Available Bandwidth Estimation from Passive TCP Measurements using the Probe Gap Model

Sukhpreet Kaur Khangura and Markus Fidler
Institute of Communications Technology
Leibniz Universität Hannover

***Abstract*—The Internet relies on congestion control protocols and adaptive applications that adjust their data rate to achieve good performance while avoiding network congestion. An essential prerequisite is the estimation of available network resources: implicitly like prevailing TCP versions that adapt their data rate iteratively; or explicitly by available bandwidth estimation techniques, as recently also adopted by TCP HyStart. Using observations of TCP throughput, applications like MPEG-DASH adapt the video quality and data rate. We discover, however, relevant conditions where TCP throughput is not a good bandwidth estimator and observe that it is outperformed by known UDP-based active probing methods. We investigate how the theory of active probing can possibly be used to extract relevant information also from passive TCP measurements. In case of TCP, the additional difficulty is found to be due to its chaotic traffic characteristics. We define a criterion to select relevant traffic samples and apply a regression technique to estimate the available bandwidth. Noteworthy, using the feedback provided by TCP acknowledgements, we can perform the estimation from sender-side measurements only. We verify the fidelity of the approach in a variety of experiments, including different types of cross-traffic, delays, and loss of data packets as well as acknowledgements.**

I. Introduction

The term *available bandwidth* denotes the residual capacity of a link that is left over by the existing traffic, in the following also referred to as *cross traffic*. The available bandwidth of a network path is determined by the *tight link*, that is the link that has the minimal available bandwidth along the path. The tight link may differ from the *bottleneck link*, that is the link with the minimal capacity. In available bandwidth estimation, a sender actively injects artificial *probe traffic* into the network. Using time-stamps of the probes, the receiver seeks to deduce the available bandwidth. To date a number of accepted active probing techniques and corresponding theories for available bandwidth estimation exist, e.g., [1]–[13].

A shortcoming is, however, the use of active probes, that are specifically tailored to the estimation method. Probes are typically sent either as *packet pairs* [14] with a defined spacing referred to as gap g_{in}, as *packet trains* [3], [15] with a fixed rate r_{in}, or as *packet chirps* [6] that are packet trains with an increasing rate. In contrast, the vast majority of the Internet traffic are TCP flows that exhibit a rather chaotic traffic pattern and it is a common practice to use the throughput of a TCP connection as an estimator of the available bandwidth. A prominent example is MPEG Dynamic Adaptive Streaming over HTTP (DASH).

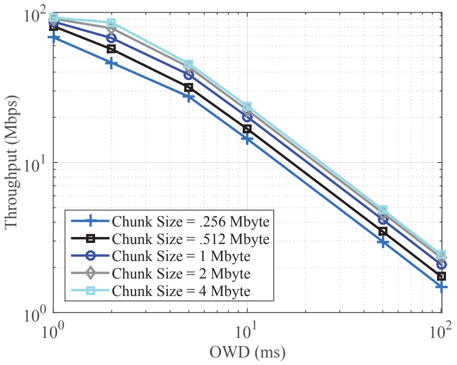


Fig. 1. Average TCP throughput. The available bandwidth of 100 Mbps is attained only if the OWD is small and the chunk size is large.

DASH is the core technology used by major Internet video providers such as YouTube and Netflix [16]. In DASH, the media content is divided into segments or chunks of a certain duration, typically in the order of a few seconds. The chunks are encoded using multiple profiles, i.e., each chunk is available in different quality levels corresponding to different bandwidth requirements. The encoded chunks are stored on HTTP servers along with a manifest file which lists the available profiles of the chunks. The client downloads the chunks one by one using HTTP GET requests, where it seeks to select the profile that matches the available resources best.

While DASH does not specify the method how to measure the available bandwidth, a typical approach is to use the average throughput achieved by TCP during the transmission of previous chunks as an estimate of the available bandwidth, e.g., [16], [17]. Formally, given a chunk of N packets each with length l, the average throughput is computed as

$$r_{\text{out}} = \frac{(N-1)l}{\sum_{k=1}^{N-1} g_{\text{out}}^k}, \tag{1}$$

where the output gap is defined as $g_{\text{out}}^k = t_{\text{out}}^{k+1} - t_{\text{out}}^k$ and t_{out}^k is the time of reception of packet k by the receiver.

To support basic insights into the relation of TCP throughput and available bandwidth, we evaluate the average throughput of TCP in controlled network experiments. Compared to [5],

This work was supported by an ERC Starting Grant (UnIQue, StG 306644).

ISBN 978-3-901882-94-4 © 2017 IFIP

[18], we use TCP CUBIC to transfer chunks of limited size. CUBIC is a high-speed TCP variant aimed at saturating networks with a large bandwidth-delay-product. In congestion avoidance, it uses a cubic function to increase the congestion window (CWND) independent of the round-trip-time [19].

Fig. 1 shows the throughput that is achieved by TCP when transmitting chunks of a fixed size in the range from 256 kbyte to 4 Mbyte. The network offers an available bandwidth of 100 Mbps and the experiments are conducted for a range of one-way-delays (OWD) from 1 to 100 ms. Further details on the network are deferred to Sec. III. We notice that TCP congestion control limits the throughput significantly below the available bandwidth for two reasons: first, the TCP transmission starts in slow start with a small CWND and, even despite the CWND is increased quickly, this initial phase has a considerable effect on the average throughput if chunks are small in size; secondly, when the OWD is non-negligible, the CWND may never reach the bandwidth-delay-product so that the actual throughput during the transmission of a finite-sized chunk generally remains below the available bandwidth.

The above limitations have motivated us to investigate in-depth: "Do observations of short TCP flows provide sufficient information to infer the available bandwidth?" And if so, "How can the available bandwidth be estimated, e.g., can techniques from active probing be adapted to passive TCP measurements?" On the way to find the answers to these questions, we make the following contributions:

- We identify the chaotic, non-packet train pattern of TCP flows and define a criterion to select traffic samples that bear relevant information. This information is encoded in the form of packet gaps that are explained by the probe gap model known in bandwidth estimation.
- We use a regression technique to obtain robust bandwidth estimates from passive measurements of these gaps. The accuracy of the method is evaluated for a variety of relevant parameter settings.
- We propose a method that can take multiple gaps as well as acknowledgement gaps as input. This extension enables bandwidth estimation using only sender-side measurements of TCP data and acknowledgement packets.

The goal of this work is to understand the information that short-lived TCP flows provide on the available bandwidth. The results may benefit adaptive applications like DASH and may contribute to new TCP versions, such as the recent Hybrid Start (HyStart) algorithm [20] that draws from capacity estimation techniques [21]. We will discuss HyStart in more detail in Sec. II.

The remainder of this paper is structured as follows. In Sec. II, we discuss the related work on available bandwidth estimation. We describe our experimental setup in Sec. III. In Sec. IV, we introduce a method to estimate the available bandwidth from passive measurements. We apply the method to short-lived TCP flows and evaluate the accuracy for a variety of relevant parameters in Sec. V. We also extend the method to use sender-side measurements of acknowledgement gaps. We provide brief conclusions in Sec. VI.

