast, if a cross-traffic packet reaches the router i just after the first packet in the pair but before the second packet, it obviously caused a delay of no more than $\tau_t$to the second packet at the outgoing interface of the i-th router. These two cases are illustrated in Figure 3.

The probability of queuing of the second packet at the i-th router is correlated to the time dispersion between the packet when they arrive at each router, which equals to $\max_{n=0\ldots i} \tau_n$. The longer the dispersion, the more likely that cross-traffic packet inserted in between the packet pair. Since $\tau_i = L/r_i$, a smaller probe size L can decrease the probability of queuing of second packet. One the other hand, the probability that the first packet suffers queuing delay is independent of the probe size.

Another weakness of the packet pair methodology is the encapsulation headers size on the wire. We control the probe size by varying the payload size at the sender side. When the probe is being transmitted through the datalink layer (layer 2), e.g. ethernet, an ethernet header is attached to the beginning of the probe packet data and thus increase the size of the frame being delivered on the network. The service time of the frame at the outgoing interface of each router depends on the total layer-2 frame size. Therefore, the header size should be considered in estimating the bandwidth. The header size of different layer-2 network topology varies, and we cannot detect the actual network topology of a specific node on

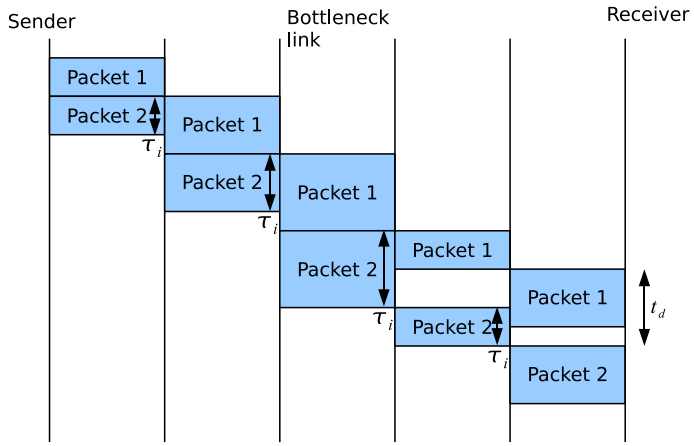

Figure 2: Illustration of packet pair dispersion

the path by end-to-end measurement. The bandwidth is likely underestimated if we only include the layer-3 packet size. The most common way to minimize the effect of encapsulation header size is to set the probe packet size to a relatively large one, say, 500 byte or more. Another method is to vary the probe packet size and obtain the capacity through linear regression [13].

There is another factor other than encapsulation header that limit the lowest packet size of probe. As the packet size (L) decreases, the packet pair dispersion, which is expressed as $t_d = \frac{L}{C}$, decreases too. The error introduced by the time resolution of the measuring machine on sending and receiving timestamp of probes would be unacceptably high if L is small and C is relatively high. For example, if C = 100Mbit/s, L = 400 bit, $t_d = \frac{400}{100{,}000{,}000} = 0.000004sec = 4\mu s$. The time resolution of 1μs will lead to a maximum of 25% error on the measured capacity.

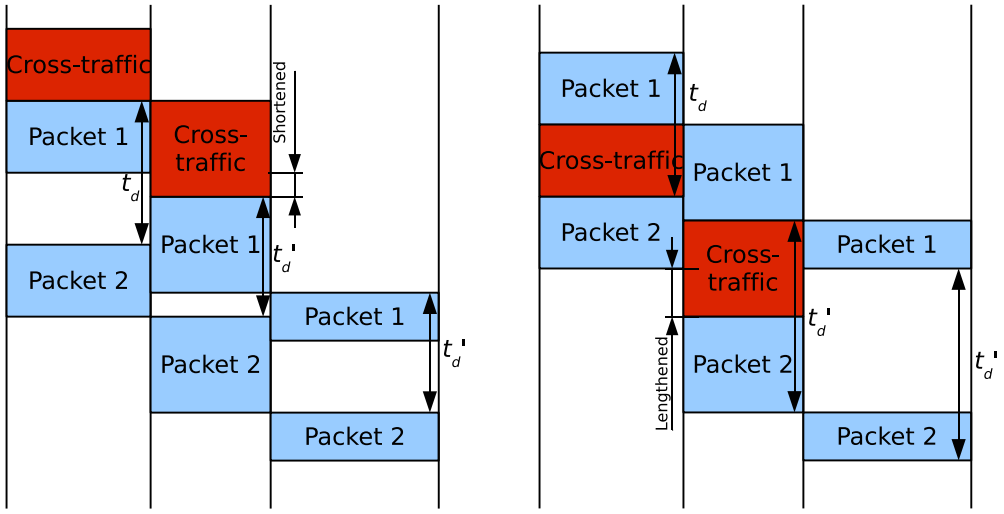

Figure 3: Graphical illustration of effect of cross-traffic on packet pair dispersion

C. CapProbe

CapProbe [6] introduced an improved technique used to estimate the end-to-end capacity. It makes use of the round-trip time data of the probe packets returned to distinguish whether the packets returned suffered unnecessary delays. The rationale is that queuing delay is the primary source of error in capacity estimation from dispersion. If the round-trip time of both packets is low, the packet pair collected incurred nearly no queuing delay. Thus the dispersion for such packet pair is unaffected by queuing delay and remains a good estimation of capacity. This method, however, cannot be applied in the loss-pair methodology. The loss-pair are expected to experience the queuing delay at the bottleneck router, which violate the principal assumption in CapProbe. There is no characteristics that let us distinguish whether the loss-pair collected are subject to queuing delay at

the bottleneck router or other routers. Therefore, to filter out other factors to the round trip delay, we normally use the mode as a representative figure of round trip delay.

D. One-Packet Techniques

Unlike the above, an approach that make use of packet delay of one packet rather than packet pair dispersion, which summarised in [9], is used to estimate the link bandwidths. The technique is also derived from the observation from [14]. Like traceroute, the tools exploit the functionality of time-to-live (TTL) value in IP header, which is mainly to avoid endless loop on routing packets. A TTL value is set by the sender, which means the upper limit of hop that the packet can transfer through. The TTL value in the header of a packet is decreased by 1 when the packet pass through a host. When the TTL is reduced to zero before it reach the destination, the host that see a zero TTL value in a packet will discard the packet and sends back an ICMP error message to the source host. By sending probing packets with TTL value from one to the number of hop to the end node, the incremental transmission delay for each hop can be collected.

The transmission delay (t_d), or service time of packet at the router, has a linear relationship with the packet size (s^k) and the bandwidth of the outgoing link of the router (b_i), i.e., $t_d = \frac{s^k}{b_i}$. By regression the per-hop incremental transmission delay and the probe size, the bandwidth for a particular link can be obtained from the slope [13]. The results obtained, however, is vulnerable to queuing delay introduced by cross-traffic. To eliminate the error caused by queuing, a simple

min-filter can be applied to the round trip delay obtained.

4 Challenges

A. Cross Traffic Intensity

The baseline traffic intensity is a key issue to loss-pair measurement. If the baseline traffic intensity is far below the capacity of the bottleneck link, the chance that the queue at the bottleneck router grow up is low. Thus, the bottleneck router's buffer will seldom be fully occupied by packets. If the real packet loss rate is low, the number of loss-pair collected in active measurement will be low. loss-pair analysis thus cannot be performed on links that the utilization is extremely low, unless the measurement traffic is sufficiently high to generate loss events.

The difficulty of obtaining good results in packet pair depends on the cross-traffic intensity. As mentioned in Section 2, heavy cross-traffic is a major obstacle for accurate bandwidth measurement. Cross-traffic adds queuing delay to all kinds of probe packets, no matter one-packet, packet pair or packet train. In addition, the packet pair or packet train methods are also vulnerable to injection of cross-traffic packet in between the probe packets. The heavier the background traffic, the more likely the results obtained are distorted.

There are some methods to overcome the interference of cross-traffic. CapProbe[6] is a tool to address this issue in packet pair, and min-filter is normally used to get rid of the distortion caused by the cross-traffic. The capacity estimation is not covered in this paper, and we assume that a reasonable estimate can be obtained through the use of end-to-end measurement tool that make use of packet pair dispersion, CapProbe, or the one-packet technique mentioned above.

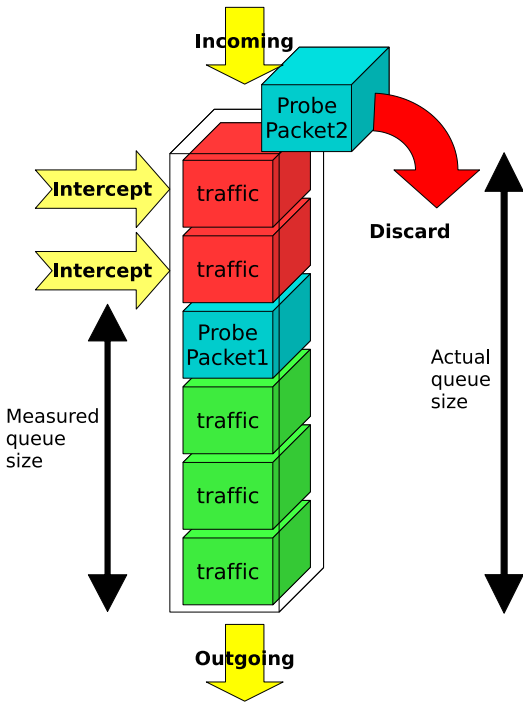

Figure 4: Illustration of loss-pair intercepted by cross-traffic packets

It is worthy to note that loss-pair is also a kind of packet pair methodology. The results obtained by loss-pair are also subject to the interference of cross-traffic. As shown in Figure 4, when cross-traffic packets are inserted in between the packet pair, the principal assumption of loss-pair is no longer valid; the packet pair experienced different situations. When we use the round trip delay in the first packet to estimate the queue size, the queue size is probably under-estimated.

B. Loss Occurence at Narrow Link

There is a principal assumption made by Liu et al in [11], which stated that majority of packet loss occurred at the bottleneck router. The queue size is then

estimated as the compound of the time required to drain a full packet queue at the bottleneck, and the outgoing bandwidth of the bottleneck router. The former one can be measured by the queuing delay observed from the loss-pair. However, if most packet loss does not happen at the narrow link of the end-to-end path, the queuing delay observed by loss-pair cannot estimate the time to drain the full queue at the bottleneck. Then, the calculated queue size is inaccurate. The second assumption of loss-pair is that the bottleneck link on the end-to-end path do not change from one to another.

Nowadays TCP traffic dominate the data transmission in Internet [17]. We expected that in the Internet most packet loss are caused by TCP traffic. According to the algorithm of TCP protocol, on the sender and receiver ends, it increase the rate of data transmitted gently to search for the available bandwidth in each transmission, and look for packet loss as a sign of congestion. Upon observing packet loss, it reduce the data transmission rate dramatically to avoid packet loss due to overflooding the link. Then it goes back to the phrase of increasing and the procedures repeat over and over again.

In the middle of the end-to-end path, the routers store packets they receive in the buffer and forward them to the next node, which is limited by the bandwidth of the link to the next node. When the total incoming traffic rate at a router is higher than the outgoing link bandwidth, the queue of packet builds up in the router's buffer, until the queue is full and the router starts to discard packets that it receive. In addition, AQM algorithm, like RED [5], is implemented in router such that the ratio of packet drop is a function of the queue length. This algorithm attempts to avoid the problem of global synchronization introduced by drop-tail

queue management.

In equilibrium, most packet loss thus should occur at the gateway to the narrow link. Figure 5 presents the relationship between queue length and packet drop . However, if a router on the measuring end-to-end path, which is not a gateway to the narrow link on the path, is actually the bottleneck router for another end-to-end path, the primary assumption of loss-pair may no long valid. As illustrated in Figure 6, C is the bottleneck router for the end-to-end path from A to Y. If the traffic intensity of path A-Y is high enough, packet loss happened at C for any traffic going to X. Little packet loss may occur at the bottleneck router D if the rate of traffic flowing through E-Y is low.

