On the Comparison of Packet-Pair and Packet-Train Measurements

Andreas Johnsson
The Department of Computer Science and Engineering
Mälardalen University
Sweden
Andreas.Johnsson@mdh.se

———-

Poster proposal for SNCNW2003

Abstract—**Measurements of link bandwidth and available bandwidth are getting increasingly important in the Internet. One example is to verify service level agreements (customer: "Do I get the bandwidth I'm paying for?").**

When developing accurate measurement methods based on active probing it is important to understand (at the packet level) how probe-packet flows and cross-traffic flows interact with each other. Several existing methods rely on packet-pair or packet-train methods. In this paper we investigate an effect that only packet trains can suffer from. Hence, we rise the thought that there might be a significant difference in measurement results when probing a network path using packet-pair and packet-train methods. We show, by an example, that the variance measure will give different results depending on the probing method.

Keywords— **Active measurement, performance, delay variation, packet trains, cross traffic.**

I. Introduction

Measurements of link bandwidth and available bandwidth on an end-to-end basis are getting increasingly important on the Internet. The measurements are done by injecting probe packets into the network and then use an algorithm to calculate link bandwidth or available bandwidth. There are two main techniques for collecting the data to be feed into the algorithms: packet-pair [1], [2], [3], [4], [5], [6] and packet-train [1], [7], [8], [9] methods. This poster will show how cross traffic is interacting with probe packets using both methods - and then ask the question whether the data from the two different methods actually can be used to measure the same characteristics. We have identified cross traffic effects on packet trains, not applicable to packet-pair techniques, that gives reason to believe that packet-train and packet-pair methods does not give the same information when collecting the same number of measurement points. Hence, more research needs to be done in this area.

II. Definitions

To analyze the behavior of probe packet trains, some mathematical definitions and derivations have to be made. In this paper, the analysis rely on a generic multiple-hop model, presented in [10]. That model focuses on expressing the delay variations of adjacent probe packets. However, in this paper only the basics of that model is needed, which is presented in the following subsection.

A. One-hop definitions

In what follows, the meaning of a hop is one router including its queue and the outgoing link from that router. This means that the arrival time of an arbitrary packet to hop h_y is equal to the departure time of the same packet from the previous hop h_x. Each router can have multiple outgoing and incoming links.

When a packet (i) arrives to the queue of hop h at time τ_i, it begins its service time $x_i > 0$ after a waiting time of $w_i \geq 0$. The packet leaves the hop after a constant propagation delay D at time τ_i^*. Thus, for the packet the one-hop delay is

$$d_i \equiv \tau_i^* - \tau_i = w_i + x_i + D. \quad (1)$$

In what follows, the index (i) corresponds to the indexing of probe packets.

Using the definition of the one-hop delay for one probe packet (Equation 1), we can compare the one way delay of two adjacent probe packets with each other. Three equations are derived (in [10])

$$\begin{aligned}
\text{inter-arrival time: } t_i &\equiv \tau_i - \tau_{i-1} \\
\text{inter-departure time: } t_i^* &\equiv \tau_i^* - \tau_{i-1}^* \\
\text{delay variation: } \delta_i &\equiv d_i - d_{i-1} \\
&= t_i^* - t_i \\
&= (x_i - x_{i-1}) + \\
&\quad (w_i - w_{i-1}).
\end{aligned}$$

When probing a network, sequences of t_i^* are measured at the receiver side. However, since t_i is part of the probe train design, and therefore known, the transformation between δ_i and t_i^* is easily done.

The waiting time of two successive packets in an infinite FIFO buffer is described by Lindley's equation

$$w_i = [w_{i-1} + x_i - t_i]^+ + c_i \tag{2}$$

where $[x]^+ = max(0, x)$. The term c_i is the waiting time caused by cross traffic entering the current hop between τ_{i-1} and τ_i.

