Path Capacity Estimation by Passive Measurement for the Constant Monitoring of Every Network Path

Naoyoshi Ohkawa and Yuji Nomura
System Software Laboratories.
FUJITSU LABORATORIES LTD.
{okawa.naoyoshi, nomura.yuji}@jp.fujitsu.com

***Abstract*—Degradation of networking quality is a serious problem for service providers (e.g., on-demand video subscription service) since it causes customer defection in their services and results in a decline of their sales. Insufficient path capacity is one of the typical causes which degrade networking quality. Especially, the recent technical trend to shift virtualized network, in which the network is dynamically and autonomically reconfigured, increases the risk of unexpected insufficiency of the path capacity in the network. To detect such an unexpected insufficiency of path capacity, a constant monitoring of the whole network is highly expected. Path capacity estimation with passive measurement method is an approach which suits to such a purpose since it adds no additional load to the target network. However, existing passive measurement methods tend to either 1) have insufficient accuracy on their estimation or 2) require heavy computational cost for their histogram analyses. In this paper, we introduce a novel path capacity estimation method by passive measurement for a constant monitoring of a network 24-hour, 365-day. Our method can estimate the capacity of each path with sufficient accuracy by eliminating the two factors of degrading the estimation accuracy, which are the influence of TCP window flow control and the influence of cross traffic on the path. We evaluated our method using packets captured from our in-company backbone network. Our method accurately estimated the capacity of the narrowest link in every 1524 connections, in which 80% of the connections were within approximately 15% of their actual values (in over 1MB traffic case), as shown in Fig. 8.**

I. INTRODUCTION

Measuring the quality of networking is important not only for user's QoE (quality of experience) but also for the businesses of content service providers (e.g., on-demand video subscription services) since degradation of network quality causes customer defection in the services and results in a decline of the service providers' sales [10]. Constant monitoring [1] of path capacity [2] is expected for such service providers to detect the degradation of their users' QoE due to the insufficient capacity of the narrowest network link between the users and the service providers. Moreover, the risk of insufficient path capacity keeps increasing since network traffic keeps growing more rapidly than the growth of network capacity and network traffic will increase threefold in the next five years due to the rapid spread of mobile devices, the wide spread of content delivery services, and also the growth of average web-page size [1].

[1]In this paper, "constant monitoring" is the unceasing measurement of the capacity of every path in a network 24-hour, 365-day.

[2]In this paper, "path capacity" is defined as the capacity of the narrowest link in the network path.

The constant monitoring of path capacity is also required for path optimization of an overlay network. Path optimization of an overlay network is a challenge to find an optimal path in the virtualized network of a cloud datacenter, which minimizes the latency of the path and maximizes the total utilization of the network. The wide spread of network virtualization technology such as Software Defined Network (SDN) increases the chance of changing paths in a network since SDN changes network paths dynamically and autonomically. In such a virtualized network, a technique to check whether the capacity of each path is configured as intended is required to avoid an unexpected insufficiency of path capacity.

The majority of existing methods for path capacity estimation are categorized into the active measurement method, which injects probe packets into a target network path to estimate its path capacity. The active measurement method can accurately estimate the capacity of a path, and further can estimate the available bandwidth [3] of the path. However, the active measurement method is unsuitable for the constant monitoring of the capacity of every path in a large-scale network 24-hour, 365-day, due to the heavy additional load for the probe packets. Additionally, some firewalls on the target path block the probe packets for a security reason and make the active measurement method unavailable.

An alternative way of path capacity estimation is the passive measurement method, which sniffs all packets that flow on a point of the target network to estimate the capacity. Although the passive measurement method suits for the constant monitoring of the capacity of every path in the network because it adds no additional load to the target network, its estimation accuracy can't be compared to that of the active measurement method due to two reasons (we will explain the reasons later in Section II-A). Several studies tackled the problem of low estimation accuracy in the passive measurement method by adopting a histogram analysis on packet pair dispersion [3], [8], [11]. However, they are also unsuitable for the constant monitoring of the capacity of every path in a large-scale network since the computational cost of their histogram analysis for all paths in the network is unacceptably high.