C. Measurement Implementation

Packet pair analysis, including packet dispersion measurement and loss-pair analysis, can be performed by active or passive measurement. Active measurement involves sending probes along the path from one end to the other end. Since the results obtained from the probes are only samples out of all traffic passing through the end-to-end path, the accuracy depends highly on the sampling methods of the probe. Passive measurement, on the other hand, normally requires privilege right to capture and analysis all end-to-end traffic.

There are a few tools that have implemented the packet pair or packet train methodology:

TCP Probe [1] incorporate the capacity measurement in TCP protocol. The TCP algorithm on the client side is customized to arrange the transmission of actual

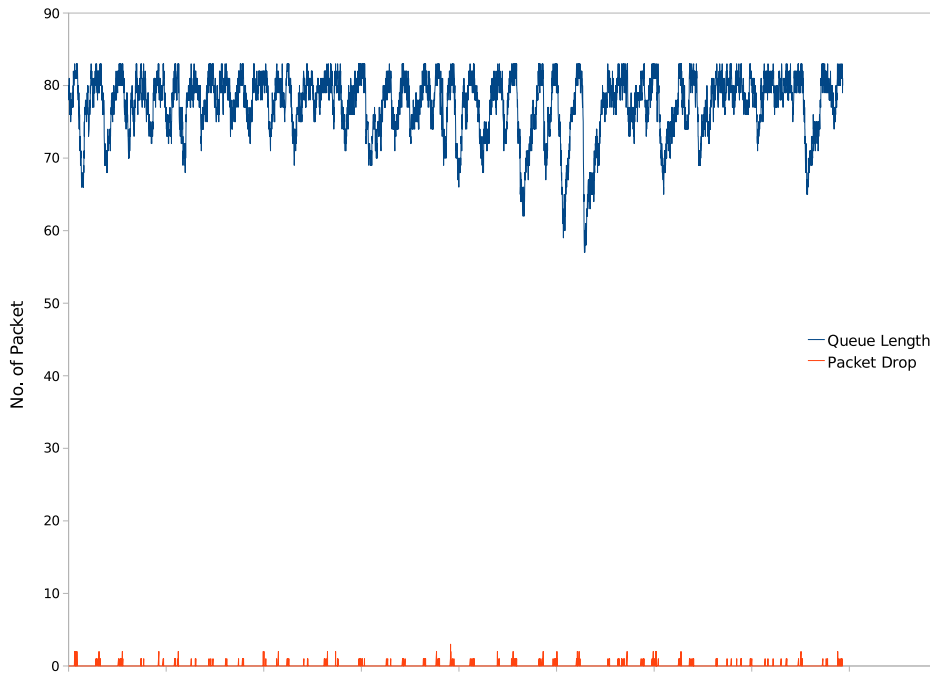

Figure 5: Queue length and packet drop timeseries measured in bottleneck router in experimental testbed, queue size at 83 packets, no AQM configured. TCP path-persistent cross-traffic generated by IPerf.

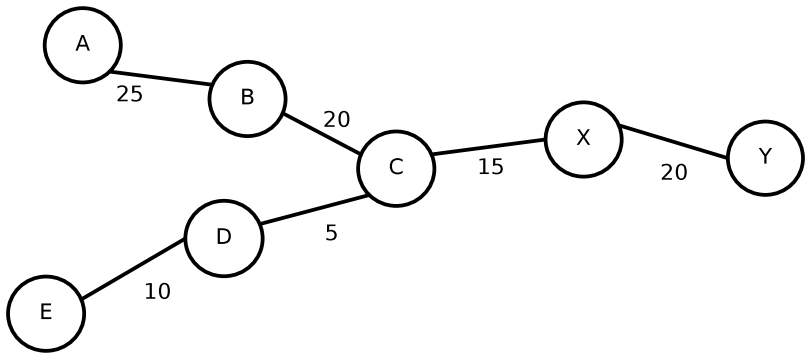

Figure 6: Illustration of situation that packet loss does not occur at narrow link in end-to-end path

data to the web server in a desired pattern. Some data packets are sent as back-to-back packet pair or in reverse order to trigger the server sending back web object as desired packet pairs to the client that the client can recognize. Therefore, the 'client' machine can send out probes to any typical web server and capture the return probes without special co-operation from the 'server' side. Based on the round trip information, the tester can estimate the path capacity from the packet pair dispersions.

Asymprobe [10] is a simple active measurement implementation of packet pair methodology. Specific programs are run on the machines at both end to send and receive probes. Probes are transmitted through a round trip from the sender to the receiver and then back to the sender, where the capacity estimation carried out. The probe size have a lower boundary of 500 bytes due to the limitation of time resolution as mentioned in previous section, and a upper boundary of 1500 bytes which is the typical setting of MTU. Thus, capacity estimation of the round-trip methodology has the ratio of forward and backward paths limited to 3:1 or 1:3, which have already broken the restriction of estimating the lowest bandwidth for the round trip as a whole.

Badabing [15] is a one-way packet train methodology designed to measure the single-way end-to-end loss episode and loss frequency. Like AsymProbe, the active measurement tool is divided into a server application and a client application that runs on each end of a path. The probes travels only one-way from the client to the server end, and the probe rate follows the poisson distribution to increase the chance of capturing the rare and irregular occurence of loss and variable duration of loss. One of the major feature of Badabing is adjusting the probe rate to

trade the accuracy of measurement with the impact on the link.

We illustrate our methodology by round-trip active measurement. Similar to TCP Probe, we exploit the TCP algorithm to send probes containing packet pair, and trigger packet pair from the web server. However, the actual implementation of our measurement is out of scope of this paper.

5 Testbed Setup

A. Hardware Configuration

We have setup the testbed as shown in Figure 7. It consists of 12 commodity computers acting as router, traffic generator and receiver, probe sender, and web server respectively as labeled in the diagram. They have 100-based Ethernet network cards installed and inter-connected by 100-based Ethernet switches. All of them had 2.4GHz Intel Pentium 4 processors running on Linux.

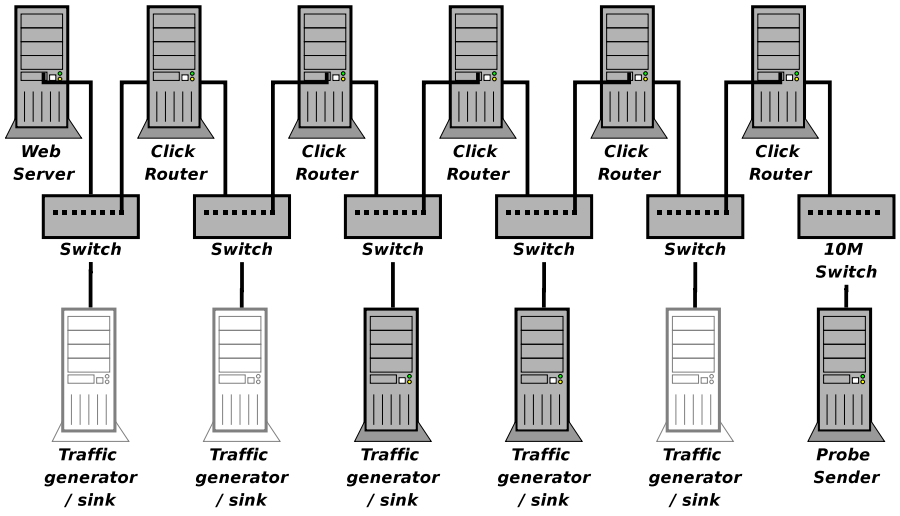

Figure 7: Testbed configuration

The five computers labelled as click router in Figure 7 run Click, and the usage of Click is summarized in the following section. Five computers that act as traffic generators / sinks are connected to the switches, which link up two adjacent routers. Each switch and the connecting cross-traffic generating machines simulates a network that both receives and transmits Internet traffic. Routers, apart

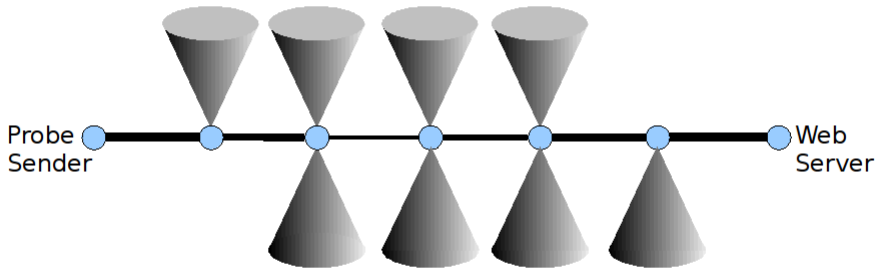

Figure 8: Logical setting of testbed

from acting as router or gateway at both adjacent networks, also simulates different bandwidth of link interconnecting different networks. The logical design is illustrated in Figure 8. The experiment testbed composed of the elements stated above tries to simulate an end-to-end path between any two hosts connecting to the Internet.

We concerns only the queue size of the bottleneck router, which can be measured by the queuing delay at the bottleneck router. By generating hop-persistent cross-traffic across the bottleneck router, packet loss and queuing delay occured at the bottleneck router. We are not going to study the effect of other cross-traffic along the path, so no traffic is generated across hops other than bottleneck router.

B. Click Modular Router

Click [8] is a open-source software implementation of conventional router. Functions are built as modules which can be loaded in the Click's architecture to customize the router's behaviour. The performance of Click has been verified by experiments and achieve a maximum loss-free forwarding rate of 330,000 64-byte packets per second on Pentium III computer. Packet flows in Click are controlled by pushing and pulling processing. Like other software router, packets are push to the next element as soon as they arrive at the incoming connection, thus unsolicited packets arrive at Click router are immediately stored to the buffer and pending further processing. In contrast, timing of pull processing are determined by the availability of the next element. For example, the transmitting device is one of the pull device that transmit packets by pull processing. It transmits one packet in the packet queue when it becomes ready.

There are several modules we have used to configure the behaviour of routers. They are summarized below:

Shaper element is a traffic shaper that sets a limit on the maximum throughput that traffic can be delivered. Such limit is achieved by pulling the packet at the specified rate to the transmitting device. The CPU scheduling in Click ensures that the pull processing are performed at the desired rate. The outgoing packets thus exhibit a time dispersion that is similar to those in a link of bandwidth equal to our preset rate. In addition, this element can be configured to add an artificial propagation delay to the traffic.

Queue element is an explicit declaration of buffer used in Click. It is necessary to

include this element in the configuration since Click do not have implicit queues on the input and output element. It is a FIFO queue and by default discard packets received when the queue is full. The queue size are in the unit of packet rather than byte.

IPClassifier is an important element to classify packet received according to the characteristics of the packet. It can classify packets of different source IP or destination IP, layer-3 protocol, flags in the header to different cases, and for each case we can assign different processes to the packets, such as store in individual queues, subject to a specific traffic shaper, etc.

C. Testbed Setting

The traffic shaper of Click in each router is set to a unique value, with the lowest bandwidth in the middle as narrow link. Refer to Figure 2, after the probe packet pair has passed through a slower link and arrives at a faster link, the dispersion between the packet pair increases. This leaves the probe vulnerable to interference of cross-traffic packet on the faster link. Therefore, it is more difficult to measure the bottleneck at the middle rather than close to server end. Apart from this, the bottleneck is not at the client's end in real-life environment, otherwise the user have already know the end-to-end capacity and no measurement are necessary. The testbed configuration of the link bandwidth are shown in Table 1. To simulate the propagation delay for an end-to-end path that span in a wide area, a 10ms delay is added to the traffic for each direction at all Click router, so the expected minimum round trip delay is not lower than $10ms \times 2 \times 5 = 100ms$.

