Analysis of a Packet-Pair Scheme for Estimating Bottleneck Bandwidth in a Network[*]

Shikha Bahl
Department of Electrical Engineering

Ashok Agrawala
Department of Computer Science

University of Maryland
College Park, MD 20742, USA
{shikha, agrawala}@cs.umd.edu

September 4, 1998

Abstract

In order to assess the performance of a connection it is important to determine the bandwidth offered by the slowest node along the path, also known as *bottleneck bandwidth*. In the past, many researchers have used *a packet-pair* technique in order to estimate the bottleneck bandwidth. In real networks, however, the measurements made using the packet pair technique do not always reflect the correct estimate of the bottleneck service time due to the presence of cross traffic. While several reasons for the observed variability have been reported, the exact nature of the impact of cross traffic on the observations has not been studied.

In this paper we present a model to explain how the measured difference of the reception time for a packet-pair can be related to the characteristic of the service time and cross traffic that the pair found along the path.

Key words: Network Performance, Bottleneck bandwidth, Packet-pair,

---

[*] This work is supported in part by NLM under contract 467M2702491and DARPA/Army.ARO under contract DABT 6396C0075 to the Computer Science Department at the University of Maryland. The views, opinions, and/or findings contained in this report are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of NLM or DARPA.

1. Introduction

A good indicator of the performance of an end-to-end connection on a network is the achieved bandwidth, which depends on many factors including the basic characteristics of the path, cross traffic encountered, etc. The maximum possible bandwidth for a connection is determined by the characteristics of the path. We may consider the path that consists of a sequence of switches and links as a sequence of servers[1], each carrying out the store and forward operation. For such a path the maximum possible bandwidth is determined by the minimum bandwidth of a server. Such a server is often referred to as the *bottleneck* server. Therefore, in order to assess the performance of a connection it becomes important to determine the bottleneck bandwidth. Note, however, that the bottleneck bandwidth is the maximum achievable bandwidth which may not be achieved in practice due to several reasons, including the presence of cross traffic, i.e. the traffic carried by the nodes or the links along this path but which is due to other sources. Several researchers have used the techniques of *packet-pair* to estimate the bottleneck bandwidth[CC96, Ke91, Pa97]. The idea behind the packet pair technique is that, if two packets are transmitted by the sender with a very small inter-send interval (smaller than the service time of the bottleneck server), then, in the absence of any cross traffic their spacing will be exactly the service time of the bottleneck server, after completing processing through the bottleneck node. Note that knowing the packet size and the bottleneck service time we can easily determine the bottleneck bandwidth. In real networks, however, cross traffic is always present. As a consequence the measurements made using the packet pair technique do not always reflect the correct estimate of the bottleneck service time. Most researchers have observed this fact and some researchers have used a variety of filtering techniques to improve their estimates [CC96, Pa97]. While several reasons for the observed variability have been reported, the exact nature of the impact of cross traffic on the observations has not been studied.

In this paper we present a model to explain how the measured difference of the reception time for a packet pair can be related to the characteristic of the service time and cross traffic that the pair found along the path. We begin our discussion by defining notations in section 2 and formulating the problem in section 3. We develop a model for a single node with cross traffic in section 4. This enables us to illustrate the effect of cross traffic on the inter-departure time of the packets as they go through a sequence of nodes. We extend the analysis to N nodes in section 5. Before discussing concluding remarks in section 7, we discuss the implications of our model on the use of the packet-pair technique as used by various researchers in section 6.

2. Notation

We use a superscript to identify a node and a subscript to identify a packet. We, however, drop the superscript when it can be done without *causing* any confusion.

a_j^i : arrival time of the jth packet at the ith node

---

[1] In this paper we use the term node and server interchangeably.

d_j^i : departure time of the jth packet from the ith node

s_j^i : service time of the jth packet at the ith node

s^b : service time of the bottleneck node

$v^i(t)$: virtual waiting time at the ith node at time instant *t*.

c^i : additional delay seen by second packet of the pair due to cross traffic at the ith node

A^i :inter-arrival distance between the two packets at the ith node.

D^i : inter-departure distance between the two packets at the ith node.

3. Problem Formulation

Let us consider an end-to-end connection from host A to host B via nodes *1* through *N* connected through direct connections between the nodes as shown in Figure 1. A packet starts from the host A and follows the path shown here to the destination B. All the subsequent packets follow the same path. Note that a server here may represent a link or a switch. The packets are processed at each node using a FIFO discipline. Further, a node processes packets using a work-conserving discipline that is if there is any packet waiting to be served the node can not go idle.

