Packet dispersion techniques and capacity estimation

Constantinos Dovrolis
University of Delaware
dovrolis@cis.udel.edu

Parameswaran Ramanathan
University of Wisconsin
parmesh@ece.wisc.edu

David Moore
CAIDA
dmoore@caida.org

ABSTRACT

The packet pair technique estimates the capacity of a path (bottleneck bandwidth) from the dispersion (spacing) experienced by two back-to-back packets [5][19][13]. It has also been claimed that the dispersion of longer packet bursts ('packet trains') can measure the available bandwidth of a path [5][9][3]. This paper examines such packet pair and packet train dispersion techniques. We first demonstrate that the dispersion of packet pairs in loaded paths follows a multimodal distribution, and discuss the queueing effects that cause multiple local modes. The path capacity is not necessarily the global mode, and so it cannot be estimated using statistical procedures for the most common bandwidth range. The effect of the probing packet size is also investigated, showing that the conventional wisdom of using maximum sized packet pairs is not optimal. On the contrary, if the probing packets of different packet pairs are of variable size, the sub-capacity local modes become wider and weaker.

We then focus on the dispersion of long packet trains. Increasing the length of the packet train reduces the measurement variance, but the estimates converge to a value, referred to as Asymptotic Dispersion Rate (ADR), that is lower than the capacity, and that is not related with the available bandwidth. We show the effect of cross traffic on the dispersion of long packet trains, and derive analytic expressions for the ADR in certain path configurations.

Putting all pieces together, we present a capacity estimation methodology that has been implemented in a measurement tool called *pathrate*. We present capacity measurements of several Internet paths using *pathrate*, and evaluate its accuracy and its robustness to cross traffic effects.

Keywords: Bandwidth estimation, bottleneck link, packet pair, packet train, available bandwidth, network measurements, multimodal distributions.

1 Introduction

The Internet is a commercial infrastructure in which users pay for their access to an Internet Service Provider (ISP), and from there to the global Internet. It is often the case that the performance level (and tariff) of these network connections is based on their bandwidth, since more bandwidth normally means higher throughput and better quality-of-service to an application. In such an environment, *bandwidth monitoring* becomes a crucial operation. Users need to check whether they get the access bandwidth that they have paid for, and whether the network 'clouds' that they use are sufficiently provisioned. ISPs also need bandwidth monitoring tools in order to plan their capacity upgrades, and to detect congested or underutilized links [4].

Parts of this paper have previously appeared at an Infocom 2001 publication [6]. This work was supported in part by the USENIX association and by the National Science Foundation under Grant No. NCR-9711092.

Network operators are increasingly using tools such as MRTG [17] to monitor the utilization of their links with information obtained from the router management software. These techniques are based on statistics maintained by the routers, and they are normally accurate. Their drawback, however, is that they can be performed only with access to the router, and such access is usually limited to network managers. Instead, in this paper we focus on an *end-to-end bandwidth monitoring* approach that requires the cooperation of only the path end-points. Even though end-to-end approaches are usually not so accurate as router-based methodologies, they are often the only feasible approach for monitoring a path that crosses several networks.

We define a network path as the sequence of store-and-forward links that transfer packets from the sender (*source*) to the receiver (*sink*). This definition assumes that the path is fixed and unique, i.e., no routing changes or multipath forwarding occur during bandwidth monitoring. Two bandwidth metrics that are commonly associated with a path are the *capacity* C and the *available bandwidth* A. *Capacity* is the maximum throughput that the path can provide to a flow, when there is no competing traffic load (cross traffic). *Available bandwidth*, on the other hand, is the maximum throughput that the path can provide to a flow, given the current cross traffic load in the path. The link with the minimum transmission rate determines the capacity of the path, while the link with the minimum spare capacity limits the available bandwidth. To avoid the term *bottleneck link*, that has been widely used for both metrics, we refer to the capacity limiting link as the *narrow link*, and to the available bandwidth limiting link as the *tight link*.

Specifically, if H is the number of hops (links) in the path, C_i is the transmission rate or *capacity* of link i, C_0 is the transmission rate of the source, then the capacity of the path is

$$C = \min_{i=0 \ldots H} C_i \tag{1}$$

Additionally, if u_i is the *utilization* of link i (with $0 \leq u_i < 1$ and u_0=0) over a certain averaging interval, the spare capacity in link i is $C_i(1 - u_i)$, and so the available bandwidth of the path in the same interval is

$$A = \min_{i=0 \ldots H} [C_i(1 - u_i)] \tag{2}$$

This available bandwidth definition assumes stationary traffic and it is based on a fluid-based model of the traffic load. The capacity and available bandwidth metrics are further discussed in Appendix I.

The *packet pair technique* is a well-known technique for measuring the capacity of a path. When a packet is transmitted in a link, it encounters a *transmission or serialization delay* due to the physical limitations of the link and the hardware constraints of the transmission equipment. In a store-and-forward link of capacity C_i, the transmission delay for a packet of size L is $\tau_i = L/C_i$. A packet pair experiment consists of two packets sent back-to-back (i.e., with a spacing that is as short as possible) from the source to the sink. Without any cross traffic in the path, the packet pair will reach the receiver with a spacing, or *dispersion*, that is the transmission delay in the narrow link $\tau_n \equiv L/C$. The receiver can calculate the capacity C from the measured dispersion Δ, as $C = L/\Delta$. Figure 1 illustrates the packet pair technique in the case of a three-link path, using the fluid analogy introduced in [10]. Even though simple in principle, the packet pair technique can produce widely varied estimates and erroneous results, mainly due to cross traffic in the path that can distort the packet pair dispersion, increasing or decreasing the capacity estimates.

A main objective of this paper is to *develop a capacity estimation methodology*, based on end-to-end measurements, that is *robust to cross traffic effects*. We first show that a straightforward application of the packet pair technique cannot, in general, produce accurate results when the cross

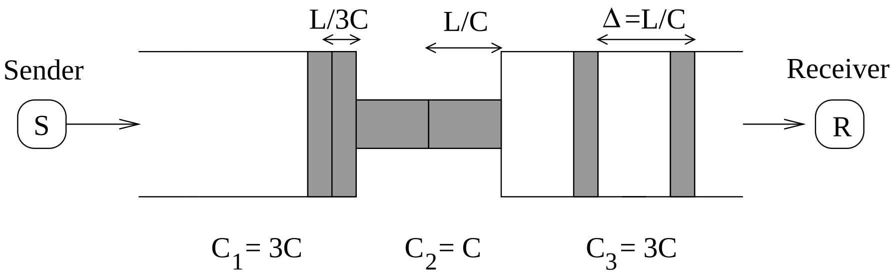


Figure 1: Graphical illustration of the packet pair technique. The width of each link corresponds to its capacity. The two packets leave the sender back-to-back, and they arrive at the receiver with a spacing that is determined at the narrowest link.

traffic effects are ignored. The reason is that the distribution of bandwidth measurements is multimodal, and some local modes, caused by cross traffic, are often stronger than the capacity mode. Consequently, the capacity of a path cannot be estimated using conventional statistical procedures for the most common measured bandwidth range. We also analyze the effect of the probing packet size, and demonstrate that the conventional wisdom of using maximum sized packet pairs is not optimal in heavily loaded paths. On the contrary, if probing packets are of variable size, the sub-capacity local modes become wider, and thus weaker.

We also focus on the dispersion of long packet trains. Earlier work [5] had assumed that the dispersion of long packet trains can be used to measure the available bandwidth in the path. We show that increasing the packet train length reduces the measurement variance, but the estimates converge to a value, referred to as *Asymptotic Dispersion Rate* (ADR), that is lower than the capacity. More importantly, we show that the ADR is *not* the available bandwidth in the path. For single hop paths, though, we can calculate the available bandwidth from the measured ADR. In the more general case of multi-hop paths, the ADR depends on the capacity and utilization of *all* links in the path, and on the routing of cross traffic relative to the measured path.

Finally, we describe a capacity estimation methodology that we have implemented in a tool called *pathrate*. The methodology uses many packet pairs to uncover the corresponding multimodal bandwidth distribution. The size of the probing packets varies, in order to reduce the strength of cross traffic induced modes. *pathrate* identifies local modes in the packet pair bandwidth distribution, and then selects the local mode that corresponds to the path capacity. This latter part relies on the dispersion of long packet trains. We present *pathrate* measurements at several wide-area Internet paths, and evaluate the accuracy of the tool and its robustness to cross traffic effects.

The rest of the paper is structured as follows. Section 2 summarizes the previous work on bandwidth estimation. Section 3 investigates the distribution of bandwidth estimates using packet pairs, while Section 4 investigates the distribution of bandwidth estimates using packet trains. An analytical model for the dispersion of long packet trains is given in Section 5. Section 6 examines the effect of the probing packets size. Based on the insight of the previous sections, Section 7 develops a capacity estimation methodology and the *pathrate* implementation. *pathrate* measurements over wide-area Internet paths are given in Section 8. We conclude in Section 9.

2 Previous work

The concept of packet dispersion, as a burst of packets traverses the narrow link of a path, was originally described in [10]. Jacobson did not consider cross traffic effects, and so did not distinguish between capacity and available bandwidth. Keshav also studied the same idea in the context of congestion control [12], but he recognized that the dispersion of packet pairs is not related to the available bandwidth when router queues are First-Come-First-Served (FCFS). He showed that if all routers use a fair queueing discipline, then the packet pair is isolated from cross traffic, and so it can estimate the available bandwidth in the path. Bolot used packet dispersion measurements to estimate the capacity of a transatlantic link and to characterize the interarrivals of cross traffic [2].

In the past, the packet pair technique was simpler to apply, mostly because capacity values tended to be determined by a few well-known links, such as dial-up modems, ISDN links, T1's, T3's, and Ethernets. Today, mainly through the use of ATM virtual circuits/paths, the bandwidth given to a path can be any value up to the physical capacity of the underlying links. For instance, ISPs often partition an OC-3 link in several fractional virtual links, leased in a granularity of a few Mbps [22].

Early work on packet pair dispersion was followed by more sophisticated variations, focusing on statistical techniques that can extract an accurate capacity estimate from 'noisy' bandwidth measurements. Carter and Crovella created *bprobe*, which uses union and intersection filtering of packet pair measurements of different sizes to produce a final capacity estimate [5]. The basic idea in [5] is that the 'correct' measurements, i.e., measurements not distorted by cross traffic packets, create a mode in the packet pair bandwidth distribution that remains prevalent as the size of the probing packets varies. Lai and Baker, on the other hand, used a kernel density estimator as their statistical filtering tool [13]. The underlying assumption in [13] is that the capacity of a path is related to the most common range of bandwidth measurements, i.e., to the strongest local mode in the underlying distribution[1].

Paxson was the first to observe that the distribution of bandwidth measurements is multimodal, and he elaborated on the identification and final selection of a capacity estimate from these modes [18]. He used packet trains of different lengths to detect multichannel links. The complete methodology is called 'Packet Bunch Modes' (PBM) [19], but as Paxson notes in his dissertation [18] (p.267): *"It is unfortunate that PBM has a large heuristic component, as it is more difficult to understand. (..) We hope that the basic ideas underlying PBM –searching for multiple modes and interpreting the ways they overlap in terms of bottleneck changes and multi-channel paths– might be revisited in the future, in an attempt to put them on a more systematic basis"*. The techniques discussed in our paper also rely on some heuristics, but contrary to Paxson's work, we explain the observed multimodalities based on cross traffic effects.

Dispersion techniques using packet trains instead of packet pairs have also been proposed for the estimation of the available bandwidth in a path. Carter and Crovella developed a tool called *cprobe* which estimates the available bandwidth from the dispersion of eight-packet trains [5]. Other researchers have proposed that the *ssthresh* variable in TCP's slow-start phase, which should ideally be set to the product of the connection's RTT with the available bandwidth, can be determined from the dispersion of the first three or four ACKs [9, 3]. The underlying assumption in [5, 9, 3] is that the dispersion of packet trains is inversely proportional to the available bandwidth. However, as we show here, this is *not* true; the available bandwidth is not directly related to the dispersion of packet trains.

[1] In [6], we incorrectly stated that the *bprobe* also makes this assumption.

Finally, several tools that measure the capacity of *every link in a path* were recently developed: Jacobson's *pathchar* [11], Downey's *clink* [7], Mah's *pchar* [15]. For a study of such tools, see [8]. The underlying idea here is not based on the dispersion of packet pairs or trains, but on the variation of the one-way delay as the packet size increases. Unfortunately, because these tools require the generation of ICMP replies from the routers, which is a process that normally follows different processing paths in a router, resulting measurements are sometimes inaccurate. Some erroneous measurements with *pathchar* are reported in [14]. More recently, Lai and Baker proposed a technique called *packet tailgating* which avoids the need for ICMP replies from the path routers [14]. However, reported capacity measurements are still often inaccurate. A possible explanation is that errors in the link capacity estimates accumulate as measurements proceed along the path. Another explanation is that the presence of 'hidden' layer-2 store-and-forward devices (e.g., Ethernet or ATM switches) creates additional transmission delays that are not accounted for in these tools. These additional delays would cause underestimation of the link capacities.

3 Packet pair dispersion

Consider an H-hop path defined by the sequence of link capacities $\mathcal{P} = \{C_0, C_1, \ldots, C_H\}$. Two packets of size L are sent back-to-back from the source of the path to the sink. These packets are called *packet pair* or *probing packets.* The *dispersion* of the packet pair is the interval from the instant that the last bit of the first packet is received at a certain path point to the instant that the last bit of the second packet is received at that point[2]. The dispersion after the source is $\Delta_0 = \tau_0 = L/C_0$, and Δ_i after link i. When the packet pair reaches the sink, the receiver measures the packet pair dispersion Δ_H, and then computes a bandwidth estimate $b = L/\Delta_H$. Since Δ_H can vary in general, if we repeat this experiment many times the b values will form a distribution $\mathcal{B}$. Our goal, then, is to infer a path capacity estimate $\hat{C}$ from the distribution $\mathcal{B}$.

