Passive Access Capacity Estimation through the Analysis of Packet Bursts

Martino Fornasa and Massimo Maresca

Centro di ricerca sull'Ingegneria delle Piattaforme Informatiche
University of Genova, University of Padova - Italy
`{martino.fornasa,massimo.maresca}@dei.unipd.it`

Abstract. Downlink capacity is the most advertised quality parameter of broadband Internet access services, as it significantly influences the user perception of performance. This paper presents an automatic computation method of such a capacity from a measurement point located inside the network. The method is fully passive as it takes advantage of existing TCP connections. It does not inject additional traffic in the network and does not require end-host collaboration. The method takes advantage of the bursty nature of TCP to apply the packet-dispersion approach to TCP segment sequences (packet trains) rather than to segment pairs. This results in a sensible reduction of noise impact on rate estimation. We present an analysis of the effects of the interfering traffic in the access link on rate estimation. We show that it is possible to detect and drop TCP packet trains affected by interfering traffic and to identify and process the packet trains that are not affected by interfering traffic. The proposed method has been validated by means of a set of experiments on ADSL and fibre Internet access services, which are described in the paper. Applications of the proposed method are i) to provide a passive SLA verification method to Internet Service Providers toward Access Service Providers, ii) to support widespread Internet access capacity measurement campaigns, and iii) to perform constant monitoring of access links for fault detection.

Keywords: Broadband Access Service, Capacity, Passive Estimation.

1 Introduction

Downlink capacity, i.e. the maximum achievable downlink network-layer rate, is the most advertised quality parameter of broadband Internet access service, as it significantly influences the user perception of application service performance. This paper proposes a passive method to estimate the downlink capacity of an access link to a TCP/IP network from a measurement point located inside the network by taking advantage of the existing TCP connections.

The method suits a variety of scenarios, the most relevant of which is the one in which a service provider wants to estimate the quality of the access service provided by another provider called Access Service Provider. This is what very often happens in Internet service provisioning, in which the access service is often operated by an Access Service Provider (ASP), typically the incumbent operator or a local telephone

R. Szabó et al. (Eds.): AccessNets 2010, LNICST 63, pp. 83–99, 2011.

company. The method proposed in this paper enables Internet Service Providers (ISPs) to estimate the downlink access capacity provided by the ASP passively and autonomously, without the cooperation of the ASP and without the cooperation of the final customer. Additionally, the method supports large-scale measurement campaigns aimed at characterizing broadband access link capacity and supports access link fault detection.

The method takes advantage of the bursty nature of TCP and applies the packet-dispersion technique to the acknowledgement segments (ACK) generated by TCP data segment sequences (packet trains) rather than by TCP packet pairs. To our knowledge, the method is the first effective narrow-link capacity estimation method that is both *passive* (it does not inject traffic on the network) and *remote-based* (it relies on the ACK packet passing times measured in a different location with respect to the narrow link)[1].

In order to obtain a method which is both passive and remote we process packet trains rather than packet pairs, as longer packet sequences allow to reduce the impact of the noise corresponding to the delay jitter of the ACK upstream path. However, packet trains last longer than packet pairs and are therefore more subject to cross-traffic than packet pairs. We propose a method to detect and drop the packet trains affected by interfering traffic, both in the uplink access queue and in the downlink access queue.

The proposed approach was validated through a set of experiments performed over ADSL and fibre access lines under different traffic conditions. This work is a continuation and extension of the author's earlier work presented in [11].

2 Background

Among the many capacity estimation methods proposed in the past we focus on the *packet dispersion method* [1, 2, 5, 6]. Such a method is based on the observation that the dispersion (i.e., the time difference between the last bit of the first packet and the last bit of the second packet) of a pair of equally-sized back-to-back packets traversing a link can be modified along the source-destination path. In general, the dispersion (d) of a back-to-back pair after a link of capacity r is $d = w / r$, where w is the size of the two back-to-back packets. Using such a formula, it is possible to calculate the link capacity as $r = w / d$. The formula is valid assuming that no interfering traffic is transported over the link. On the contrary an interfering traffic on the link changes the packet dispersion and leads to a rate estimation error. In absence of interfering traffic the dispersion of two back-to-back packets that traverse a path is the one induced by the path "narrow link" (i.e., the link having the smallest capacity on the path). The packet dispersion method has also been used in TCP Westwood [10] in order to estimate the fair share bandwidth for a TCP connection.

Capacity estimation techniques can be active or passive. Active techniques rely on active probing and therefore require the injection of traffic on the network, whereas passive techniques only rely on traffic observation (traffic traces). As a consequence passive techniques enable non-invasive capacity estimation of large numbers of access links as well as long lasting measurement campaigns.

---

[1] Active-remote techniques have been proposed in the past ([8, 9]).

Passive packet dispersion methods can be classified depending on the point where the measurement equipment is placed. *Receiver-side* techniques are based on measurements taken at the receiver, *sender-side* techniques are based on measurements taken at the sender, *network-side* techniques are based on measurements taken at one network node located in the path between the sender and the receiver.

If the measurement point is on the sender side or on the network side the packet dispersion must be estimated by observing the existing TCP connections, and more specifically by using the timestamps of the TCP acknowledgement segments taken at the measurement point to estimate the interarrival times of the corresponding forward data segments at the receiver.