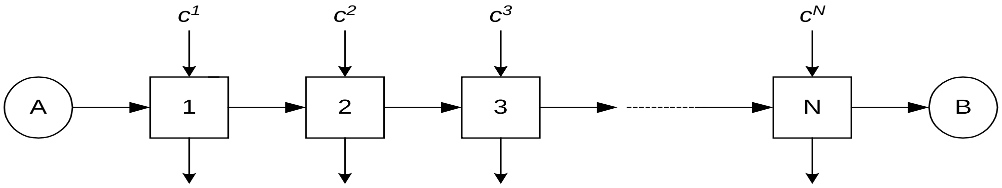


Figure 1: Flow of traffic between a pair of hosts along a given path

The nodes may also process packets that either originate elsewhere or are destined for other hosts. Such packets which are related to other source and destination host pairs, but which are processed by one or more intermediate nodes along the path, are treated as *cross traffic*. Note that each node along the path is an independent server, which may process the packets at its own speed. We, however, assume that the packets along the primary path have the *same* processing time s^i at node *i*. The server *b*, which takes the longest to process a packet, is the bottleneck server, taking s^b seconds to process the packets of the primary path. The basic *problem we want to study is that of determining the bottleneck service time, s^b by observing* the inter-departure times of packets. In the packet pair technique we send a pair of packets from host A with a known time interval between them, and measure the time interval between their arrival at host B. In a typical experiment we may send a series of such packet pairs and use the observations to estimate the values for s^b.

4. Single-Node Analysis

Let us consider a single node and two arrivals. Packet 1 arrives at the server at time instant a_1 and requires service for time duration s. Let $v(a_1)$ [2]be the virtual waiting time that a packet experiences at time instant a_1 (i.e., the pack t had to wait for $v(a_1)$ amount of time before being served by the server).

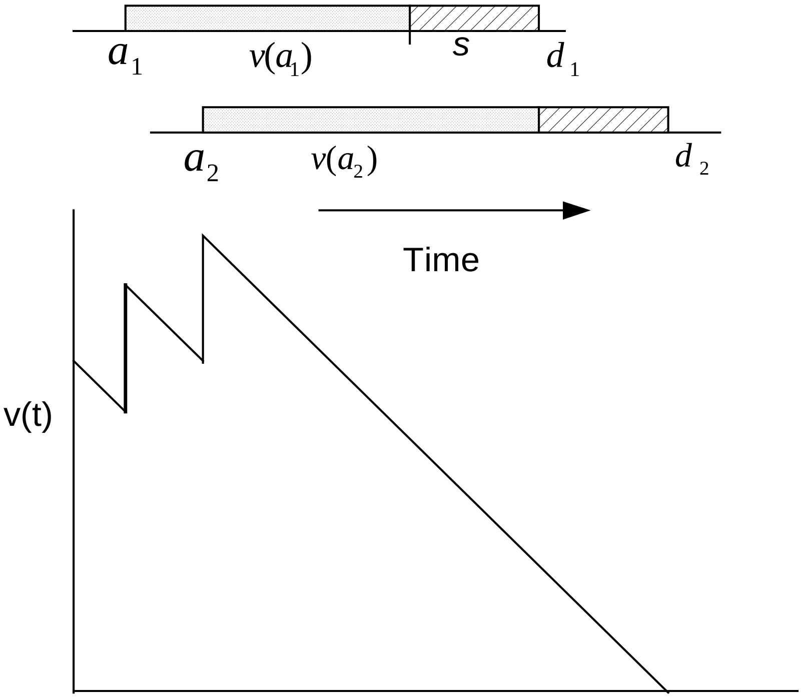


Figure 2: Packet Timing with no cross traffic

Then, due to the FIFO discipline followed by the server, it will depart the server at time instant d_1, where $d_1 = a_1 + v(a_1) + s$. Similarly, the second packet arrives at a_2 and departs the server at time instant d_2, where $d_2 = a_2 + v(a_2) + s$.

