An Empirical Evaluation of Wide-Area Internet Bottlenecks

Aditya Akella, Srinivasan Seshan
Carnegie Mellon University
Pittsburgh, PA 15213
{aditya,srini+}@cs.cmu.edu

Anees Shaikh
IBM T.J. Watson Research Center
Hawthorne, NY 15213
aashaikh@watson.ibm.com

ABSTRACT

Conventional wisdom has been that the performance limitations in the current Internet lie at the edges of the network – *i.e* last mile connectivity to users, or access links of stub ASes. As these links are upgraded, however, it is important to consider where new bottlenecks and hot-spots are likely to arise. In this paper, we address this question through an investigation of *non-access* bottlenecks. These are links within carrier ISPs or between neighboring carriers that could *potentially* constrain the bandwidth available to long-lived TCP flows. Through an extensive measurement study, we discover, classify, and characterize bottleneck links (primarily in the U.S.) in terms of their location, latency, and available capacity.

We find that about 50% of the Internet paths explored have a non-access bottleneck with available capacity less than 50 Mbps, many of which limit the performance of well-connected nodes on the Internet today. Surprisingly, the bottlenecks identified are roughly equally split between intra-ISP links and peering links between ISPs. Also, we find that low-latency links, both intra-ISP and peering, have a significant likelihood of constraining available bandwidth. Finally, we discuss the implications of our findings on related issues such as choosing an access provider and optimizing routes through the network. We believe that these results could be valuable in guiding the design of future network services, such as overlay routing, in terms of which links or paths to avoid (and how to avoid them) in order to improve performance.

Categories and Subject Descriptors

C.2 [**Computer Systems Organization**]: Computer-Communication Networks; C.2.5 [**Computer-Communication Networks**]: Local and Wide-Area Networks

General Terms

Measurement, Performance

1. INTRODUCTION

A common belief about the Internet is that poor network performance arises primarily from constraints at the edges of the network. These narrow-band access links (e.g., dial-up, DSL, etc.) limit the ability of applications to tap into the plentiful bandwidth and negligible queuing available in the interior of the network. As access technology evolves, enterprises and end-users, given enough resources, can increase the capacity of their Internet connections by upgrading their access links. The positive impact on overall performance may be insignificant, however, if other parts of the network subsequently become new performance bottlenecks. Ultimately, upgrades at the edges of the network may simply shift existing bottlenecks and hot-spots to other parts of the Internet. In this study, we consider the likely location and characteristics of future bottleneck links in the Internet. Such information could prove very useful in the context of choosing intermediate hops in overlay routing services [1, 31] or interdomain traffic engineering, and also to customers considering their connectivity options.

Our objective is to investigate the characteristics of links within or between carrier ISP networks that could *potentially* constrain the bandwidth available to long-lived TCP flows, called *non-access* bottleneck links. Using a large set of network measurements, we seek to discover and classify such links according to their location in the Internet hierarchy and their estimated available capacity. By focusing on interior links, we try to avoid access links near the source and destination (*i.e.*, first-mile and last-mile hops), as these are usually obvious bottlenecks in the current Internet. This paper makes two primary contributions: 1) a methodology for measuring bottlenecks links and 2) a classification of existing bottleneck links.

Methodology for measuring non-access Internet bottleneck links: Our main challenge in characterizing Internet bottlenecks is to measure paths that are representative of typical routes in the Internet, while avoiding biases due to a narrow view of the network from few probe sites, or probes which themselves are poorly connected. Our results are based on measurements from 26 geographically diverse probe sites located primarily in the U.S., each with very high speed access to the Internet. We measure paths from these sites to a carefully chosen set of destinations, including paths to all Tier-1 ISPs, as well as paths to a fraction of Tier-2, Tier-3, and Tier-4 ISPs, resulting in 2028 paths in total. In addition, we identify and measure 466 paths passing through public Internet exchange points in order to explore the common perception that public exchanges are a major source of congestion in the Internet.