III. Cross-traffic effects on probe streams

Methods that probes a network path are typically divided into two categories. Either a sequence of well separated packet pairs or a number of packet trains are injected into the network. The dispersion (i.e. separation) of the probe packets at the receiver side is then used in different bandwidth estimation algorithms (using, for example, dispersion mean values or dispersion variance). In the following subsections we will show that there is a difference between using dispersion values from packet-train and packet-pair methods.

A. Packet train

We have identified three different cross-traffic effects on packet trains: mirror effects, chain reactions and quantification effects. However, in this paper we only describe mirror effects.

A.1 Mirror effects

The mirror effect arises if a packet train consists of at least three probe packets $(i-1), (i)$ and $(i+1)$. Assume that packet $(i-1)$ and $(i+1)$ are unaffected by cross traffic (i.e. $w_{i-1} = w_{i+1} = 0$). Then, if packet (i) experiences a waiting time of $w_i > 0$, δ_i will get a positive value. Now, packet $(i+1)$ will have a delay variation described by $\delta_{i+1} = (x_{i+1} - x_i) + (w_{i+1} - w_i) = (w_{i+1} - w_i)$, since the two probe packets have the same size, i.e. the same service time. Of course, in the expression for δ_i the service times can also be eliminated. Since neither packet $(i-1)$ nor packet $(i+1)$ are delayed by cross traffic, the following holds

$$\begin{aligned}
\delta_{i+1} &= w_{i+1} - w_i \\
&= -w_i \\
\delta_i &= w_i - w_{i-1} \\
&= w_i \\
&\Longrightarrow \\
\delta_{i+1} &= -\delta_i . \qquad (3)
\end{aligned}$$

We define equation (3) as *perfect mirroring*. An example of perfect mirroring is shown in Figure 1. The vertical packets above the time line show when in time a probe packet (white box) or a cross traffic packet (shaded boxes) arrives at the router. The arc shows when in time all bits of a packet have been received. When all bits are received, the router can to start send the packet, which is shown below the time line. The probe packet (i) is delayed w_i time units and hence creates a perfect mirror effect, since the probe packets $(i-1)$ and $(i+1)$ are unaffected.

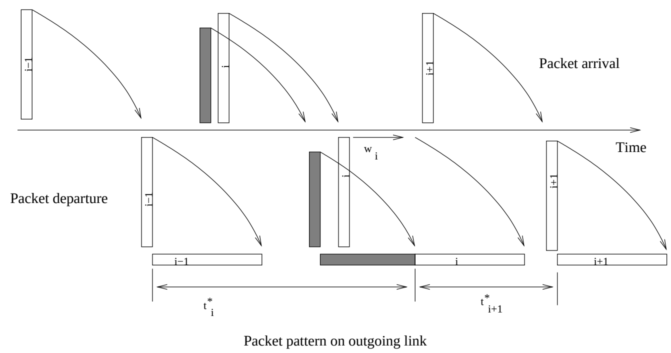

Fig. 1. Arrival and departure times for cross traffic (shaded boxes) and probe packets (white boxes) entering a router.

In addition to the fact that packet (i) can be delayed, there is a possibility that one, or both of the probe packets $(i-1)$ and $(i+1)$ are affected by cross traffic. This will of course blur the perfect mirror effect, i.e. equation (3) does not hold. It is obvious that this possibility grows with increasing cross traffic and/or probe send rate.

Exactly how Equation 3 is modified when more than one probe packet is affected is left out in this paper.

B. Packet pair

In packet pair methods well separated probe pairs are injected into the network. These pairs are independent of each other, and therefore do not affect each other. Hence, mirror effects can not occur. This means that the dispersion between two packets either increases or decreases depending on weather the first or the second probe packet was affected by cross traffic.

An example of when the separation increases can be seen in Figure 1 if the white probe packet $(i+1)$ is ignored. While, if probe $(i-1)$ is ignored the figure show when the dispersion between two packets in a pair decrease.

C. Comparison

The main difference between packet-train and packet-pair methods is that one cross-traffic packet affects at least two dispersion values in the train case[1], while in the packet pair case it only affects one dispersion value. In the packet train case the two dispersion values are dependent of each other. This might be exploitable, or perhaps may lead to mistakes in calculations.