Let us define $u = Max[v(a_1) + s - (a_2 - a_1), 0]$. Note that if no cross traffic occurs between a_1 and a_2 then, virtual waiting time at time instant $a_2, v(a_2) = u$. Clearly, the impact of cross traffic that occurs between the arrivals of these two packets is in terms of the additional delay, if any, that is sustained before the processing of the second packet begins. We use the term *c* to indicate this cross traffic. Note that this term does not represent the true cross traffic; when some packets arrive and get processed before a_2, the

---

[2] Note that the virtual waiting time *v*(t) is a non-negative function of time, representing the amount of time a packet has to wait before its servicing can begin if it were to arrive at time *t*. In between the arrival, the virtual waiting time decreases at the rate of one second per second. At the arrival instant it jumps up by the service time for the newly arriving packet.

arrival instant of the second packet, there will be no additional delay seen by the second packet resulting in c and u to be zero.

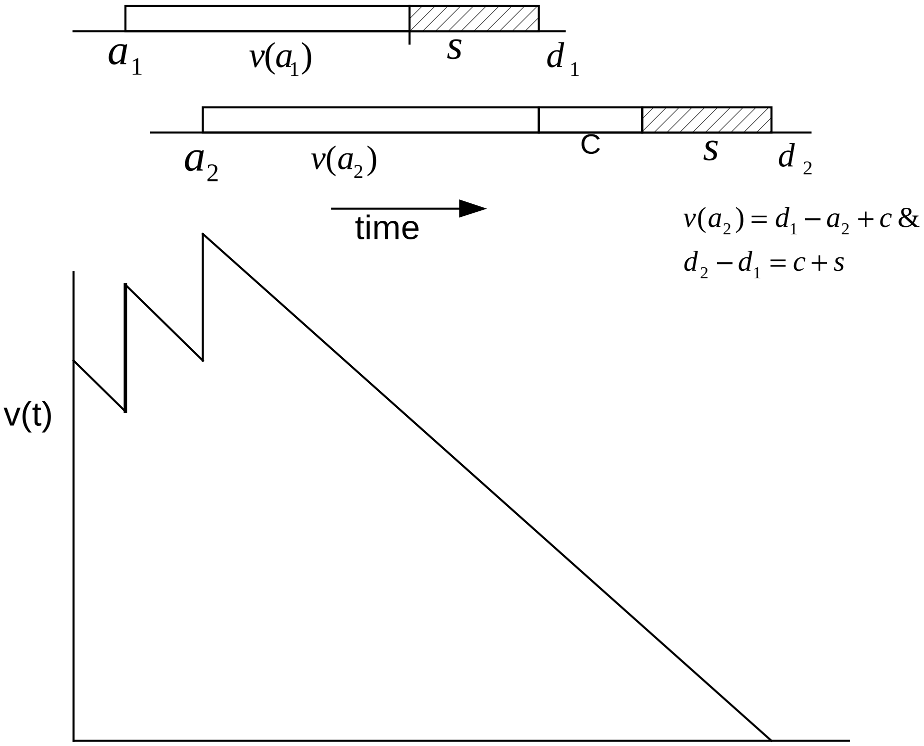
In general, $v(a_2) = u + c$

$= Max[v(a_1) + s - (a_2 - a_1),0] + c$

$= Max[v(a_1) + s + a_1 - a_2),0] + c$

$= Max[(d_1 - a_2),0] + c$ (1)

Note that, when the second packet arrives before the departure of the first packet, (see Figure 3) i.e., $a_2 < d_1$, then $v(a_2) = d_1 - a_2 + c$.



$v(a_2) = d_1 - a_2 + c$ &

$d_2 - d_1 = c + s$

Figure 3: Packet timing with $a_2<d_1$

When the second packet arrives after the departure of the first (see Figure 4), then $v(a_2) = c$.