In general, the capacity estimation methods based on the ACK interarrival times are more complex than the receiver-based methods, because of the following reasons:

1. The traffic that passes through a path that does not include the measurement point (interfering traffic) can disturb the measurement.
2. The jitter on the upstream path delay may modify the ACK dispersion.
3. The TCP ACKs are sent according to the delayed ACK scheme, i.e., the TCP protocol acknowledges more than one data packet at a time.
4. Congestion on the uplink queue leads to a decreased ACK pair dispersion, namely the *ACK compression* phenomenon.

3 Proposed Approach

A general issue impacting the accuracy of the packet pair dispersion approaches is that the *w/r* ratio is small compared to the network delay jitter. This is caused by the fact that the Maximum Segment Size (MSS) of TCP is about 1500 bytes for legacy reasons, irrespective of the ever increasing capacity of networks. This issue can be mitigated by the adoption of a more general approach that consists of considering longer packet sequences, usually called packet trains. The dispersion of a packet train composed by n packets [0, 1, … n-1] of size w_i will be:

$$d_{0,n-1} = \frac{\sum_{i=1}^{n-1} w_i}{r}$$

The above formula is valid assuming no interfering traffic on the link. Considering packet trains allows obtaining capacity estimates less influenced by the measurement noise. However, some issues arise when considering packet trains, as stated in [2]. The authors correctly argue that the longer a packet sequence, the larger the probability of the influence of interfering traffic causing increased dispersion.

We solve such an issue by means of a method aimed at discarding the packet trains affected by the interfering traffic, both in the downstream access link and in the upstream access link. We show that it is possible to detect and drop the packet trains affected by interfering traffic. The proposed method is based on the following assumptions:

1. There are no post-narrow links, since we are considering the access link, that is the last downstream link (and usually the narrow link of the downstream path).
2. The capacity of the access link is far below the capacities of the backbone links.

3. The majority of TCP data segments are approximately 1500 bytes long. In fact, as stated in [4], the packets size distribution on the Internet is mostly bimodal at 40 bytes (TCP ACK segments) and 1500 bytes (TCP MSS segments).

Assumption 1 excludes the increased dispersion of packet trains caused by post-narrow links. Assumption 2 allows isolating the effect of the upstream path delay jitter as a symmetric noise. Assumption 3 allows coming up with heuristics aimed at filtering out the packet trains influenced by interfering traffic on the downstream path.

4 Reference Model

We consider a reference model (shown in Fig. 1) in which a passive Traffic Monitoring System (TMS) is placed on a specific interface of a Measurement Node (MN). The MN is connected to a Customer Premises Gateway (CPG) by means of a chain of links and nodes. The MN can be placed somewhere inside an Internet Service Provider network, at the border of such a network (for example in a Neutral Access Point facility), or at a network endpoint, for example in a content provider premises. We consider the access service in place between the Service Provider Remote Access Service (SP-RAS) and the CPG.

The access link downstream and upstream capacities can be equal (symmetric access, such as a fibre or HDSL access) or different (asymmetric access, such as an ADSL). We are interested in measuring access downstream capacity taking advantage of any existing TCP connections, so we consider the TCP half connection in which the end-user host (the one attached downstream the CPG) acts as a receiver, and a host placed upstream with respect to the MN acts as a sender, i.e., we consider TCP data segments flowing toward a CPG and the TCP ACK segments coming from the CPG. This corresponds to the usual case in which the end-user host acts as a client of a server on the Internet.

The TMS captures the TCP segments passing through the MN interface, detects the TCP data segment with the corresponding ACK segments based on the TCP sequence number, and fills out an array of (Packet size, TCP ACK passing time timestamp) pairs:

$$\begin{gathered}(w_0, t_0^a)\\(w_1, t_1^a)\\(w_2, t_2^a)\\\dots\\(w_{N-1}, t_{N-1}^a)\end{gathered}$$

where w_i is the data segment IP total size, t_i^a is the ACK segment passing timestamp and N is the total number of TCP ACK segments during the observation period. Such an array is the input of the capacity estimation algorithm presented in next section.

In some cases TCP does not send an ACK segment for each data segment received due to the *delayed acknowledgment* technique. This issue will be discussed in Section 7.

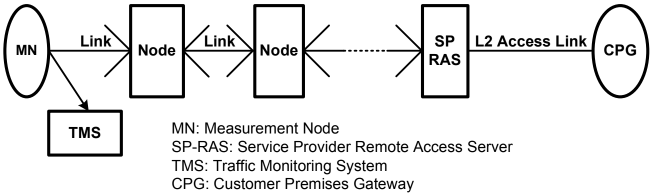

Fig. 1. Description of the reference model

5 Analysis: No Interfering Traffic

In this section we analyze the behaviour of the reference model described in Section 4 in order to devise a downlink access capacity estimation method. In order to do so, we make two simplifying assumptions, which will be removed in the following sections:

1. *No interfering traffic on the access downlink queue,* i.e., all the traffic that passes through the access link passes also through the TMS.
2. *No congestion on the access uplink queue,* i.e., acknowledgment segments never queue on the uplink access queue, thus the uplink access link gives a fixed contribution to data/acknowledgment pair round-trip time[2].