First, suppose that there is no cross traffic in the path. It is easy to see that the dispersion Δ_i cannot be lower than the dispersion at the previous hop Δ_{i-1} and the transmission delay $\tau_i = L/C_i$ at hop i, i.e., $\Delta_i = \max\{\Delta_{i-1}, \tau_i\}$. Applying this model recursively from the sink back to the source, we find that the dispersion at the receiver is

$$\Delta_H = \max_{i=0\ldots H} \tau_i = \frac{L}{\min_{i=0\ldots H}\{C_i\}} = \frac{L}{C} = \frac{L}{C_n} = \tau_n \tag{3}$$

where C_n and τ_n are capacity and transmission delay of the narrow link, respectively. Consequently, when there is no cross traffic, all bandwidth estimates are equal to the capacity ($b = C$).

In presence of cross traffic in the path, probing packets can experience additional queueing delays due to cross traffic. Let d_i^1 be the queueing delay of the first probing packet at hop i. Also, let d_i^2 be the queueing delay of the second probing packet at hop i, *after the first packet has been transmitted at that link* (see Figure 2). Dispersion after hop i is

$$\Delta_i = \begin{cases} \tau_i + d_i^2 & \text{if } \tau_i + d_i^1 \geq \Delta_{i-1} \\ \\ \Delta_{i-1} + (d_i^2 - d_i^1) & \text{otherwise} \end{cases} \tag{4}$$

Note that when $\tau_i + d_i^1 < \Delta_{i-1}$ and $d_i^2 < d_i^1$, the dispersion from hop $i-1$ to hop i decreases ($\Delta_i <$

[2] We refer to IP packet boundaries, and so we measure IP-level throughput.

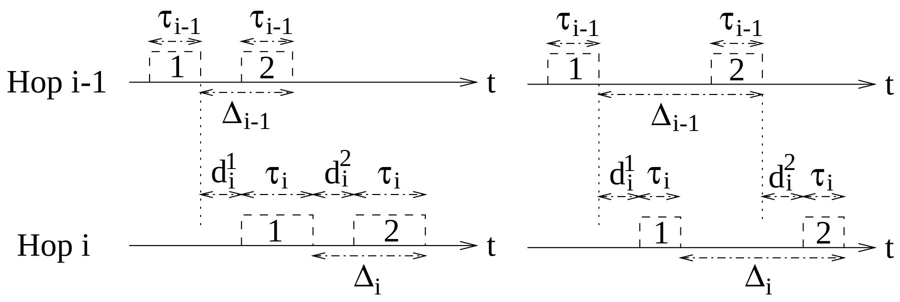
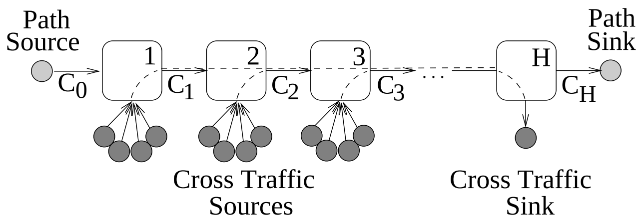
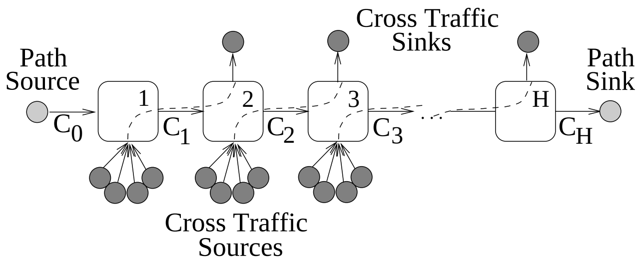


Figure 2: The two cases of Equation 4.



(a) Path persistent cross traffic



(b) One-hop persistent cross traffic

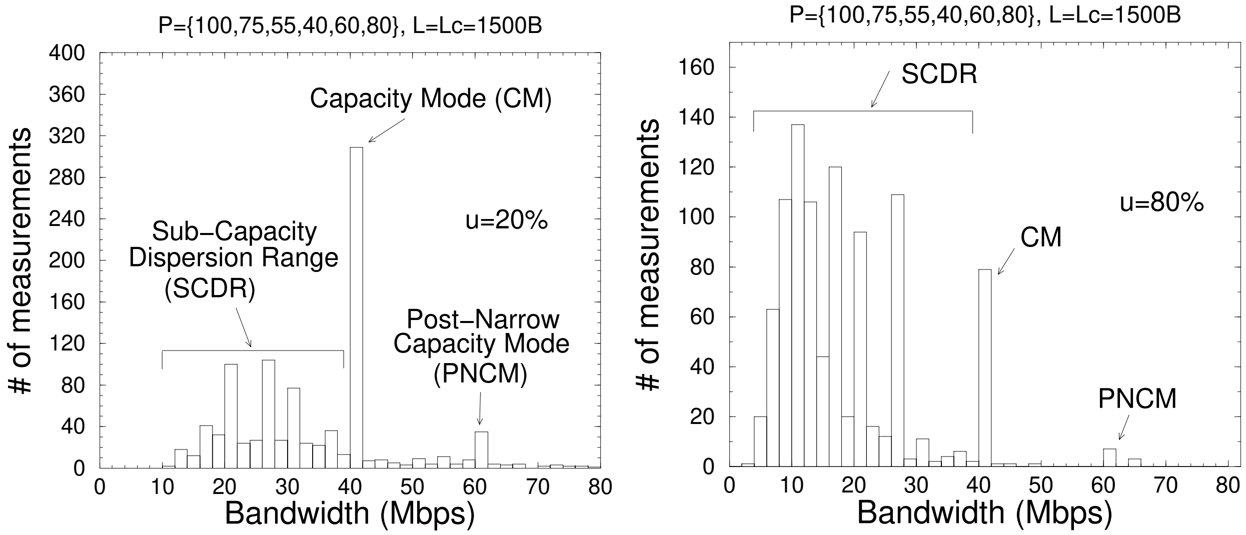
Figure 3: The two extreme cases of cross traffic routing.

Δ_{i-1}). This effect can cause a dispersion at the receiver that is lower than the dispersion at the narrow link ($\Delta_H < L/C_n$), *when there are additional hops after the narrow link*[3]. We refer to such links as *post-narrow links*. The last observation implies that *the capacity of the path cannot be estimated simply from the minimum measured dispersion* (or from the maximum bandwidth measurement), because that minimum dispersion could have resulted from a post-narrow link.

In order to examine the properties of the packet pair bandwidth distribution $\mathcal{B}$ in a controllable and repeatable manner, we used the Network Simulator [1]. Simulations allow us to investigate the cross traffic effects in packet pair dispersion, avoiding issues such as route changes, multichannel links, timestamping accuracy and resolution, that can distort the measurements. Together with simulations, however, we present measurements from real-world Internet paths as well.

The simulated model follows the description given earlier, i.e., the source sends packet pairs

[3]If there is more than one link with capacity C, the narrow link is the last of them in the path.



(a) Light load conditions

(b) Heavy load conditions

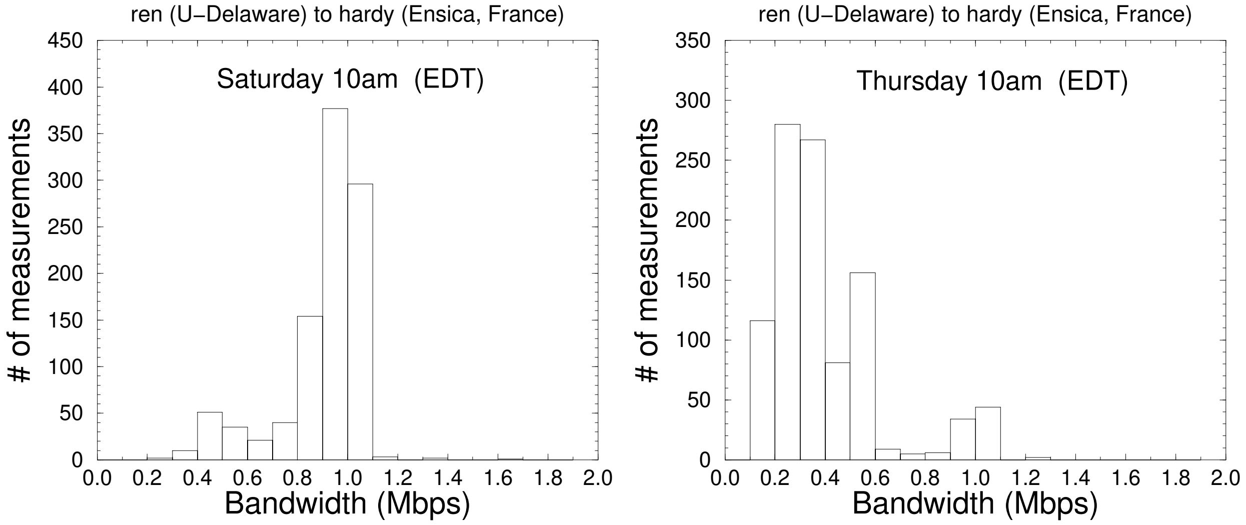
Figure 4: The effect of the network load in the packet pair distribution $\mathcal{B}$ (simulations).

and the sink computes bandwidth estimates b from the measured dispersions Δ_H. Cross Traffic (CT) in the simulations is generated from sixteen Pareto sources at each hop with shape parameter α=1.9, i.e., interarrivals have 'infinite' variance. The CT packet size is L_c, which is either constant or follows a certain random distribution (described later). The packet scheduling discipline in the network links is First-Come First-Served (FCFS), which is the predominant forwarding algorithm in Internet routers today. In the following experiments, the bandwidth distribution $\mathcal{B}$ is formed from 1000 packet pair measurements.

An important issue is the routing of CT packets relative to packet pairs. Figure 3 shows two extreme cases. In Figure 3-a, CT packets follow the same path as the packet pairs, a situation that we refer to as *path persistent cross traffic*. In Figure 3-b, the CT packets always exit one hop after they enter the path, that we refer to as *one-hop persistent cross traffic*. The effect of CT routing will be discussed in §5. For now, we mention that the simulations use one-hop persistent CT.

Let us first examine the effect of the network load in the packet pair distribution $\mathcal{B}$. Figure 4 shows the histogram of $\mathcal{B}$ for a path $\mathcal{P}$ ={100,75,55,40,60,80} (all capacities in Mbps), with a bin width of 2 Mbps. Note that the path capacity is C=40Mbps, while the post-narrow links have capacities of 60 and 80 Mbps, respectively.

In Figure 4-a all links are 20% utilized, whereas in Figure 4-b all links are 80% utilized. When the path is lightly loaded (u=20%) the capacity value of 40 Mbps is prevalent in $\mathcal{B}$, and it forms the *Capacity Mode (CM)*, which in this case is the global mode of the distribution. Bandwidth estimates that are lower than CM are caused by CT packets that interfere with the packet pair, increasing its dispersion, and causing a *Sub-Capacity Dispersion Range (SCDR)*. For instance, the SCDR in Figure 4-a is between 10 and 40 Mbps. The reason that we observe several local modes in the SCDR is discussed later in this section.



(a) Light load conditions

(b) Heavy load conditions

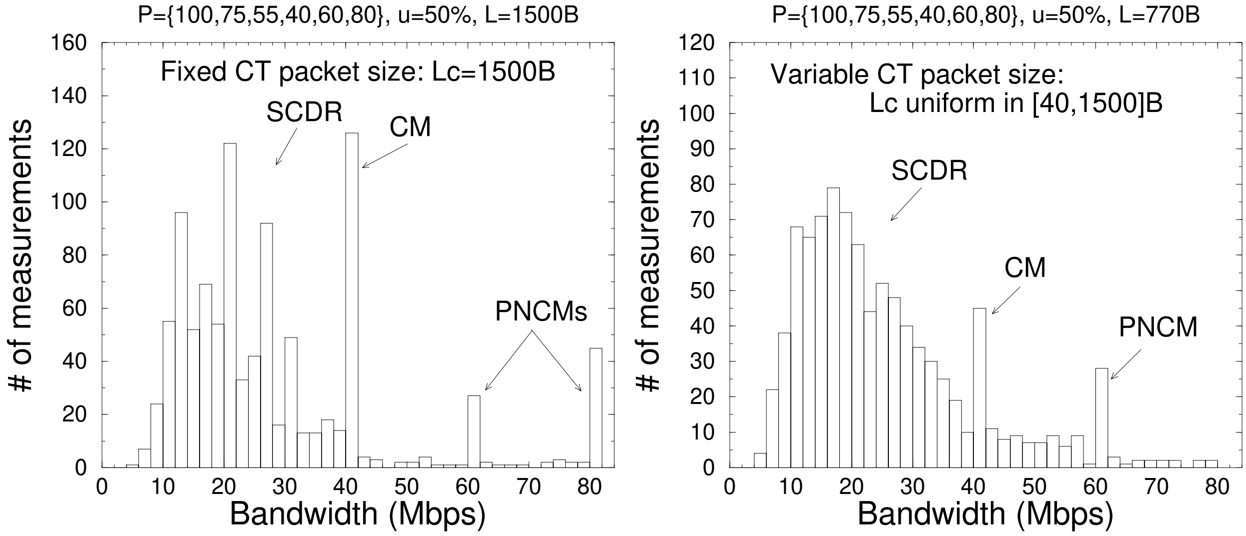
Figure 5: The effect of the network load in the packet pair distribution $\mathcal{B}$ (measurements).