The queue sizes for all routers in our testbed are set in the Queue element in the Click router. Except for the bottleneck router, the queue size of other routers are configured based on the rule-of-thumb from [18], which means the queue size in byte should be equal to the round trip delay of traffic flow multiplied by the bandwidth of the network interface of the router. For convenience, we assume an average packet size of 770 byte (mean of 40 byte and 1500 byte) and calculate the estimated queue size in number of packet, i.e., $n = 100ms \times r_i/(770 \times 8)$. The queue size for each router are summarized in Table 1.

The probe sender is located at one end of the network with web server at the other end. Apache 2 is running on the server machine acting as a conventional web server. When the probing packet pair from the probe sender reaches the web server machine, it triggers a pair of packets sending back to the probe sender machine. The packet pair collected at the probe sender machine then provides traces of path characteristics such as round trip delay, re-ordering and loss of packet pair.

Our methodology, which will be introduced in the coming section, is sensitive to the packet size of traffic flowing through the bottleneck router, so measures must be taken to ensure that experiment data remains unaffected by other traffic such as control packets. We configured packet classifier, IPClassifier in Click, to separate probing packets and cross-traffic packets from other TCP/IP control packets that are used by our probing machanism, such as SYNC, FIN and ACK. Only the probing packets and baseline traffic flow through the concerned buffer. All other packets are stored in a separate buffer, and are not subject to the abovementioned traffic shaper. We can then strictly control the packet size and packet rate through

Router	Capacity (both direction) in Mbit/s	Queue Size in packet	Delay (each direction) in ms
1	7.5	122	10ms
2	5.5	89	10ms
3	3.0	50	10ms
4	6.0	97	10ms
5	8.0	130	10ms

Table 1: Configuration of testbed illustrated in Figure 7

the queue concerned. We can also ensure that the probe packet and baseline traffic are fairly treated and goes to the same buffer.

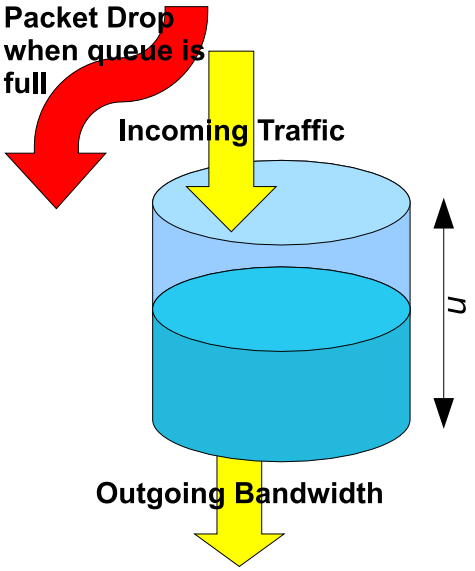

Figure 9: Model of router's buffer size in unit of byte

6 Queue Size and Queuing Delay

A. The Models

Figure 9 illustrates the model used by [11]. The router's buffer, which managed as a FIFO queue, can be viewed as a bucket with a definite volume. Water, thus incoming traffic, are pumped into the bucket at an uncooperative rate. Similar to the store-and-forward routers, the water is temporary stored in the buffer, and then flow out of the bucket at a fixed rate depending on the outgoing bandwidth. The buffer will be gradually filled up, thus the queue length will increase, if the uncontrollable incoming traffic rate is higher than the fixed outgoing traffic.

The incoming packets gradually fill the queue until it reach the limit, i.e. queue

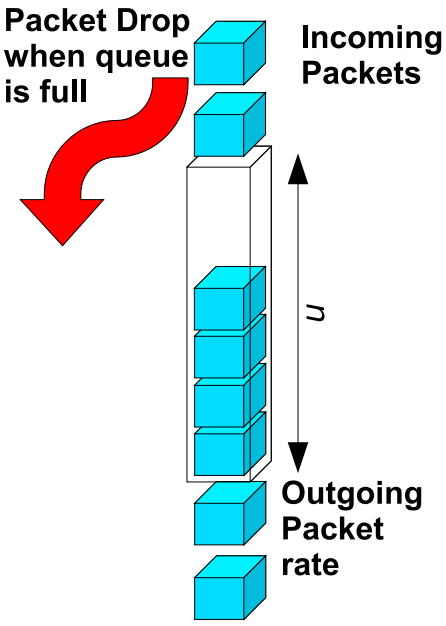

Figure 10: Model of router's buffer size in unit of byte

size (n). Just like water being added to a full bucket, any packet that arrived when the buffer is full will be discarded and is lost. A loss-pair, in which one out of two packets traveled closely together, is generated when one of the packet arrived at the router when the queue is nearly full, and another packet is dropped since the buffer is full. Unlike water stored in a bucket that each molecules is not identifiable, packets stored in the buffer goes out of the router in FIFO fashion. The successfully transmitted packet, which initially buffered at the very last bits of the queue when $q_l = n$, experienced the longest queuing delay time (w) which equals to the length of the queue size divided by the outgoing traffic rate (r), i.e. $w = \frac{n}{r}$.

However, the model does not apply to all routers. There are two type of buffer management in routers: in unit of byte and in unit of packet. The former one

means that the buffer is partitioned according to the size of each packet in the queue, such that every byte in the buffer is filled by incoming packets back-to-back. The later one would first have the buffer partitioned into a fixed number of slots with the maximum packet size, and each incoming packet is stored in one slot no matter how large is the packet size.

The above-mentioned model is valid only for the former one. For the later one, since each buffered packet may have different size, the total size of buffered packets may varies, therefore the time taken to discharge a full queue of packets is not fixed. To estimate the buffer size, we must first find the average packet size. Then by performing experiment to obtain the average queue delay time, we can calculate the buffer size $n = w \times r$.

We proposed an amended model as shown in Figure 10 to solve the above problem. The queue size (n) and queue length are in unit of packet, and the outgoing traffic rate ($rate_{out}$) is in unit of packet per second.

B. Assumptions on Fairness

Here we make some assumptions on the traffic and probes, which are necessary to maintain the fairness between probing and baseline traffics:

1. Assume that the probability of loss for both traffic and probes are the same.
2. Assume that the router queue are managed in FIFO, and all traffic in different size and protocol are treated fairly.

3. Assume that no active queue management method such as RED is applied in the router, so a simple drop-tail approach is adopted.

4. Assume that average packet rate and packet size of baseline traffic are not affected by the probes.

Probe packets may not be treated the same way as the baseline traffic for all routers. For example, like what have been implemented in the Click router, the driver let the administrator to classify packets according to their characteristics, such as the layer-3 protocols and flags within the headers. Vendors try to tune the router in the way to optimize the performance that user perceive while the detailed machanism remains hidden. For load balancing or fairness of different traffic protocol, packets of different characteristics are discarded not under the same circumstances. For example, UDP and TCP packets, or packet in different size range, may be stored in different queues and have different priority of transmission. Algorithms like these challenges any kinds of measurements. Our methodology is more sensitive to the fairness on packets of different size range. We perform the experiment on testbed consists of "fair" routers only. On the other hand, we assume the fairness on packet drop probability of different packet sizes in the network components apart from the routers.

By injecting probe packets of different size at different rate, we try to alter the average packet size of all traffic flowing through the concerned router. Assume that the baseline traffic remains constant, the weighted average packet size becomes a linear function of the probe size and rate. If the underlying traffic is, in fact, change while we inject probe of different size and rate, we cannot base on the

abovementioned linear relationship to measure the effect of average packet size on queuing delay. To validate this assumption, experiments must be carried out in a real end-to-end path. Although we have not performed an Internet experiment, since our methodology injects low intensity traffic to the end-to-end path being measured, no effect on the actual baseline traffic is expected.

The third assumption is just inherited from the original Loss-Pair [11]methodology. The arithmetics below are based on the assumptions mentioned. The validity of these assumptions will be tested in the testbed experiment.

C. Framework on size estimation of router's buffer in unit of packet

Here we focus on the second model that manage router's buffer in unit of packet. Similar to the previous model but in unit of packet instead of byte, the relationship between queue size (n), queuing delay (w) and outgoing packet rate ($rate_{out}$) of the router can be written as:

$$n = w \times rate_{out} \tag{1}$$

In terms of packet, for a given outgoing packet rate $rate_{out}$, the time (t) required to drain a full buffer can be simply expressed as:

$$t = \frac{n}{rate_{out}} \tag{2}$$

For a given queue size of the bottleneck router (n) and the average packet size of queued packets (s_{avg}), we can calculate the total size, in unit of byte, of queued packets in a full buffer by $n \times s_{aMethodologyonvg}$. Then, given the outgoing bandwidth (r), t can be written, in unit of byte, as:

$$t = \frac{n \times s_{avg}}{r} \tag{3}$$

Therefore, combine (2) and (3) to eliminate t, we have:

$$\frac{n}{rate_{out}} = \frac{n \times s_{avg}}{r}$$

Re-arrange the equation, we got:

$$rate_{out} = \frac{r}{s_{avg}} \tag{4}$$

Substitute (4) into (1), we have:

$$n = w \times \frac{r}{s_{avg}} \tag{5}$$

Or, re-arrange the equation in terms of w,

$$w = \frac{n \times s_{avg}}{r} \tag{6}$$

This equation summarize the relationship between queue size (n), queuing delay (w), outgoing bandwidth (r) and average traffic packet size (s_{avg}). We note that s_{avg} is introduced to reflect the change of unit of queue size from byte to number of packet. The bandwidth and queue size of router is normally fixed. From the equation, we can see that the change of average packet size will result in a change of queuing delay (w).

Our next step is to exploit the effect of change of average packet size on the queuing delay. According to our assumptions, the change of probe rate (α_p) and probe size (s_p) do not affect the traffic packet rate (α_t) and packet size (s_t), and the loss rate for traffic and probe are the same. Therefore, we can calculate the weighted average packet size as the following:

$$\begin{aligned} s_{avg} &= \frac{\alpha_t}{\alpha_t + \alpha_p} \times s_t + \frac{\alpha_p}{\alpha_t + \alpha_p} \times s_p \\ s_{avg} &= \frac{\alpha_t \times s_t + \alpha_p \times s_p}{\alpha_t + \alpha_p} \end{aligned} \tag{7}$$

By Substituting (7) into (5), we have:

$$n = \frac{w \times r \times (\alpha_t + \alpha_p)}{\alpha_t \times s_t + \alpha_p \times s_p} \tag{8}$$

As we know, the queue size (n) and bandwidth (r) are fixed. Re-arrange the equation to express queuing delay in terms of other variables, we got:

$$
\begin{aligned}
w &= \frac{n}{r} \times \frac{\alpha_t \times s_t + \alpha_p \times s_p}{\alpha_t + \alpha_p} \\
w &= \frac{n \times \alpha_p}{r \times (\alpha_p + \alpha_t)} \times s_p + \frac{n \times \alpha_t \times s_t}{r \times (\alpha_p + \alpha_t)}
\end{aligned} \tag{9}
$$

The equation (8) is in slope-intercept form. For queue size (n), bandwidth (r), average baseline traffic packet size (s_t) and traffic rate (α_t) remains constant, and we fix probe rate (α_p) to a specific value, there is a linear relationship between the queuing delay (w) and the probe size (s_p). In addition, the slope of the line depends on the probe rate (α_p) we choose, although the rate of change of the slope, i.e. that of $\frac{\alpha_p}{\alpha_p+\alpha_t}$, is not directly proportional to the rate of change of α_p. By adjusting the probe rate (α_p), we can change the slope of the line of results.