We consider a data segment sequence [*i, i+1, ..., j-2, j-1*] where all but the first segment arrive at a non-empty queue. We call such a sequence a 'Packet Burst' (PB). More precisely, we suppose now that:

- Before the arrival of data segment *i*, the access downlink queue is empty (Fig. 2a).
- The queue does not empty up to the arrival of segment *j-1* (Fig. 2b).
- Before the arrival of segment *j* the queue is empty (Fig. 2c).

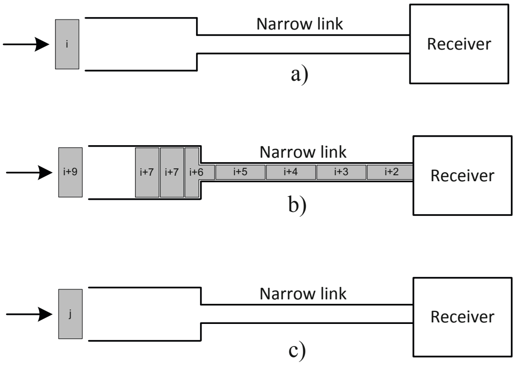

Fig. 2. Packet burst

---

[2] This hypothesis is justified by the fact that the size of a data packet can be up to 1500 bytes, whereas the size of an ACK packet is around 40 bytes. So, in the absence of data traffic originating downstream the access link, symmetric access links never show queueing on the uplink, and also asymmetric access links are always correctly dimensioned in order to avoid such a phenomenon.

We split the analysis in two phases. First, we write an expression for the arrival time at the MN of the ACK segments triggered by the data segments forming the PB (t^a). Second, we write an expression for the arrival time of the *j*-th ACK segment, the first that does not belong to the PB.

During the PB. During a PB we can write an expression for the interarrival time to the receiver (dispersion) between data segment *k*-1 and data segment *k* as:

$$t_k^r - t_{k-1}^r = \frac{w_k}{r} \qquad \forall k : i < k < j$$

where t_k^r is the arrival time of the *k*-th data segment to the receiver, w_k is its total IP size and *r* is the downlink access capacity. Now, we can write an expression for the arrival time of the generic *k*-th segment to the receiver as:

$$t_k^r = t_i^r + \frac{1}{r}\sum_{l=i+1}^{k} w_l \qquad \forall k : i < k < j$$

It is worth noting that during a PB the interarrival times at the receiver are not influenced by the downstream delay jitter (i.e., the jitter on the delay needed by a data segment to travel from the MN to the access downlink queue).

The arrival of a data segment at the TCP receiver causes the generation of an ACK segment that flows back to the sender. The arrival time of the *k*-th ACK segment at the MN (t_k^a) can be written as the sum of the arrival time of the data segment at the receiver (t_k^r), plus the network upstream delay (*T*), plus a noise component due to the upstream delay jitter (ξ_k):

$$t_k^a = t_k^r + T + \xi_k = t_i^r + \frac{1}{r}\sum_{l=i+1}^{k} w_l + T + \xi_k \qquad \forall k : i < k < j$$

The above formula can by simplified by subtracting the arrival time of the first ACK of the PB (t_i^a):

$$\underbrace{t_k^a}_{y_k} = t_i^a - \xi_i + \frac{1}{r}\underbrace{\sum_{l=i+1}^{k} w_l}_{x_k} + \xi_k \qquad \forall k : i < k < j \qquad (1)$$

Now we define ($x_k \equiv \sum_{l=i+1}^{k} w_l$) and ($y_k \equiv t_k^a$). Thus, as long as the queue does not empty, the (x_k, y_k) points are approximately arranged on a line with slope *1/r* and y-intercept (t_i^a- ξ_i). The reciprocal of the slope of such a line represents the capacity of the downlink access queue (*r*). In order to devise the fitting line parameters, it is possible to apply the linear regression method to such points. The fitting line is represented in Fig. 3 as a dotted line. So, during a PB, the capacity of the downlink access queue can be obtained as the inverse of the slope obtained by applying the linear regression on the PB points:

$$r = 1/lin\_regr\_slope(\{x_i, x_{i+1}, \ldots, x_{j-1}\}, \{y_i, y_{i+1}, \ldots, y_{j-1}\})$$

After the Packet Burst. The data segment *j* arrives at an empty queue. So point (x_j, y_j) is not aligned with the previous points, while it is shifted upward a Δ quantity (see Fig. 3). In general, given a sequence of pairs (w_k, t_k^a), it is possible to identify a set of PBs where a linear relationship exists. As shown in Fig. 4, such PBs form a set of fitting lines, at different y-intercepts, but with same slope (*1/r*).