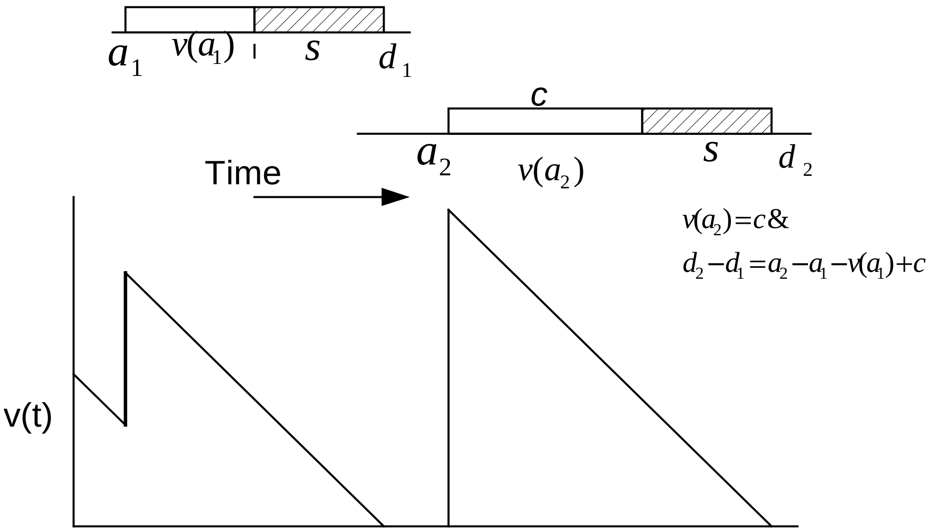


Figure 4:Packet timing with $a_2 > d_1$

Now, let us look at the inter-departure time, *D*, of the two packets.

$$
\begin{aligned}
D = d_2^1 - d_1^1 &= [a_2 + v(a_2) + s] - [a_1 + v(a_1) + s] \\
&= [u + c] - v(a_1) + A \\
&= Max[v(a_1) + s - A, 0] + c - v(a_1) + A \\
&= Max[s, A - v(a_1)] + c \qquad (2)
\end{aligned}
$$

From this equation we note that *D* can take two values. When there is no cross traffic (*c=0*), then *D* is either equal to *s* or *A*-*v*(a₁), whichever is larger. In the presence of cross traffic the value of *D* is increased by, *c*.

In the packet pair technique, a pair of packets are sent back to back, leading to a small value for *A*. Clearly, when there is no cross traffic, both *v*(a₁) and *c* will be zero, resulting in *D* = *s*. The presence of cross traffic, however, can distort this estimate even for a single server by making *D* = *s* + *c*.

Another way to write the expression for *D* is:

$$
D^1 = \begin{cases} s + c^1, & a_2 < d_1 \\ c^1 - v^1(a_1) + A & a_2 > d_1 \end{cases} \qquad (3)
$$

When the first packet departs after the arrival of the second packet, the inter-packet distance is the service time, *s*. When the first packet departs before the arrival of the second packet the inter-departure time is governed by the inter-arrival time as adjusted by the virtual waiting time found by the first packet. Cross traffic term affects both these situations. Clearly, the smallest inter-departure time from any server will be the service time of that server, irrespective of the inter-arrival time and the virtual waiting time found on the server by the first packet. Also as the virtual waiting time of the first packet increases (irrespective of the second packet) inter-packet distance may decrease, and it increases with the cross-traffic.

5. N Node Analysis

Next, let us consider the flow of traffic between the pair of hosts through N nodes. For N nodes in succession, departure time from the *previous* node becomes the arrival time at the next node, i.e., $d_j^i = a_j^{i+1}$.

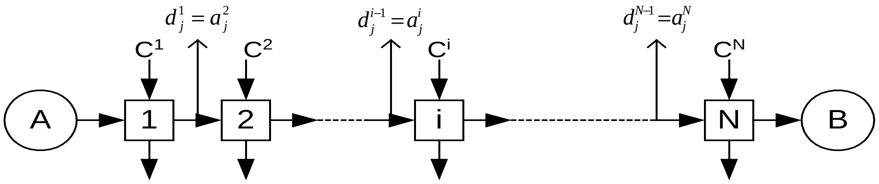


Figure 5: Flow of traffic through N nodes in tandem

Let us define $f^i = c^i - v^i(a_1)$ and let $A^j = (a_2^j - a_1^j)$. From above, we know that the inter-packet distance after the first node is:

$$\begin{aligned} D^1 &= Max[s^1, A^1 - v^1(a_1)] + c^1 \\ &= Max[s^1 + c^1, A^1 - v^1(a_1) + c^1] \\ &= Max[s^1 + c^1, A^1 + f^1] \qquad (4) \end{aligned}$$

Similarly, inter-packet distance after the second node is: $D^2 = Max[s^2 + c^2, A^2 + f^2]$

But, $A^2 = D^1$, so D^2 becomes

$$\begin{aligned} D^2 &= Max[s^2 + c^2, Max[s^1 + c^1, A^1 + f^1] + f^2] \\ &= Max[s^2 + c^2, s^1 + c^1 + f^2, A^1 + f^1 + f^2] \qquad (5) \end{aligned}$$

As shown here, inter-departure distance D^2 after the second node is the maximum of three terms. The first two terms represent the service time of the server, adjusted by the cross traffic encountered at that server. In addition, the second term is adjusted by the f^2 term. As noted above, the inter-departure time *increases* due to the cross traffic and decreases due to the virtual waiting time encountered by the first packet. The f terms represent the adjustments due to these two factors as encountered at the downstream servers. The last term shows the adjustments to be made to the inter-arrival time, A^1, at the two servers.