One more interesting phenomenon is noted from the equation. If the probe packet size equals to the average baseline traffic packet size, i.e., $s_p = s_t$, from equation (9),

$$
\begin{aligned}
w &= \frac{n}{r} \times \frac{(\alpha_t + \alpha_p) \times s_t}{\alpha_t + \alpha_p} \\
w &= \frac{n}{r} \times s_t
\end{aligned} \tag{10}
$$

Therefore, the queuing delay (w) does not vary with probe rate (α_p) when the probe size (s_p) equals to the baseline traffic packet size (s_t). As we have previously revealed that the queuing delay is a linear function of probe size (s_p),

and the slope of the line of result depends on α_p, we will expect a unique common intersection point of lines of results obtained from different probe rate (α_p). Therefore, queuing delay measured in different probe rate (α_p) are equal if and only if the probe size (s_p) equals to the average baseline traffic packet size (s_t). This observation will be validated with testbed experiment in the coming section.

7 Methodology

Base on the relationship as modeled in the above section, we propose a methodology for active measurement summarised as follows:

1. Inject probes of different packet sizes at two probe rates;
2. The average packet sizes for all packets that flow through the bottleneck router is then changed slightly for each case;
3. The queuing delay observed for each case varies due to different average packet size;
4. By exploiting the relationship between probe packet size and queuing delay formulated in the above section, we can estimate the queuing delay in unit of packet.

In passive measurement, the packet size of traffic flowing through the bottleneck router cannot be estimated, and the loss-pair analysis proposed in [11] does not rely on the average packet size to estimate the bottleneck router's queue size. However, the average packet size is one of the key value in estimating the queue size in the unit of packet. No passive measurement is expected to observe this value.

Our philosophy is to actively change the average packet size by controlling the packet size of probes being injected into the end-to-end path, and then observe the change of the resulted queuing delay. The packet injected should resulted in

change of queuing delay that is measurable, while the impact on the undergoing traffic is minimal. Based on the relationship shown in equation (9), we should be able to estimate the baseline traffic average packet size and then the queue size of the bottleneck router.

Our methodology will be illustrated with testbed results in the coming section.

8 Validation with Testbed Results

A. Other Considerations on Probe Packet Round Trip Delay

Before we can validate the formulae in the previous section, we must first consider all factors that contribute to the round trip delay from probe packets. As mentioned in previous sections, despite the inherit propagation delay in an end-to-end path, there are many factors contributing to the RTT (d_p) of a packet flowing along the path. Let the packet size of probe be s_p, the propagation delay along a path be d, bandwidth at the i-th link be r_i, then the packet service time (d_i^s) for a packet being forwarded onto the i-th link at a router can be expressed as:

$$d_i^s = \frac{s_p}{r_i}$$

The queuing delay at the bottleneck router just before the narrow link, w', thus the time to discharge the cross-traffic packets queued in front of the probe in the bottleneck router's buffer, is directly proportional to queue length at the bottleneck router when the packet entered the router. We can use equation (6)to estimate the queuing delay, with queue length (q_l) instead of queue size since the queue may not be full. Therefore,

$$w' = \frac{q_l \times s_{avg}}{r}$$

Without concerning the queue size of other routers, the round trip delay of a packet travelling along an end-to-end path can be written as:

$$
\begin{aligned}
d_p &= d + \sum_{1}^{n} d_i^s + w' + rand \\
d_p &= d + \sum_{1}^{n} \left\{ \frac{s_p}{r_i} \right\} + \frac{q_l \times s_{avg}}{r} + rand \\
d_p &= d + s_p \times \sum_{1}^{n} \left\{ \frac{1}{r_i} \right\} + \frac{q_l \times s_{avg}}{r} + rand
\end{aligned} \tag{11}
$$

where *rand* denotes the random components out of the scope of this formula, such as the queuing delay at other router.

Therefore, with base propagation delay (d), bandwidths of all links along the path (r_i), and average baseline traffic packet size (s_{avg}) remains constant, the round trip delay for a probing packet is a function of probe size (s_p) with third and fourth components randomly change from time to time. The random factors, however, can be eliminated by statistics. To eliminate both random components is simple: if the experiment are conducted long enough, we can apply a min-filter to the data collected to obtain a minimum round trip delay, which should be closed to:

$$
min(d_p) = d + s_p \times \sum_{i=1}^{n} \left\{ \frac{1}{r_i} \right\} \tag{12}
$$

For a loss-pair, the buffer of the bottleneck router is nearly full when the probe arrive at the router, thus $w' = w$. Substitute $w' = w$into (11),

$$
d_{p(losspair)} = d + s_p \times \sum_{i=1}^{n} \left\{ \frac{1}{r_i} \right\} + w + rand \tag{13}
$$

Measuring queuing delay of loss-pair (w) is key to the estimation of queue size (n) in linear regression of straight lines represented by equation (9). Therefore, we have to remove the random component denoted by *rand* while retaining the key figure w. Unlike the minimum round trip delay for normal packets ($min(d_p)$), we cannot apply a min-filter to remove the random factors for loss-pair.

The major reason that prevent us from eliminating the random component by min-filter can be traced to the situation illustrated in Figure (4). Packet loss occurred when the bottleneck router is under heavy cross-traffic. The likehood that packet pairs are interferred by cross-traffic packet is high. The min-filter will probably return a distorted value which is lower than the actual figure, and the resulting measured buffer size is significantly under-estimated. Similar to [11], the statistical mode value of round trip delay returned from loss-pair is the best option currently available. We will further discuss the validity in later section.

Assume the modal value of loss-pair round trip delay ($d_{p(losspair)}$) eliminates the last random component, we have

$$mode(d_{p(losspair)}) = d + s_p \times \sum \left\{ \frac{1}{r_i} \right\} + w \tag{14}$$

Therefore, to estimate the queuing delay, we can subtract (12) from (14):

$$mode(d_{p(losspair)}) - min(d_p) = w \tag{15}$$

B. Packet Size and Queuing Delay

We set the bandwidth through the bottleneck router (r) to 256Kbit/s, queue size (n) to 10, and no baseline traffic through the nodes. Probes of packet sizes from 200 byte to 1400 byte, without considering ethernet and TCP/IP header, are sent from the Probe Sender machine to the Web Server at the other end through the bottleneck router. The probe rate is high enough to fill the buffer and generate packet loss at the bottleneck router.

[11] assumes that the minimum round trip delay along the path has been estimated in some way, such as observing the RTT and min-filtering, although the mode of RTT of normal packets traversing the bottleneck link is used as illustration. queuing delay is estimated by the difference of the RTT of packet pair that incurred loss and the minimum RTT along the path.

Unlike [11], the relationship between probe packet size and minimum round trip delay is important to our experiment. First we have to estimate the minimum round trip delay ($min(d_p)$) and modal loss-pair round trip delay ($mode(d_{p(losspair)})$) where probe packet size equals to the average baseline traffic packet size, then by following equation (15), we calculate the difference of the two value to estimate the queuing delay (w) experienced by baseline traffic. Based on the queuing delay obtained we can determine the queue size (n) of the bottleneck router.

To limit the quantity of results to be obtained and the duration of experiment, we send probe packet of sizes range from 200 to 1400 bytes, and a difference of 100 bytes is maintained between adjacent values of packet size. Linear regression is performed to look for an intersection point foreseen in equation (10), which

occurred when probe packet size (s_p) equals to baseline traffic packet size (s_t).

Figure 11 shows the expected minimum RTT according to equation (12) and the actual data obtained from our testbed experiment. We ignore the actual propagation delay in the 100-based Ethernet testbed and effect of packet service time in the 100-based Ethernet switches in calculating the expected values, as they are negligible. The result is generally consistent to the predicted value, with some minor difference.

To determine a modal RTT of loss-pair for each probe size, it is necessary to specify a bin width. We have set bin width to 0.25ms, as it gives a reasonable granularity while the probability of providing a unique mode on RTT is high for the experiment data. All our results thereafter use the same bin width.

The statistical mode of RTT of probes returned, in which exactly the second packet in the back-to-back packet pair is lost, is chosen as the estimator of RTT of packet that incurred the full queuing delay at the bottleneck router. The case where ther first packet is lost is not chosen because the RTT of the second packet in a packet pair includes a effect of packet dispersion that is used for capacity estimation in [4]. This deviates from the methodology in [11] where both cases are considered valid. In our testbed setting with the bottleneck set to 3Mbit/s, for a probe packet of 700 bytes in size, from equation a time dispersion of $\frac{700\times 8}{3000000} = 1.867ms$ is expected. Assume that the packet size of baseline traffic equals to 700 bytes too, by equation (10), for a queue size of 50 packets, expected queuing delay equals to $\frac{50}{3000000} \times 700 \times 8 = 93ms$. A percentage error of 2% is resulted. We will try to avoid the error by excluding the results from loss packet in which the first packet is lost.

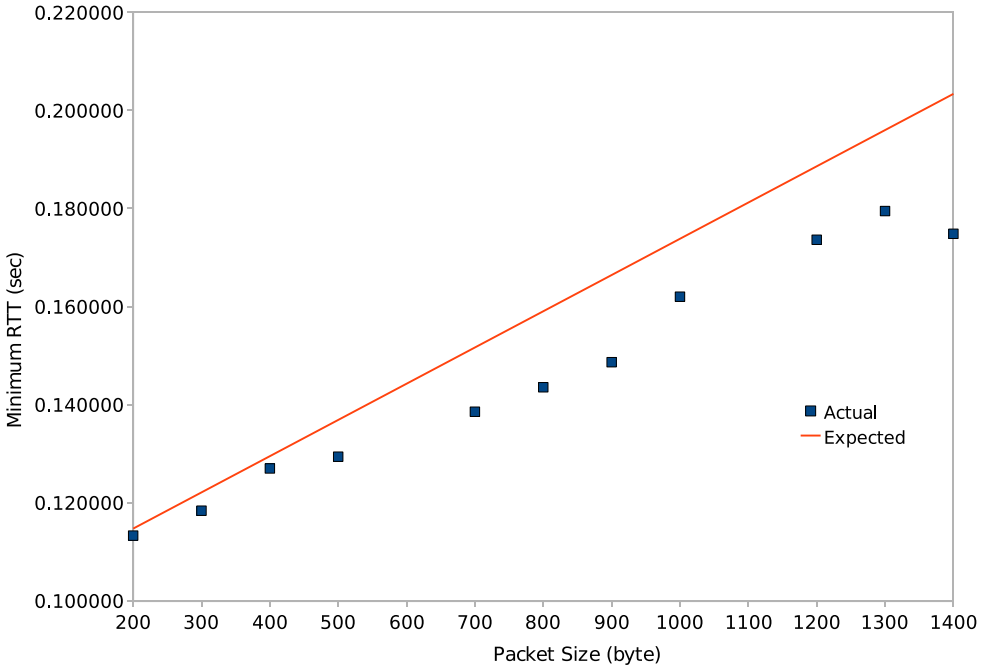

Figure 11: Minimum RTT for different traffic packet size

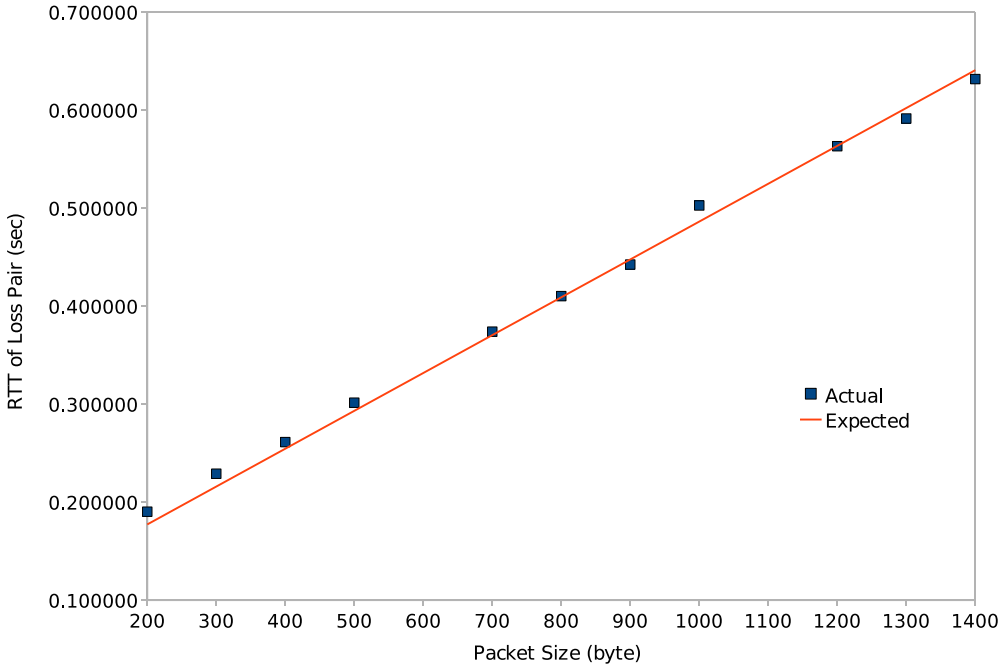

Figure 12: Modal loss-pair RTT for different traffic packet size

Figure 12 shows the loss-pair RTT obtained from experiment and the respective expected value from equation (14). The observation is strictly consistent to our prediction.