In this paper we introduce a novel accurate method of path capacity estimation by passive measurement which suits for the constant monitoring of the capacity of every path in a large-scale network 24-hour, 365-day. Our method, which is based

[3]Available bandwidth is the maximum rate at which a new connection can send without congestion collapse.

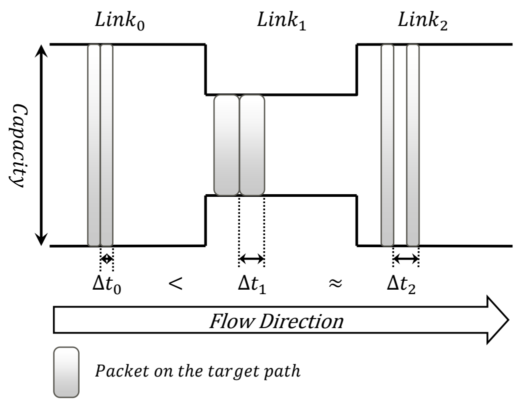


Fig. 1: Packet-pair technique commonly used in existing methods

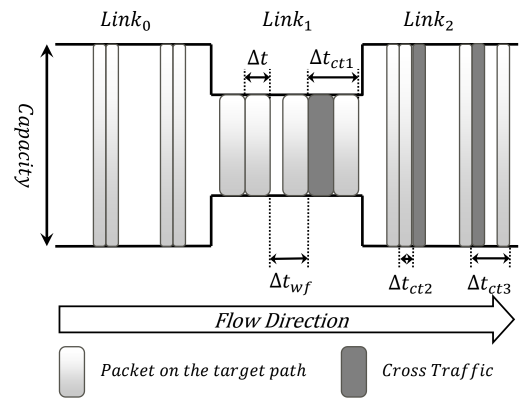


Fig. 2: Two cases that unsatisfy the two assumptions of the packet-pair technique.

on the mechanism of window flow control and the distribution characteristic of packet-pair dispersion caused by cross traffic, can estimate the capacity of every path in the network with an acceptable computational cost.

The remainder of the paper is organized as follows. Section II provides background knowledge. Section III presents our accurate method of path capacity estimation by passive measurement. We evaluate the accuracy of our method on our company's backbone network in Section IV. Related works are summarized in Section V, and Section VI concludes the paper.

II. BACKGROUND KNOWLEDGE

In this section, we first explain the packet-pair technique, which has been used to estimate the capacity of a path by passive measurement method, and two factors impact its estimation accuracy in Section II-A. Our method is based on this technique. We next explain TCP window flow control in Section II-B, which is used in our Filter 1 shown later in Section III-A. we finally explain the distribution characteristic of the packet-pair dispersion influenced by cross traffic in Section II-C, which is used in our Filter 2 shown later in Section III-B.

A. Packet-Pair Technique and Factors Impacting its Estimation Accuracy

The packet-pair technique estimates the capacity of a path from the interval between a packet and its subsequent packet, and the size of the latter packet. Assuming that there is no cross traffic and the two packets are sent close enough in time so that they are queued together at the narrowest link in a path, the packet-pair technique calculates the capacity of the path (C) as:

$$C = \frac{S}{\Delta t}, \tag{1}$$

where S is the packet size of the latter packet, and Δt is the interval between the two packets at the destination end (see Fig. 1). In Fig. 1, the path consists of three links ($Link_0$, $Link_1$ and $Link_2$). The width of each link indicates the capacity of the link (C_0, C_1 and C_2). In this example, the narrowest link in the path is $Link_1$, and the capacity of the path is C_1. When the two packets are sent to the narrowest link from the wide link ($Link_0$), the interval between the two packets (Δt_1) increases due to the transmission delay in the narrowest link. When the two packets are sent to the wide link ($Link_2$) from the narrowest link, the interval between the two packets keeps the same amount ($\Delta t_2 = \Delta t_1$). Thus the interval between the two packets can be formulated as:

$$\Delta t_i = \max\left(\Delta t_{i-1}, \frac{S}{C_i}\right). \tag{2}$$

In the packet-pair technique, there are two factors that degrade the accuracy of path capacity because the packet-pair technique makes two assumptions which may not be held in practice. When the assumptions are unsatisfied, the estimation accuracy of path capacity is degraded significantly [2]. The first assumption is that the two packets are transmitted consecutively so that the packets are queued together at the narrowest link. The second assumption is that there is no cross traffic on the path. Fig. 2 shows the cases where the assumptions are unsatisfied.