A second challenge lies in actually measuring the bottleneck link and reporting its available bandwidth and location. Due to the need for control at both ends of the path, we were unable to leverage any of the existing tools to measure the available bandwidth. Hence, we developed a tool, *BFind*, which measures available capacity using a bandwidth probing technique motivated by TCP's behavior, and operates in a single-ended mode.

Classification of bottleneck links: We apply our measurement methodology to empirically determine the locations, estimated avail-

This work was supported by the Army Research Office under grant number DAAD19-02-1-0389. Additional support was provided by IBM.

IMC'03, October 27–29, 2003, Miami Beach, Florida, USA.

able bandwidth, and delay of non-access bottleneck links. In classifying these links, we draw extensively on recent work on characterizing AS relationships [33, 8]. Our results show that nearly half of the paths we measured have a non-access bottleneck link with available capacity less than 50 Mbps. Moreover, the percentage of observed paths with bottlenecks grows as we consider paths to lower-tier destinations. Surprisingly, the bottlenecks identified are roughly equally split between intra-ISP links and peering links between ISPs. Also, we find that low-latency links, both within and between ISPs have a significant probability of constraining available bandwidth. Of the paths through public exchanges that had a bottleneck link, the constrained link appeared at the exchange point itself in nearly half the cases.

Our work complements and extends the large body of work on measuring and characterizing the Internet. In particular, several recent efforts have focused on end-to-end Internet path properties, as these can have a significant impact on application performance and transport protocol efficiency. For example, recent wide-area measurement studies focus on performance metrics like delay, loss, and bandwidth [23, 36], packet reordering [15], routing anomalies [24, 11, 32], and path stability [16]. In addition, a number of measurement algorithms and tools have been developed to measure the capacity or available bandwidth of a path (see [13] for examples). Our focus is on identifying and characterizing potential bottleneck links through the measurement of a wide variety of Internet paths.

We believe that our observations provide valuable insights into the location and nature of performance bottlenecks in the Internet, and in some cases, address common impressions about constraints in the network. In addition, we hope that our work could help improve the performance of future network protocols and services in terms of which bottlenecks to avoid (and how to avoid them).

In the next section we describe our measurement methodology with additional details on our choice of paths and the design and validation of BFind. Section 3 presents our observations of non-access bottlenecks, and Section 4 offers some discussion about the implications of our findings. In Section 5 we briefly review related work in end-to-end Internet path characterization and measurement tools. Finally, Section 6 summarizes the paper.

2. MEASUREMENT METHODOLOGY

The Internet today is composed of an interconnected collection of Autonomous Systems (ASes). These ASes can be roughly categorized as carrier ASes (e.g. ISPs and transit providers) and stub ASes (end-customer domains). Our goal is to measure the characteristics of potential performance bottlenecks that end-nodes encounter that are not within their own control. To perform this measurement we need to address several issues, described below.

2.1 Choosing a Set of Traffic Sources

Stub ASes in the Internet are varied in size and connectivity to their carrier networks. Large stubs, *e.g.* large universities and commercial organizations, are often multi-homed and have high speed links to all of their providers. Other stubs, *e.g.* small businesses, usually have a single provider with a much slower connection.

At the core of our measurements are traffic flows between a set of sources, which are under our control, and a set destinations which are random, but chosen so that we may measure typical Internet paths (described in detail in Section 2.2). However, it is difficult to use such measurements when the source network or its connection to the upstream carrier network is itself a bottleneck. Hence, we choose to explore bottleneck characteristics by measuring paths from well-connected end-points, i.e. stub ASes with very high speed access to their upstream providers. Large commercial and academic organizations are example of such end-points. In addition to connectivity of the stub ASes, another important factor in choosing sources is diversity, both in terms of geographic locations, and carrier networks. This ensures that the results are not biased by repeated measurement of a small set of bottlenecks links.