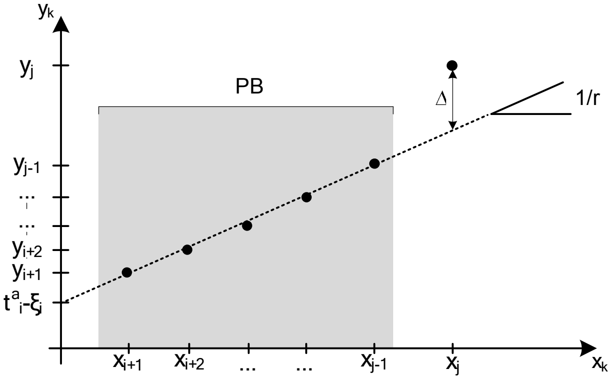

Fig. 3. Linear relationship

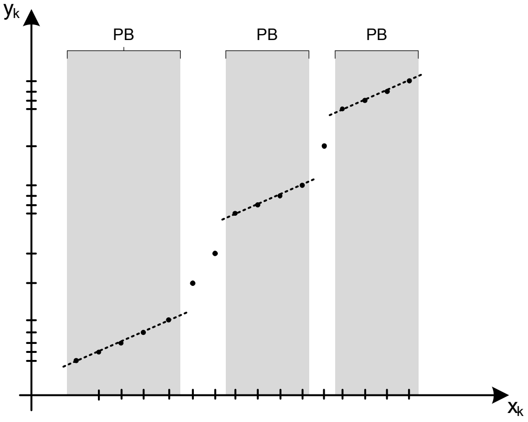

Fig. 4. Linear relationship: multiple PBs

5.1 Packet Burst Identification Algorithm

We propose an algorithm to identify the PBs during a given observation period taking advantage of the linear relationship devised in Equation (1). The algorithm input is the (t_i^a, w_i) array obtained at the TMS, while the output is the set of the capacities (r_0, r_1, …, r_{P-1}) associated with the maximum-sized PBs. The capacity associated to each PB is obtained by linear regression over the PB (i.e., the reciprocal of the slope of the fitting line).

The algorithm (see the pseudocode on Fig. 5) consists of successive tests over increasing sequences of pairs, to find the maximum-sized sequence of pairs showing a 'good' fit to the linear model described by Equation (1). It starts considering the subsequence composed of the first three pairs ([m, n], m = 0, n = 2). At each iteration, the algorithm:

- Calculates the $\{x_k, y_k\}$ points over the considered interval, i.e., $\{x_k, y_k\}_{k=m \text{ to } n}$ according to x_k and y_k definitions provided in the previous section.
- Performs a fit test on such points. Then:
  - If the fit is bad, the interval is shifted up by one (m←m+1; n←n+1) and the next iteration is started.
  - If the fit is good, the capacity value associated with such an interval is saved, the interval is enlarged by one (n←n+1), and a new iteration is started; if the fit on the larger interval is good, the interval is enlarged another time, and so on. However, if the fit on the larger interval is bad, the last valid capacity value (the one found in a previous iteration) is retained, and the next three-element interval is selected.

At the end of the iterations, the algorithm has identified several PBs, each characterized by a capacity value.

Goodness of fit. In principle it could be possible to use the linear regression coefficient of determination (R^2) calculated over the considered interval in order to discriminate between a good fit and a bad fit to the linear model[3]. However, we noticed that the use of R^2 has some drawbacks on long ($n > 4$) TCP segment sequences, as it tends to join successive PBs that are split by a non-PB point, because a single nonlinearity can be hidden summing a large number of squared residuals. In order to overcome this problem, we apply a different fit evaluation method, based on the definition of the *instantaneous rate* (ρ), defined as the ratio between the increase in x and the increase in y of the PB points:

$$\rho_k = \frac{x_k - x_{k-1}}{y_k - y_{k-1}} \qquad \forall k : m < k \le n$$

The linearity condition can be detected by checking that all the instantaneous rates on the PB are equal. So, for every $[m,n]$ segment interval, we calculate the above ratios and consider a fit good when the difference between each of the ρ_k values and the mean of the ρ_k over the interval is below a given threshold; in particular the condition is:

$$|\rho_k - \text{mean}(\rho_k)| \le C \cdot \text{mean}(\rho_k) \qquad \forall k : m < k \le n$$

In this way we define a range around the mean value of the instantaneous rates, and all the rate values on the interval have to fall within that range. The C value must be tuned; in our test we found that a 0.2 value (giving a ±20% range) can be appropriate.

In summary, the algorithm exploits the linearity of the ACK generation time during a PB in order to identify all the PBs during the measurement period and to estimate a downlink capacity for each of them.

---

[3] The coefficient of determination is defined for a $[m, n]$ interval as follows: $R^2 \equiv 1 - \frac{\sum_{k=m}^{n} (y_k - \hat{y}_k)^2}{\sum_{k=m}^{n} (y_k - \bar{y})^2}$ where $\hat{y}_k$ is the value predicted by the linear model (i.e. $\hat{y}_k \equiv t_m^a + \frac{1}{r} x_k$); and $\bar{y}$ is the mean of the x_k ($\bar{x} \equiv \sum_{k=m}^{n} y_k / (m - n + 1)$). The coefficient of determination value is between 0 and 1, where 1 means that the fit line passes exactly through the measured points.

```
**Input:** $\{t_k^a, w_k\}_{k=0 \text{ to } (N-1)}$
**Output:** $r_0, r_1, \ldots, r_{P-1}$
$m \leftarrow 0$
$n \leftarrow 2$
$p \leftarrow 0$
$good \leftarrow$ false
repeat
    $x_m \leftarrow w_m$
    $y_m \leftarrow t_m^a$
    for $i = m + 1$ to $n$ do
        $x_i \leftarrow x_{i-i} + w_i$
        $y_i \leftarrow t_i^a$
    end for
    $r \leftarrow$ capacity obtained by linear
          regression on $(\{x_m, \ldots, x_n\}, \{y_m, \ldots, y_n\})$
    if the fit is good then
        $last\_valid\_capacity \leftarrow r$
        $good \leftarrow$ true
        $n \leftarrow n + 1$  //Enlarge the window by one
    else if $good$ = true then
            $r_p \leftarrow last\_valid\_capacity$
            $p \leftarrow p + 1$
            $good \leftarrow$ false
            $m \leftarrow n$
            $n \leftarrow m + 2$  //Select next three-element interval
        else
            $m \leftarrow m + 1$
            $n \leftarrow n + 1$  //Shift the window by one
        end if
    end if
until the end of dataset
```