- Factor 1: When the first assumption that the two packets are transmitted consecutively is unsatisfied, the interval between a packet and its subsequent packet becomes larger than the transmission delay of the narrowest link ($\Delta t_{wf} > \Delta t$), which results in an underestimation of the capacity of the path. For example, in the case that the capacity of the narrowest link is 100Mbps and the packet size is 1,500 bytes, the two packets need to be transmitted less than 120 microseconds apart.
- Factor 2: When the second assumption that there is no cross traffic is unsatisfied, the interval between the two packets changes. When there is cross traffic between the two packets at the narrowest link, the second packet experiences additional queueing delays by the cross traffic, and the interval between the two packets increases ($\Delta t_{ct1} > \Delta t$). In this case, it results in an underestimation of the capacity of the path. When the first packet of the two packets experience additional queueing delays due to cross traffic in the links after the narrowest link, the interval between the two packets decreases ($\Delta t_{ct2} < \Delta t$). This case results in an overestimation of the capacity of the path. When the latter packet of the two packets experiences additional queueing delays due to cross traffic in the

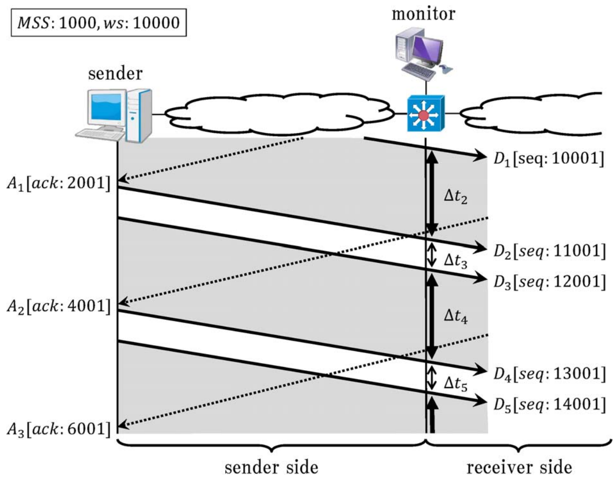


Fig. 3: Mechanism of TCP window flow control

links after the narrowest link, the interval between the two packets increases ($\Delta t_{ct3} > \Delta t$). This case results in an underestimation of the capacity of the path.

B. TCP Window Flow Control

In this subsection, we explain TCP window flow control, which our Filter 1 (see Section III-A) is based on to eliminate the influence of Factor 1 described in Section II-A.

TCP window flow control is the process to control the transmission rate of a TCP communication. TCP window flow control uses a sequence number (seq), an acknowledgment number (ack) and a window size (ws). seq identifies the order of the packet transmitted from the sender node in order to reconstruct the data at the receiver node. ack notifies the sender node about the order of the packet that the receiver node received correctly. ws decides the number of the packets that the sender node can transmit without waiting for an ACK packet. The sender node can transmit packets until a window becomes full ($seq \leq ack + ws$) without waiting for an ACK packet. TCP window flow control achieves high throughput and avoids congestion collapse by controlling the amount of ws. In Fig. 3, A_i are ACK packets and D_i are data packets. Δt_i are the intervals between D_i and D_{i-1}. Maximum Segment Size (MSS) limits the payload size of TCP packets. In the case of Fig. 3, MSS is 1,000 and ws is 10,000. When the sender node receives $A_1[ack : 2001]$, the sender node can transmit two data packets: $D_2[seq : 11001]$ and $D_3[seq : 12001]$. Also, when the sender node receives $A_2[ack : 4001]$, then, the sender node can transmit two data packets: $D_2[seq : 13001]$ and $D_3[seq : 14001]$.