Bandwidth estimates that are higher than CM are caused at the post-narrow links when the first probing packet is delayed more than the second. These estimates are referred to as *Post-Narrow Capacity Modes (PNCMs)*. Note a PNCM at 60 Mbps, which is the capacity of the link just after the narrow link. That local mode is created when the first probing packet is delayed long enough for the packet pair to be serviced back-to-back in that link.

In the case of heavy load (u=80%), the probability of CT packets interfering with the probing packets is large, and *the CM is not the global mode of* $\mathcal{B}$ (see Figure 4-b). Instead, the global mode is in the SCDR, which now dominates the bandwidth measurements. In fact, in quite congested paths the capacity may not even appear as a local mode, making any capacity estimation approach futile. A key point to take from Figure 4 is that *the path capacity cannot be always correctly estimated by statistical techniques that extract the most common bandwidth value or range*. Instead, we must examine the resulting bandwidth distribution in queueing terms, analyze what causes each of the local modes, and what differentiates the CM from the rest of the local modes.

Figure 5 shows packet pair bandwidth distributions that are similar to those in Figure 4, but this time based on Internet measurements rather than simulations. The measured path is from a host (*ren*) at the University of Delaware to a host (*hardy*) at Ensica in Toulouse, France. The capacity of the path is 1 Mbps, limited by Ensica's access link. The distribution of Figure 5-a resulted from measurements on a Thursday morning in Delaware (afternoon in France), while the distribution of Figure 5-b resulted from measurements on a Saturday morning in Delaware. It is reasonable to expect that the path is much more loaded on a weekday than on a Saturday. The impact of the network load is obvious in Figure 5. *When the path is lightly loaded, the capacity mode is prevalent and thus easier to measure. In heavier loads, on the other hand, the SCDR is prevalent, and the CM is only a local mode.* There are no significant PNCMs in these measurements.

Let us now investigate what causes the local modes in the SCDR. Figure 6 shows $\mathcal{B}$ for the same



(a) Fixed CT packet size (b) Variable CT packet size

Figure 6: The effect of the cross traffic packet size L_c in the packet pair distribution $\mathcal{B}$.

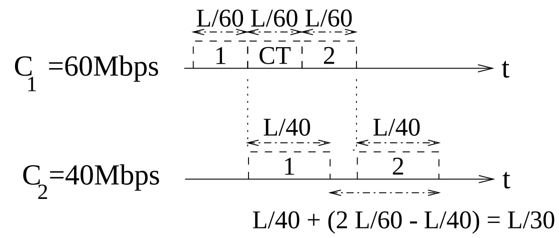


Figure 7: Generation of a 30 Mbps measurement in a path $\mathcal{P}=\{100, 60, 40\}$.

path as in Figure 4, when the CT packet size L_c is fixed to 1500 bytes (Figure 6-a), and when it varies uniformly in the range [40, 1500] bytes (Figure 6-b). In the first case, the probing packet size L is also 1500 bytes, while in the second case it is 770 bytes, i.e., the average of the [40, 1500] range.

When all packets have the same size (L_c=L=1500B), it is simple to explain the local modes in the SCDR of Figure 6-a. The basic idea is that *the SCDR local modes are created when a certain number of CT packets interferes with the packet pair in specific links of the path.* For instance, consider the path $\mathcal{P}=\{100, 60, 40\}$. A local mode at 30 Mbps can be caused by a CT packet interfering with packet pairs at the 60 Mbps link, since in that case the dispersion after the narrow link is $\Delta_n = \frac{L}{40} + (2\frac{L}{60} - \frac{L}{40}) = \frac{L}{30}$ (see Figure 7). Similarly, a mode at 20 Mbps can be caused by a single packet interfering with the packet pair at the 40 Mbps link, or by two packets interfering at the 60 Mbps link, and so on.

When the CT packet size varies uniformly in the range [40,1500]B though (Figure 6-b), the resulting dispersion is less predictable, since a single CT packet interfering with the packet pair can produce a range of dispersion values, depending on its size L_c. However, *the CM and some of the PNCMs are still distinct in the distribution*, as they are caused by the probing packets being serviced back-to-back at the narrow or at post-narrow links, respectively.

Several measurement studies have shown that the packet size distribution in the Internet centers around three or four common values [21, 16]. Specifically, about 50% of the packets are 40 bytes, 20% are 552 or 576 bytes, and 15% are 1500 bytes. These dominant packet sizes would cause a packet pair bandwidth distribution that is more similar to the 'discrete dispersion' effects of Figure 6-a, rather than the 'continuous dispersion' effects of Figure 6-b.

To summarize the observations of this section, *we can think of the packet pair distribution $\mathcal{B}$ as a sequence of local modes imposed on an underlying noise distribution. Each local mode is caused by a 'common queueing scenario' that comprises a fraction of the overall packet pair dispersion. Some of the local modes are below the capacity (SCDR), some of them above the capacity (PNCMs), and one of them is normally the capacity (CM). The challenge of a capacity estimation procedure is to detect which local mode is the CM.* We present a methodology for doing so, after examing in the next two sections the dispersion of packet trains and the effect of the probing packet size.

4 Packet train dispersion

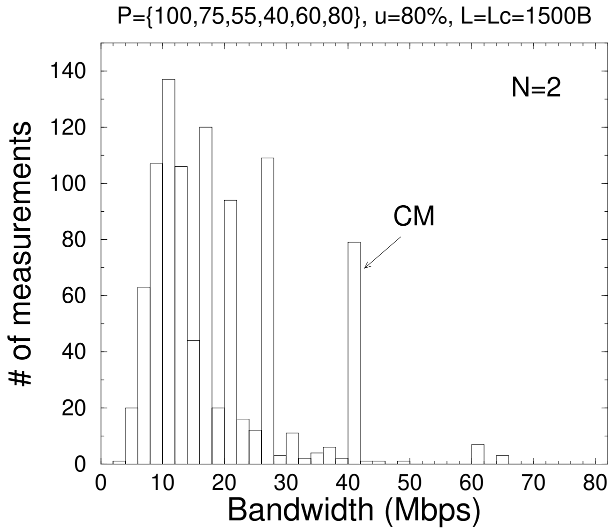
Extending the packet pair technique, the source can send N back-to-back packets of size L to the sink; we refer to these packets as a *packet train of length N*. The sink measures the total dispersion $\Delta(N)$ of the packet train, from the first to the last packet, and computes a bandwidth estimate as $b(N) = \frac{(N-1)L}{\Delta(N)}$. Many such experiments form the packet train bandwidth distribution $\mathcal{B}(N)$.

In the absence of cross traffic, bandwidth estimates will be equal to the capacity C, as in the packet pair case. We need to measure the capacity of a path using packet trains when the narrow link is multichanneled [19]. In a k-channel link of total capacity C, the individual channels forward packets in parallel at a rate of C/k, and so we can measure the link capacity from the dispersion of packet trains with N=k+1. Packet trains are also required to measure the *sustainable rate* of a traffic shaper[4].

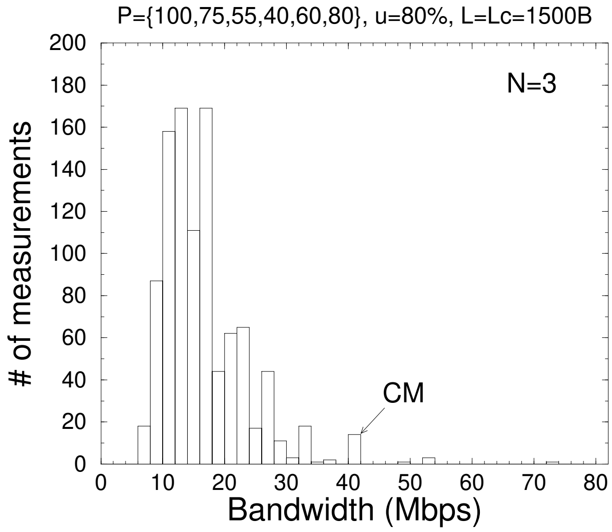
It may seem that use of packet trains rather than packet pairs would render capacity estimation more robust to random noise caused by cross traffic. One can argue such a hypothesis because packet trains lead to larger dispersion values, which are less sensitive to measurement noise. However, this is not the case for the following reason. Although the dispersion $\Delta(N)$ becomes larger as N increases, so does the 'noise' in the dispersion $\Delta(N)$, since it becomes more likely that CT packets will interfere in the packet train. This issue was briefly mentioned in [18] (p.259), noting that packet trains should be less prone to noise, since individual packet variations are smoothed over a single large interval rather than N-1 small intervals. The larger the N, however, the greater the likelihood that a packet train will be dispersed by cross traffic, leading to bandwidth underestimation.

In this section we present simulation and experimental results illustrating the effect of N in the bandwidth distribution $\mathcal{B}(N)$, and make some general observations about this relation. In the next section, we explain these empirical observations with a model of cross traffic interference in packet trains. Specifically, Figure 8 shows the histograms of $\mathcal{B}(N)$, with four increasing values of N, for a simulated path $\mathcal{P}$ ={100,75,55,40,60,80} (u=80% in all links). Figure 9 shows the histograms of $\mathcal{B}(N)$, with four increasing values of N again, for an Internet path from *nova* (at Univ-Wisconsin, Madison) to *ren* (at Univ-Delaware, Newark) in June 2001.

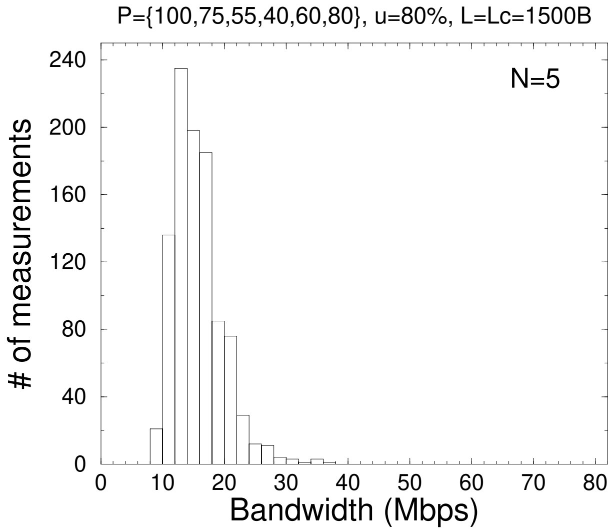
[4]Traffic shapers, usually in the form of a leaky bucket, limit the capacity of a (virtual) link from a peak rate to a sustainable rate after a certain burst size.



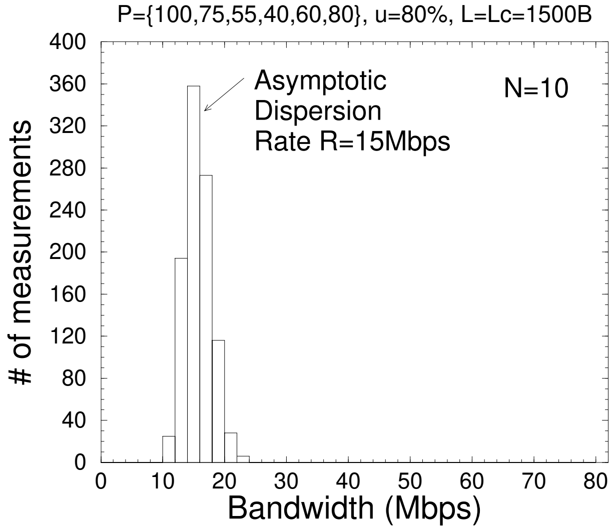
(a) Packet pairs (N=2)