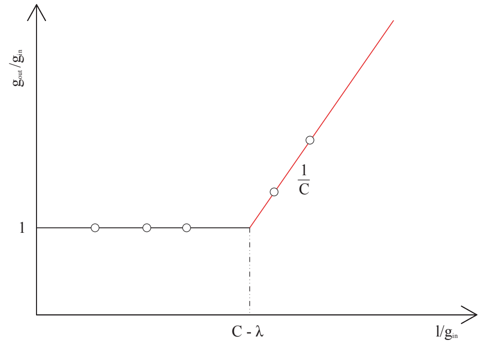


Fig. 2. Gap response curve. The turning point marks the available bandwidth.

II. State-of-the-Art in Bandwidth Estimation

In this section, we introduce the basic probe rate, respectively, probe gap model of a First-In First-Out (FIFO) multiplexer that is used in bandwidth estimation to derive its characteristic response curve. Further, we discuss state-of-the-art bandwidth estimation methods. We start with a definition of available bandwidth. Given a link with capacity C and cross traffic with long-term average rate λ, where $\lambda \in [0, C]$, the available bandwidth A is defined as the residual capacity that is left over after cross traffic is served, i.e., $A = C - \lambda$ [10].

The available bandwidth of a network path is defined to equal the available bandwidth of the tight link, that is the link that has the minimal available bandwidth along the path [11]. This leads to the abstraction of a network path as a single tight link, that is applicable in the long-term average [13].

A common assumption in bandwidth estimation is that cross traffic has a constant rate λ and behaves like fluid, i.e., it is infinitely divisible. FIFO multiplexing of probe traffic with input rate r_{in} leads to a rate-proportional capacity sharing and the output rate of the probe traffic is determined as

$$r_{\text{out}} = \frac{r_{\text{in}}}{r_{\text{in}} + \lambda} C, \tag{2}$$

if $r_{\text{in}} + \lambda > C$ and $r_{\text{out}} = r_{\text{in}}$ otherwise [1]. After some reordering the so-called *rate response curve*

$$\frac{r_{\text{in}}}{r_{\text{out}}} = \max\left(\frac{r_{\text{in}} + \lambda}{C}\right) = \begin{cases} 1 & \text{if } r_{\text{in}} \leq C - \lambda, \\ \frac{r_{\text{in}} + \lambda}{C} & \text{if } r_{\text{in}} > C - \lambda, \end{cases} \tag{3}$$

is obtained. The utility of Eq. (3) is due to the clear bend at $r_{\text{in}} = C - \lambda$ that identifies the available bandwidth.

An equivalent representation as *gap response curve* is obtained by insertion of $r_{\text{in}} = l/g_{\text{in}}$ and $r_{\text{out}} = l/g_{\text{out}}$ into Eq. (3), where l is the constant packet size of the probe traffic and the gaps g_{in} and g_{out} denote the time difference between probe packets that are input and output, respectively. The resulting gap response curve

$$\frac{g_{\text{out}}}{g_{\text{in}}} = \begin{cases} 1 & \text{if } \frac{l}{g_{\text{in}}} \leq C - \lambda, \\ \frac{l}{g_{\text{in}} C} + \frac{\lambda}{C} & \text{if } \frac{l}{g_{\text{in}}} > C - \lambda, \end{cases} \tag{4}$$

has the same characteristic bend, for illustration see Fig. 2.

Active techniques for estimation of the available bandwidth can be classified to be either iterative or direct. We use the rate response curve to illustrate the difference. The same conclusions can be made if the gap response curve is used.

Iterative probing techniques basically search for the turning point of the rate response curve by sending repeated probes at increasing rates, as long as $r_{\rm in}/r_{\rm out} = 1$. When $r_{\rm in}$ reaches $C - \lambda$, the available bandwidth is saturated and increasing the probe rate further results in $r_{\rm in}/r_{\rm out} > 1$. As a consequence, a queue builds up at the multiplexer. This causes increasing OWDs that can be detected by the receiver. The procedure is implemented, e.g., by Pathload [4] and Pathchirp [6].

Direct probing techniques estimate the upward line segment of the rate response curve for $r_{\rm in} > C - \lambda$. The line is determined by C and λ. If C is known, a single probe $r_{\rm in} = C$ yields a measurement of $r_{\rm out}$ that is sufficient to estimate $\lambda = C(C/r_{\rm out} - 1)$ from Eq. (3). Spruce [7] implements this approach. If C is also unknown, a minimum of two different probing rates $r_{\rm in} > C - \lambda$ is sufficient to estimate the two unknown parameters of the line. This approach is taken, e.g., by TOPP [1], DietTOPP [2], and BART [9].

In practice, methods for bandwidth estimation have to deal with noisy measurement data, e.g., due to inaccurate time-stamping. Further, significant deviations are due to random cross traffic that is not considered by the fluid flow model. To deal with the randomness, different post-processing techniques are used. A typical approach is to repeat measurements several times to compute average values, as done by Pathchirp [6] and Spruce [7], or to perform a majority decision, as in case of Pathload [4] that also reports an undecided bandwidth region. BART [9] uses a Kalman filter to estimate the available bandwidth from repeated measurements and to track changes of the available bandwidth online. TOPP [1] and DietTOPP [2] use linear regression to determine the parameters of the upward line segment of the rate response curve using several probes with rates in an interval $[r_{\rm in}^{\rm min}, r_{\rm in}^{\rm max}]$.

To determine a minimal probing rate $r_{\rm in}^{\rm min}$ that is larger than $C - \lambda$, DietTOPP performs an initial phase where probes are sent back-to-back at the maximal possible rate $r_{\rm in}^{\rm max}$ to measure the corresponding output rate $r_{\rm out}^{\rm max}$. Given that $r_{\rm in}^{\rm max} \geq C - \lambda$, it follows from Eq. (2) that $r_{\rm out}^{\rm max} \geq C - \lambda$, too, so that $r_{\rm in}^{\rm min}$ can be safely chosen as $r_{\rm in}^{\rm min} = r_{\rm out}^{\rm max}$.

A detailed analysis of the impact of random cross traffic on the properties of rate and gap response curves is provided by [10], [11]. The main finding is an elastic deviation from the fluid flow model that may cause biased estimates. The bias is significant in the middle part of the probing range around $r_{\rm in} = C - \lambda$, where it blurs the characteristic bend of the curve. For intuition, the bend of the curve at $C - \lambda$ can be thought of as fluctuating along the x-axis if the intensity of cross traffic λ is random. Further, [10], [11] connect the concept of rate and gap response curves with known bandwidth estimation tools.

Typically, bandwidth estimation tools use active probing where probes are sent as packet pairs, i.e., $N = 2$ packets sent with a defined gap $g_{\rm in}$, packet trains, i.e., a larger number of packets $N > 2$ sent at a defined rate $r_{\rm in}$, or packet chirps that are trains with an increasing rate. Correspondingly, either the gap response curve or the rate response curve applies. Compared to a packet pair, a packet train is generally less susceptible to random fluctuations as the computation of $r_{\rm out}$ from packet time-stamps by means of Eq. (1) averages over several $g_{\rm out}$ at the expense of a larger number of probe packets. Spruce [7] and IGI [8] are examples that use packet pairs. TOPP [1], DietTOPP [2], Pathload [4], PTR [8], and BART [9] use packet trains, and Pathchirp [6] uses packet chirps.