Without the influence of cross traffic, the packets are classified into two groups for the purpose of estimating path capacity: The first group is the packets transmitted consecutively, which interval depends on the the capacity of the path. Meanwhile, the second group is the packets transmitted after waiting for an ACK packet, which interval does not depend on the capacity of the path. In the case of Fig 3, D_3 and D_5 are transmitted consecutively without waiting for an ACK packet, whereas Δt_3 and Δt_5 depend on the capacity of the path. On the other hand, D_2 and D_4 are transmitted after waiting for an ACK packet. Thus, Δt_2 and Δt_4 are larger than the

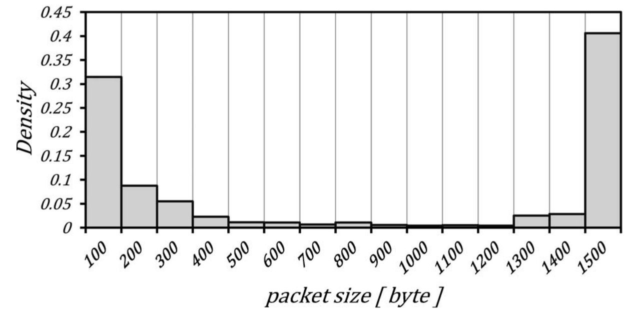


Fig. 4: Distribution of the size of packets observed in our in-company backbone network

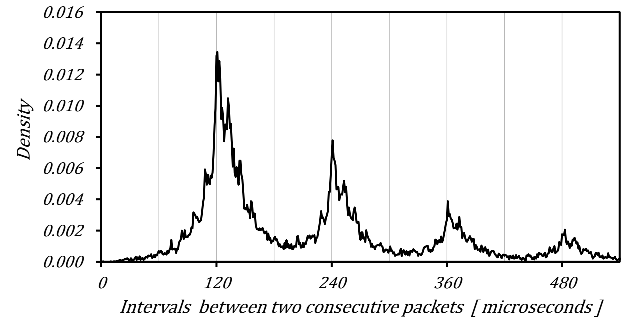


Fig. 5: Distribution of the intervals between two consecutively transmitted packets under the influence of cross traffic

transmission delay at the narrowest link. These intervals do not depend on the capacity of the path.

In order to accurately estimate path capacity, it is necessary to eliminate the packets transmitted after waiting for an ACK packet. If we can estimate ws correctly, we can distinguish the data packets transmitted after waiting for an ACK packet. However, it is difficult to estimate ws correctly because ws is dynamically changed in the sender node by various control methods (e.g., Reno, Vegas, and so on) and the information of ws and its control method can not be obtained from the header of TCP packets.

C. Distribution of Intervals under the Influence of Cross Traffic

In this subsection, we show how cross traffic impacts the distribution of the interval between two consecutive packets. Our Filter 2 (see Section III-B) uses the characteristics of the distribution of the intervals to eliminate the influence of Factor 2 described in Section II-A.

We investigated the distribution of TCP packet size in our in-company backbone network as shown in Fig. 4. 1,500-byte packets occupied more than 40% of the total packets. 1,500 bytes is the MTU of Ethernet frame. Packets of less than 100 bytes account for more than 30% of the total packets. These packets were ACK packets. The two kind of packets above account for more than 70% of the total packets. The ACK packets have a low impact on intervals because their transmission delay are low. Accordingly, the 1,500-byte packets have a dominant impact on intervals between two consecutive packets.

Next, we investigated the distribution of intervals between two consecutive packets under the influence of cross traffic as

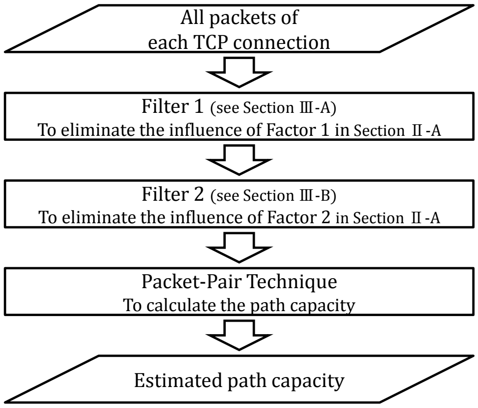


Fig. 6: Block diagram of our path capacity estimation method

shown in Fig. 5. In the path, the capacity of the narrowest link was 100 Mbps and the capacity of the other links were 1 Gbps. The transmission delay of a 1,500-byte packet on a 100 Mbps link is 120 microseconds. When packets of cross traffic intervene between two packets at the narrowest link, the interval becomes the integral multiple of the transmission delay at the narrowest link, such as 240, 360 and 480 microseconds. When there is some cross traffic on the links except the narrowest link, the mode of intervals is changed to 12 microseconds intervals, based on the queuing model $M_S^X/M/1$ [14]. 12 microseconds is the transmission delay of a 1,500-byte packet transmitted on 1 Gbps link.