(b) Packet trains with N=3

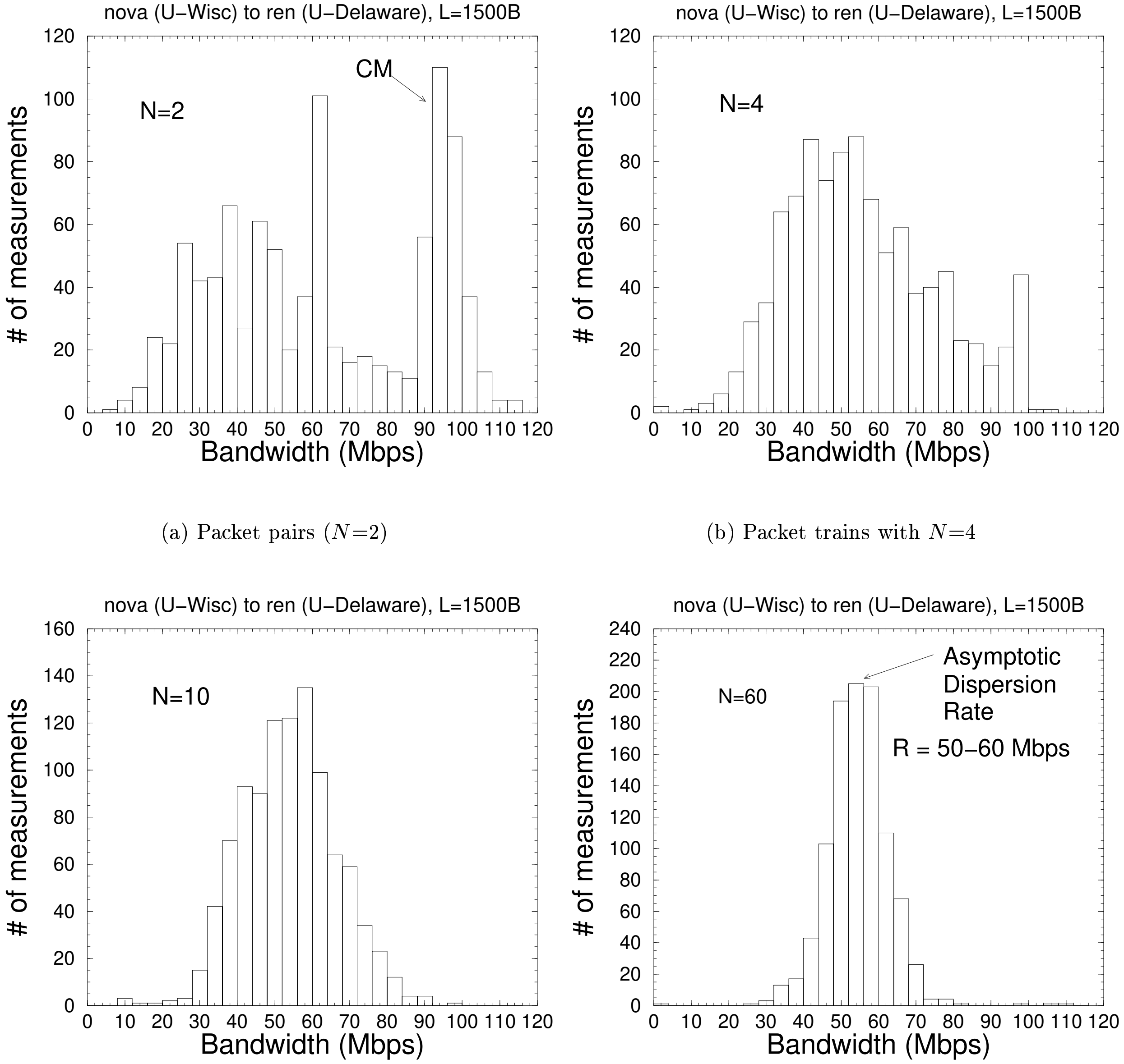


(c) Packet trains with N=5



(d) Packet trains with N=10

Figure 8: The effect of the packet train length N (simulations).



(a) Packet pairs (N=2)

(b) Packet trains with N=4

(c) Packet trains with N=10

(d) Packet trains with N=60

Figure 9: The effect of the packet train length N (measurements).

A first observation is that, *as N increases, the CM and PNCMs become weaker, until they disappear, and the SCDR prevails in the bandwidth distribution $\mathcal{B}(N)$*. This implies that, as N increases, almost all packet trains encounter additional dispersion due to CT packets. In other words, *the best value of N for generating a strong capacity mode is N=2, i.e., to use packet pairs. Anything longer than packet pairs is more likely to get additional dispersion due to cross traffic.*

A second observation is that, *as N increases, $\mathcal{B}(N)$ becomes unimodal.* So, when N is sufficiently large, the dispersion of packet trains is *not* determined by distinct CT interference scenarios, such as the interference by a single CT packet of size x at a link y, but by the aggregate amount of CT bytes that interfere with the packet train.

A third observation is that *the range of the distribution, which is related to the measurement variance, decreases as N increases.* This decrease in the distribution implies that the variance in the amount of cross traffic interfering with the packet train decreases, as the length of the packet train increases.

A fourth observation is that, *when N is sufficiently large and $\mathcal{B}(N)$ is unimodal, the center of the (unique) mode is independent of N.* We refer to the center of this unique mode as the *Asymptotic Dispersion Rate (ADR) R.* The fact that ADR does not depend on the packet train length means that, for sufficiently large N, the dispersion of the packet train $\Delta(N)$ becomes proportional to N-1, and thus, the packet train length factor $N-1$ cancels out at the bandwidth estimation formula $b(N)=\frac{(N-1)L}{\Delta(N)}$.

In the next section, we present a model for the dispersion of packet trains that explains and supports these empirical observations, taking into account the intensity of the cross traffic in the path.

5 Asymptotic Dispersion Rate

Let us first consider a single-hop path $\mathcal{P}=\{C_0, C_1\}$ with $C_0 \geq C_1$, i.e., the narrow link ('link-1') has capacity C_1. A packet train of length N is sent from the source to the sink, with all probing packets being of size L. The initial dispersion of the packet train at link-0 is $\Delta_0 = L(N-1)/C_0$. We assume that the network queues serve packets in a FCFS manner. Let r_1 be the average arrival rate of cross traffic in link-1. The average amount of cross traffic that arrives in link-1 during Δ_0 is $\bar{X}_1 = \Delta_0 r_1$. This cross traffic amount interferes with the transmission of probing packets, and so the average dispersion at the exit of the narrow link is

$$\bar{\Delta}_1 = \frac{(N-1)L + \bar{X}_1}{C_1} = \frac{(N-1)L}{C_1}(1 + u_1\frac{C_1}{C_0}) \tag{5}$$

where $u_1 = r_1/C_1$ is the load (or utilization) of the narrow link due to cross traffic.

The average bandwidth estimate at the receiver, that we refer to as *Asymptotic Dispersion Rate* R, is

$$R \equiv \frac{(N-1)L}{\bar{\Delta}_1} = \frac{C_1}{1+u_1\frac{C_1}{C_0}} < C_1 \tag{6}$$

which is lower than the capacity of the path. Note that, as observed in §4, *the ADR is independent of N*, because the average amount of interfering cross traffic $\bar{X}_1$, and thus the overall dispersion $\bar{\Delta}_1$, is proportional to N-1. As shown in Figures 8-d and 9-d, even with heavy-tailed Pareto cross traffic or with real Internet traffic, a train length of several tens of packets is usually sufficient to give an estimate of R.

We continue with some comments on Equation 6. First, if we know the capacities C_0 and C_1, we can measure R from the average dispersion of long packet trains, compute the cross traffic utilization u_1 from Equation 6, and then compute the available bandwidth as $A = C_1(1-u_1)$. So, *we can estimate the available bandwidth of single hop paths using the dispersion of packet trains that are sufficiently long to produce a narrow estimate of R.* This analysis also shows that the available bandwidth is not inversely proportional to the dispersion of long packet trains, as was assumed for instance in [5], even for single hop paths. For example, in the path of Figure 8-d we have that R=15 Mbps, while the available bandwidth is A=40(1-0.8)=8 Mbps.

Second, for capacity estimation purposes, it helps to inject probing packets in the path from a higher bandwidth interface (higher C_0), since the cross traffic term $(u_1\, C_1/C_0)$ is then reduced. For instance, suppose that we measure the capacity of a path with C_0=100Mbps and C_1=100Kbps using packet trains. Even if the narrow link is almost saturated ($u \approx 1$), the cross traffic term $u_1\, C_1/C_0$ will introduce an underestimation error that is less than 0.1%. With the same token, the effect of cross traffic in the dispersion of packet trains becomes critical when there is a relatively small difference between the link capacities of the path.

Third, note that the term $u_1\, C_1/C_0$, which represents the effect of cross traffic on the dispersion of the packet train, is equal to $\bar{X}_1/[(N-1)L]$. This value also represents the average number of CT bytes interfering with two successive packets of the probing train. The exact size of the interfering CT packets or the specific location of the packet train at which the CT packets interfere do not matter, as long as the aggregate amount of interfering cross traffic adds up to $\bar{X}_1$. This is why the distribution $\mathcal{B}(N)$ becomes unimodal as N increases. Also, the variance of $\bar{X}_1$ decreases as N increases, since the actual CT arrival rate during the packet train transmission period tends to the average value r_1. So, the distribution $\mathcal{B}(N)$ becomes narrower as the packet trains becomes longer.

The previous model considered a single-hop path with $C_0 \geq C_1$. In the following, we derive exact expressions or upper/lower bounds for the ADR in some more general path configurations.

Path persistent cross traffic with decreasing link capacities: Suppose that the link capacities are decreasing along the path, i.e., $C_0 \geq C_1 \geq \ldots \geq C_H$. Let r_i be the average rate of cross traffic that enters the path in link i. If the cross traffic is path persistent (see Figure 3-a), the total cross traffic rate in link i is $\sum_{k=1}^{i} r_k$. The average dispersion, then, at the exit of link i is

$$\bar{\Delta}_i = \bar{\Delta}_{i-1}(C_{i-1} + r_i)/C_i \tag{7}$$

because the total size of the packet train at the exit of link $i-1$ (including any interfering cross traffic) is $\bar{\Delta}_{i-1}C_{i-1}$ bytes, and $\bar{\Delta}_{i-1}r_i$ additional bytes of cross traffic interfere at link i. Applying Equation 7 recursively from $i = H$ to $i = 1$ gives that

$$R \equiv \frac{(N-1)L}{\bar{\Delta}_H} = \frac{C_H}{\prod_{i=1}^{H}(1 + \frac{r_i}{C_{i-1}})} \tag{8}$$

For instance, the ADR for the path $\mathcal{P}= \{C_0, C_1, C_2\}$ and $C_0 \geq C_1 \geq C_2$ with path persistent cross traffic is

$$R = \frac{C_2}{1 + \frac{r_1}{C_0} + \frac{r_2}{C_1} + \frac{r_1 r_2}{C_0 C_1}} \tag{9}$$

Note that the ADR is determined, at least in this path configuration, by the cross traffic rate and the capacity *of each link in the path.* This is fundamentally different than the available bandwidth A, which depends on the utilization and capacity of only the tight link.

Path persistent cross traffic (general case): When the capacities do not decrease along the path, an exact analysis is more complicated. To see why, consider two probing packets that depart a link of capacity C_{i-1} back-to-back, with dispersion $\Delta_{i-1} = L/C_{i-1}$, and enter a link of capacity $C_i > C_{i-1}$. It is easy to check that an *empty space* of duration $\Delta_{i-1}(1 - C_{i-1}/C_i)$ will be created between the two probing packets at link i. The cross traffic at link i can fill this empty space in the packet train, or cause additional dispersion without filling in the empty space. Unfortunately, we cannot predict how much of this empty space will be covered with cross traffic. But we can derive a *lower bound* on the dispersion Δ_i, and thus an upper bound on the ADR, *if we assume that the cross traffic at link i increases the dispersion beyond Δ_{i-1} only after it fills in all empty space in the packet train.*

Consider, first, a single-hop path $\mathcal{P}= \{C_0, C_1\}$ with $C_0 < C_1$. The dispersion of the packet train when it leaves link-0 is $\Delta_0{=}(N-1)L/C_0$. Based on the previous assumption for the CT interference, the dispersion at the output of link-1 is

$$\bar{\Delta}_1 = \Delta_0 + \max\left\{\frac{\bar{X}_1}{C_1} - \frac{(N-1)L}{C_0}(1 - \frac{C_0}{C_1}), \quad 0\right\} \tag{10}$$

because the total duration of empty space in the train is $\Delta_0(1 - C_0/C_1)$, and the total amount of interfering cross traffic at link-1 is $\bar{X}_1 = \Delta_0 r_1$. We can re-write the previous equation as

$$\bar{\Delta}_1 = \Delta_0 + \max\left\{\frac{\Delta_0}{C_1}\left[r_1 - (C_1 - C_0)\right], \quad 0\right\} \tag{11}$$

If the cross traffic rate at link-1 is sufficiently low ($r_1 < C_1 - C_0$), the dispersion is not increased ($\bar{\Delta}_1{=}\Delta_0$), and so the ADR is equal to the capacity of the path ($R = C_0$). Otherwise, the final dispersion becomes

$$\bar{\Delta}_1 = \Delta_0(u_1 + \frac{C_0}{C_1}) \tag{12}$$

which yields the same ADR as in Equation 6.

We can extend this result to multiple-hop paths, when the cross traffic is path persistent. We again derive the lower bound on the dispersion Δ_H assuming that the cross traffic increases the packet train dispersion only after it fills in all the empty space between probing packets. Specifically, we can show with similar reasoning as in the previous paragraph that

$$\bar{\Delta}_i = \begin{cases} \bar{\Delta}_{i-1}\frac{C_{i-1}+r_i}{C_i} & \text{if } C_{i-1} \geq C_i \\ & \text{or } r_i \geq C_i - C_{i-1} > 0 \\ \bar{\Delta}_{i-1} & r_i < C_i - C_{i-1} \end{cases} \tag{13}$$

Given the capacity and cross traffic rate in each hop, and since $\Delta_0 = L(N-1)/C_0$, we can solve recursively for $\bar{\Delta}_H$, and then calculate an upper bound on the ADR from $R = L(N-1)/\bar{\Delta}_H$.

General cross traffic routing: When the cross traffic is not path persistent, i.e., CT packets can exit the path at any link, the dispersion of packet trains is hard to analyze for the same reason as previously: CT packets can interfere in the packet train increasing its dispersion, and then exit the path at a subsequent link leaving empty space, or 'bubbles', between probing packets. These bubbles can be filled in by other CT packets in subsequent hops, or they can persist until the packet train reaches the sink.

When nothing is a priori known about the cross traffic routes, we can derive only lower and upper bounds on the ADR. For simplicity, we present these bounds for the special case of a multi-hop path in which all links have the same capacity C. Specifically, when $C_i = C$ for $i = 1 \ldots H$, the ADR is

$$\frac{C}{\prod_{i=1}^{H}(1+\frac{r_i}{C})} \leq R \leq \frac{C}{1+\frac{1}{C}\max_{i=1\ldots H} r_i} \tag{14}$$

Note that r_i is, as previously stated, the average rate of cross traffic entering the path in link i (and *not* the total cross traffic rate in link i).