The advantage of active probing is that it enables obtaining specific points of the rate or gap response curve selectively. The injection of probe traffic contributes, however, to the load of the network. Hence, it is favorable if the available bandwidth can also be estimated from passive measurements of existing network traffic. The additional difficulty of passive measurements is that the input rate cannot be controlled so that it is hard to extract the desired information [12].

Recently, passive techniques have been developed that make use of the fact that a TCP sender in slow start likely transmits the packets of entire CWNDs in a row [20], [22], i.e., as a packet train. In this case packet dispersion techniques from [21] apply to each CWND. A prominent example is TCP HyStart [20] that seeks to estimate the available bandwidth to find a safe exit point from slow start before packets are lost. In essence, HyStart uses $r_{\rm out}$ obtained from Eq. (1), where CWND is substituted for N, as an estimate of the available bandwidth. Further, HyStart measures acknowledgements where $g_{\rm ack}^k$ is the acknowledgement gap that takes the place of the output gap $g_{\rm out}^k$.

HyStart provides, however, only a rough estimate of the available bandwidth since the output rate of a packet train $r_{\rm out}$ defined by Eq. (1) is known to provide a lower bound of the bottleneck capacity and an upper bound of the available bandwidth of a network path [21], i.e., $C - \lambda \leq r_{\rm out} \leq C$. This is also verified by Eq. (2) using $r_{\rm in} \geq C - \lambda$. Moreover, acknowledgement-based bandwidth estimation techniques have been found to be challenging due to the interaction of acknowledgements with cross traffic [23] that may lead to acknowledgement compression [24]. Compared to HyStart, we investigate methods that can report the exact available bandwidth instead of a bound. Further, we do not rely on the assumption that TCP transmits packet trains when in slow start, and use the entire traffic of a TCP connection as input.

III. Experimental Setup

Before we investigate how to estimate the available bandwidth from passive TCP measurements, we provide a brief overview of our testbed network that is used to obtain the experimental results presented in this paper. The experiments are conducted in a controlled network at Leibniz Universität Hannover that is managed by the Emulab software [25].

We use a dumbbell topology with a single bottleneck link as shown in Fig. 3 and emulate the transmission of DASH video chunks in the presence of downstream as well as upstream cross traffic. The downstream cross traffic is used to load the bottleneck link so that a defined amount of bandwidth remains available, whereas uplink cross traffic interferes with TCP acknowledgements and alters their spacing.

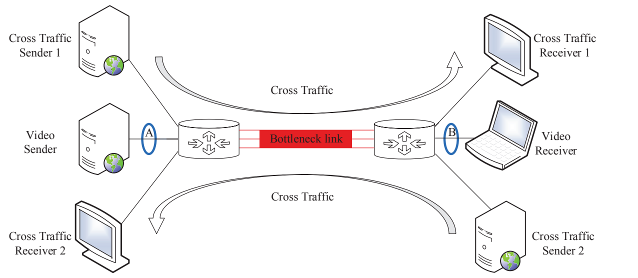


Fig. 3. Dumbbell topology set up in Emulab. The bottleneck link has a capacity of 100 Mbps and a configurable delay and loss rate. Cross traffic in downstream and upstream direction is used to load the bottleneck link to enable controlled bandwidth estimation experiments.

For our experimental purposes, we consider the animated movie named "Sintel" by Blender Foundation. We downloaded the movie from YouTube and obtained the chunk statistics. The modal value of the chunks of about 1 Mbyte is chosen as a reference. To transmit chunks of data of a defined size via TCP and UDP, we use the traffic generators iPerf [26] and RUDE & CRUDE [27], respectively. Cross traffic of different types and intensities is generated using D-ITG [28].

Since the PCs in our Emulab testbed are connected via physical Ethernet links of 1 Gbps and 10 Gbps, respectively, we use the token bucket filter [29] to emulate a 100 Mbps bottleneck link. In addition, a delay node is used at the bottleneck to emulate a wide area link to investigate the effect of different OWDs on the bandwidth estimation. The delay node can also be configured to create packet loss with a defined probability in downstream and upstream direction. The access links are configured to have 100 Mbps capacity, too. We note that the emulation has limited accuracy and hence contributes additional noise to the measurements.

We disable the segmentation offloading by the network interface card using ethtool [30]. Hence, the TCP/IP stack is responsible for segmenting chunks into datagrams of 1500 byte size, that is the maximum transmission unit carried by the Ethernet links. Including Ethernet header and trailer, the packet size is 1514 byte. Packet time-stamps at the video sender and receiver are generated at points A and B, respectively, using libpcap at the hosts. We also use a specific endace DAG measurement card to obtain accurate reference time-stamps.

IV. Estimation from Passive Measurements

As already discussed in Sec. I, TCP throughput is not generally a good estimator of the available bandwidth. Two reasons that we have investigated are non-negligible OWDs and short-lived TCP flows, as caused by small to medium chunk sizes, see Fig. 1. Now the following questions arise:

- Is it possible to estimate the available bandwidth in such scenarios where TCP throughput is limited?
- How can the required information be extracted from the rather chaotic traffic patterns of TCP?

Before we investigate the specifics of TCP traffic in Sec. V, we first develop a method for available bandwidth estimation from general passive measurements. We verify the method in controlled experiments.

A. Passive Estimation Method

We construct a method that is based on the probe gap model. It uses techniques from direct probing together with a threshold test to select relevant packet gaps. To motivate our design decisions, we start with a discussion of the different options that arise before we give details on the implementation.

Given passive measurements, we have to deal with non-structured traffic that cannot be assumed to take certain patterns, like CWND-sized packet trains as used by HyStart [20]. In order to be able to apply packet train models nevertheless, an option is to filter the sender-side measurement data for clusters of back-to-back packets that exceed a certain threshold. This approach is used in [22], where the specific requirement is that packet trains span several scheduling periods of a cellular network. A drawback of the approach is that a potentially large number of samples that do not pass the threshold test may be discarded. To avoid the dependence on any kind of traffic structure, we will work with individual packet gaps. Hence, the probe gap model applies.

Using the probe gap model, the task is to estimate the parameters of the gap response curve from passive measurements of g_{in} and g_{out}. The difficulty is due to the fact that we cannot assume evenly spaced g_{in} as achievable by active probing. For example, we may not have any samples close to the bend of the gap response curve at $l/g_{\text{in}} = C - \lambda$, see Fig. 2. Consequently, iterative techniques that search for the turning point may not apply, as relevant data may be missing. Further, in the presence of random cross traffic, it has been found that deviations blur the turning point, unless long packet trains are used [10], [11].

Techniques from direct probing, on the other hand, require that $g_{\text{in}} < l/(C-\lambda)$ where C and λ are unknown. Filtering out all $g_{\text{in}} \geq l/(C-\lambda)$ beforehand may seem to be an easy task, given that $g_{\text{out}} = g_{\text{in}}$ in this case, see Eq. (4). In practice, g_{out} may, however, be significantly distorted. Important reasons for this are: inaccurate time-stamping, the packet granularity and the randomness of non-fluid cross traffic, and the interaction with cross traffic on links other than the bottleneck link.