To summarize, there are two characteristics of the distribution of intervals between two packets transmitted consecutively under the influence of cross traffic. Characteristic 1 is that the intervals increase to an integral multiple of the transmission delay at the narrowest link when there is cross traffic between two packets at the narrowest link. Characteristic 2 is that the intervals distribute based on $M_S^X/M/1$ when there is some cross traffic on the links except the narrowest link.

III. Path Capacity Estimation Method

This section describes our path capacity estimation method by passive measurement. Our estimation method consists of three steps: Filter 1, Filter 2 and the packet-pair technique as described in Fig. 6. The two filters are shown in Section III-A and Section III-B, respectively. After eliminating unnecessary packets by the two filters, the packet-pair technique is applied to derive the capacity of the target path. Note, our estimation method estimates the path capacity of only the sender side from the monitoring point in the network.

A. Filter 1: To Eliminate the Influence of Factor 1

To eliminate the influence of Factor 1, we eliminate the packets that are transmitted after waiting for an ACK packet, based on the mechanism of TCP window flow control. Filter 1 consists of two steps: first clustering step and then comparing step. The clustering step classifies the all captured packets in a connection into a group of the packets transmitted after waiting for an ACK packet and groups of the packets transmitted consecutively without waiting for an ACK packet. The comparing step distinguishes the group of packets transmitted after waiting for an ACK packet from the others to eliminate the packets in the group.

In the first clustering step, to gather packets transmitted after waiting for an ACK packet into one group, all captured packets in a connection are partitioned into N groups in order of arrival. A group consists of only the packets transmitted after waiting for an ACK packet. This group is referred to as *invalid group* in this section. Other $N-1$ groups consist of only the packets transmitted consecutively. N is determined by the number of packets transmitted consecutively without waiting for an ACK packet. The number of packets transmitted consecutively can be estimated by MSS and the increment of ack, as follows:

$$N = floor\left(\frac{A_i - A_{i-1}}{MSS}\right). \quad (3)$$

In Fig. 3, the amount of increase in ack equals 2,000 ($A_i - A_{i-1} = 2,000$) and MSS equals 1,000, then N becomes 2. When N remains constant, the packet transmitted after waiting for an ACK packet is sent every $N-1$ packets. Therefore, Filter 1 clusters all packets into N groups based on the order of capturing. The clustering rule is defined as:

$$g = G(i) = \mod(i, N), \quad (4)$$

where g and $G(i)$ are group number ($g = 0, ..., N-1$). In the case of Fig. 3, Group 0 then consists of D_2 and D_4, and Group 1 consists of D_3 and D_5.

In the second comparing step, we identify the *invalid group* from N groups clustered at the clustering step, and eliminates the packets in the *invalid group*. We first compare the path capacity calculated by the packets in each group. The capacity calculated by the packets of the *invalid group* is smaller than the others, because the packets of the *invalid group* are transmitted after waiting for an ACK packet, and the intervals of the *invalid group* are wider than those of other groups which consist of packets transmitted consecutively without waiting for an ACK packet. Therefore, the *invalid group* (g_{wf}) is derived as Equation 5.

$$g_{wf} = \arg\min_g C^g, \quad (5)$$

where C^g is the capacity calculated by the packets in the Group g. To calculate path capacity by using packet-pair technique, the sum total of packet size and the sum total of each interval between two packets are calculated for each group whenever a new packet is captured.

$$\begin{aligned} Size^{G(i)} &\mathrel{+}= S_i \\ Gap^{G(i)} &\mathrel{+}= \Delta t_i, \end{aligned} \quad (6)$$

where $Size^g$ is the sum total of packet size in group g, and Gap^g is the sum total of intervals in group g. Using Equation 1 and Equation 6, we can obtain the *invalid group* number in Equation 5 as:

$$g_{wf} = \arg\min_g \left(\frac{Size^g}{Gap^g}\right). \quad (7)$$

In the case of the example in Fig. 3, Group 0 is derived as the *invalid group* ($g_{wf} = 0$).