Fig. 5. Pseudocode of the Packet Burst identification algorithm

6 Analysis: Interfering Traffic

In this section we remove the simplifying assumptions of no interfering traffic on the access queue made at the beginning of Section 5. In particular, in Section 6.1 we examine the effects of interfering traffic on the downlink access queue while in Section 6.2 we examine the effects of congestion on the uplink access queue.

6.1 Downlink Access Queue Interfering Traffic

In case of interfering traffic on the access downlink queue, i.e., TCP data segments that arrive at the access downlink queue through a path that does not include the MN, the queue might contain interleaved traffic coming from different paths, possibly invalidating the Packet Burst identification algorithm described in Section 5.1. Consider for example, at a given moment, the case in which the queue contains the traffic pattern:

`M M I M I I I M M . . .`

where `M` denotes a segment passing through the MN, and `I` denotes an interfering traffic segment. Such a pattern destroys the linear relationship described by Equation (1), thus invalidating the PB identification method. The interfering traffic on the downlink has two possible outcomes:

1. A given segment sequence is identified as a PB. This can be due to two causes:
   - There is no interfering traffic and the measured traffic produces a PB on the downlink queue. So, the capacity obtained by the algorithm is correct.
   - The interfering traffic and the measured traffic are shaped to cause a false positive, i.e., a pattern of measured and interfering traffic leading to a PB condition associated to an incorrect capacity. We discuss how to detect false positives later.
2. A given segment sequence is not identified as a PB. This can be due to:
   - The fact that there is no interfering traffic but the measured traffic does not produce a PB on the downlink queue.
   - The fact that the interfering traffic destroys the linearity on the downlink queue.

False positive detection. Equation (1), which provides an expression for the passing time of the *k*-th ACK segments belonging to a PB, can be modified to take into account the interfering traffic on the access downlink queue. Being v_i the sum of the sizes of the interfering segments that arrive at the queue between the arrival of the (*k-1*)-th measured segment and the *k*-th measured segment, we obtain:

$$t_k^a = t_i^a - \xi_i + \frac{1}{r}\sum_{l=i+1}^{k}(w_l + v_l) + \xi_k \qquad \forall k : i < \mathrm{k} < j$$

However, as we stated before, the capacity estimation algorithm is only able to monitor the traffic that passes through the MN. So, the system equation seen by the algorithm is the following:

$$t_k^a = t_i^a - \xi_i + \frac{1}{r^*}\sum_{l=i+1}^{k}(w_l) + \xi_k \qquad \forall k : i < \mathrm{k} < j$$

with a different capacity (r^*) with respect to the actual capacity. It is easy to devise the necessary condition for a false positive:

$$\frac{v_{i+1}}{w_{i+1}} = \frac{v_{i+2}}{w_{i+2}} = \frac{v_{i+3}}{w_{i+3}} = \ldots$$

Under such conditions, the under-estimated (wrong) capacity is:

$$r^* = \underbrace{\frac{w_i}{v_i + w_i}}_{<1} r$$

Moreover, as the length of most TCP data segments is about 1500 bytes [4], the denominator of the aforementioned expression can assume only a value that is an integer multiple of 1500. The segment patterns that can cause a false positive and their corresponding wrong rates are summarized Table 1.

Table 1. False positives segment patterns

Segment pattern	$\mathbf{r}^*$
`M I M I M I`	1/2 r
`M I I M I I M I I`	1/3 r
`M I I I M I I I M I I I`	1/4 r
…	…

In summary, the probability of false positives depends on the segment pattern of the measured and interfering traffic on the access downlink queue. So a false positive always causes a capacity under-estimation, with a capacity less than or equal to half of the real one.

An exception to this, as we show in Section 7, is that the delayed acknowledgement mechanism implemented by TCP can raise the maximum wrong capacity to $(2/3)r$.

So, the interfering traffic on the downlink access queue can cause a capacity underestimation (false positive) with a value less than 2/3 of the actual capacity value.

6.2 Uplink Access Queue Interfering Traffic

In Section 5 we assumed that the ACK segments never meet congestion on the access uplink queue, i.e., that the access ACK segments takes a constant amount of time to traverse such a queue. This means that, if we ignore the effect of the upstream path delay jitter, the ACK segments arrive at the MN at the same rate at which they are sent by the receiver.