To determine a threshold $g_{\text{in}}^{\max}$ up to which g_{in} may safely be used by techniques from direct probing, we adapt a criterion from DietTOPP [2] to the probe gap model. We identify the minimal input gap denoted $g_{\text{in}}^{\min}$ in the passive measurement data and extract the corresponding output gap $g_{\text{out}}^{\min}$. It can be shown that $g_{\text{out}}^{\min} < l/(C-\lambda)$, so that we can use $g_{\text{in}}^{\max} = g_{\text{out}}^{\min}$ as a threshold to filter out all $g_{\text{in}} \geq g_{\text{in}}^{\max}$.

To verify that $g_{\text{out}}^{\min} < l/(C-\lambda)$, we use Eq. (2) to derive the corresponding gap representation

$$g_{\text{out}}^{\min} = \frac{g_{\text{in}}^{\min}\lambda + l}{C}, \tag{5}$$

where we assumed that $g_{\text{in}}^{\min} < l/(C-\lambda)$. The condition is satisfied if there exist samples g_{in} on the right, upward slope of the gap response curve. Otherwise, if there are no samples in this region, the measurement data does not provide sufficient

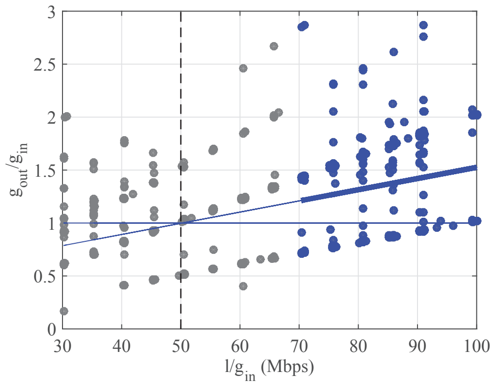


Fig. 4. $(g_{\rm in}, g_{\rm out})$ samples obtained by transmission of a chunk of 1 Mbyte via UDP. The sender varies $g_{\rm in}$ so that the samples are evenly distributed. Samples that are marked blue are used to determine the regression line. The intersection of the regression line with the horizontal line at 1 marks the available bandwidth of 50 Mbps.

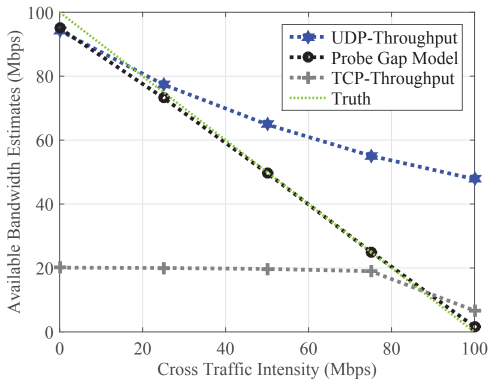


Fig. 5. Available bandwidth estimates for varying cross traffic rates. The estimates obtained from UDP $(g_{\rm in}, g_{\rm out})$ measurements closely match the ground truth. The TCP throughput underestimates the available bandwidth, mainly due to the OWD of 10 ms and the limited chunk size of 1 Mbyte. The UDP throughput overestimates the available bandwidth since a greedy UDP sender can preempt cross traffic at a FIFO multiplexer.

information to estimate the available bandwidth. The intuition behind Eq. (5) is that during $g_{\rm in}^{\rm min}$ an amount of fluid cross traffic of $g_{\rm in}^{\rm min}\lambda$ is accumulated that is transmitted in FIFO order between the two packets that span $g_{\rm in}^{\rm min}$. The condition $g_{\rm in}^{\rm min} < l/(C-\lambda)$ ensures that the FIFO multiplexer does not becomes idle during this interval. By insertion of $g_{\rm in}^{\rm min} < l/(C-\lambda)$ into Eq. (5), it follows that $g_{\rm out}^{\rm min} < l/(C-\lambda)$.

A lower bound of $g_{\rm out}^{\rm min}$ can be obtained from Eq. (5) if we let $g_{\rm in}^{\rm min} \to 0$. It follows that $g_{\rm out}^{\rm min}$ is bounded in the interval $l/C < g_{\rm out}^{\rm min} < l/(C-\lambda)$, i.e., using the threshold $g_{\rm in}^{\rm max} = g_{\rm out}^{\rm min}$ may filter out usable samples that satisfy $g_{\rm in} < l/(C-\lambda)$. We ignore these samples since they are close to the middle part of the gap response curve at $l/g_{\rm in} = C - \lambda$ that has been found to be biased if cross traffic is random [10], [11].

In practice, we cannot rely on a single sample $g_{\rm in}^{\rm min}$ to determine $g_{\rm out}^{\rm min}$. Instead, we consider a bin of the x smallest gaps $g_{\rm in}$ and compute the average of the corresponding $g_{\rm out}$ to obtain a robust estimate of $g_{\rm out}^{\rm min}$. In our experiments we configure x so that 10% of the gaps are used to estimate $g_{\rm out}^{\rm min}$.

Once we have selected samples that satisfy $g_{\rm in} < l/(C-\lambda)$, we can apply any technique from direct probing to estimate the available bandwidth. Here, we use linear regression to determine the upward segment of the gap response curve, as this method does not require any specific distribution of the $g_{\rm in}$ that are used. The available bandwidth estimate is determined from Eq. (4) as the x-axis intercept where the regression line intersects with the horizontal line at 1, see Fig. 2.

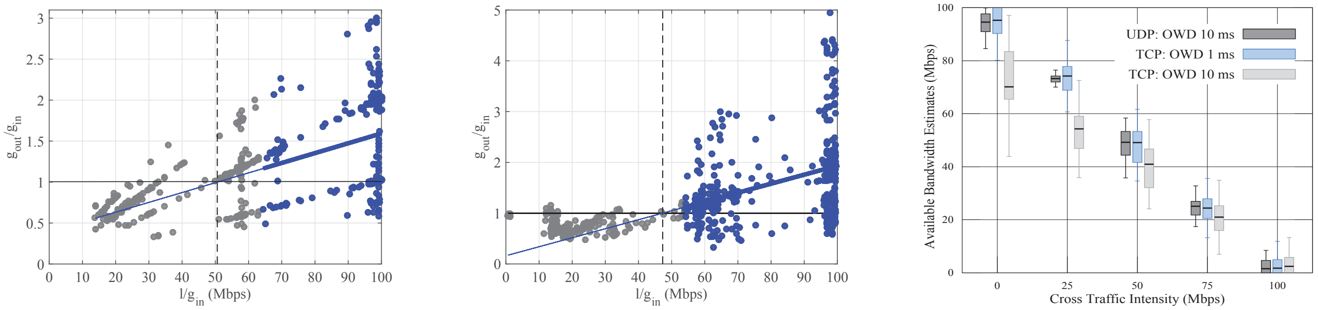
B. Experimental Verification

For a first experimental verification of the estimation method, we use $(g_{\rm in},\, g_{\rm out})$ samples that are evenly distributed over the range 30 Mbps $\le l/g_{\rm in} \le$ 100 Mbps. The samples are obtained in our experimental testbed using the tool RUDE & CRUDE that can emit UDP packets with a defined gap. The packet size is $l = 1514$ byte on the Ethernet and we transmit chunks of 1 Mbyte, corresponding to 660 packets. A set of $(g_{\rm in},\, g_{\rm out})$ samples obtained by transmission of one chunk is shown in Fig. 4. In the experiment, constant bit rate (CBR) cross traffic with rate $\lambda = 50$ Mbps is used.