From equation (15), queuing delay is estimated by the difference of minimum RTT of probes not incurring loss or re-ordering and the modal RTT of loss-pair probes. Figure 13 shows the result calculated from the experiment data is again consistent to the predicted value. We conclude that a linear relationship between the queuing delay and the packet size exists.

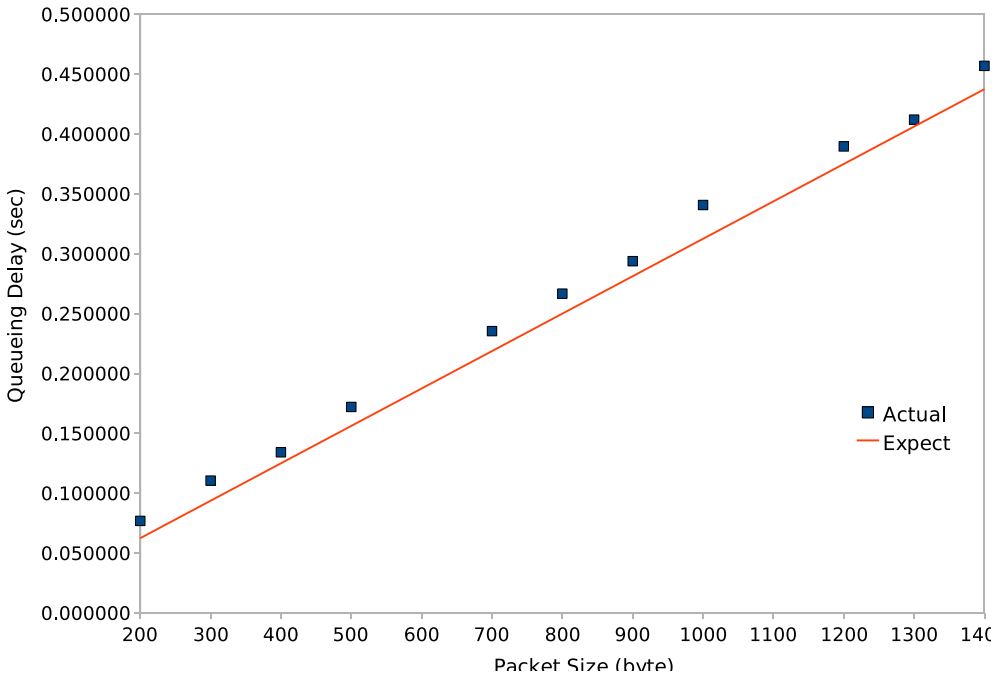

Figure 13: Queuing delay for different traffic packet size

C. Probe Packet Size and Baseline Traffic Packet Size

Next, we configure the bandwidth through the router 3 back to 3Mb/s, and generate hop-persistent baseline traffic across the router using a traffic generator and a sink next to the router. The stream of traffic fixed at a rate of 1150 packet/s and each packet is 342 byte in size including IP and Ethernet header. Same as in previous experiment, probes of size varies from 200 to 1400 byte are sent through the testbed at a rate of 10 probes per second. The experiment is then repeated at a probe rate of 30 probes per second.

As stated in equation 9, and reinforced by previous experiment results, there is a linear relationship between loss-pair RTT and probe packet size for any given probe rate. The slope of the line of results increases with increasing probe rate (α_p). Moreover, as predicted by equation 10, an intersection point exists for the lines of results at $s_p = s_t$. It is because w for different probe rate (α_p) are equal if and only if $s_p = s_t$., and so are $mode(d_{p(losspair)})$ for different α_p.

The modal round trip delay of loss-pair of different probe packet size in the two sets of experiment are shown in Figure 14. The two sets of experiment data, as we foresee, follow two linear trends with different slopes, and only intercept at one point where $s_p = s_t \sim 354$ bytes (including ethernet and TCP/IP header).

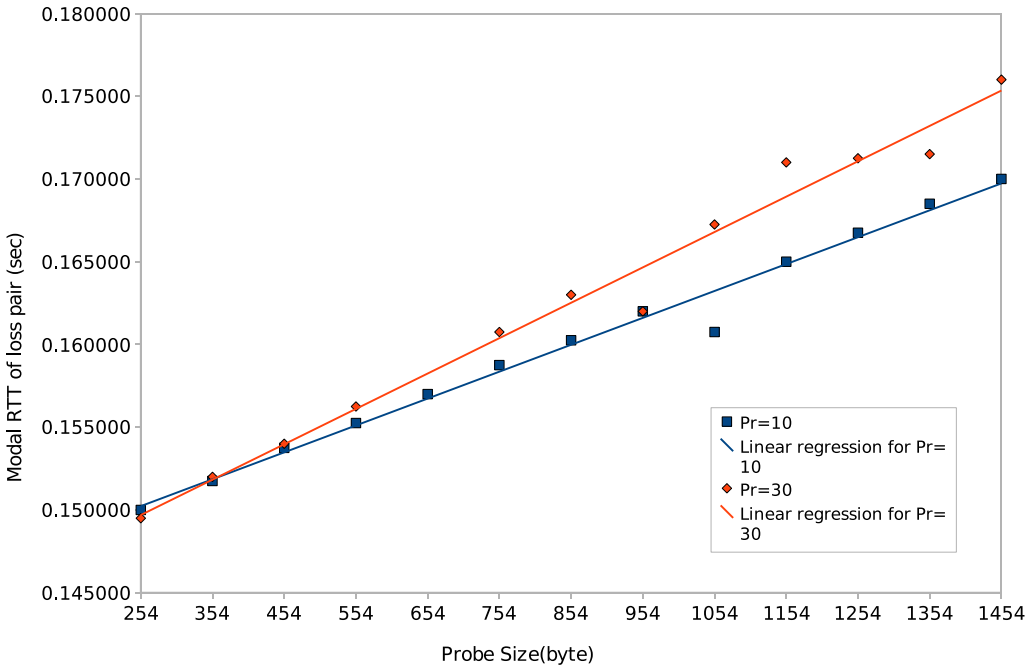

Figure 14: Loss-pair RTT for two probe rates under fixed baseline traffic packet size and rate

D. Diverseness of Baseline Traffic Packet Size

To study the effect of diverseness of baseline traffic packet size, we configure the cross-traffic of UDP packets with packet size uniformly distributed from 500 to 1000 bytes, and at a fixed rate of 480 packet/s. We have send probes of size varies from 200 to 1400 bytes, at the rate of 15 and 30 packet/s.

Figure 16 shows the lines of modal loss-pair RTT of probe rate 10 and 30 probe/s respectively under baseline traffic of uniformly distributed packet size from 500 to 1000 bytes. The two lines intersects at s_p = 717 bytes, which is near the true s_t of 784 bytes.

Figure 17 and 18 are the comparison of loss-pair RTT frequency distributions under baseline traffic of fixed and uniformly distributed packet size. Under the fixed-size baseline traffic, the mode is highly distinguishable, and the other data are densely packed near the mode. On the other hand, under variable-size baseline traffic, the frequency distribution forms a bell shape, with the mode sits near the middle of the peak. Clearly, the distribution of baseline traffic packet size do not prevent us from obtaining a clear mode for our analysis.

We then study the effect of strong multi-modalities baseline traffic packet size (s_t). Research shows that for real Internet traffic, packet size of traffic are centered in a few number of values, namely 40, 576, and 1500 bytes [17]. Figure 15 shows the

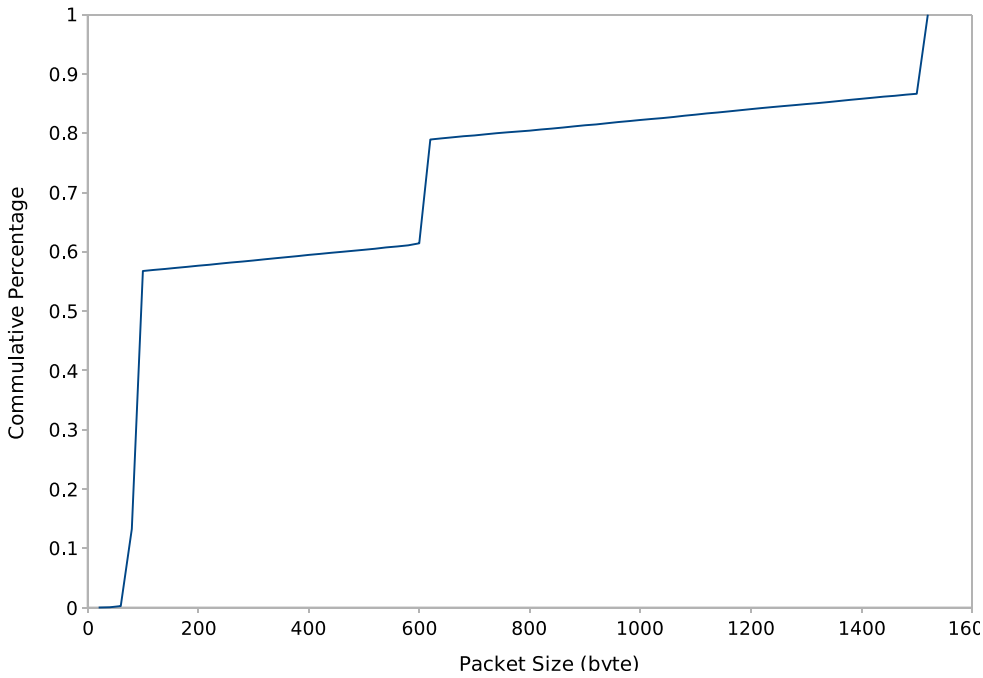

Figure 15: Cumulative percentage of simulated Internet traffic

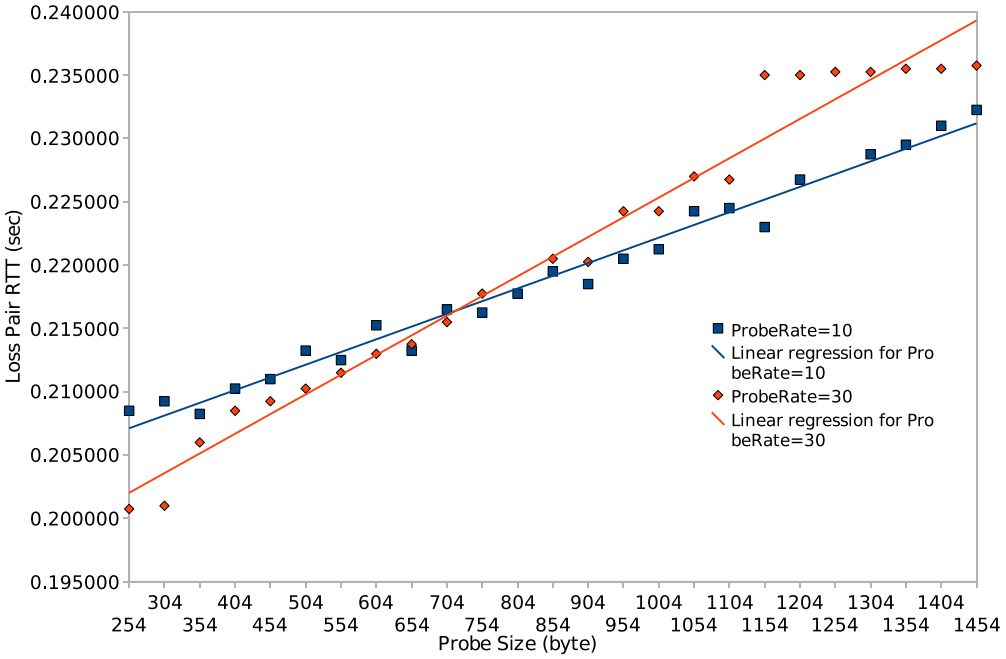

Figure 16: Loss-pair RTT for different s_p under baseline traffic of s_t from 500 to 1000 bytes and α_t= 480 packet/s