Note that as $f = c - v$ and that c and v terms can only be non-negative, f can be positive or negative. As a consequence, depending on the cross traffic and the conditions encountered by the first packet, any of the three terms can result in D^2 and be the maximum term.

Similarly, the inter-departure distance after three nodes is given by the following equation.

$$\begin{aligned} D^3 &= Max[s^3 + c^3, D^2 + f^3] \\ &= Max[s^3 + c^3, Max[s^2 + c^2, s^1 + c^1 + f^2, A^1 + f^1 + f^2] + f^3] \\ &= Max[s^3 + c^3, s^2 + c^2 + f^3, s^1 + c^1 + f^2 + f^3, A^1 + f^1 + f^2 + f^3] \qquad (6) \end{aligned}$$

And after the fourth node is

$$\begin{aligned} D^4 = Max[&s^4 + c^4, s^3 + c^3 + f^4, s^2 + c^2 + f^3 + f^4, s^1 + c^1 + f^2 + f^3 + f^4, \\ &A^1 + f^1 + f^2 + f^3 + f^4] \qquad (7) \end{aligned}$$

As a generalization, the inter-packet distance after N nodes is:

$$D^N = Max[s^N + c^N, [s^{N-i} + c^{N-i} + \sum_{k=N-i+1}^{N} f^k]_{i=1}^{i=N-1}, A^1 + \sum_{k=1}^{N} f^k] \qquad (8)$$

The inter-packet distance after *N* nodes is maximum of *N+1* terms. The i^{th} term is based on its service time adjusted by the cross traffic and the conditions found at the downstream node. The last term is based on the inter-arrival time, modified by the conditions encountered at the nodes along the path.

5.1. Implications of the result

5.1.1. Case 1: No Cross-traffic

In the absence of cross traffic, all $f^i \,\&\, c^i$ terms go to zero and the inter-packet distance after the N^{th} node becomes $D^N = Max[s^N, [s^{N-i}]_{i=1}^{i=N-1}, A^1]$. From this equation, we can infer that the inter-packet distance after *N* nodes *will* reflect the service time of the bottleneck node provided the inter-arrival distance of the packets at the first node is less than the service time of the bottleneck node.

From the way packets are serviced in this case, the inter-departure time from a node is always the maximum of the inter-arrival time and the service time at this node. When packets are processed at a sequence of nodes, their inter-departure time is governed by either the inter-arrival time at the first node or the maximum service time seen along the path so far(see Figure).

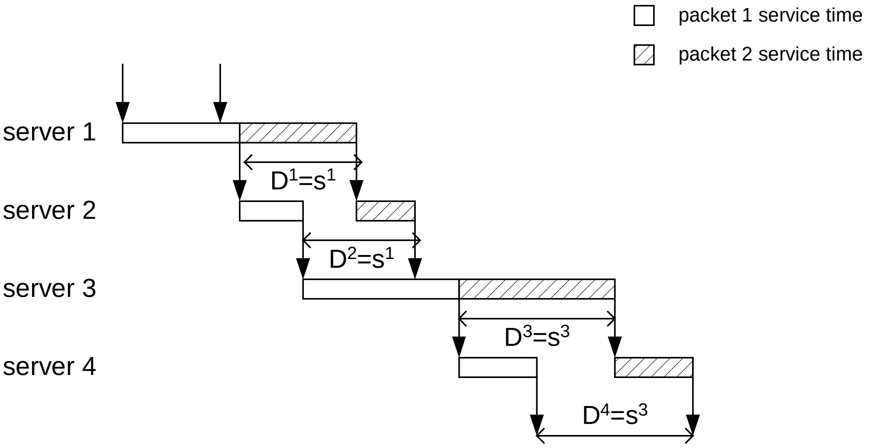


Figure 6 :Processing of packets without cross traffic.