The cross traffic deviates, however, from the fluid flow assumption as it uses packets of 1514 byte. The effects of the packet granularity become visible as a vertical spread of the $g_{\rm out}/g_{\rm in}$ points in Fig. 4. To illustrate an example, we consider $l/g_{\rm in} = 50$ Mbps that corresponds to $g_{\rm in} = 0.24$ ms. The transmission time of a packet at $C = 100$ Mbps is 0.12 ms, so that a cross traffic packet can fit exactly into the gap. This results in $g_{\rm out}/g_{\rm in} = 1$ as also predicted by the fluid model. Cross traffic packets can arrive, however, at arbitrary points in time and if a cross traffic packet arrives right before the first or second packet that constitute $g_{\rm in}$, it delays this packet by 0.12 ms so that $g_{\rm out}/g_{\rm in} = 0.5$ or 1.5, respectively.

The method for estimation of the available bandwidth from the samples proceeds in two steps. First, it estimates $g_{\rm in}^{\rm min}$ and the corresponding $g_{\rm out}^{\rm min}$ based on the 10% of the samples with the smallest $g_{\rm in}$. Using $g_{\rm in}^{\rm max} = g_{\rm out}^{\rm min}$ as a threshold for $g_{\rm in}$, only the samples with $l/g_{\rm in} > l/g_{\rm in}^{\rm max} = 70$ Mbps, that are marked blue in Fig. 4, are used in the second step to perform the linear regression. The regression line is shown as a thick blue line. Extending this line until it intersects with the horizontal line at 1, reveals an available bandwidth estimate of 50 Mbps. Further, it follows from Eq. (4) that the slope of the regression line provides an estimate of $1/C$. In Fig. 4, the slope is approximately 0.01 corresponding to $C = 100$ Mbps.

We repeated the above experiment with different cross traffic intensities of $\lambda \in \{0, 25, 20, 75, 100\}$ Mbps. For each case we conducted 100 repeated measurements. We report the median value in Fig. 5. The available bandwidth estimates closely match the ground truth that is marked in the figure by a green line. For comparison, we also include the throughput that is achieved by a TCP sender and a UDP sender, respectively, that transmit the same amount of data of 1 Mbyte. Clearly, the TCP throughput underestimates the available bandwidth, as soon as more than 20 Mbps are available. To understand



(a) OWD = 1 ms, λ = 50 Mbps

(b) OWD = 10 ms, λ = 50 Mbps

(c) Bandwidth Estimates

Fig. 6. $(g_{\text{in}}, g_{\text{out}})$ samples obtained from passive TCP measurements and bandwidth estimates obtained thereof. The values of g_{in} are a result of TCP congestion control and depend on network parameters such as the OWD. With increasing OWD a clustering of g_{in} samples is observed and the variability of bandwidth estimates increases.

this effect, we note that the testbed was configured to have a OWD of 10 ms. As a consequence, the TCP throughput is limited by the CWND. For further details see the discussion of Fig. 1. The UDP throughput, on the other hand, overestimates the available bandwidth. This is due to the fact that a greedy UDP sender can preempt the cross traffic at a FIFO multiplexer and monopolize the link. The effect is expressed by Eq. (2). Given UDP traffic is injected at line rate $r_{\text{in}} = C$, it achieves a throughput of $r_{\text{out}} = C^2/(C + \lambda)$ that equates to $\{100, 80, 67, 57, 50\}$ Mbps for the given λ. Similar values are observed in the measurement results shown in Fig. 5.

V. Estimation from TCP Measurements

In this section, we investigate the $(g_{\text{in}},\ g_{\text{out}})$ characteristics of passive TCP measurements and evaluate the available bandwidth estimates that can be obtained thereof. Further, TCP offers a unique opportunity to estimate the available bandwidth from sender-side measurements only, using the feedback that is provided by the spacing of the acknowledgements. We extend the estimation method to include multiple packet gaps as well as acknowledgement gaps. For these, we also develop a technique that deals with packet loss.

A. TCP $(g_{\text{in}},\ g_{\text{out}})$ characteristics

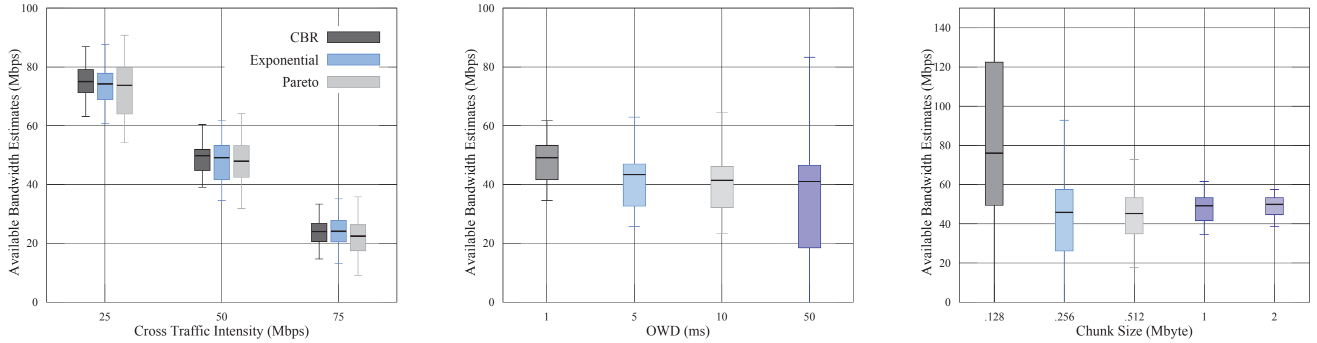
The $(g_{\text{in}},\ g_{\text{out}})$ characteristics of TCP traffic are largely affected by TCP congestion control and related parameters such as the OWD. Since the input gap is not fixed as in case of active probing, we extend the notation by superscript k and write $g^k_{\text{in}} = t^{k+1}_{\text{in}} - t^k_{\text{in}}$ whenever we refer to a specific input gap. Above, t^k_{in} is the send time-stamp of packet k.

In Fig. 6, we show two characteristic sets of $(g_{\text{in}},\ g_{\text{out}})$ samples obtained from TCP traffic. The OWD is 1 ms in Fig. 6(a) and 10 ms in Fig. 6(b). The remaining parameters are as in Fig. 4. For an OWD of 1 ms, the values of g_{in} are spread more or less evenly over a wide range, similar to Fig. 4. If the OWD is increased to 10 ms, we observe, however, a clustering of the g_{in} values. Roughly three clusters are formed: in the left part, large g_{in} in the range of up to two OWD occur if the sender has to wait for acknowledgements after transmitting a full CWND; in the middle part, the self-clocking of TCP by the acknowledgements causes g_{in} that correspond roughly to the available bandwidth; and in the right part, back-to-back packets can be found that are triggered, e.g., by cumulative acknowledgements.