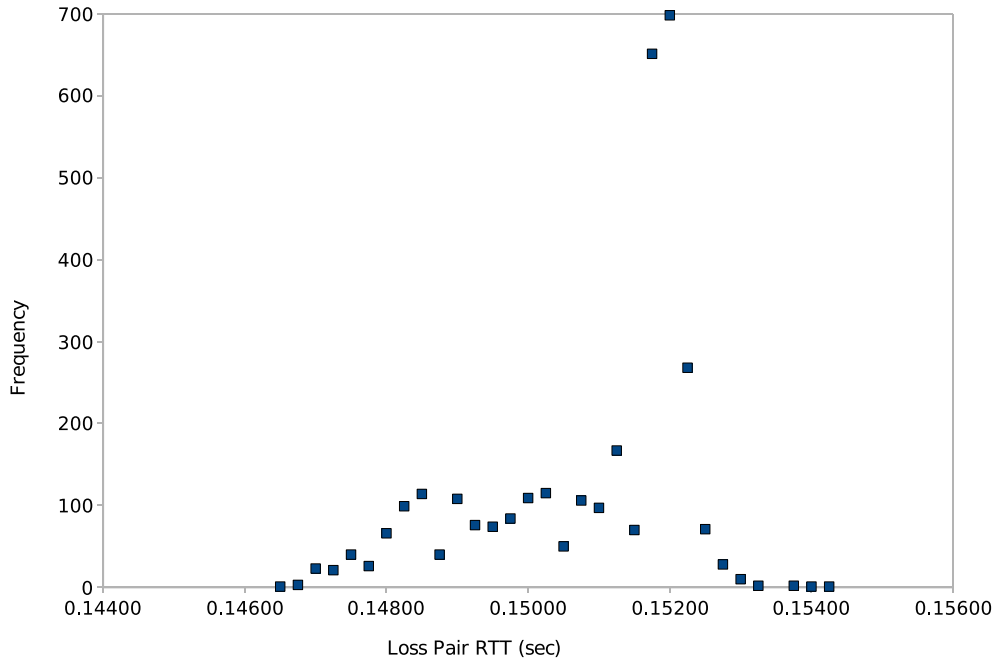

Figure 17: Loss-pair RTT frequency distribution for α_p= 30, s_p= 300, α_t= 480, s_t= 354 (fixed)

cumulative percentage of the simulated cross-traffic that goes through the bottle-neck router, and Figure 19 shows the loss-pair RTT frequency distribution. The frequency distribution again exhibits a clear bell shape with sharp cental tendency.

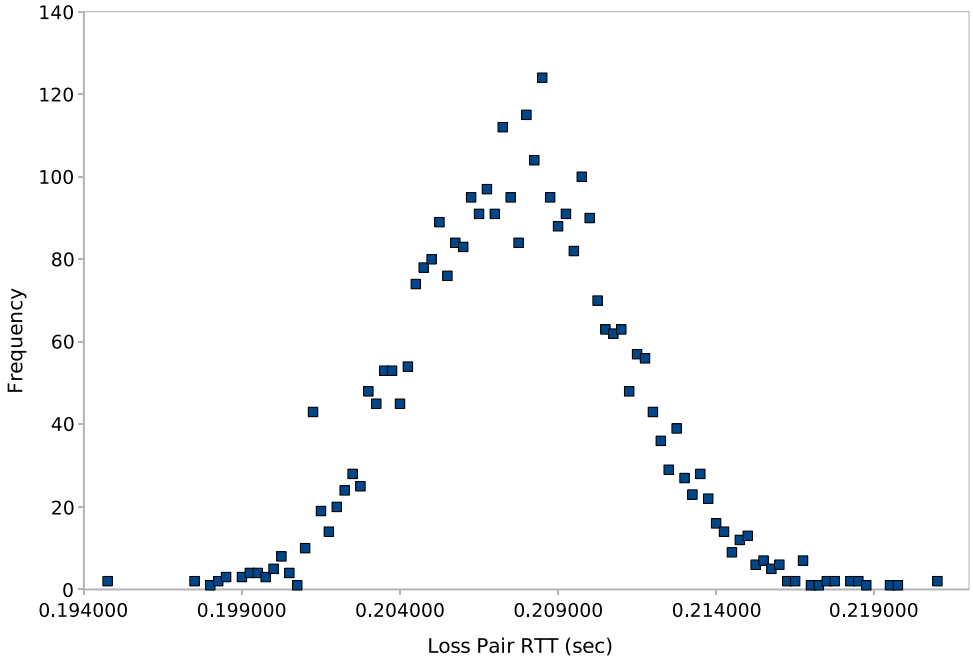

Figure 18: Loss-pair RTT frequency distribution for α_p= 30, s_p= 350, α_t= 480, s_t from 554 to 1054 (uniformly distributed)

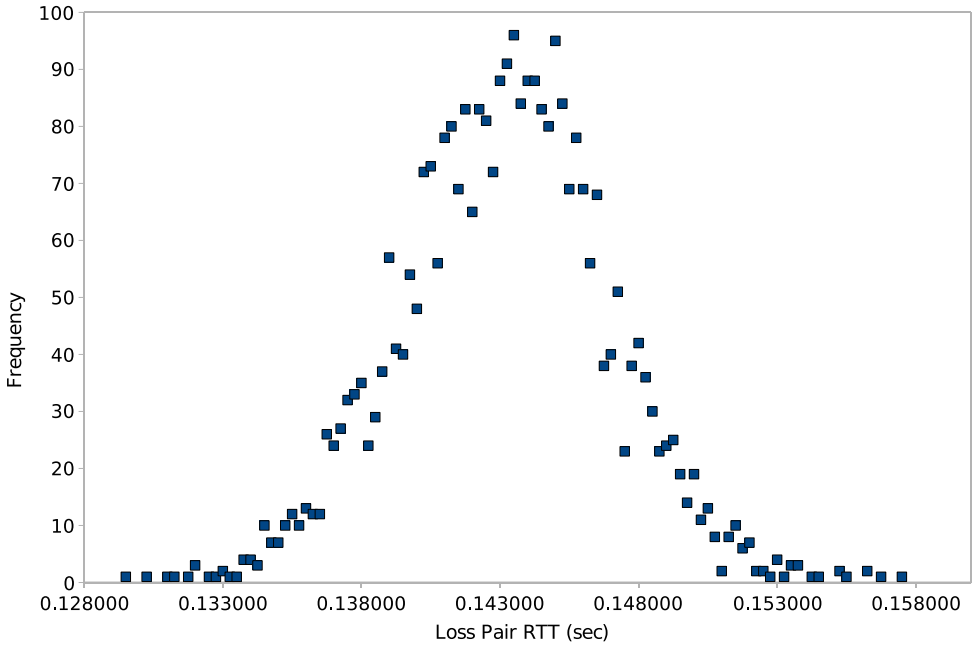

Figure 19: Loss-pair RTT frequency distribution for cross-traffic illustrated in Figure 15, s_p = 254, α_p = 30

E. Estimation of Queue Size

After estimation of average traffic packet size, s_t, the next step is to calculate the queue size n. In equation (5), assume that the bandwidth can be found by methodology such as [4] and [6], the only unknown is the queuing delay (w), which can be computed in equation (15).

For experiment of constant rate baseline traffic in Section VII (C), we cannot obtain a representative minimum RTT from the probes. It is because the traffic is too high, such that in most of the time the queue length remains at a high level and seldom reach zero. It only happens to constant bit rate traffic and heavy traffic. In real-life, there always enough room for probing pair to reach the bottleneck router when the queue is empty.

We send packet pairs through the testbed while no cross-traffic is configured to obtain the minimum RTT for our calculation. It represents the best value we can obtain in experiment. The results is shown in Figure 20.

The steps to estimate the queue size (n) is summarised as follow:

1. Estimate the baseline traffic packet size (s_t);
2. Perform linear regression to obtain the loss-pair RTT for $s_p = s_t$;
3. Perform linear regression to obtain the minimum RTT for $s_p = s_t$;
4. Calculate w as loss-pair RTT minus minimum RTT inferred in step 2 and 3;
5. Calculate n from equation (1).

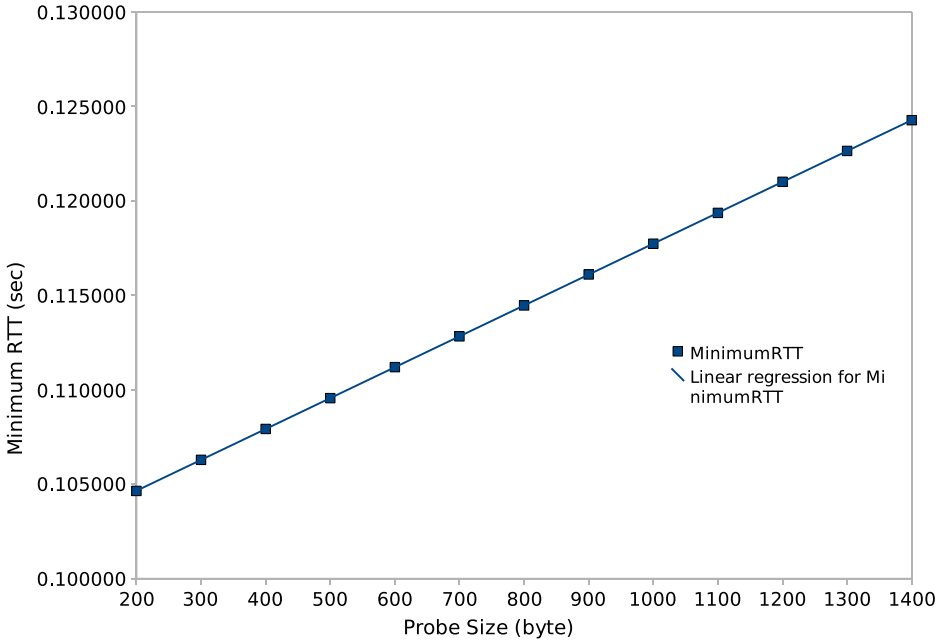

Figure 20: Minimum RTT for different probe packet size

We use the results shown in Figure 14 for illustration. The loss-pair RTT according to the regression line at the intersection point (359 bytes) is 0.151932 s. The respective minimum RTT from regression line in Figure 20 is 0.107254 s. queuing delay, w, which is the difference of the two, equals to 0.044678 s. Queue size, n, is then calculated as:

$$n = \frac{w \times BW}{s_t} = \frac{0.107254 \times 3000000}{359 \times 8} = 46.66$$

This estimated value is quite close to the true value of 50. The error is 6.67%.