The lower bound of (14) can result when *bubbles created at a link are not filled in subsequent links.* In that case, an additional dispersion $(\Delta_{i-1} r_i)/C$ is introduced in each link i, i.e.,

$$\Delta_i = \Delta_{i-1}(1+\frac{r_i}{C}) \tag{15}$$

and so the dispersion at the receiver becomes

$$\Delta_H = \Delta_0 \prod_{i=1}^{H}(1+\frac{r_i}{C}) = \frac{L(N-1)}{C_0}\prod_{i=1}^{H}(1+\frac{r_i}{C}) \tag{16}$$

The upper bound of (14) can result with one-hop persistent cross traffic, when the CT packets at each link fill in the bubbles created at previous links before introducing any additional dispersion. In that case, the dispersion at a link i is

$$\Delta_i = \max\left\{\Delta_{i-1}, \frac{N(L-1)}{C}(1+\frac{r_i}{C})\right\} \tag{17}$$

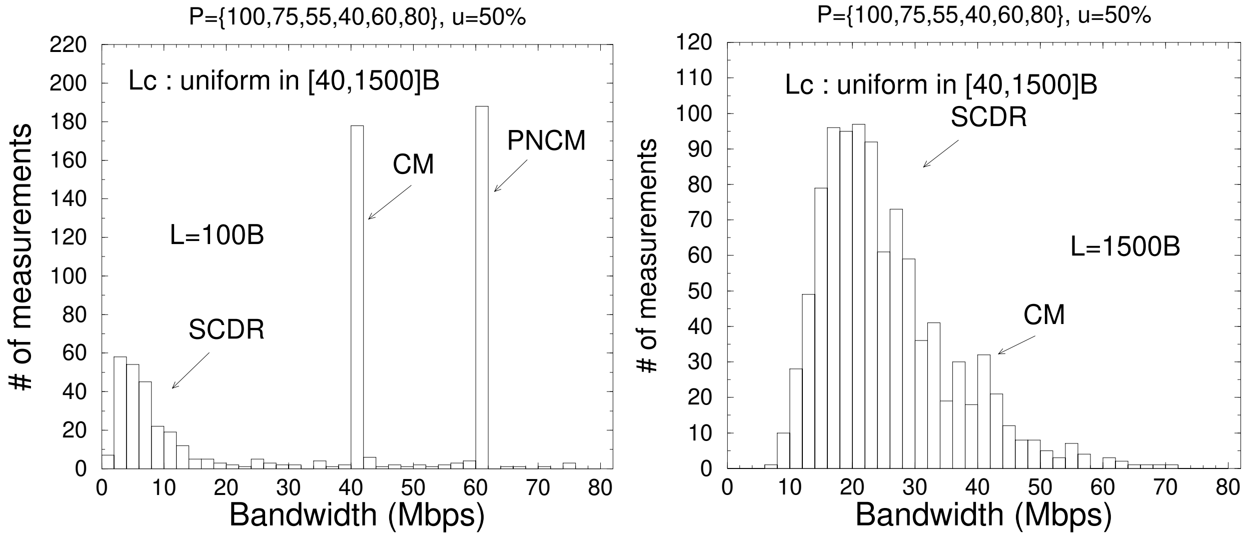
because Δ_i, in this case, cannot be larger than the dispersion that would be caused if the packet train had arrived back-to-back at link i. Applying Equation 17 recursively from $i = H$ to $i = 1$, we get that

$$\Delta_H = \Delta_0(1+\max_{i=1\ldots H}\frac{r_i}{C}) \tag{18}$$

where $\Delta_0 = L(N-1)/C_0$.

6 The effect of the probing packet size

In this section, we focus on the effect of the packet size L in packet pair probing. The 'conventional wisdom', as reflected for instance in [13], is that the best value of L is the maximum non-fragmented packet size, i.e., the *path Maximum Transmission Unit (MTU)* size. The reason is that a higher L leads to larger dispersion, which in turn is easier to measure, more robust to queueing delay noise, and less sensitive to the timestamping resolution at the receiver. Here, we first investigate the effect of the probing packet size on the strength of the PNCM, CM, and SCDR modes. Then, we show that if the size of the probing packets in different packet pair experiments varies, the SCDR modes become wider and weaker.



(a) L=100 bytes

(b) L=1500 bytes

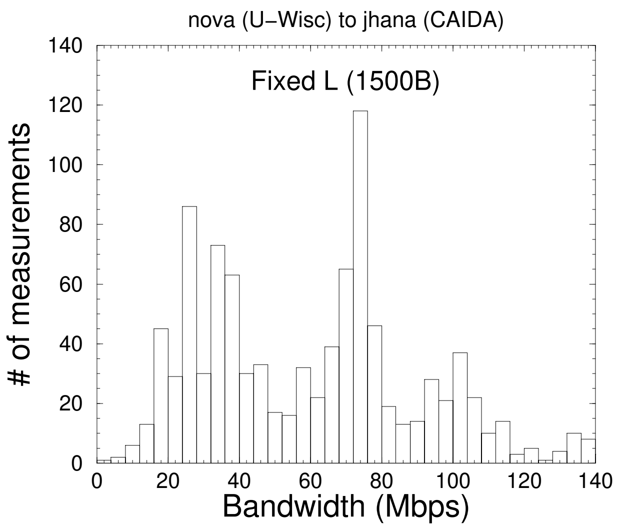
Figure 10: Small versus large packet size in packet pair probing.

Let us first examine the effect of a fixed L on the strength of the PNCM, CM, and SCDR modes. A larger packet size L leads to a wider time interval in which a CT packet can interfere with the packet pair. Suppose that a packet pair arrives at a link i with capacity C_i. If a CT packet arrives at link i in the time interval between the arrival of the first and the second probing packets, which is of length L/C_i, it will interfere with the probing packets, increasing the dispersion Δ_i above τ_i. The larger the L, the higher the likelihood of an interfering CT arrival, and thus the SCDR becomes more prevalent in the bandwidth distribution $\mathcal{B}$. This effect becomes clear in Figure 10, where $\mathcal{B}$ is shown for a small packet (L=100B) and for a large packet (L=1500B), with the same path configuration, load, and CT packet size distribution in both cases. Note that the size L of the probing packets is qualified as 'small' or 'large', relatively to the size of the cross traffic packets. The SCDR in the case of small L is much weaker than in the case of large L, because it is less likely for CT packets to interfere in the 'small-sized' packet pairs.

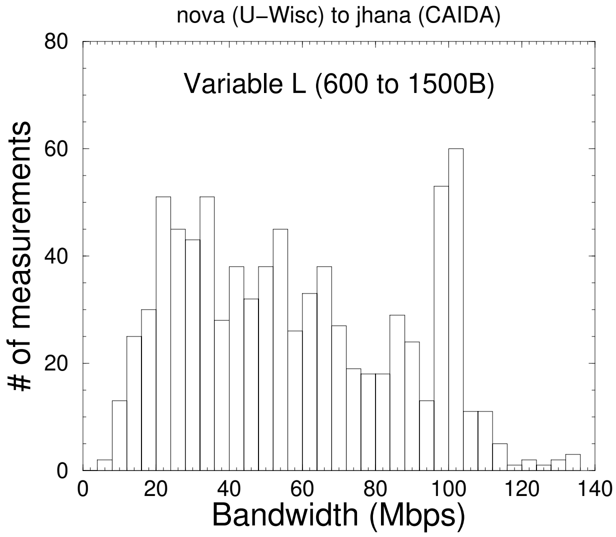
As L decreases, on the other hand, the dispersion decreases proportionally, and it becomes more susceptible to distortion at the post-narrow links. Suppose that L=100B, $\mathcal{P}$={40,80} and that a packet pair leaves the narrow link back-to-back, i.e., with Δ_0=20μs. It only takes one CT packet, larger than 100 bytes, at the 80 Mbps link to delay the first probing packet so much that the packet pair dispersion is determined by link-1, i.e., Δ_1=10μs. In other words, when L is small, the formation of PNCMs becomes more likely. This is shown in Figure 10. Note the PNCM at 60 Mbps in Figure 10-a, which is actually stronger than the CM at 40 Mbps. On the other hand, there are no significant PNCMs when L=1500B, as shown in Figure 10-b.

The next question that we consider is: is it better to use fixed or variable-sized probing packets in different packet pairs?[5] If L is the same in all packet pairs, an SCDR mode can be created when a cross traffic packet with a 'common' size interferes between packet pairs. For example, if L=1500B,

[5]Note that the two probing packets of the same packet pair have to have the same size.



(a) L = 1500B



(b) L: 600-1500B

Figure 11: Fixed versus variable-sized packet pair bandwidth measurements.

many interfering 552B packets at a 40Mbps link will cause an SCDR mode at about 29Mbps. As noted in §3, there are some quite common packet sizes in the Internet today, and so the creation of strong SCDR modes should be expected if all packet pairs have the same size L (see also Figure 6).

On the other hand, consider what would happen when probing packets of different packet pairs have variable sizes, say between 50B and 1500B. The interference of a 552B cross traffic packet between a packet pair at the 40Mbps link can cause a bandwidth measurement from 3.3Mbps up to 29Mbps. In general, the use of variable-sized probing packets 'spreads' the sub-capacity measurements in a larger bandwidth range, making the SCDR modes wider, and thus weaker.

The trade-off between fixed and variable probing packet sizes is shown in Figure 11. The two bandwidth distributions resulted from 1000 packet pair experiments in the path from *nova* (U-Wisc) to *jhana* (CAIDA). In the first case (Figure 11-a), all packet pairs consist of probing packets with size L=1500 bytes. Note the presence of strong SCDR modes around 72Mbps, 35Mbps, and 26Mbps. The 100Mbps CM appears only as a weak mode, compared to these SCDR modes. In Figure 11-b, on the other hand, the size of probing packets in different packet pairs varies uniformly from 600 to 1500 bytes. Notice that the sub-capacity measurements have been 'spread' throughout a wide interval, and thus the local SCDR modes are weaker compared to Figure 11-a. The CM has accumulated roughly the same number of measurements in both Figures 11-a and 11-b, but it is easier to detect in the latter, because it is stronger compared to the SCDR modes.

So, what should be the probing size L if we are estimating the capacity of a path with many packet pair experiments? Obviously, our objective is to make CM relatively stronger than the 'erratic' PNCM and SCDR modes. We showed that using a range of L, instead of a fixed value, spreads the sub-capacity measurements making the SCDR modes wider, and thus, weaker. We also showed that using probing packets that are too small can create intense PNCMs, while using probing packets that are too large can lead to a prevalent SCDR and a weak CM. Combining these effects, we see that a good compromise for the probing packet size is to use a range of sizes, say from L_{min} to L_{max}, where L_{min} is not a very small packet and L_{max} is not a very large packet, relatively to the common cross traffic packets. An additional constraint on L_{min} is given in the next paragraph.

There is a practical issue that we need to consider, finally, related to the *minimum dispersion that the receiving host can measure.* A receiving host can only measure the dispersion of a packet pair when it is higher than Δ_m. This lower bound Δ_m is determined by the latency to receive a packet in the OS, move the packet from kernel to user space through a *recvfrom* system call, timestamp the arrival, and perform any other operations of the receiving program before waiting for the second probing packet. For *pathrate*, we measured Δ_m in several different platforms, including Sun Ultra-10 and Pentium-II workstations running Solaris 2.6 or Free-BSD 3.2, and the minimum dispersion Δ_m is in the order of 30 to 40 μs. Given Δ_m for a particular host, the maximum possible capacity that can be measured for a packet size L is $C = L/\Delta_m$. For example, with Δ_m=40μs and L=800B the maximum capacity that can be measured is 160 Mbps. On the other hand, when a rough estimate $\bar{C}$ of the capacity is known, the packet size should be at least $L > \bar{C}\Delta_m$.

7 A capacity estimation methodology

In this section, we present a capacity estimation methodology based on the insight developed earlier in the paper. We have implemented this methodology in a tool called *pathrate*[6]. We describe here the main characteristics and operation of *pathrate* version 2.1. The reader should note, however, that evolution of *pathrate* is an ongoing project, and that new versions of *pathrate* may deviate from the following description.

The main observations upon which we have developed *pathrate* are:

- The optimal train length for detecting the Capacity Mode CM is N=2, i.e., to use packet pairs.
- Using packet pairs with variable-sized packets makes the SCDR modes wider and weaker.
- Using relatively larger (but still variable-sized) packets makes the PNCMs weaker.
- The Asymptotic Dispersion Rate R can be measured with long packet trains. The capacity C is larger or equal than R.

Pathrate requires the cooperation of both the source and the sink, i.e., it is a *double end-point methodology.* More flexible approaches require access only to the source of the path, 'forcing' the sink to reply to each received packet using ICMP, UDP-echo, or TCP-FIN packets. The drawback of those approaches is that the reverse path from the sink to the source, through which the replies are forwarded, affects bandwidth measurements, making it hard to decouple the characteristics of the two paths. We prefer the double end-point methodology, even though it is less flexible, because it is more accurate.

Pathrate uses UDP for transfer of probing packets. Additionally, *pathrate* establishes a TCP connection, referred to as the *control channel*, between the sender and the receiver. The control channel acknowledges every correctly received packet pair/train, and it is also used for commands sent from the receiver to the sender regarding the probing packet size, train length, number of measurements, train spacing, and end of each measurement phase.

[6] *Pathrate* is publicly available at http://www.cis.udel.edu/~dovrolis/bwmeter.html.