B. Filter 2: To Eliminate the Influence of Factor 2

To eliminate the influence of cross traffic, we eliminate the influence of cross traffic in the narrowest link using by Characteristic 1 described in Section II-C, and eliminate the influence of cross traffic in the links except the narrowest link using Characteristic 2 described in Section II-C. Our method can accurately estimate the capacity of every path by the packets that conformed to the two assumptions of the packet-pair technique.

First, to eliminate the influence of cross traffic in the narrowest link, the capacity calculated from the latest two packet is compared with the capacity calculated from all packets of the group. According to the Characteristic 1, when the captured packet experiences additional queueing delays by cross traffic in the narrowest link, the calculated capacity becomes the half of the actual capacity of the path or less. Therefore, if the calculated capacity is larger than the constant rate [4]of the estimated capacity of the group, the sum total of packet size and the sum total of each interval between two consecutive packets are calculated for each group as Equation 9 and 9, respectively.

$$\frac{S_i}{\Delta t_i} > c \times \frac{SizeCap^{G(i)}}{GapCap^{G(i)}} \tag{8}$$

$$\begin{aligned} SizeCap^{G(i)} &\mathrel{+}= S_i \\ GapCap^{G(i)} &\mathrel{+}= \Delta t_i. \end{aligned} \tag{9}$$

Second, to eliminate the influence of cross traffic in the links except the narrowest link, our method uses the central limit theorem. According to the Characteristic 2, each interval between two consecutive packets is distributed based on $M_S^X/M/1$. When the number of sample packets is enough large, $M_S^X/M/1$ approximates to a Gaussian distribution. Therefore, the influence can be decreased by increasing the number of sample packets [5].

Using the packet-pair technique (Equation 1), the estimated capacity (C_{est}) is derived as Equation 10.

$$C_{est} = \frac{\sum\limits_{g=0, g \neq g_{wf}}^{N} SizeCap^g}{\sum\limits_{g=0, g \neq g_{wf}}^{N} GapCap^g}. \tag{10}$$

IV. Evaluation

In this section, we evaluate our method using packets captured in our in-house backbone network. Each of network path at the sender side (the paths between sender nodes and the monitoring point) consists of 5 to 15 hops, and RTTs of each path were between 1 to 300 ms, and the average RTT was 10 ms. We selected 211 paths which we knew each of their path capacity is either 100 Mbps (174 paths) or 10 Mbps (37 paths). The number of sampled TCP connections was over 1,400,000 in the captured data during 3 hours.

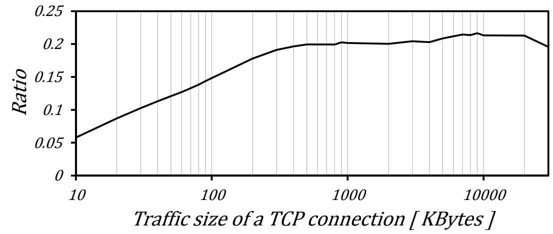
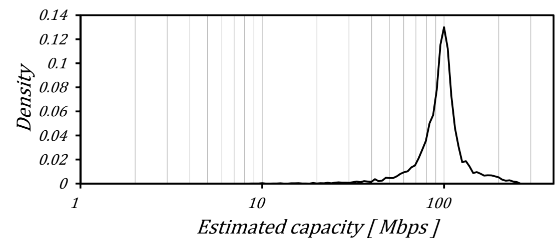
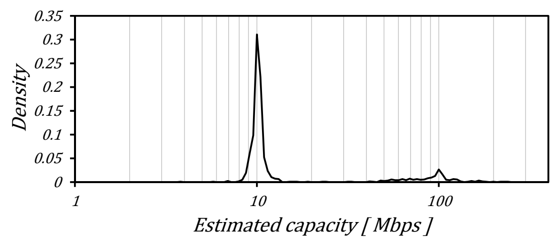


Fig. 7: The ratio between the number of TCP connections which could estimate the path capacity and the number of all TCP connections, aggregated by the amount of traffic



(a) The estimated capacity of every connection which actual value is 100 Mbps



(b) The estimated capacity of every connection which actual value is 10 Mbps

Fig. 8: Distributions of the estimated capacity of every connection which two different actual value