We configure the queue size at the bottleneck router from 30 to 60 packets. Baseline traffic are generated across the bottleneck router, with uniformly distributed

n	Measured t_s	Measured w	Estimated n
30	803.42	0.061295	28.61
40	855.49	0.082657	36.23
50	717.12	0.103304	54.02
60	794.19	0.124853	58.95

Table 2: Summary of queue size estimation

packet size (s_t)from 500 to 1000 bytes and fixed traffic rate (α_t) of 480 packet/s. Queue size of the bottleneck router (n) is set to 30, 40, 50, and 60 respectively for each set of experiment. The result of queue size estimation is summarised in Table 2.

Result shows that the error of estimation ranges from 2% to 9%, which is generally acceptable.

9 Further Study on More Complex Distribution of Baseline Traffic

To further study the effect of a complex mix of baseline traffic on our methodology, we configure the packet size and inter-departure time of the baseline traffic as follows:

1. Multi-modal packet size (40, 576, and 1500 bytes) and fixed inter-departure time.
2. Multi-modal packet size (40, 576, and 1500 bytes) and uniformly distributed inter-departure time.
3. Both packet size and inter-departure time follow Pareto distribution.

The first configuration serves as an control experiment to study the effect of distribution of packet size. The later two configurations are effectively causing some burst of traffic in different packet size, which we anticipate that has most impact on the results obtained.

Figure 23 shows a typical frequency distribution of loss-pair RTT observed. The bell-shaped graph is deformed, with a significant peak departed from the middle. The cumulative frequency distribution for different probe size is shown in Figure 24. We noted that there exists a distorted portion at around 202ms for a number of probe rates. The modal loss-pair RTT for high probe rate are therefore affected. In Figure 21, 22 and 25, we see that due to such distortion, the linear regression on the modal loss-pair RTT fails to locate the average baseline traffic packet size.

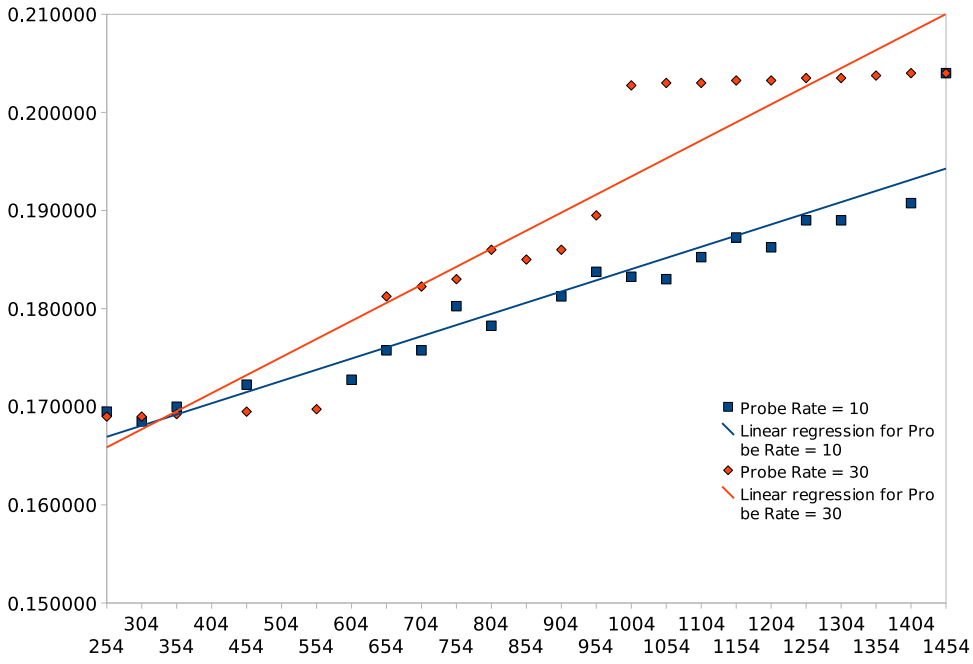

Figure 21: Loss-pair RTT under baseline traffic of multi-modal size and fixed rate

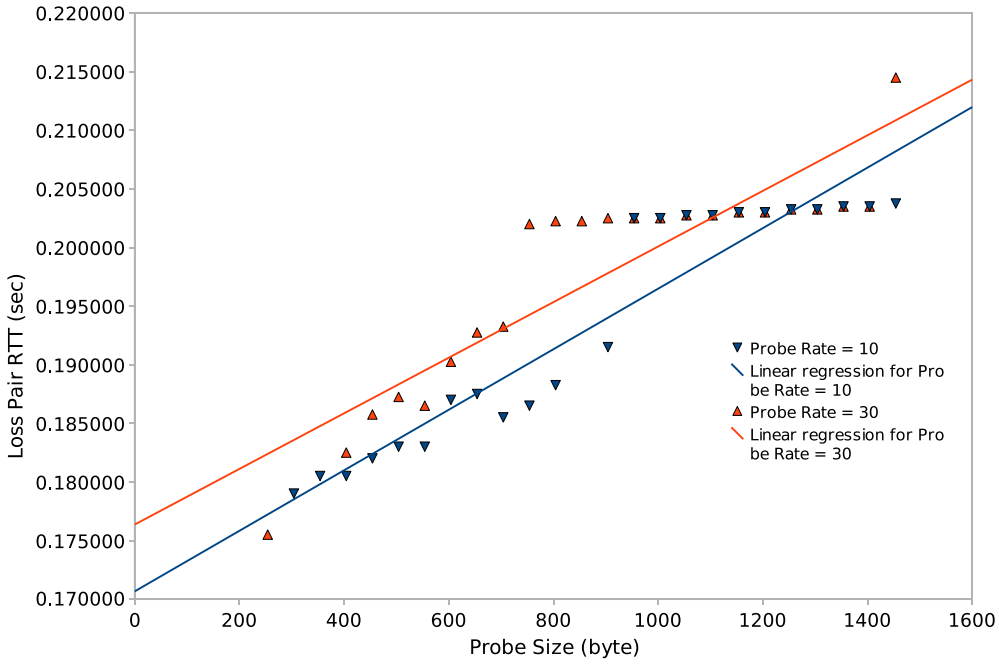

Figure 22: Loss-pair RTT under baseline traffic of multi-modal size and uniformly distributed rate

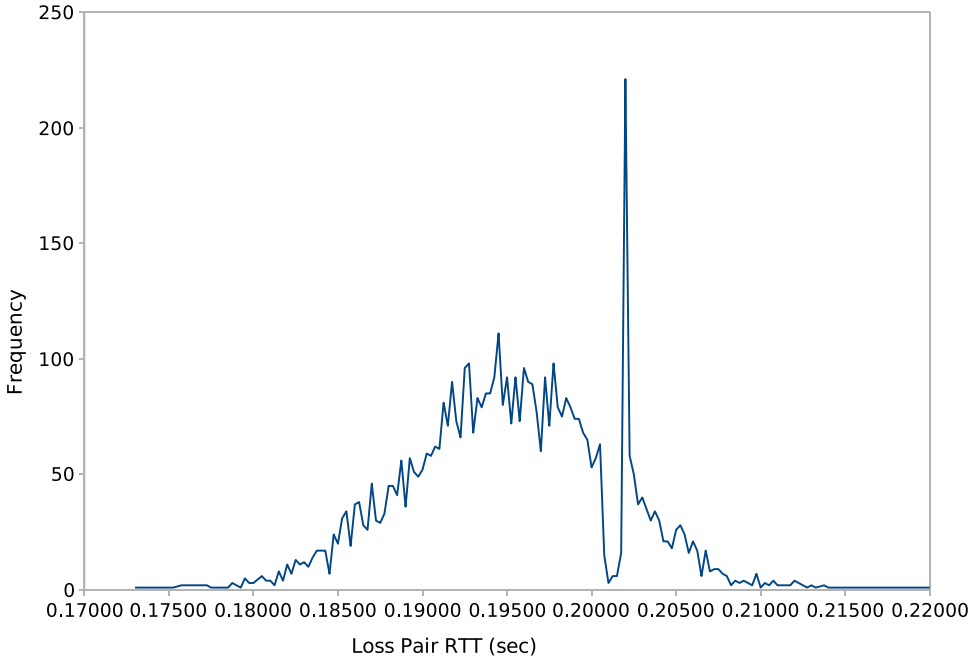

Figure 23: Loss-pair RTT frequency distribution of cross-traffic with multi-modal size and uniformly distributed rate, probe size = 754 bytes

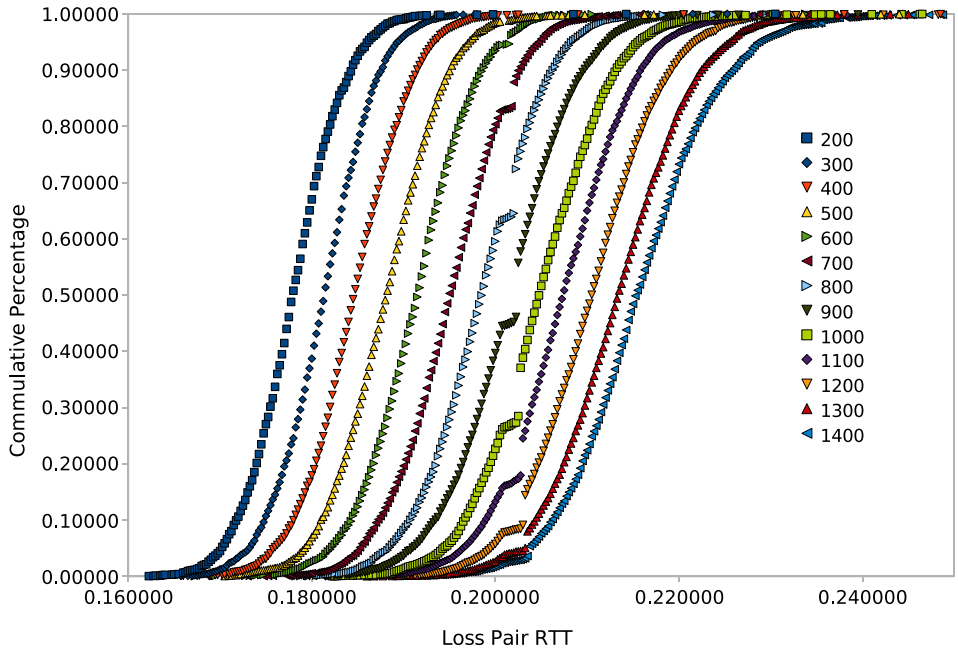

Figure 24: Cumulative frequency distribution of loss-pair RTT under baseline traffic of multi-modal size and uniformly distributed inter-departure time

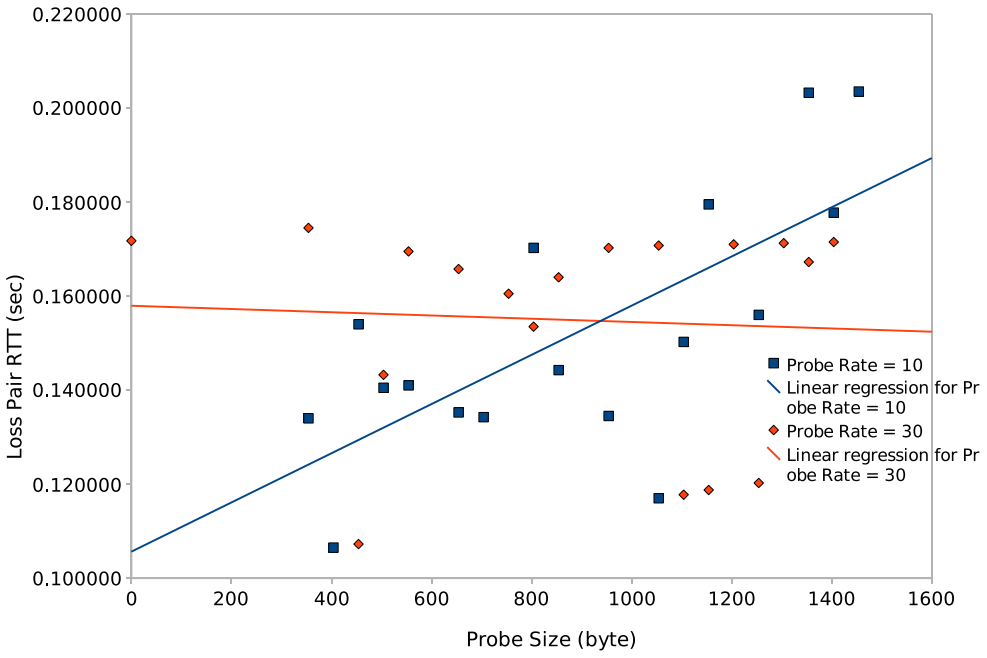

Figure 25: Loss-pair RTT under baseline traffic of size and inter-departure time in Pareto distribution

For all of the above, the packet sizes of a significant portion of baseline traffic packets are much smaller than the probe packets. The impairment is mainly due to the probability that small-size cross-traffic packets being inserted in between packet pair increases with decreasing relative baseline traffic packet size. This effect is studied in [4].