We use hosts participating in the PlanetLab project [26], which provides access to a large collection of Internet nodes that meet our requirements. PlanetLab is a Internet-wide testbed of multiple high-end machines located at geographically diverse locations. Most of the machines available this time are in large academic institutions and research centers in the U.S. and Europe and have very high-speed access to the Internet. Note that although our traffic sources are primarily at universities and research labs, we do not measure the paths *between* these nodes. Rather, our measured paths are chosen to be representative of typical Internet paths (e.g., as opposed to paths on Internet2).

Initially, we chose one machine from each of the PlanetLab sites as the initial candidate for our experiments. While it is generally true that the academic institutions and research labs hosting PlanetLab machines are well-connected to their upstream providers, we found that the machines themselves are often on low-speed local area networks. Out of the 38 PlanetLab sites operational at the outset of our experiments, we identified 12 that had this drawback. In order to ensure that we can reliably measure non-access bottlenecks, we did not use these 12 machines in our experiments.

	tier-1	*tier-2*	tier-3	tier-4
Total #unique providers	11	11	15	5
Avg. #providers per PlanetLab source	0.92	0.69	0.81	0.10

Table 1: First-hop connectivity of the PlanetLab sites

The unique upstream providers and locations of the remaining 26 PlanetLab sites are shown in Table 1 and Figure 1(a), respectively. We use a hierarchical classification of ASes into four *tiers* (as defined by the work in [33]) to categorized the upstream ISPs of the different PlanetLab sites. ASes in tier-1 of the hierarchy, for example AT&T and Sprint, are large ASes that do not have any upstream providers. Most ASes in tier-1 have peering arrangements with each other. Lower in the hierarchy, tier-2 ASes, including Savvis, Time Warner Telecom and several large national carriers, have peering agreements with a number of ASes in tier-1. ASes in tier-2 also have peering relationships with each other, however, they do not generally peer with any other ASes. ASes in tier-3, such as Southwestern Bell and Turkish Telecomm, are small regional providers that have a few customer ASes and peer with a few other similar small providers. Finally, the ASes in tier-4, for example rockynet.com, have very few customers and typically no peering relationships at all [33].

2.2 Choosing a Set of Destinations

We have two objectives in choosing paths to measure from our sources. First, we want to choose a set of network paths that are representative of typical paths taken by Internet traffic. Second, we wish to explore the common impression that public network exchanges, or NAPs (network access points), are significant bottlenecks. Our choice of network paths to measure is equivalent to choosing a set of destinations in the wide-area as targets for our testing tools. Below, we describe the rationale and techniques for choosing test destinations to achieve these objectives.

2.2.1 Typical Paths

(a) Sources

(b) Destinations (mapped to the closest source)

Figure 1: Locations of PlanetLab sources (a) and destinations (b): Each destination location is identified by the PlanetLab source with minimum delay to the destination. Three of our sources and seven destinations are located in Europe (shown in the inset). The size of the dots is proportional to the number of sites mapped to the same location.

Most end-to-end data traffic in the Internet flows *between* stub networks. One way to measure typical paths would have been to select a large number of stub networks as destinations. However, the number of such destinations needed to characterize properties of representative paths would make the measurements impractical. Instead, we use key features of the routing structure of the Internet to help choose a smaller set of destinations for our tests.

Traffic originated by a stub network subsequently traverses multiple intermediate autonomous systems before reaching the destination stub network. Following the definitions of AS hierarchy presented in [33] (and summarized earlier), flows originated by typical stub source networks usually enter a tier-4 or a higher tier ISP. Beyond this, the flow might cross a sequence of multiple links between ISPs and their higher-tier upstream carriers (*uphill path*). At the end of this sequence, the flow might cross a single peering link between two peer ISPs after which it might traverse a *downhill path* of ASes in progressively lower tiers to the final destination, which is also usually a stub. This form of routing, arising out of BGP policies, is referred to as *valley-free* routing. We refer to the portion of the path taken by a flow that excludes links within the stub network at either end of the path, and the access links of either of the stub networks, as the *transit path*.