Bandwidth estimates for CBR cross traffic in the range $\lambda \in \{0, 25, 20, 75, 100\}$ Mbps are summarized in Fig. 6(c), where we show box-plots comprising the 0.05, 0.25, 0.5, 0.75, and 0.95-quantiles obtained from 100 repeated measurements each. As before the capacity is $C = 100$ Mbps and the chunk size 1 Mbyte. We compare the case of TCP measurements with an OWD of 1 and 10 ms, respectively with the UDP measurements presented in Fig. 5, before. The results confirm that it is possible to estimate the available bandwidth from passive TCP measurements using the probe gap model. Moreover, we are able to obtain rational available bandwidth estimates in those scenarios of non-negligible OWDs where TCP throughput as bandwidth estimator is limited. While in case of small OWDs, TCP and UDP measurements perform comparably, the bandwidth estimates show more variability as well as a certain underestimation if the intensity of the cross traffic is low and the OWD is increased. In case of low cross traffic intensity, fewer samples pass the threshold test and contribute to the regression line.

B. Parameter Evaluation

We proceed with an evaluation of the effects of relevant parameters, including the intensity and distribution of cross traffic, the OWD, and the chunk size, on the quality of bandwidth estimates that are obtained from passive TCP measurements. We use cross traffic of different burstiness: CBR as assumed by the probe gap model; a moderate burstiness due to exponential inter-arrival times; and a strong burstiness due to Pareto inter-arrival times with infinite variance, caused by a shape parameter of $\alpha = 2$. The packet size is $l = 1514$ byte and the average rate of the cross traffic is $\lambda \in \{25, 50, 75\}$ Mbps. The chunk size is 1 Mbyte, the capacity $C = 100$ Mbps, and the OWD is 1 ms. The results of 100 repeated experiments for each configuration are plotted as box plots in Figure 7(a). We notice that the median of the estimates corresponds well with the true available bandwidth,



(a) OWD = 1 ms, chunk size 1 Mbyte (b) $\lambda = 50$ Mbps exponential, chunk size 1 Mbyte (c) $\lambda = 50$ Mbps exponential, OWD = 1 ms

Fig. 7. Parameter evaluation. Bandwidth estimates for different types of cross traffic burstiness (a), OWDs (b), and chunk sizes (c). The quality of the estimates is good for small to medium OWDs and improves with the chunk size. The burstiness of cross traffic mostly influences the variability of the estimates.

regardless of the type of cross traffic. Effects of the cross traffic can be observed in the variability of the estimates that increases if the burstiness is increased.

In Fig. 7(b), we evaluate the impact of the OWD in a wide range of $\{1, 5, 10, 50\}$ ms for exponential cross traffic with average rate $\lambda = 50$ Mbps. The results quantify the effects of the OWD on the $(g_{\text{in}}, g_{\text{out}})$ characteristics that we observed already in Fig. 6. The quality of the bandwidth estimates obtained from passive TCP measurements decreases if the OWD is increased. We note that this effect is specific to TCP congestion control. A fixed increase of the OWD will not alter the $(g_{\text{in}}, g_{\text{out}})$ characteristics of UDP traffic.

The impact of the chunk size, that determines the number of samples that are obtained for estimation of the available bandwidth, is evaluated in Fig. 7(c). The cross traffic is exponential with $\lambda = 50$ Mbps and the OWD is 1 ms. Clearly, the quality of the estimates and specifically the variance of the estimates improves significantly if more samples are available. Moreover, as the chunk size is increased, the quality of the samples changes, too. This is due to the growth of the CWND during the course of the transmission that causes less stalling. Considering the case of a chunk size of 128 kbyte that corresponds to about 85 packets, we conclude that it is challenging to obtain an estimate of the available bandwidth already during the slow start phase, as HyStart does.

In our figures, we generally include all outliers to show unaltered results. In practice, a number of sanity checks can be performed to filter such outliers, e.g., a decreasing regression line implies a contradiction as it indicates that the available bandwidth estimate is larger than the capacity.

C. Acknowledgement Gaps

TCP offers the option to perform the estimation based only on sender-side measurements of data and acknowledgement packets, i.e., no specific cooperation of the receiver is required. This feature is used for example by TCP HyStart. Here, we will advance the probe gap model to use acknowledgement gaps. Two aspects have to be considered: TCP uses delayed acknowledgements and typically only every other packet is acknowledged, i.e., the acknowledgement process effectively performs a sub-sampling; and secondly, the cross traffic in the reverse path may alter the spacing of acknowledgements and hence increase the measurement noise. In order to evaluate the impact of the two aspects one at a time, we first investigate the sub-sampling only. For this purpose, we define a multi-gap as $g^{j,k}_{\text{out}} = \sum_{i=j}^{k-1} g^{i}_{\text{out}} = t^{k}_{\text{out}} - t^{j}_{\text{out}}$ and $g^{j,k}_{\text{in}}$ accordingly. Subsequently, we make the transition from multi-gaps to the corresponding ack-gaps denoted $g^{j,k}_{\text{ack}}$. Fig. 8 illustrates the concepts of multi-gap and ack-gap. The packets are numbered by i and ACK i denotes a cumulative acknowledgement of all packets up to and including packet $i - 1$.

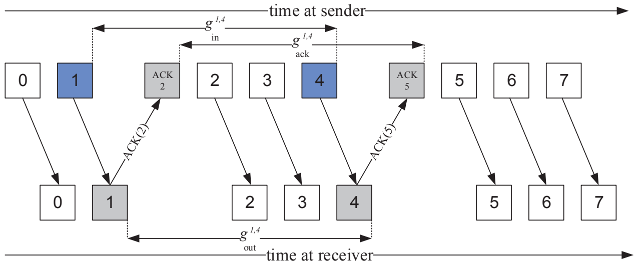


Fig. 8. Multi-gap and ack-gap models. The ack-gap enables available bandwidth estimation using sender-side measurements only.

We note that combining several gaps to form a multi-gap does not imply constant rate packet train models, since the individual input gaps from passive measurements are random. In fact, the derivation of a multi-gap response curve by repeated application of Eq. (4) requires a condition for each individual input gap. Considering only the relevant, upward segment of the gap response curve, we use Eq. (4) to derive

$$\frac{g^{j,k}_{\text{out}}}{g^{j,k}_{\text{in}}} = \frac{\lambda}{C} + \frac{(k-j-1)l}{C g^{j,k}_{\text{in}}}, \tag{6}$$

if $l/g^{i}_{\text{in}} > C - \lambda$ for all $i \in \{j, k-1\}$. The multi-gap response curve shows the same characteristic slope as the gap response curve with one difference: the average input gap $\overline{g}^{j,k}_{\text{in}} = g^{j,k}_{\text{in}}/(k-j-1)$ and average output gap $\overline{g}^{j,k}_{\text{out}} = g^{j,k}_{\text{out}}/(k-j-1)$ take the place of g_{in} and g_{out}, respectively.