Due to the size difference between TCP forward data segments and TCP ACK segments, congestion on the uplink access queue is likely to be triggered by data traffic on the uplink (e.g., during a file upload or when a file sharing application is active). If we consider the TCP data segments flowing through the uplink there are two possibilities:

1. Data traffic on the uplink modifies the ACK spacing of a PB and destroys its linearity. In this case, the segment sequence is discarded by the PB identification algorithm, without causing incorrect capacity estimation.
2. Data traffic on the uplink causes the ACK-compression phenomenon [3], i.e., a reduction of the time spacing between successive ACKs due to a congested uplink access queue.

The following example shows the impact of ACK compression on the capacity estimation algorithm. Let us consider a PB composed by segments $[i, i+1, \ldots, j-1]$ having the following properties:

- The access uplink queue contains w_{int} bits at the moment of the arrival of the first ACK at the uplink queue.

- w_{int} is large enough to cause the buffering of all the burst ACK segments in the uplink queue.

For simplicity, let us suppose that all the PB data segment sizes are equal to w and ignore the effect of the uplink delay jitter. Let r^u be the uplink capacity and let w_{ack} be the ACK segment size. Under such conditions, the ACK interarrival times do not depend on the downlink capacity, while they only depend on the uplink capacity.

$$t_k^a - t_{k-1}^a = \frac{w_{ack}}{r^u} \qquad \forall k : i < \mathrm{k} < j$$

The capacity value (r*) obtained is:

$$r^* = \frac{w}{t_k^a - t_{k-1}^a} = \frac{w}{w_{ack}} r^u$$

Such a value is independent of the actual downstream access capacity, and as a consequence it cannot be correct. In the typical case in which w=12000 bit (1500 Bytes) and w_{ack}=320 bit (40 Bytes), the estimated capacity is:

$$r^* = 37.5 r^u$$

On a symmetric access link (r=r^u), the estimated capacity is almost 40 times the actual capacity. On an asymmetric link (e.g., ADSL), the downlink/uplink capacity ratio is usually less than 20, which makes the estimated capacity nearly twice the actual capacity.

In summary, in presence of interfering traffic on the uplink, a PB can be subject to the ACK compression phenomenon due to uplink delay queue congestion. In this case the algorithm can estimate a ‘wrong’ rate at a value which is at least twice the actual downlink access capacity.

7 Delayed Acknowledgements

TCP receiver implementations may employ the *delayed acknowledgement* technique, which consists of sending less than one ACK segment for every received data segment. Specifications allow a host to send an ACK every two incoming data segments [7]. In presence of delayed ACKs, the PB identification algorithm can not measure the passing time of the ACKs of every TCP data segment, while it can only measure the ACK passing time every two data segments. In this case it is possible to consider a (w, t^a) pair for every ACK, with the timestamps taken from the second data segment (the one that received the ACK) and a size that is the sum of the sizes of the two data segments:

$$\begin{gathered}(w_0 + w_1, t_1^a)\\(w_2 + w_3, t_3^a)\\(w_4 + w_5, t_5^a)\\\ldots\end{gathered}$$

So, in this case a minimum of six successive TCP segments forming a PB is necessary to perform capacity estimation using the proposed method.

The delayed ACK mechanism also influences the effect of the downlink interfering traffic on the capacity estimation described in Section 6.1. In fact, in presence of delayed ACKs, there are more combinations of interfering traffic and measured traffic that can lead to false positives. In particular, the maximum capacity estimation associated to a false positive in the presence of delayed ACK is caused by the following traffic pattern on the downlink queue:

`M I M* M I M* M I M* M` …

where `M*` represents a measured packet that does not receive an ACK, `M` represents a measured packet that does receive an ACK, and `I` is an interfering traffic packet. It can be shown that in this case the estimated capacity is 2/3 of the actual capacity.

8 Experiments

We performed a set of experiments on ADSL and fibre access services to the Internet aimed at validating the proposed capacity estimation technique. We placed the TMS and a Web Server on a host attached to a well-provisioned link, with the path from the TMS to the access link composed by about 20 hops, traversing the backbone network of two service providers.

The experiments consisted in 100 HTTP downloads of a small file (50 KB) from the Web Server, with a 2-second interval. The file size was chosen in order to obtain short TCP connections to simulate Web surfing activities. Due the to TCP slow-start mechanism, such downloads did not exploit the full access capacity, but only a small fraction of it. We considered three possible traffic conditions:

A. *No interfering traffic,* obtained by running only the 50-KB file downloads on the access host.
B. *Interfering traffic on the downlink,* obtained by means of a large persistent file download from a third-party Web Server to the access host. The large file download was started before the experiments and kept active during the execution of the experiments, in such a way that the short downloads of the experiment had to compete with it.
C. *Interfering traffic on the uplink,* obtained by means of a large file upload to a third party server from the access host. The large file upload was started before the experiments and kept active during the execution of the experiments in such a way that the ACKs generated by the short downloads of the experiment had to compete with it.

Fig. 6 and 7 show the histograms representing the distribution of the capacity values obtained by the PB identification algorithm.

