# Estimating Achievable Throughput

Lili Wang, James Griffioen, Kenneth L. Calvert
Laboratory for Advanced Networking
University of Kentucky
http://protocols.netlab.uky.edu/ lwang0/metaverse.html

## 1. INTRODUCTION

A growing number of applications have begun to leverage the flexibility, control, and robustness offered by overlay and grid networks. For example, to achieve better throughput or reliability between a source and destination than the throughput or reliability offered by the standard IP shortest path route, applications have been turning to application-selected overlay paths. Unfortunately, the underlying Internet over which overlays are constructed offers very little information to help applications make such routing decisions. In particular, it is difficult to obtain accurate information about the path capacity (of "virtual" overlay links), the current traffic level across a path, the available bandwidth along a path, or the expected throughput a transport protocol can expect to get if it were to use the path. Because many applications can benefit from information about the available bandwidth along an overlay path, a variety of available bandwidth estimation techniques have been developed – where available bandwidth is defined as the unused capacity along a path. Examples include Spruce, IGI, Delphi, ABwE, Pathload.

A common technique used by many of these tools is the packet pair dispersion technique, in which a sender sends successive probing packets to a receiver using a known time gap between packets and then uses the observed change in the time gap at the receiver to estimate the available bandwidth. The accuracy of this technique depends on whether the queue at the bottleneck router becomes empty between the departure of the first packet and the arrival of the second packet. Thus, several algorithms have been designed that attempted to reduce the impact of invalid probing measurements.

Moreover, several available bandwidth estimation techniques rely on knowledge of the path capacity – which must itself be estimated. Although there exists many tools to estimate path capacity, our analysis of these tools shows that they are highly inaccurate or only work under certain conditions. Using inaccurate capacity values as the basis for one's available bandwidth estimate simply compounds the errors and inaccuracies. Furthermore, the capacity tools estimate the "narrowest" link capacity (i.e., the link with the smallest capacity) while available bandwidth estimates need to know the capacity of the most congested link.

We present a new approach to bandwidth estimation that extends the packet gap model to include additional information and does not rely on (inaccurate) path capacity estimates. Unlike past approaches, our approach incorporates information about the end-to-end queuing delays to explicitly analyze how competing traffic interacts with probe traffic. In addition, our proposed technique does not incur the high overhead associated with techniques that must iteratively search for an optimal interpacket gap.

Another key feature of our approach is that it attempts to predict the *achievable throughput* rather than the *available bandwidth* as this is more often the information desired by the application. Assuming the application-level transport protocol operates in a TCP-friendly manner, the throughput achieved by the transport protocol can be different from the available bandwidth, because the throughput of TCP-like protocols depends on several factors including loss rate, round trip time and the type of competing traffic. In addition, as networks become more complex with rate limiters becoming common, it is possible that the path capacity and the available bandwidth are in no way related to the achievable throughput.

## 2. PACKET PAIR GAP MODEL

Packet pair dispersion techniques estimate available bandwidth by measuring the gap change between two probe packets. If the queue at the bottleneck router does not become empty between the departure of the first packet and the arrival of the second packet, the output packet gap includes the transmission time of the first probe packet along with transmission time of all background traffic. Given the path capacity and the size of the probing packet, the amount of competing traffic can be easily estimated.

However, the above condition may not be true during the probing period. Under other condition, observed change in interpacket gap may have little relationship with the actual cross traffic that occurred during the probe period. Knowing this, several techniques have been developed to avoid using such probes in the available bandwidth estimation com-

*SIGCOMM* '05 Poster Session

putation. For example, some approaches try to carefully controlling the interpacket gap value to ensure the probes meet the condition needed to be “good/useful” probes. Unfortunately, it is impossible to avoid “bad” probes. Other approaches try to identify and discard “bad” probes using statistical methods which have their own error factors. Both approaches ignore the information about competing traffic that is available from certain “bad” probes. Our approach avoids these problem by using the knowledge of both queuing delay and interpacket gaps to accurately distinguish the different cases and then deals with each case in a proper way.