We ignore any packet pairs or trains that encountered losses during the measurement process. As a simple form of congestion avoidance, *pathrate* aborts the measurement process when it detects a number of consecutive losses in the path. We ensure that the time interval between successive packet pairs or trains is larger than the round-trip time of the path, and never less than 500 msec. When *pathrate* detects a low-capacity link or a congested path, it dynamically increases the packet train spacing (depending also on the length of the packet trains) so that it does not overload the path with measurement traffic.

pathrate consists of mainly three execution phases. We describe the major tasks in each phase next.

Preliminary measurements and tests (Phase 0). Initially, *pathrate* detects the maximum train length N_{max} that the path can transfer without experiencing significant losses. N_{max} provides an upper bound on the length of the packet trains that will be used in the estimation of the ADR, in Phase II.

Then, *pathrate* generates about 60 packet trains of gradually increasing length, from $N = 2$ to $N = \min(10, N_{max})$ packets. *Pathrate* uses these measurements in three ways. First, *pathrate* uses these measurements to detect whether the narrow link consists of parallel channels. If there is a steep bandwidth decrease when N increases from c to $c + 1$, we can infer that the narrow link consists of c channels. In none of our experiments we detected this pattern, however.

Second, these measurements give a rough estimate $\tilde{C}$ of the capacity. $\tilde{C}$ is computed as the average of the preliminary measurements, after we remove the five largest and five smallest values (that may be outliers). Given $\tilde{C}$, we then compute a reasonable *bandwidth resolution*, or *bin width*, ω. The bandwidth resolution ω is an important parameter for the detection of local modes in a distribution of bandwidth measurements (see Appendix II). The final capacity estimate that *pathrate* produces will be a bandwidth range of width ω. We calculated the bandwidth resolution from a piece-wise linear function of $\tilde{C}$, which was determined empirically. Roughly speaking, ω is about 10% of $\tilde{C}$ when $\tilde{C}$ is about 1Mbps, 7.5% when $\tilde{C}$ is about 10Mbps, down to 2% when $\tilde{C}$ is about 150Mbps.

Third, we used the preliminary measurements in a 'Quick-Estimate' mode, when there is extremely low variation in the measurements. This happens with paths that are quite lightly loaded. Specifically, *pathrate* measures the coefficient of variation CV of the preliminary measurements (again, ignoring the five largest and smallest values). If CV is less than a certain threshold, *pathrate* exits, providing $\tilde{C}$ as the final capacity estimate.

Phase I: Packet pair probing. As shown in §4, one is more likely to observe the capacity mode CM using packet pairs rather than packet trains. Consequently, in Phase I we use a large number of packet pairs to 'uncover' all local modes in the bandwidth distribution $\mathcal{B}$. We expect one of these modes to be the Capacity Mode CM. Also, as shown in §6, the use of variable-sized probing packets widens the SCDR modes, while the use of relatively large probing packets weakens the PNCMs. Based on these observations, Phase I consists of K_1=1000 packet pair measurements with variable sized probing packets. L varies between L_{min} and L_{max} bytes. The minimum size L_{min} results from the host-related timestamping constraints discussed at the end of the §6, and is always larger than 550 bytes so that the PNCMs are relatively weak. The maximum size L_{max} is set to the TCP connection's (control channel) Maximum Segment Size. If the path supports MTU-discovery, the Maximum Segment Size will be constrained by the MTU size.

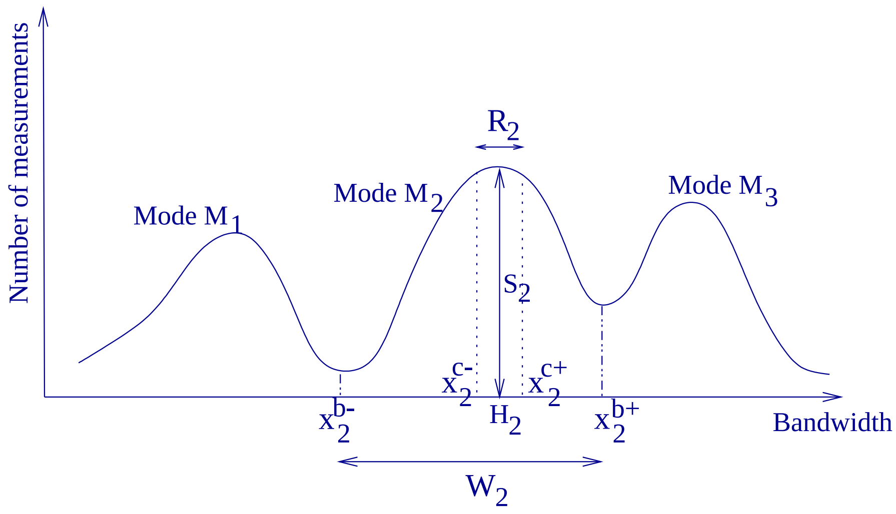


Figure 12: The characteristics of a local mode.

At the end of Phase I, we estimate the local modes in the distribution of packet pair measurements. Appendix II describes the numerical procedure for the identification of local modes. For each mode M_i, this procedure returns the central modal bin R_i, the number of measurements S_i in R_i, the range of the modal bell W_i, and the number of measurements B_i in the modal bell W_i. The average bandwidth in the central modal bin R_i is denoted as H_i. These modal characteristics are shown in Figure 12. The sequence of all local modes in the Phase I measurements is $\mathcal{M}= \{M_1, M_2, \ldots M_{\rm II}\}$, where the local modes are ordered so that $H_i < H_{i+1}$.

We expect that one of these local modes, say M_k, is the Capacity Mode CM, i.e., $C = H_k$. Modes M_i with $i > k$ are PNCMs, while modes M_i with $i < k$ reside in the SCDR. The challenge then is to select the right local mode M_k. We do so in Phase II.

Phase II: ADR estimation. As N increases, the CM and the PNCMs are eliminated from the bandwidth distribution $\mathcal{B}(N)$, and the SCDR accumulates all measurements. For a large value of N, $\mathcal{B}(N)$ becomes unimodal, centered at the Asymptotic Dispersion Rate R. Additionally, the ADR R is less or equal than the capacity C. *pathrate* uses these facts in Phase II to estimate R, and then to eliminate the Phase I modes that are less than R from the capacity estimation procedure.

Specifically, Phase II consists of K_2=500 packet trains with length N_{max} (the maximum possible length) and with maximum sized packets ($L = L_{max}$). Because the train lengths are quite large, the resulting measurements normally form a unimodal distribution. Using the same numerical procedure as in Phase I, *pathrate* determines the unique mode of this distribution and estimates the ADR R from the central modal bin of this unique mode. Given R, we can ignore any Phase I modes M_i with $H_i < R$, as they are definitely SCDR modes.

If there are more than one Phase II modes that are larger than R, the next challenge is to select the one that is more likely to be the Capacity Mode. The insight that we use here is that the CM is a relatively narrow and strong Phase I mode. The reason is that Phase I uses variable sized probing packets, 'spreading' the SCDR modes, and relatively large probing packets, weakening the presence of PNCMs. To evaluate the 'narrowness' and 'strength' of the Phase I modes, we compute

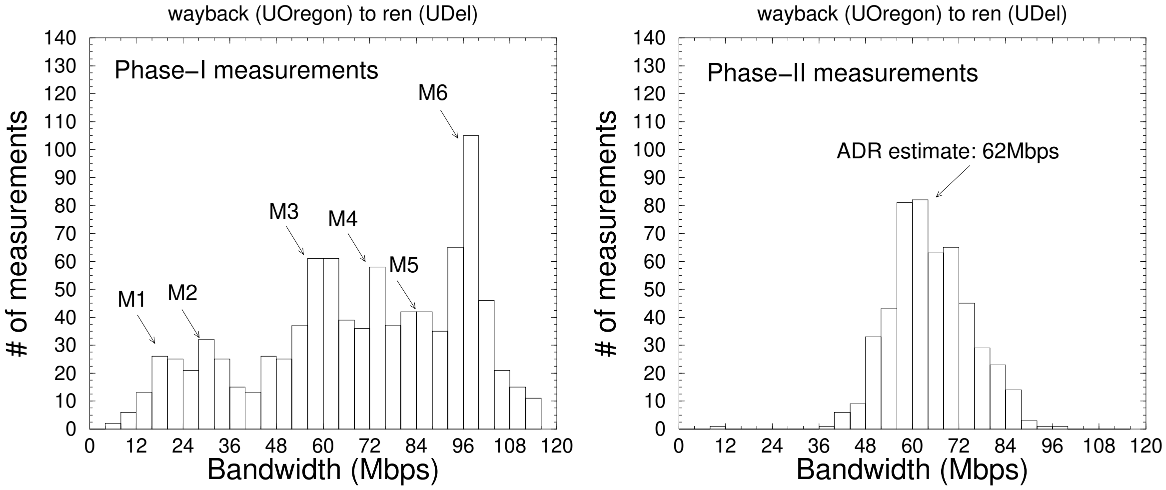


Figure 13: Histograms of Phase I and Phase II measurements in an Internet path.

the following *figure of merit* F_i for each Phase I mode:

$$F_i = 100\,\frac{S_i}{K_1}\,\frac{S_i}{W_i} \tag{19}$$

where K_1 is the number of measurements in Phase I. The fraction $\frac{S_i}{K_1}$ measures the 'strength' of mode M_i, while the fraction $\frac{S_i}{W_i}$ measures the 'narrowness' of mode M_i. Note that $F_i \leq 100$.

The final capacity estimate $\hat{C}$ that *pathrate* makes is the Phase I mode M_k that is larger than R, and has the maximum figure of merit F, i.e.,

$$k = \arg \max_{i=1 \ldots \Pi} F_i : \;\text{ such that } H_i > R \tag{20}$$

Finally, we mention a couple of special cases. First, if there is no Phase I mode that is larger than R, the ADR is the final capacity estimate. This happens when the path is quite lightly loaded. Second, if the ADR estimate R is near the same value as the estimate $\tilde{C}$ of the preliminary measurements, and additionally, if the Phase II measurements show little variation, the final capacity estimate is the ADR. This circumstance can happen when the narrow link is lightly loaded, and of much lower capacity than the rest of the links in the path.

To illustrate the operation of *pathrate*, Figure 13 shows the histograms of bandwidth measurements in Phase I and in Phase II across a path that connects *wayback* to *ren* (for more details about this path see §8). The local modes in Phase I are shown with arrows, while the ADR estimate in Phase II is about 62Mbps. Note that the histograms of these Figures are made with static 'bin-partitioning', and so they do not show the actual local modes that *pathrate* would estimate. The three Phase I modes that are larger than the ADR are centered around 74Mbps, 83Mbps, and 98Mbps. The figures of merit for each of these modes is 2.45, 2.21, and 3.55, respectively. Consequently, the final *pathrate* estimate will be the local mode that is centered at 98Mbps.

In the next section we examine the accuracy and robustness of the *pathrate* methodology, using a number of experiments across Internet paths of known capacities.

Host	Location
nova	Univ.Wisconsin, Madison WI
jhana	CAIDA, SDSC, San Diego CA
zamboni	CMU, Pittsburgh PA
ren	Univ.Delaware, Newark DE
wayback	Univ.Oregon, Eugene OR
hardy	Ensica, Toulouse FRANCE

Table 1: Measurement hosts and their locations.

8 Capacity measurements with *pathrate*

In this section, we present 12 capacity measurements using *pathrate* (version 2.1) for each path in a mesh of six hosts. The host names and their location are given in Table 8. Due to a firewall at Ensica, the measurements that involve that one site (*hardy*) could only be performed from and to the University of Delaware (*ren*). Consequently, we eventually had 22 instead of 30 paths to experiment with.

We performed measurements during several weekdays in July 2001. We chose weekdays, and specifically daytime hours at both ends of the path, in order to test *pathrate* during the heaviest load conditions possible in each path. We note that measurements would be more accurate if one used *pathrate* during lower load conditions (e.g., weekend nights). Our objective, however, was to check the accuracy and robustness of *pathrate* when measured paths carry their 'rush hour' type of cross traffic.

All hosts are connected to a Fast Ethernet interface, with the exception of *zamboni* which is connected to a 10Mbps Ethernet interface. Routes between the US-located hosts all used the Abilene backbone, which consists of OC-48 and OC-12 links. Local-area and access links at each site were Fast Ethernet, OC-3, or OC-12 links. Consequently, the capacity of all paths between US sites was limited to the Fast Ethernet *nominal* bandwidth (100Mbps), with the exception of paths that involved *zamboni*, for which the capacity was limited to Ethernet's *nominal* 10Mbps.

We note that these 10 and 100Mbps values are just 'nominal' capacities, and that the actual capacity of an Ethernet interface may deviate from these values by a small amount that depends on the specific Ethernet configuration and Network Interface Card vendor. Specifically, *zamboni* could not provide more than 9.8Mbps, independent of whether we were using TCP bulk transfers or UDP flooding. Similarly, the Fast Ethernet interface of *nova* would never give more than 96Mbps, even with UDP flooding at that LAN. The two paths between *hardy* and *ren* were limited by Ensica's 1Mbps access link. Those routes were passing through the Voicenet, Level3, OpenTransit, and Renater networks during our measurements. We are not aware of the link capacities in these networks, but we are confident that they are more than 1Mbps.

Figures 14 to 17 show *pathrate* measurements, which vary from extremely accurate and robust (e.g., the path from *ren* to *hardy*, or from *ren* to *zamboni*), to mostly accurate and robust but with occasional errors or fluctuations (e.g., the path from *zamboni* to *ren*, or from *rem* to *jhana*), while in some cases the measurements are often wrong or significantly different (e.g., the path from *nova* to *jhana*, or the path from *hardy* to *ren*). Note that we measured all paths to *zamboni* using the 'Quick-Estimate' *pathrate* mode, while paths in the reverse direction (from *zamboni*) used 'Quick-Estimate' in about 50% of the measurements. The rest of the paths were measured using Phase I and Phase II.