First, we evaluated the necessary amount of traffic to estimate the capacity of each path. Fig. 7 shows the ratio of TCP connections for which our method can estimate the path capacity to all TCP connections, by the traffic size of the connection. The ratio monotonically increases until traffic size reaches 500 KB. When traffic size exceeds 500 KB, the ratio stays relatively constant at approximately 20%[6]. The result shows our method requires more than 500 KB traffic on the path to estimate the capacity of a path. We consider this condition is reasonable since the average traffic of a HTTP access was 1.6 MB in Jan 2014 according to HTTP Archive [15].

Second, we evaluated the estimation accuracy of the capacity of each path. Fig. 8 shows the distributions of the estimated capacity of every connection at two different actual value. In Fig. 8(a) shows, in 70% of connections, their estimated capacities are within approximately 20% of their actual value (100 Mbps). In Fig. 8(b) shows, in 80% of connections their estimated capacities are within approximately 20% of their actual value (10 Mbps). Those results show the estimation accuracy of our method is sufficiently high for out propose which is the constant monitoring of the capacity of every

[4] Threshold $c = 0.75$ is adopted to cut-off the second and follow-on peaks in the distribution of intervals as shown in Fig. 5.

[5] In Section IV, to reduce the influence of cross traffic on the links except the narrowest link, the necessary amount of traffic (the sum total of packet size) to sufficient accurately estimate the path capacity is limited to 9000 bytes

[6] In our in-house backbone network, 20% is the guessed ratio of TCP connections which is appropriate for our proposed method. N is more than 1, other than for interactive communication and unstable communication traffic.

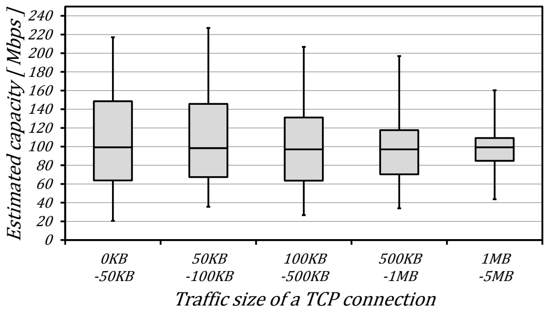


Fig. 9: Box plot of the estimated capacity of each path, aggregated by traffic size. The top line of the box is the 90^{th} percentile of the distribution, the bottom line of the box is the 10^{th} percentile of the distribution, and the central line of the box is the median of the distribution. In this case, the top end of the whiskers represents 99^{th} percentile of the distribution, and the bottom end of the whiskers represents the 1^{th} percentile of the distribution.

path in a network 24-hour, 365-day. The variance of estimated capacities in 100 Mbps cases is larger than that in 10 Mbps cases. We consider this wider variance in 100 Mbps cases is caused by the observation error of the time stamp of each packet. For example, when the observation error of each time stamp is 30 microseconds, and the actual value is 100 Mbps, the estimation error becomes 20%. On the other hand, when the actual value is 10 Mbps with the same observation erro of each time stamp, the estimation error becomes 0.26%. In Fig 8(b), There is another small peak at 100 Mbps although the actual value is 10 Mbps. We consider this small peak is caused by bursty cross traffic on the links except the narrowest link. How to remove these error cases is one of our future works.

Finally, we evaluated the impact of traffic size on estimation accuracy of our path estimation method. Fig. 9 shows the box plot of the estimated capacity of every connection, which actual value is 100 Mbps, by traffic size. As traffic size increases, the variance of the estimated capacity becomes smaller. With the traffic size of 1 MB to 5 MB, in 80% of connections, their estimated capacities are within approximately 15% of their actual value. This result shows that our method can achieve sufficiently high accuracy when it is applied for a connection which traffic size exceeds 1 MB.