In the Figure 6, packets are processed through a sequence of four servers. We observe that in the absence of cross traffic inter-departure time after a server is the maximum service time seen along the path so far. For example, the inter-packet distance after the second server, D^2, is equal to the service time of the first server, s^1, and not s^2, because $s^1 > s^2$. Similarly, the inter-departure distance after the fourth server, D^4, is equal to the service time of the third server since $s^3 > s^4$ and $s^3 > s^1$.

5.1.2. Case 2: With Cross-traffic

When we consider the path of *N* nodes, with cross traffic (Eq. 8), we note that the observed D^N is the maximum of $N+1$ terms. The first term is the sum of the service time and cross traffic encountered at the N th node. Of next $N-1$ terms, each i^{th} term is the service time s^i adjusted by c^i and one *f* term for each of the nodes from $i+1$ to *N*. The last term is based on A^i adjusted by *f* terms for each of the servers. Note *that* c^i and $v(a)$ are non-negative terms, but *f* terms can be positive or negative. As a consequence, depending on the cross traffic any one of the N+1 terms can become the largest and hence determine D^i. Further, in the presence of cross traffic D^i can hardly be considered a good estimate of s^b.

From the way packets are serviced in this case, the inter-departure time from a node is always the maximum of the inter-arrival time and the service time at this node. When packets are processed at a sequence of nodes, their inter-departure time is governed by either the inter-arrival time at the first node or the maximum service time seen along the path so far.

A graphical representation of this statement is shown in Figure 7. In presence of cross traffic, inter-departure time is no longer the maximum service time seen by the packets along the path. For example, $D^3 \neq s^3$ but, in fact, is larger than s^3 due to non-zero cross traffic. Server 3 not only has the largest service time, but also has some cross traffic, so, the inter-departure distance after the fourth server is much larger than the service time of the third server, s^3.

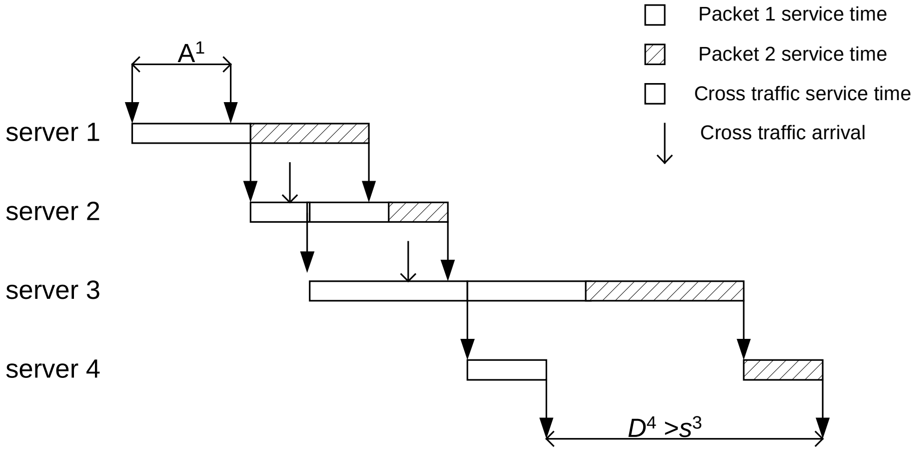


Figure7 :Processing of packets with cross traffic.

6. Using the Packet-pair Technique in Experimental Studies

The Packet pair technique has been used by several researchers [CC96, Ke91, Pa97] in their studies of Internet performance, and, in particular, for determining bandwidth offered by the slowest node along the path. The primary approach in these studies has been to send packet pairs in a sequence and measure D^Ns from which the estimate of the bottleneck bandwidth is determined. A common observance in all these experimental studies was that the measured values of D^N showed a significant variability when a sequence of packet pairs was sent. Each author tried to forward several hypotheses regarding the causes of the variability, and used various filtering and estimation techniques to come up with better estimates of the bottleneck bandwidth [CC96, Ke91].

As we have seen from Eq. 8, that the observed value of D^N is maximum of $N+1$ terms and depending on the cross traffic condition the value can be much larger than s^b. The only condition under which $D^N= s^b$ is when no cross traffic is present anywhere along the path. Clearly, the kind of variability observed in the experimental evaluation is primarily due to cross traffic present along the path.