Due to space constraints we cannot go through each inaccurate measurement and analyze what caused it. In certain cases, the error was caused because the Capacity Mode was not detected during Phase I. This circumstance can occur in heavily loaded paths, or when the Capacity Mode is 'merged' with a SCDR mode. In other cases, there was more than one Phase I mode larger than the ADR (Asymptotic Dispersion Rate), and the figure of merit of the CM mode was lower than the figure of merit of one of the PNCMs or SCDR modes. In this latter case, *pathrate* displays all *possible capacity modes*, i.e., all Phase I modes that are larger or equal to the ADR, together with their figure of merit.

Normally, *pathrate* manages to give the capacity of a path as a *possible* mode, even though that mode is not always the tool's final choice. When a user observes multiple possible capacity values with similar figures of merit, he/she should run *pathrate* sometime later.

9 Conclusions

This paper focused on the dispersion of packet pairs and packet trains, in the presence of cross traffic. We investigated the effects of the packet train length and of the probing packet size, and showed that the dispersion of long packet trains is not related to the available bandwidth in the path, but to another bandwidth metric that we call Asymptotic Dispersion Rate. As an application of this study, we developed a capacity estimation methodology that has been implemented in a tool called *pathrate*.

The insight gained from the proposed dispersion models is also applicable to several areas of networking research, e.g., congestion control mechanisms, server selection algorithms, quality of service monitoring. The ADR metric, related to the utilization of all links in the path, may be a useful metric for monitoring the quality of service that the path offers, and it would be interesting to examine its dynamic variations over both short and long timescales. Also, it is important to examine the implications of our results for TCP. Specifically, TCP assumes that a packet burst sent by the sender will be dispersed at the 'bottleneck' link, causing ACKs to flow back to the sender with a rate that is related to the path's available bandwidth (the so called 'self-clocking TCP operation'). As we showed, however, the dispersion of a packet burst is not related to the available bandwidth, but to the ADR. This important issue requires further investigation.

Appendix I: Comments on the capacity and available bandwidth definitions

The definition of the path capacity C in Equation 1 is straightforward, but two issues require consideration. First, additional headers in layer-2 technologies can result in an IP-layer capacity estimate that is lower than the 'advertized' nominal bandwidth of a link. For instance, a 155 Mbps IP-over-ATM OC-3 link can normally provide about 130 Mbps at the IP layer. Second, in certain multichannel links the router performs hashing based on the destination address of the packet, or based on the 5-tuple header fields, in order to determine the specific sub-link across which the packet should be forwarded. In that case, all probing packets will go across the same sub-link, and so a measurement tool such as *pathrate* will be able to estimate the capacity of only that sub-link. This

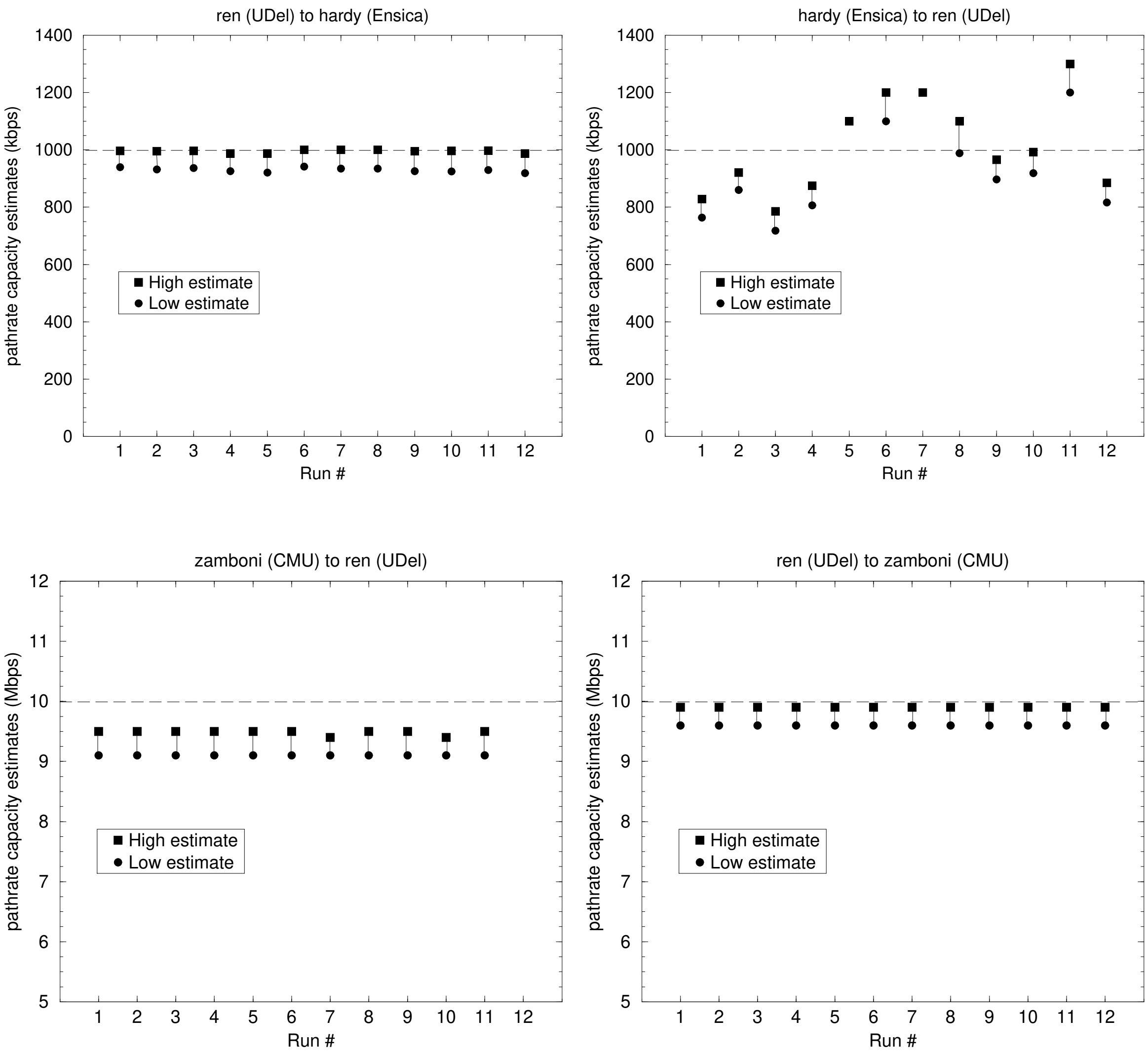


Figure 14: Twelve *pathrate* measurements for each of four paths.

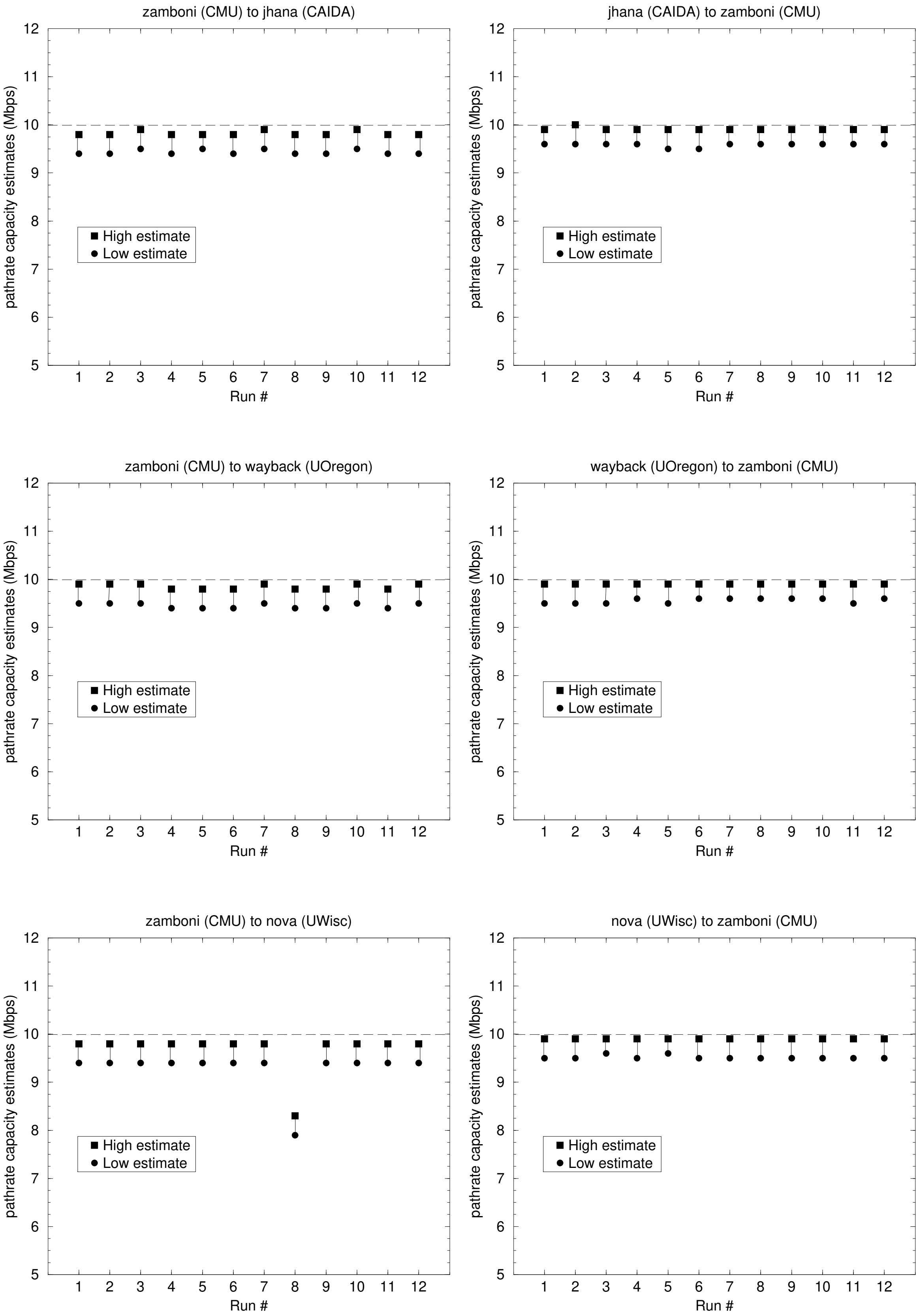


Figure 15: Twelve *pathrate* measurements for each of six paths.

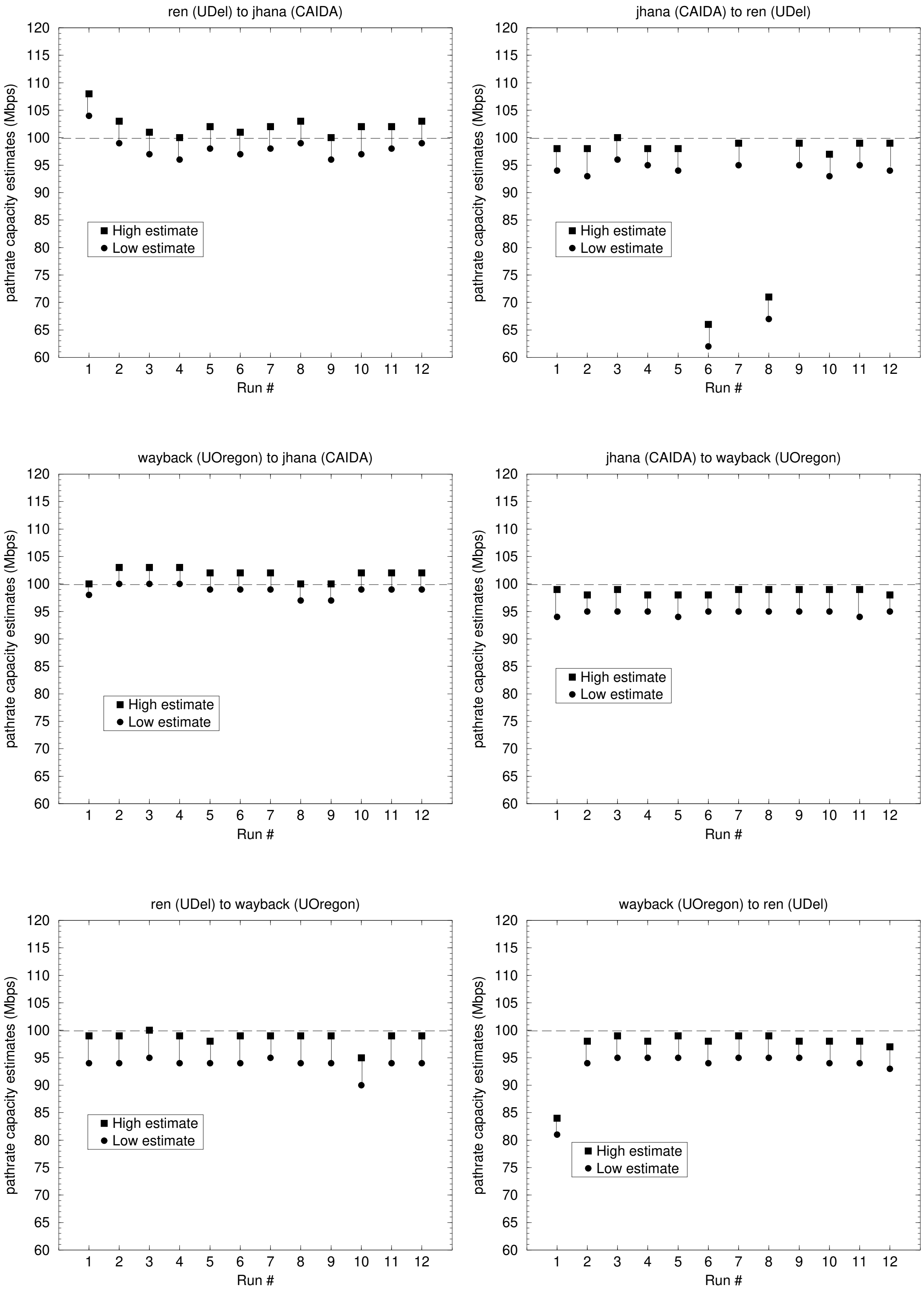


Figure 16: Twelve *pathrate* measurements for each of six paths.