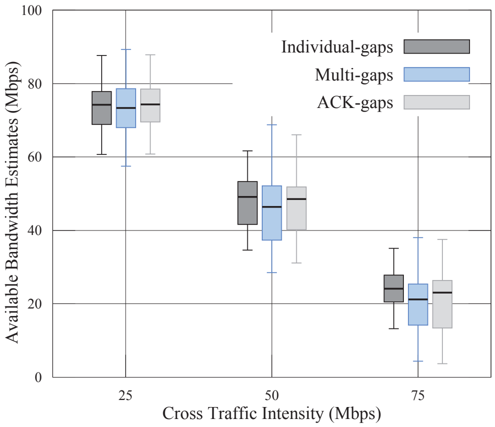


Fig. 9. Available bandwidth estimates obtained from individual gaps, multi-gaps, and ack-gaps, respectively. The use of multi-gaps and ack-gaps results in an increased variability of the estimates. The estimates are reasonable in all three cases.

In order to estimate the available bandwidth from the multi-gap response curve, we use the method defined in Sec. IV-A. A slight modification is required to identify samples that satisfy the condition of Eq. (6). Given the average input gaps $\overline{g}_{\text{in}}^{j,k}$ we find the minimal average input gap $\overline{g}_{\text{in}}^{\min}$ and the corresponding average output gap $\overline{g}_{\text{out}}^{\min}$. We select $g_{\text{in}}^{\max} = \overline{g}_{\text{out}}^{\min}$ as a threshold for g_{in}^{i} to test that $g_{\text{in}}^{i} < g_{\text{in}}^{\max}$.

To verify that $\overline{g}_{\text{out}}^{\min}$ is a valid threshold with respect to Eq. (6), i.e., $\overline{g}_{\text{out}}^{\min} < l/(C-\lambda)$, we apply Eq. (5) repeatedly to obtain

$$\overline{g}_{\text{out}}^{\min} = \frac{\overline{g}_{\text{in}}^{\min}\lambda + l}{C}, \tag{7}$$

where we assumed that all g_{in}^{i} that are part of $\overline{g}_{\text{in}}^{\min}$ satisfy $g_{\text{in}}^{i} < l/(C-\lambda)$. By insertion of $\overline{g}_{\text{in}}^{\min} < l/(C-\lambda)$ it follows that $\overline{g}_{\text{out}}^{\min} < l/(C-\lambda)$.

The estimation method applies in the same way if acknowledgements are used to avoid receiver-side measurements. In this case $g_{\text{ack}}^{j,k}$ takes the place of $g_{\text{out}}^{j,k}$.

We present bandwidth estimates obtained from multi-gap and ack-gap measurements compared to the use of individual gaps in Fig. 9. The cross-traffic is exponential with average rate $\lambda \in \{25, 50, 75\}$ Mbps. Cross traffic is generated both in downstream and upstream direction. The chunk size is 1 Mbyte, the capacity $C = 100$ Mbps, and the OWD is 1 ms. Box-plots of 100 repeated measurements are shown. In all cases the median of the estimates closely matches the true available bandwidth. The variability is, however, increased if multi-gaps or ack-gaps are used. The reason is due to a smaller number of samples that pass the threshold test. Compared to the multi-gap results, we do not notice a significant change of the accuracy when ack-gaps are used.

We note that the estimation may be enhanced using a weighted regression that takes the number of individual gaps that are comprised by a multi-gap into account. We did not use this option since TCP typically acknowledges every other packet so that most of the multi-gaps or ack-gaps are of the same size, i.e., they comprise two individual gaps.

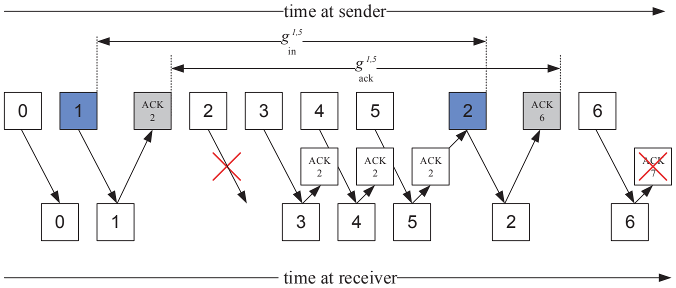


Fig. 10. Use of the ack-gap model in the presence of packet loss. Duplicate acknowledgements are ignored and the ack-gap is closed by the next higher cumulative acknowledgement. The corresponding input gap has to consider the retransmission that triggered this acknowledgement.

D. Evaluation of Loss

The fluid flow models that are used in available bandwidth estimation assume lossless systems and few methods for bandwidth estimation consider loss. The iterative packet train method Pathload [4] uses loss as an indication that $r_{\text{in}} > C-\lambda$. The work [13] models lost packets as incurring an infinite delay. Further, it is possible to define the output rate r_{out} of a packet train in the presence of loss, considering only the packets that are received. The output gap g_{out} of a packet pair is, however, void if any of the two packets is lost. This may cause estimation bias, since packet pairs that encounter congestion have a higher loss probability.

If acknowledgement gaps are used, packet loss has to be taken into account since it causes retransmissions and perturbs the sequence. Fig. 10 shows an example, where a packet is lost and three duplicate acknowledgements trigger a fast retransmit. To deal with this case, we define the following procedure: first, when determining the ack-gaps, duplicate acknowledgements are ignored; also packets that are retransmitted later are ignored; second, the packets that triggered the remaining acknowledgements are identified; these are used to compute the corresponding input gaps. In the example in Fig. 10, ACK 2 and ACK 6 remain, resulting in $g_{\text{ack}}^{1,5}$. The acknowledgements have been triggered by packet 1 and by the retransmission of packet 2, respectively. Hence, the corresponding $g_{\text{in}}^{1,5}$ is determined as the difference of the send time-stamps of these two transmissions.

While the procedure can deal with single packet losses, we note that burst losses can result in more intricate constellations that may not be resolvable unambiguously. The loss of acknowledgements, on the other hand, is less an issue as it is typically resolved by the next cumulative acknowledgement.

Available bandwidth estimates that are obtained from ack-gaps with loss in downstream and upstream direction are shown in Fig. 11. The cross traffic has exponential inter-arrival times and an intensity of 50 Mbps. The OWD is 1 ms and the chunk size 1 Mbyte. Loss rates of 0%, 0.1%, and 1% are evaluated. We notice an increase of the variability of the

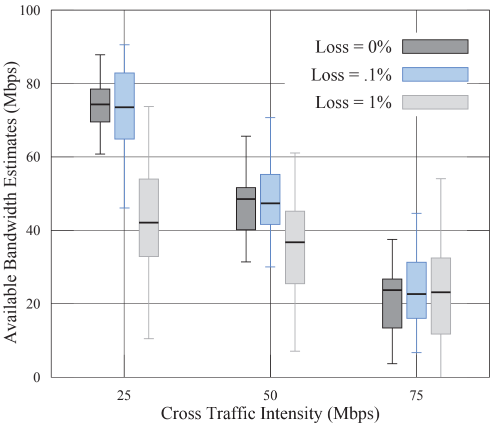


Fig. 11. Available bandwidth estimates obtained from ack-gaps in the presence of loss in downstream and upstream direction. The accuracy of estimation decreases with increasing loss rate.

estimates in case of packet loss. Further, for a loss rate of 1%, the available bandwidth is underestimated, particularly if the intensity of the cross traffic is low. In this case, few samples that pass the threshold test remain for the regression step to estimate the upward segment of the gap response curve.