Vern Paxson, in his study, used packet-pair at the TCP level and measured the inter-packet distance at the receiver (RBPP) as well as inter-packet distance for *acks* at the source [Pa97]. He observed a much larger variability at the source compared to that observed at the receiver. Paxson attempted to improve the estimate of the bottleneck bandwidth by sending a stream of packets in a bunch and observed their arrivals at the receiver. He then, used heuristic techniques to filter the raw measurements and develop the estimates for the bottleneck bandwidth.

Let us consider the path from a sender to a receiver, which consist of a set of nodes along the forward path. The return path from the receiver to the sender as used by the acks may or may not follow the same set of nodes. However, the path from the sender to the receiver clearly has a smaller number of nodes compared to the path from the sender, via receiver, back to the sender. From the Eq. 8, we note that as the number of nodes along the path increases, more effects of cross traffic are likely to be observed. It is not surprising that a larger variability is observed in sender-based measurements because of the larger number of nodes along the path. Clearly, in this analysis, the receiver has to be considered as one of the nodes. Because it does not carry out the store-and-forward operations, servicing time at the receiver has to include the time to process the packet and generate the acks.

While working on the design of flow control schemes using control theoretic formulation, Keshav [Ke91] used packet-pairs. He also observed a significant variability in the measured values of D^Ns, and accounted for this variability as observation noise in his formulations.

Carter and Crovella used packet pairs to provide an estimate of the bandwidth between two servers [CC96]. They developed a tool, *bprobe*, to provide an estimate of the bottleneck bandwidth by sending a sequence of 10 ICMP echo packets, of varying sizes and observed their arrivals. They used heuristic techniques to develop their estimates. Their measurements also show the variability which is most likely due to the cross traffic.

Bolot formulated a model in which he sent a stream of UDP packets at regular intervals [Bo93]. In order to study the end-to-end packet characteristic in the Internet he measured the inter-arrival time difference at the sender and formulated a mathematical model to obtain the bottleneck bandwidth. He also observed a significant variability in D^Ns and used a phase plot of the round trip time of two adjacent packets in the sequence to estimate the bottleneck bandwidth.

Jain et al considered a model very similar to the model we have used, except he used a fluid approximation for cross traffic [JAS92]. As a consequence, in the time duration a_1 and a_2, the total cross traffic that arrived is $\rho(a_2-a_1)$ where ρ is the rate at which cross traffic arrives at a server. Using this model he showed that the inter-packet arrival distance could be modeled as a piecewise-linear function. It is not surprising that his experimental studies also showed a significant variability. His model was developed in an idealized condition that the cross traffic strictly followed the fluid flow model. In practice, however, this assumption does not hold, giving the variability he observed.

7. Concluding Remarks

In this paper we presented a model for the inter-departure time for packets when they are processed by a sequence of store and forward nodes. This model takes into account the cross traffic at the nodes and shows that the only condition in which we can be assured that the inter-departure time is equal to the service time at the bottleneck node is when there is no cross traffic present. In the presence of cross traffic the inter-departure time is significantly effected.

The model in this study used a single server at each node. If a pair of switches have several links connected between them, they have to be treated as multiple servers. While our analysis does not extend to that case directly, we believe that the appropriate extensions of the model will explain the observed behavior. Also, our model assumes a FIFO discipline and does not support any reordering of packets.

In our analysis, we assume that the time can be measured with arbitrarily high precision. However, in practice, the clocks used have finite precision and therefore an appropriate clock discretization error (bounded b clock interval is added to all of the measurements.

8. Reference

[Bo93] J.C. Bolot, “End-to-End Packet Delay and Loss Behavior in the Internet,” *Proceedings of SIGCOMM ’93*, pp. 289-298, September 1993.

[CC96] R. Carter and M. Crovella, “Measuring Bottleneck Link Speed in Packet-Switched Networks,” *Performance Evaluation*, Vol.27-8, pp.297-318, Oct. 1996.

[JAS92] B.N. Jain, A. Agrawala, and D. Sanghi., “Model and Analysis of a Virtual Circuit with Cross Traffic”, *Journal of Internetworking*,Vol. 4, No. 2, pp. 81-98, June 1993.

[Ke91] S. Keshav, “A Control-Theoretic Approach to Flow Control,” *Proceedings of SIGCOMM ’91*, pp.3-15, Sep. 1991.

[Pa97] V. Paxson, “Measurements and Analysis of End-toEnd Internet Dynamics,” Ph.D. dissertation, University of California, Berkeley, April 1997.