Another problem of the packet pair model is that they just use the gap value to estimate the competing traffic and ignore the queuing delay of the probing packets. In fact, queuing delay can not only help distinguish the different cases, it can also be used to estimate the competing traffic. If we just ignore the amount of background traffic which does not interleave with the probing packets nicely, it may under-estimate the amount of background traffic during the probing period. Although the output gap values obtained in bad cases might not reflect the amount of background traffic, the queuing delay can be used to infer the competing traffic during these probing periods.

## 3. OUR APPROACH

We are designing a bandwidth estimation technique that extends existing packet dispersion techniques by incorporating information about end-to-end queuing delay. In particular, by observing end-to-end delays over time, we can identify the minimal (i.e., optimal) end-to-end delay corresponding to the case in which a packet experienced no queuing delay anywhere in the network. By comparing the end-to-end delays of the probe packets against the minimal delay, we can classify packet pairs into one of five different cases. Each case corresponds to one possible queuing scenario at the bottleneck router. Like Spruce and IGI, our current work assumes a single bottleneck.

Figure 1 shows the different ways in which the competing traffic affects the output gap values. The left graph shows the queue state when the first packet arrives at the router. The right graph shows the queue state when the second packet arrives at the router.

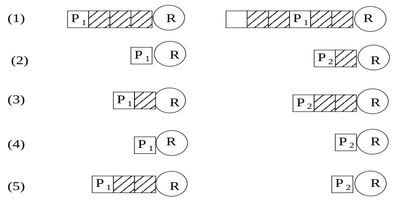

**Figure 1: Output gap analysis**

We classifies five cases to three categories.

1. All cross traffic, if any, is captured by the two probing packets. This case occurs if the first probing packet does not departure from the router before the arrival of the second packet. The output gap value is the transmission time of the first probing packet plus the transmission time of the background packets inserted between the two probing packets. This case is shown in Figure 1.
2. Partial cross traffic is captured by the two probing packets. The first packet departures from the router before the arrival of the second probing packet. In case (2) from Figure 1 the first packet is not delayed by cross traffic, while in case (3), the first packet is delayed by cross traffic. We do not know whether any transmission of background packets occur during the period from the departure of the first packet to the arrival of the second packet. Given this uncertainty, we use a parameter to control what percent of the unknown period (from the departure of the first packet to the arrival of the second packet) is assumed to transmit background packets. The parameter is dynamically adjusted based on the current load.
3. No cross traffic is captured by the two probing packets. The first packet departures from the router before the arrival of the second probing packet and the second packet arrives at an empty queue. This is case 4 and case 5 shown in Figure 1.

We determine whether the first probing packet departures from the router before the arrival of the second packet by checking whether the queuing delay of the first packet is less than the input gap value. To decide whether a probing packet is delayed, we check whether the queuing delay of the probing packet is greater than a threshold value.

To get a random “look” at a path, we send a sequence of packet pair probes using a Poisson distribution. The threshold value mentioned above depends on a (scaled) estimate of the path’s capacity, $C$, which corresponds to the maximum path throughput rather than the capacity. The value $C$ is discovered in an offline process that iteratively refines $C$ to match the observed throughput (which only needs to be measured once). In cases where multiple measurements of the throughput are available, we plan to use (offline) linear regression analysis to update the scaled capacity, $C$.

We classify the probing measurements using the above classification scheme and then estimate the path load. Then we estimate the achievable throughput based on the path load and the scaled capacity, $C$. This is shown in Figure 2.

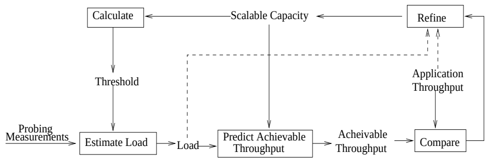

**Figure 2: Design**

We have tested our tool using Emulab, Planetlab, and regular Internet nodes. Our experiments show that our techniques accurately estimate the achievable throughput in a wide range of environments. Experimental results are available on our web page.