An important difference between the two cases is when the variance σ^2 is calculated. Variance is important, for example, when creating models of cross traffic. We define the variance as

$$\sigma^2 = 1/n * \sum_{i=1}^{n}(t_i^* - \overline{t_i^*})^2$$

where n is the number of measured values and $\overline{x}$ is the mean. In the packet train case we refer to the perfect mirror example in the previous section, where one probe packet was affected by one cross traffic packet. We have two measured values, δ_i and δ_{i+1} where $\delta_i = -\delta_{i+1}$. The variance in this case will be $\sigma^2 = 1/2*((\delta_i^2 - (0)^2) + (\delta_i^2 - (0)^2)) = \delta_i^2$ (remember, $\delta_i = t_i^* - t_i$).

An comparable example using probe pairs is constructed by letting two well separated probe pairs traverse the network path. Then, let only one of the two probe pairs be affected by a cross traffic (only one cross traffic packet, equivialent to the packet-train case). The variance will be $\sigma^2 = 1/2 * ((\delta_i^2 - \overline{\delta^2}) + (0^2 - \overline{\delta^2})) = 1/2*((\delta_i^2 - (\delta_i/2)^2) + (0^2 - (\delta_i/2)^2) = 1/4*\delta_i^2$ since one of the pairs traverse the path unaffected (i.e. $\delta = 0$). That is, there is a difference when using packet pair and packet train techniques.

IV. Conclusions

In this paper the mirror effect on packet trains that are traversing a network path has been investigated. This effect does not apply to packet pair methods, which gave rise to the question whether packet pair and packet train methods really measure the same characteristics of cross traffic and link bandwidth. We have shown by a simple example that the variance is not the same when using input data from pairs and trains, respectively. These topics need further research, and hopefully discussions will arise when this information is displayed as a poster.

References

[1] Robert Carter and Mark Crovella, "Measuring bottleneck link speed in packet-switched networks," Tech. Rep. 1996-006, Boston University Computer Science Department, Boston, MA, USA, Mar. 1996.

[2] Constantinos Dovrolis, Parameswaran Ramanathan, and David Moore, "What do packet dispersion techniques measure?," in *Proceedings of IEEE INFOCOM*, Anchorage, AK, USA, Apr. 2001, pp. 905–914.

[3] Srinivasan Keshav, "A control-theoretic approach to flow control," in *Proceedings of ACM SIGCOMM*, Zürich, Switzerland, Sept. 1991, pp. 3–15.

[4] Kevin Lai and Mary Baker, "Measuring bandwidth," in *Proceedings of IEEE INFOCOM*, New York, NY, USA, Mar. 1999, pp. 235–245.

[5] Bob Melander, Mats Björkman, and Per Gunningberg, "Regression-based available bandwidth measurements," in *Proceedings of the 2002 International Symposium on Performance Evaluation of Computer and Telecommunications Systems*, San Diego, CA, USA, July 2002.

[6] Jiri Navratil and R. Les. Cottrell, "ABwE: A practical approach to available bandwidth estimation," in *Passive and Active Measurement (PAM) Workshop 2003 Proceedings*, La Jolla, CA, USA, Apr. 2003.

[7] Ningning Hu and Peter Steenkiste, "Evaluation and characterization of available bandwidth probing techniques," *IEEE Journal on Selected Areas in Communication*, 2003.

[8] Manish Jain and Constantinos Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput," in *Proceedings of ACM SIGCOMM*, Pittsburg, PA, USA, Aug. 2002.

[9] Manthos Kazantzidis, "How to measure available bandwidth on the Internet," Tech. Rep. 010032, Department of Computer Science, University of California, Los Angeles, Los Angeles, CA, USA, Sept. 2001.

[10] Attila Pásztor and Darryl Veitch, "The packet size dependence of packet pair like methods," in *Tenth International Workshop on Quality of Service (IWQoS 2002)*, Miami Beach, USA, May 2002.

[1] This holds if the cross traffi c packet did not affect the last packet in the train, then only one dispersion value will change.