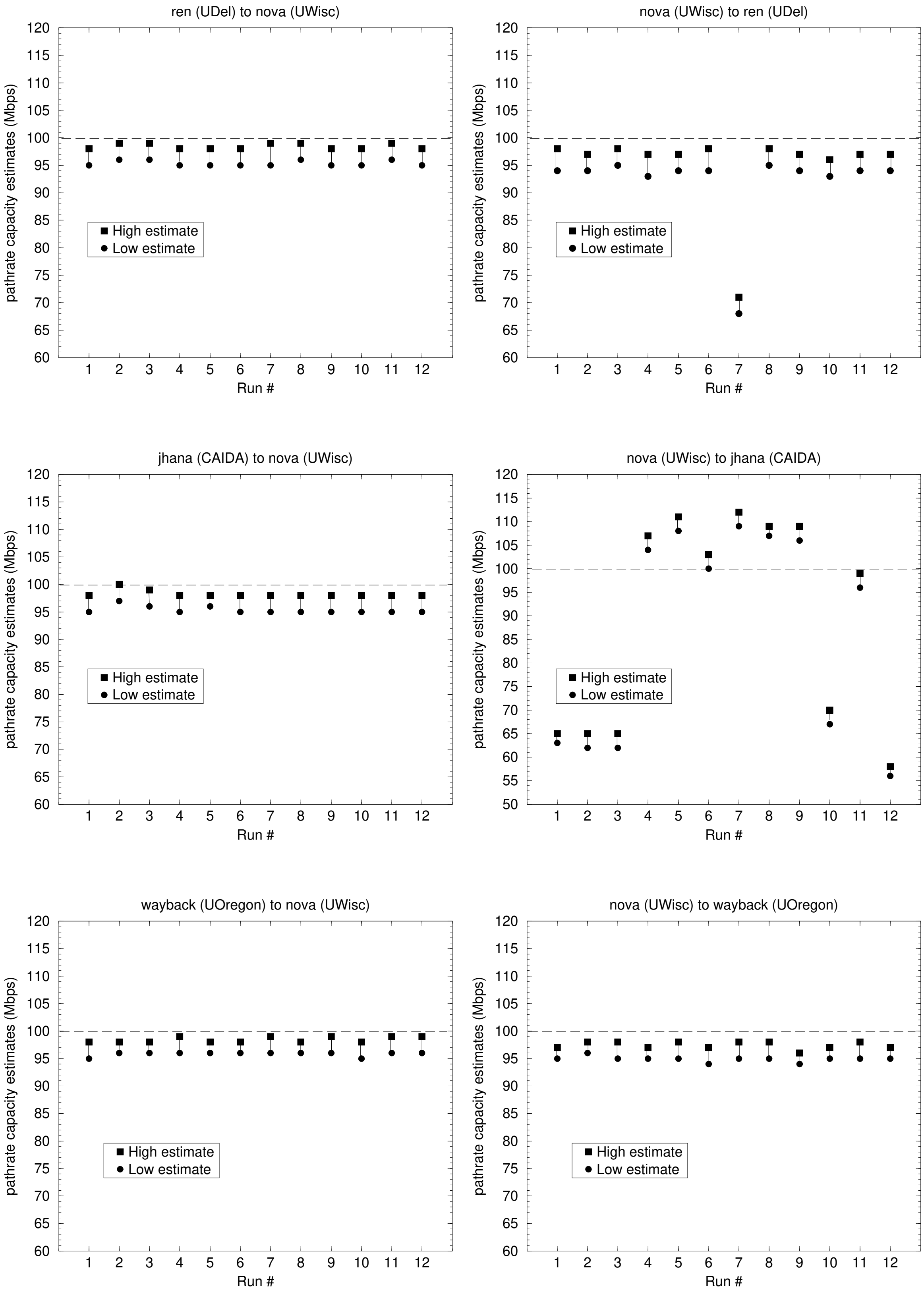


Figure 17: Twelve *pathrate* measurements for each of six paths.

value is also, however, the maximum throughput that a certain IP microflow would be able to get in that path.

Regarding the available bandwidth A, defined in Equation 2, we make the following remarks. Equation 2 is based on a fluid model of the network traffic, and so the utilization of the tight link can be up to 100% before the path is saturated. Considering the traffic burstiness of Internet traffic, a utilization of almost 100% would imply large queueing delays, and in the case of limited buffers, a certain loss rate. Consequently, the available bandwidth A of Equation 2 is an upper bound on the available bandwidth in practice.

A second point is that A should *not* be interpreted as the maximum throughput that an individual application or a certain transport protocol could obtain across the path. Obviously, a flow can get more than A if it is *congestion unresponsive*, i.e., if it does not reduce its rate in the face of packet loss. Also, a transport protocol such as TCP may get a higher or lower throughput than A, depending on operating conditions. For instance, even if the available bandwidth A is zero in a saturated path, a TCP connection would still obtain some throughput, because it shares the tight link's capacity with the rest of the TCP connections in the path. A TCP connection may also be unable to obtain A because of TCP's required multiplicative reduction of the congestion window upon the loss of a packet.

Appendix II: Estimation of local modes in a multimodal distribution

In this appendix, we describe a numerical algorithm that detects all local modes in a set of measurements. The algorithm bares some similarities with a numerical procedure described in [20]. In general, the underlying distribution that generates a set of measurements is multimodal, with the local modes being the local maxima of the corresponding probability density function. Since we only have a finite set of measurements, we can only approximate the probability density function, and must use heuristics in detection of local maxima.

Specifically, suppose that we have a set of K measurements. We order the measurements in increasing sequence $x_1 \leq x_2 \leq \ldots \leq x_K$. The *rank* of a measurement is $r(x_m) = m$; ties for equal measurements are broken in an arbitrary manner. An important input parameter in the mode detection algorithm is the *resolution* (or *bin width*) ω. We estimate each local mode as a range of width ω.

The following algorithm estimates iteratively all local modes $\{M_1, M_2, \ldots, M_\Pi\}$ in the distribution that generated these K measurements. For each mode M_l, the following *modal characteristics* are produced (see Figure 12):

- The range of the *central modal bin*, $R_l = [x_l^{c-}, x_l^{c+}]$. The width of this range is no larger than ω.
- The number of measurements in R_i, $S_l = r(x_l^{c+}) - r(x_l^{c-}) + 1$.
- The range of the *modal bell*, $W_l = [x_l^{b-}, x_l^{b+}]$. The modal bell includes all measurements in local mode M_l that are distributed around the central modal bin R_l in a 'bell-shaped' fashion.
- The number of measurements in the modal bell W_l, denoted as $B_l = r(x_l^{b+}) - r(x_l^{b-}) + 1$.

The values x_l^{c-}, x_l^{c+}, x_l^{b-}, and x_l^{b+} for each local mode M_l are determined with the following algorithm.

Initially, all K measurements are unmarked, and l=1.

1. First, we estimate the range R_l and the number of measurements S_l in the central modal bin of mode M_l. The defining property of the central modal bin is that it includes *the maximum number of consecutive and unmarked measurements in a range with width that is at most* ω. More formally,

$$S_l = \max_{1 \leq i \leq K} \left\{ t = j - i + 1, \quad \text{such that } j = \max_{i \leq j \geq K} \{x_j \leq x_i + \omega\}, \quad \text{and } x_i \ldots x_j : \text{ unmarked} \right\} \tag{21}$$

   x_l^{c-} and x_l^{c+} are the values of x_i and x_j, respectively, that maximize the previous expression for S_l.

2. Next, we find the right extent of the modal bell of M_l, i.e., we estimate x_l^{b+}. This part of the algorithm is iterative. In each step, we determine *the window at the right of the currently rightmost bin of* M_l *that includes the maximum number of measurements in a range with width that is at most* ω. If this window has more measurements than the rightmost bin of M_l, it is outside the modal bell of M_l, and so it belongs to a different local mode. Otherwise, we set that bin as the rightmost bin of M_l, and repeat the iteration. The rightmost modal bin at the start of the iteration is the central modal bin.

   More formally, suppose that the rightmost modal bin at iteration step k is $[x_i, x_j]$ with $s = j - i + 1$ measurements. We then find the bin $[x_m, x_n]$ that resides at the right of $[x_i, x_j]$ and that overlaps with $[x_i, x_j]$, such that

$$\hat{s} = \max_{i+1 \leq m \leq j+1} \left\{ t = n - m + 1, \quad \text{such that } \; n = \max_{n \leq m \geq K} \{x_n \leq x_m + \omega\} \right\} \tag{22}$$

   If $\hat{s} < s$, the bin $[x_m, x_n]$ is in the modal bell of M. In that case, we set $m = i$, $n = j$, $s = \hat{s}$, and repeat the iteration. Otherwise, $[x_i, x_j]$ marks the rightmost bin of the modal bell of M, and we set $x_l^{b+} = x_j$.

3. Using the same approach as in the previous step, we determine the leftmost bin of the modal bell of M_l, and compute x_l^{b-}.

4. We mark all measurements in the modal bell of M_l (from x_l^{b-} to x_l^{b+}). Then, set $l = l + 1$ and repeat the iteration (i.e., go to step 1). The algorithm terminates when all measurements are marked.

Note that when searching for the next local mode, marked measurements will be skipped from the estimation of the central modal bin, which implies that the estimated local modes cannot have overlapping central modal bins, even though their modal bells may overlap.

Acknowledgments

The authors would like to thank K. C. Claffy for her comments on this paper. We are also grateful to the following people for providing us with computer accounts: Paul Amer (Ensica), Andy Myers and Hui Zhang (CMU), and David Meyer (Univ-Oregon).

References

[1] *Network Simulator.* http://www-mash.cs.berkeley.edu/ns/.

[2] J. C. Bolot, "Characterizing End-to-End Packet Delay and Loss in the Internet," In *Proceedings ACM SIGCOMM*, pp. 289–298, 1993.

[3] L. S. Brakmo and L.L.Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet," *IEEE Journal on Selected Areas of Communications*, vol. 13, no. 8, , October 1995.

[4] R. Caceres, N.Duffield, and A.Feldmann, "Measurement and Analysis of IP Network Usage and Behavior," *IEEE Communications Magazine*, pp. 144–152, May 2000.

[5] R. Carter and M.E.Crovella, "Measuring Bottleneck Link Speed in Packet-Switched Networks," *Performance Evaluation*, vol. 27,28, pp. 297–318, 1996.

[6] C. Dovrolis, P. Ramanathan, and D. Moore, "What do Packet Dispersion Techniques Measure?," In *Proceedings of IEEE INFOCOM*, April 2001.

[7] A. Downey. *clink: a Tool for Estimating Internet Link Characteristics.* http://rocky.wellesley.edu/downey/clink/, June 1999.

[8] A. Downey, "Using Pathchar to Estimate Internet Link Characteristics," In *ACM SIGCOMM*, September 1999.

[9] J. Hoe, "Improving the Start-up Behavior of a Congestion Control Scheme for TCP," In *Proceedings ACM SIGCOMM*, September 1996.

[10] V. Jacobson, "Congestion Avoidance and Control," In *Proceedings ACM SIGCOMM*, pp. 314–329, September 1988.

[11] V. Jacobson. *pathchar: A Tool to Infer Characteristics of Internet Paths.* ftp://ftp.ee.lbl.gov/pathchar/, April 1997.

[12] S. Keshav, "A Control-Theoretic Approach to Flow Control," In *Proceedings ACM SIGCOMM*, September 1991.

[13] K. Lai and M.Baker, "Measuring Bandwidth," In *Proceedings IEEE INFOCOM*, April 1999.

[14] K. Lai and M.Baker, "Measuring Link Bandwidths Using a Deterministic Model of Packet Delay," In *Proceedings ACM SIGCOMM*, September 2000.

[15] B. A. Mah. *pchar: a Tool for Measuring Internet Path Characteristics.* http://www.employees.org/~bmah/Software/pchar/, June 2000.

[16] S. McCreary and K. C. Claffy, "Trends in Wide Area IP Traffic Patterns," Technical report, CAIDA, February 2000.

[17] T. Oetiker. *MRTG: Multi Router Traffic Grapher.* http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html.

[18] V. Paxson, *Measurements and Analysis of End-to-End Internet Dynamics*, PhD thesis, University of California, Berkeley, April 1997.

[19] V. Paxson, "End-to-End Internet Packet Dynamics," *IEEE/ACM Transaction on Networking*, vol. 7, no. 3, pp. 277–292, June 1999.

[20] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes*, Cambridge University Press, 1986.

[21] K. Thompson, G. J. Miller, and R. Wilder, "Wide-Area Internet Traffic Patterns and Characteristics," *IEEE Network*, pp. 10–23, November 1997.

[22] UUNET. *UUNET Technologies.* http://www.uunet.net/, November 2000.