There are two possible measures to lower such effect. One is to lower the probe packet size. Another one is to apply the linear regression on the mean or median of loss-pair RTT. The former method simply lower the relative probe size to traffic packet size to lower the probability that cross-traffic packet interfere with the packet pair. The later assumes that the ingenuous mode should be located at the middle of the bell shape graph. Mean and median is less likely affected by

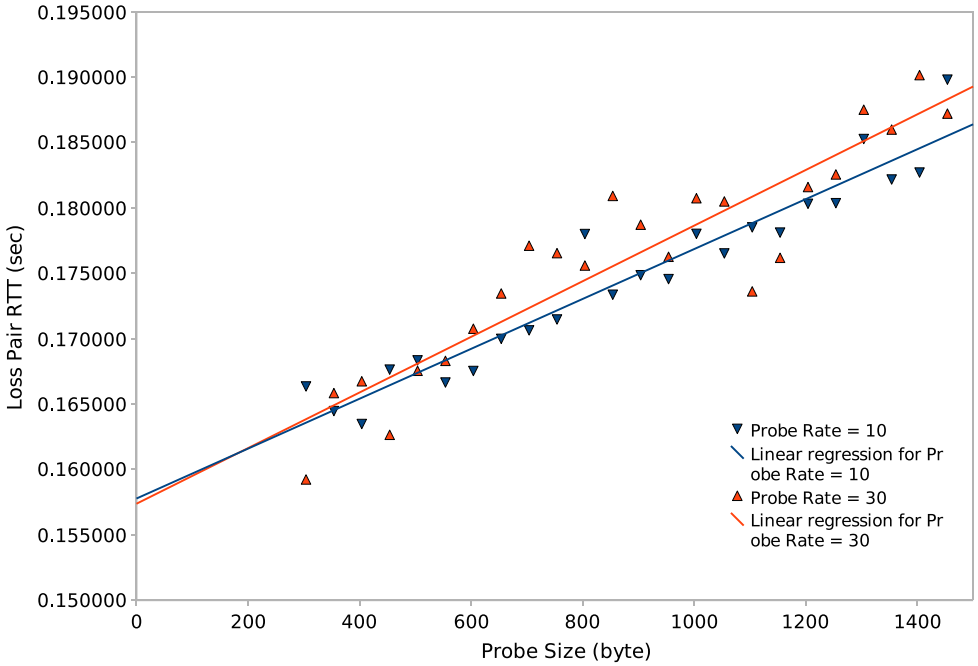

Figure 26: Linear regression on mean loss-pair RTT

outliers data. Due to the constraint of tool, we cannot adopt the first method. We demonstrate the second one in Figure 26 and 27 for mean and median of loss-pair RTT respectively.

The estimation of bottleneck router queue size is summarized in Table 3. The measured queue size is much closer to the true value of 60 for estimation using mean and median.

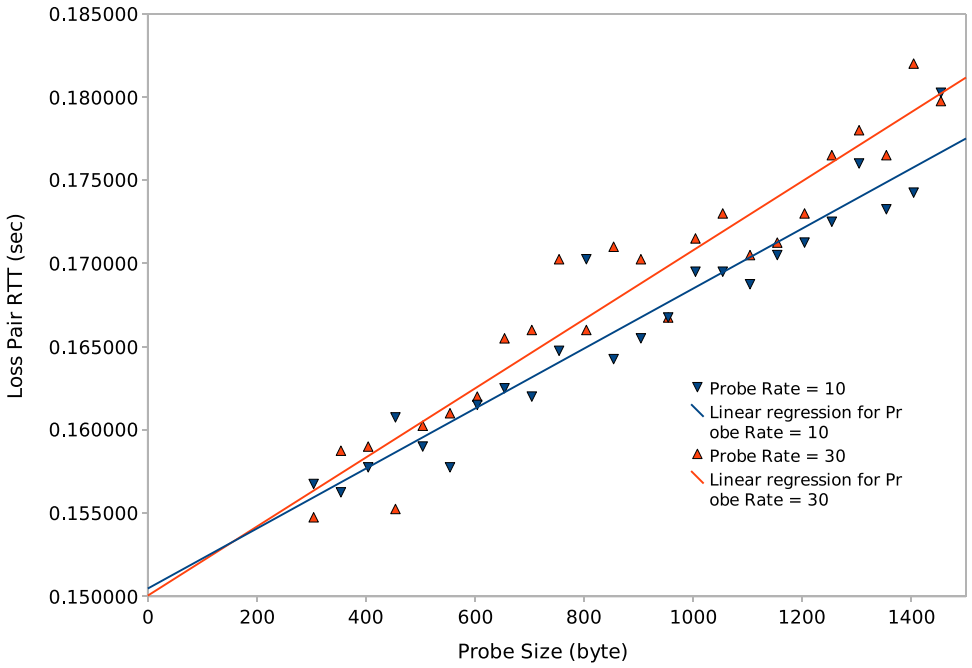

Figure 27: Linear regression on median loss-pair RTT

Method	*Measured* s_t	*Measured* w	*Estimated* n
Mode	889.45	0.038873	16.39
Mean	264.96	0.058136	82.28
Median	251.95	0.050593	75.3

Table 3: Comparison of queue size estimation by regression on mode, mean and median

10 Limitation and Future Works

As mentioned in the previous section, when there exists a significant portion of small-size packets in baseline cross-traffic, the lines of regression of modal RTT obtained from testbed experiment are distorted. To provide a stronger basis for actual Internet measurement, there are still room for further study on the interaction of probe packet size and cross-traffic packet size.

11 Conclusion

Loss-pair passive measurement has been shown to be able to measure the queue size in the bottleneck router along a path, but the application is limited to router for which the queue is managed in the unit of byte [11]. We show that the passive measurement is not applicable to router that have queue length allocated in unit of packet. This renders the methodology useless in Internet measurement, since the approach cannot apply to at least one of the popular buffer allocation method deployed in routers.

To address the issue, we have proposed an active measurement methodology, which is based on a model that relates queuing delay, average packet size, bandwidth and queue size in unit of packet. By sending probe packets of different size at different rate, we can change the average packet size through the bottleneck router. We can then apply our model to estimate the average packet size of the baseline traffic, and eventually the queue size of the bottleneck router.

Our methodology use linear regression to estimate the average packet size and delay. In this way, random error can be effectively filtered out. Testbed experiment results show that our estimation on the queue size of the bottleneck router in the unit of packet is fairly accurate.

However, we found that our methodology is seriously affected by intensive and relatively small-size cross-traffic packets. To cope with this problem, we can send probes with small size, and perform linear regression on the mean or median of loss-pair RTT. Under heavy network utilization, we show that the later method

has significantly improve the accuracy of measurement. The interaction of probe packet size and cross-traffic packet size will be studied further.

References

[1] L.-J. Chen M. Y. Sanadidi A. Persson, C. A. C. Marcondes and M. Gerla. Tcp probe: A tcp with built-in path capacity estimation. 2005.

[2] D. Barman, G. Smaragdakis, and I. Matta. The effect of router buffer size on highspeed tcp performance. *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, 3:1617–1621 Vol.3, Nov.-3 Dec. 2004.

[3] Amogh Dhamdhere and Constantine Dovrolis. Open issues in router buffer sizing. *SIGCOMM Comput. Commun. Rev.*, 36(1):87–92, 2006.

[4] Constantinos Dovrolis, Parameswaran Ramanathan, and David Moore. Packet-dispersion techniques and a capacity-estimation methodology. *IEEE/ACM Trans. Netw.*, 12(6):963–977, 2004.

[5] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. Netw.*, 1(4):397–413, 1993.

[6] Rohit Kapoor, Ling-Jyh Chen, Li Lao, Mario Gerla, and M. Y. Sanadidi. Capprobe: a simple and accurate capacity estimation technique. *SIGCOMM Comput. Commun. Rev.*, 34(4):67–78, 2004.

[7] S. Keshav and R. Sharma. Issues and trends in router design. *Communications Magazine, IEEE*, 36(5):144–151, May 1998.

[8] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M. Frans Kaashoek. The click modular router. *ACM Trans. Comput. Syst.*, 18(3):263–297, 2000.

[9] Kevin Lai and Mary Baker. Measuring link bandwidths using a deterministic model of packet delay. In *SIGCOMM '00: Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 283–294, New York, NY, USA, 2000. ACM.

[10] Guang Yang-Medy Y. Sanadidi Ling-Jyh Chen, Tony Sun and Mario Gerla. End-to-end asymmetric link capacity estimation. *Networking 2005: Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, pages 780–791, 2005.

[11] Jun Liu and Mark Crovella. Using loss pairs to discover network properties. In *IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 127–138, New York, NY, USA, 2001. ACM.

[12] M. May, J. Bolot, C. Diot, and B. Lyles. Reasons not to deploy red. *Quality of Service, 1999. IWQoS '99. 1999 Seventh International Workshop on*, pages 260–262, 1999.

[13] A. Pasztor and D. Veitch. The packet size dependence of packet pair like methods. *Quality of Service, 2002. Tenth IEEE International Workshop on*, pages 204–213, 2002.

[14] Vern Paxson. End-to-end internet packet dynamics. *IEEE/ACM Trans. Netw.*, 7(3):277–292, 1999.

[15] Joel Sommers, Paul Barford, Nick Duffield, and Amos Ron. Improving accuracy in end-to-end packet loss measurement. In *SIGCOMM '05: Proceedings of the 2005 conference on Applications, technologies, architectures, and*

protocols for computer communications, pages 157–168, New York, NY, USA, 2005. ACM.

[16] Tammo Spalink, Scott Karlin, Larry Peterson, and Yitzchak Gottlieb. Building a robust software-based router using network processors. In *SOSP '01: Proceedings of the eighteenth ACM symposium on Operating systems principles*, pages 216–229, New York, NY, USA, 2001. ACM.

[17] K. Thompson, G.J. Miller, and R. Wilder. Wide-area internet traffic patterns and characteristics. *Network, IEEE*, 11(6):10–23, Nov/Dec 1997.

[18] Curtis Villamizar and Cheng Song. High performance tcp in ansnet. *SIGCOMM Comput. Commun. Rev.*, 24(5):45–60, 1994.