VI. Conclusions

Motivated by the shortcomings of TCP throughput as available bandwidth estimator, we investigated how techniques from active probing can benefit TCP bandwidth estimation. The difficulty is due to the uncontrollable traffic patterns emitted by TCP that do not match typical active probes, such as packet trains. To solve the issue, we used individual packet gaps and applied a linear regression technique to estimate the gap response curve. We performed a comprehensive measurement study to evaluate the accuracy of the available bandwidth estimates, where we investigated the impact of relevant parameters including type and intensity of cross traffic, and the OWD. We also considered the effect of the number of samples on the variability of the estimates. While it turned out that obtaining bandwidth estimates already during the initial slow start phase is challenging, the transmission of a typical DASH chunk of 1 Mbyte or more can provide stable estimates. Taking advantage of the feedback that is provided by TCP acknowledgements, we enhanced the estimation method to use sender-side measurements only.

References

[1] B. Melander, M. Bjorkman, and P. Gunningberg, “A new end-to-end probing and analysis method for estimating bandwidth bottlenecks,” in *IEEE Globecom*, 2000, pp. 415–420.

[2] A. Johnsson, B. Melander, and M. Björkman, “Diettopp: A first implementation and evaluation of a simplified bandwidth measurement method,” in *Second Swedish National Computer Networking Workshop*, 2004.

[3] C. Dovrolis, P. Ramanathan, and D. Moore, “What do packet dispersion techniques measure?” in *IEEE INFOCOM*, 2001, pp. 905–914.

[4] M. Jain and C. Dovrolis, “Pathload: A measurement tool for end-to-end available bandwidth,” in *Passive and Active Measurements (PAM) Workshop*, 2002.

[5] ——, “End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput,” *IEEE/ACM Transactions on Networking (TON)*, vol. 11, no. 4, pp. 537–549, 2003.

[6] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell, “pathChirp: Efficient available bandwidth estimation for network paths,” in *Passive and Active Measurement (PAM) Workshop*, 2003.

[7] J. Strauss, D. Katabi, and F. Kaashoek, “A measurement study of available bandwidth estimation tools,” in *ACM SIGCOMM Conference on Internet Measurement*, 2003, pp. 39–44.

[8] N. Hu and P. Steenkiste, “Evaluation and characterization of available bandwidth probing techniques,” *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, pp. 879–894, 2003.

[9] S. Ekelin, M. Nilsson, E. Hartikainen, A. Johnsson, J.-E. Mangs, B. Melander, and M. Bjorkman, “Real-time measurement of end-to-end available bandwidth using Kalman filtering,” in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2006, pp. 73–84.

[10] X. Liu, K. Ravindran, and D. Loguinov, “A queueing-theoretic foundation of available bandwidth estimation: single-hop analysis,” *IEEE/ACM Transactions on Networking*, vol. 15, no. 4, pp. 918–931, 2007.

[11] ——, “A stochastic foundation of available bandwidth estimation: Multi-hop analysis,” *IEEE/ACM Transaction on Networking*, vol. 16, no. 1, pp. 130–143, 2008.

[12] J. Liebeherr, M. Fidler, and S. Valaee, “A system theoretic approach to bandwidth estimation,” *IEEE/ACM Transactions on Networking*, vol. 18, no. 4, pp. 1040–1053, 2010.

[13] R. Lübben, M. Fidler, and J. Liebeherr, “Stochastic bandwidth estimation in networks with random service,” *IEEE/ACM Transactions on Networking*, vol. 22, no. 2, pp. 484–497, 2014.

[14] S. Keshav, “A control-theoretic approach to flow control,” in *Proc. ACM SIGCOMM*, Sep. 1991, pp. 3–15.

[15] V. Paxson, “End-to-end internet packet dynamics,” *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 277–292, 1999.

[16] S. Akhshabi, A. C. Begen, and C. Dovrolis, “An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP,” in *ACM Conference on Multimedia Systems*, 2011, pp. 157–168.

[17] Q. Lin, Y. Liu, Y. Shen, H. Shen, L. Sang, and D. Yang, “Bandwidth estimation of rate adaption algorithm in DASH,” in *IEEE Globecom Workshops*, 2014, pp. 243–247.

[18] M. Jain and C. Dovrolis, “Ten fallacies and pitfalls on end-to-end available bandwidth estimation,” in *ACM Internet Measurement Conference*, 2004, pp. 272–277.

[19] S. Ha, I. Rhee, and L. Xu, “Cubic: a new TCP-friendly high-speed TCP variant,” *ACM SIGOPS Operating Systems Review*, vol. 42, no. 5, pp. 64–74, 2008.

[20] S. Ha and I. Rhee, “Taming the elephants: New TCP slow start,” *Computer Networks*, vol. 55, no. 9, pp. 2092–2110, 2011.

[21] C. Dovrolis, P. Ramanathan, and D. Moore, “Packet-dispersion techniques and a capacity-estimation methodology,” *IEEE/ACM Transactions On Networking*, vol. 12, no. 6, pp. 963–977, 2004.

[22] F. Michelinakis, G. Kreitz, R. Petrocco, B. Zhang, and J. Widmer, “Passive mobile bandwidth classification using short lived TCP connections,” in *IFIP Wireless and Mobile Networking Conference (WMNC)*, 2015, pp. 104–111.

[23] L. Zhang, S. Shenker, and D. D. Clark, “Observations on the dynamics of a congestion control algorithm: The effects of two-way traffic,” *ACM SIGCOMM Computer Communication Review*, vol. 21, no. 4, pp. 133–147, 1991.

[24] A. Capone, L. Fratta, and F. Martignon, “Bandwidth estimation schemes for TCP over wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 3, no. 2, pp. 129–143, 2004.

[25] D. S. Anderson, M. Hibler, L. Stoller, T. Stack, and J. Lepreau, “Automatic online validation of network configuration in the emulab network testbed,” in *IEEE International Conference on Autonomic Computing*, 2006, pp. 134–142.

[26] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs, “Iperf: The TCP/UDP bandwidth measurement tool,” *https://iperf.fr/*, 2005.

[27] J. Laine, S. Saaristo, and R. Prior, “Real-time udp data emitter (rude) and collector for rude (crude),” 2000.

[28] S. Avallone, S. Guadagno, D. Emma, A. Pescape, and G. Ventre, “D-ITG distributed internet traffic generator,” in *Quantitative Evaluation of Systems*, 2004, pp. 316–317.

[29] K. Wagner, “Short evaluation of linuxs Token-Bucket-Filter (TBF) queuing discipline,” 2001.

[30] “ethtool,” ://linux.die.net/man/8/ethtool, accessed: 05-01-2017.