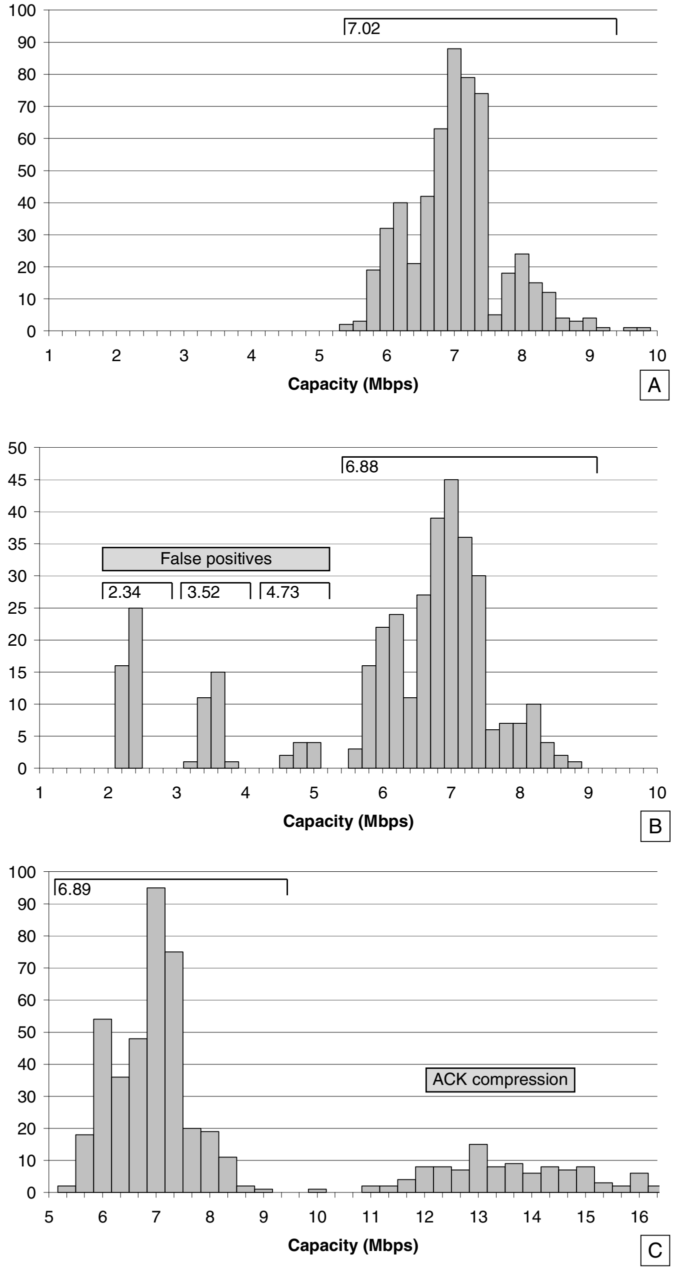

Fig. 6. Capacity distributions (7-Mbps ADSL)

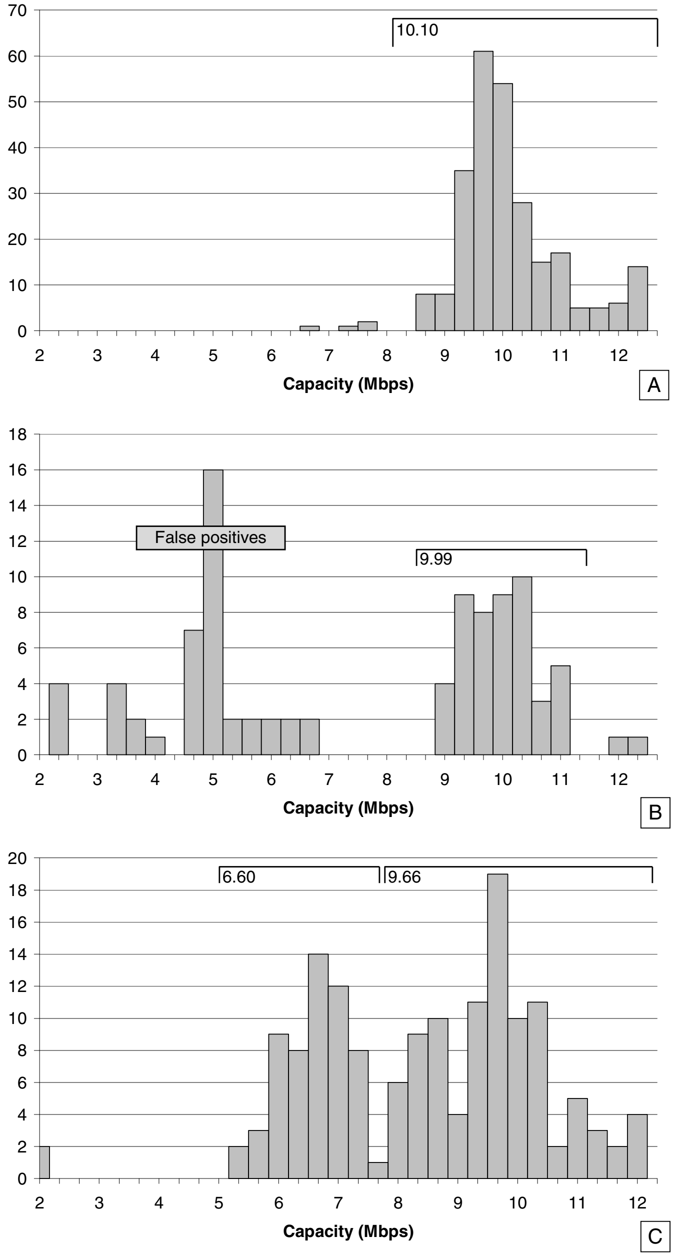

Fig. 7. Capacity distributions (10-Mbps fibre)

8.1 ADSL

Fig. 6 shows the histograms of the experiment performed on a 7-Mbps ADSL access line over the three traffic conditions previously described.