V. Related Works

Active measurement methods designed for the Probe Rate Model (RPM). Pathload [6], IGI [4] and Pathchirp [12] use this model. This model is based on the concept of self-induced congestion, and it transmits packet-train, a sequence of consecutive probe packets, at varies rates. If the probing rate on the sender side is lower than the available bandwidth, the arrival rate of the packet-train at the receiver side will match the rate at the sender side. If the probing rate of the sender side is greater than or equal to the available bandwidth, the arrival rate at the receiver side will be lower than the sender side, because the packet train will congest the queues at the narrowest link. The estimated accuracy of the active method is high, and the active method can estimate not only path capacity but also available bandwidth. However, active measurement methods can have a significant impact during an estimation [13], and are not suitable for large scale networks. Active methods are suitable for rate control [5] and path selection in an overlay network [7] of multimedia streaming.

Passive methods designed for the Probe Gap Model (RGM). Nettimer [11], PPrate [3], MultiQ [8] and L. Kevin (2004) [9] use this model. The above methods are based on packet dispersion and using a histogram. Nettimer is the oldest passive tool, the estimate accuracy is low. MultiQ is the first passive method that discovers the capacity of multiple congested links along a path from a single flow trace, and this method frequently overestimates. PPrate uses algorithms similar to Pathrate [2] to passively discover capacities, The estimate accuracy of this method is high, and sometimes underestimates when working on the ack stream.

VI. Conclusion

We introduced a novel path capacity estimation method for passive measurement, which eliminates the negative influence of TCP window flow control and the negative influence caused by cross traffic on the path, and can estimate the capacity of each path with sufficient accuracy. Through the evaluations on our in-company backbone network, we showed our method could accurately estimate the capacity of the narrowest link in every 1524 connections, in which 80% of the connections were within approximately 15% of their actual values (in over 1MB traffic case). The computational cost of our estimation method is so reasonable that we can use the method for the constant monitoring of the capacity of every path in a large-scale network 24-hour, 365-day.

References

[1] Cisco Visual Networking Index, "The Zettabyte Era" Technical report, Cisco, 2012.

[2] C. Dovrolis, P. Ramanathan, and D. Moore, "Packet-dispersion techniques and a capacity-estimation methodology" IEEE/ACM Trans, Netword, 12(6):963-977, 2004.

[3] En-Najjary, T and Urvoy-Keller, G, "PPrate: A passive capacity estimation tool" In IEEE E2EMON, 82-89. 2006.

[4] N. Hu and P. Steenkiste, "Evaluation and Characterization of Available Bandwidth Probing Techniques" IEEE Journal on Selected Areas in Communications, 21(6):879-894, 2003.

[5] D. Ikegami, H. Nakazato, K. Asatani, and H. Tominaga, "Rate control on correlation coefficient of network delay" In GLOBECOM, 3118-3122, 2003.

[6] M. Jain and C. Dovrolis, "Pathload: A Measurement Tool for End-to-End Available Bandwidth" In PAM, 14-25, 2002.

[7] M. Jain and C. Dovrolis, "Path Selection Using Available Bandwidth Estimation in Overlay-Based Video Streaming" In Networking, 628-639, 2007.

[8] S. Katti and e. a. Katabi, D, "MultiQ: Automated Detection of Multiple Bottleneck Capacities along a Path" In SIGCOMM, 245-250, 2004.

[9] L. Kevin and B. Mary, "Measuring Link Bandwidths Using a Deterministic Model of Packet Delay" In SIGCOMM, 30(4):283-294, 2000.

[10] S. Krishnan and K. Sitaraman, "Video Stream Quality Impacts Viewer Behavior: Inferring Causality Using Quasi-Experimental Designs" In ACM Internet measurement conference, 211-224, 2012.

[11] K. Lai and M. Baker, "Nettimer: A Tool for Measuring Bottleneck Link, Bandwidth" In Proceedings of the 3rd Conference on USENIX Symposium on Internet Technologies and Systems, 3(01), 11-11, 2001.

[12] V. J. Ribeiro and e. a. Rudolf H. Riedi, "pathChirp: Efficient Available Bandwidth Estimation for Network Paths" In PAM, 2003.

[13] A. Shriram and J. Kaur, "Empirical Evaluation of Techniques for Measuring Available Bandwidth" In INFOCOM. 2162-2170, 2007.

[14] M. Garetto and D. Towsley, "Modeling, simulation and measurements of queuing delay under long-tail internet traffic" In ACM SIGMETRICS. 47-57. 2003.

[15] HTTP Archive, http://httparchive.org/.