Experiment A (no interfering traffic) shows a unimodal distribution with a mean value of 7.02 Mbps, estimating the downlink capacity with a 0.3% error.

Experiment B (interfering traffic on the downlink) results in a quadrimodal distribution, with a mode placed on the actual capacity value (with a mode mean value of 6.88 Mbps, 1.7% error), and other three modes placed at about 2/3 (4.73 Mbps), at 1/2 (3.52 Mbps) and at 1/3 (2.34 Mbps) of the actual capacity value, consistently with the results presented in Section 6.1.

Experiment C (interfering traffic on the uplink) shows a strong mode with a mean value of 6.89 Mbps (1.57% error), and a number of samples that largely overestimates the correct capacity value, consistently with the results presented in Section 6.2.

8.2 Fibre

Fig. 7 shows the results of the experiments performed on a 10-Mbps symmetric fibre access line.

Experiment A (no interfering traffic) shows a unimodal distribution with a mean value of 10.10 Mbps (1% error).

Experiment B (interfering traffic on the downlink) exhibits a main mode with a mean value of 9.99 Mbps (0.1% error), and a noticeable peak around 5 Mbps (half of the actual capacity), which can be attributed to the false positives.

Experiment C (interfering traffic on the uplink) shows a main mode with a mean value of 9.66 Mbps (3.4% error), and a second mode at 6.60 Mbps, at about 2/3 of the actual capacity value. Further investigations explained the presence of such a strong 2/3-mode with a interfering traffic on the downlink composed of UDP packets generated by an IPTV service active on the access line.

9 Conclusion

We have presented a new method to measure the downlink access capacity in a passive way from a measurement point inside the network. The proposed method is an extension of the packet dispersion method. It differs from the traditional packet dispersion method in the fact that it is applied to longer TCP segment sequences. The application of the packet dispersion method to TCP segment sequences longer than two leads to a significant reduction of noise in the measurement.

We have started our analysis by introducing the Packet Burst (PB) concept, i.e., a sequence of TCP segments that are so close in time to each other that they need to be buffered in the access downlink queue. It is during these PBs that the model computes the access capacity. We presented an algorithm that exploits the linearity of the ACK passing times time during a PB in order to identify all the PBs during the measurement period. We have shown that it is possible to identify and drop the packet sequences affected by interfering traffic. In particular: 1) the interfering traffic on the downlink access queue is detected taking advantage of the fact that the packets related to the measured traffic and the packet related to interfering traffic can be organized only around a limited number of patterns; 2) the congestion on the uplink access queue can be detected because it induces the ACK compression phenomenon.

We have presented a set of experiments carried out on ADSL and fibre broadband access services to validate the proposed approach.

Future work will include large-scale characterization of Internet capacity of a large number of access lines from a measurement point located inside the network. Such a characterization will be possible thanks to the fact that the proposed method is highly scalable as it is passive and does not inject traffic in the network.

References

1. Prasad, R., Dovrolis, C., Murray, M., Claffy, K.: Bandwidth estimation: metrics, measurement techniques, and tools. IEEE Network 17(6), 27–35 (2003)
2. Dovrolis, C., Ramanathan, P., Moore, D.: Packet dispersion techniques and a capacity estimation methodology. IEEE ACM Trans. Netw. 12(6), 963–977 (2004)
3. Zhang, L., Shenker, S., Clark, D.D.: Observations on the dynamics of a congestion control algorithm: the effects of two-way traffic. SIGCOMM Comput. Commun. Rev. 21(4), 133–147 (1991)
4. Sinha, R., Papadopoulos, C., Heidemann, J.: Internet packet size distributions: Some observations., Tech. Report ISI-TR-2007-643 (May 2007)
5. Jacobson, V.: Congestion avoidance and control. In: Proc. of SIGCOMM 1988, New York, pp. 314–329 (1988)
6. Bolot, J.C.: Characterizing end-to-end packet delay and loss in the internet. Journal of High Speed Networks 2(3), 289–298 (1993)
7. Braden, R.: RFC 1122. Requirements for Internet Hosts – Communication Layers, Updated by RFCs 1349, 4379 (October 1989)
8. Croce, D., En-Najjary, T., Urvoy-Keller, G., Biersack, E.W.: Capacity estimation of ADSL links. In: Proc. of CoNEXT 2008, Madrid, Spain (2008)
9. Dischinger, M., Haeberlen, A., Gummadi, K.P., Saroiu, S.: Characterizing Residential Broadband Networks. In: Proc. of IMC 2007, San Diego, CA, USA (2007)
10. Casetti, C., Gerla, M., Mascolo, S., Sanadidi, M.Y., Wang, R.: TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links. In: Proc. of ACM Mobicom 2001, Rome, Italy (2001)
11. Fornasa, M., Maresca, M., Baglietto, P., Zingirian, N.: Passive Access Capacity Estimation for QoS Measurement. In: Proc. of IWQoS 2009, Charleston, SC, USA (2009)