Clearly, non-access bottlenecks lie in the transit path to the destination stub network. Specifically, the bottleneck for any flow could lie either (1) *within* any one of the ISPs in the uphill or the downhill portion of the transit path or (2) *between* any two distinct ISPs in either portion of the transit path. Therefore, we believe that measuring the paths between our sources and a wide variety of different ISPs would provide a representative view of the bottlenecks that these sources encounter.

Due to the large number of ISPs, it is impractical to measure the paths between our sources and all such carrier networks. However, the *reachability* provided by these carriers arises directly from their position in the AS hierarchy. Hence, it is more likely that a path will pass through one or two tier-1 ISPs than a lower tier ISP. Hence, we test paths between our sources and *all* tier-1 ASes. To make our measurements practical, we only test the paths between our sources and a fraction of the tier-2 ISPs (chosen randomly). We measure an even smaller fraction of all tier-3 and tier-4 providers. The number of ISPs we chose in each tier is presented in Table 2.

	tier-1	*tier-2*	*tier-3*	*tier-4*
Number tested	20	18	25	15
Total in the Internet [33]	20	129	897	971
Percentage tested	100	14	3	1.5

Table 2: Composition of the destination set

In addition to choosing a target AS, we need to choose a target IP address within the AS for our tests. For any AS we choose, say `<isp>`, we pick a router that is a few (2-4) IP hops away from the machine `www.<isp>.com` (or `.net` as the case maybe). We confirm this router to be *inside* the AS by manually inspecting the DNS name of the router where available. Most ISPs name their routers according to their function in the network, *e.g.* edge (chi-edge-08.inet.qwest.net) or backbone (sl-bb12-nyc-9-0.sprintlink.net), routers. The function of the router can also be inferred from the names of routers adjacent to it. In addition, we double check using the IP addresses of the carrier's routers along the path to `www.<isp>.com` (typically there is a change in the subnet address close to the web server). We measure the path between each of the sources and the above IP addresses. The diversity of the sources in terms of geography and upstream connectivity ensures that we sample several links with the ISPs. The geographic location of the destinations is shown in Figure 1(b). Each destination's location is identified by that of the traffic source with the least delay to it.

2.2.2 Public Exchanges

The carrier ASes in the Internet peer with each other at a number of locations throughout the world. These peering arrangements can be roughly categorized as public exchanges, or NAPs, (e.g., the original 4 NSF exchanges) or private peering (between a pair of ISPs). One of the motivations for the deployment of private peering has been to avoid the perceived congestion of public exchanges. As part of our measurements, we are interested in exploring the accuracy of this perception. Therefore, we need a set of destinations to test paths through these exchanges.

We selected a set of well-known NAPs, including Worldcom MAE-East, MAE-West, MAE-Central, SBC/Ameritech AADS and PAIX in Palo Alto. For each NAP, we gather a list of low-tier (*i.e.*, low in the hierarchy) customers attached to the NAP. The customers are typically listed at the Web sites of the NAPs. As in each of the above cases, we use the hierarchy information from [33] to determine if a customer is small. Since these customers are low tier, there is a reasonable likelihood that a path to these customers from any source passes through the corresponding NAP (*i.e.*, they are not multihomed to the NAP and another provider). We then find a small set of addresses from the address block of each of these customers that are reachable via traceroute. We use the complete BGP table dump from the Oregon route server [30, 29] to obtain the address space information for these customers.

Next, we use a large set of public traceroute servers (153 traceroute sources from 71 providers) [34], and trace the paths from these servers to the addresses identified above using a script to automate finding and accessing working servers. For each NAP, we select all paths which appear to go through the NAP. For this purpose, we use the router DNS names as the determining factor. Specifically, we look for the name of the NAP to appear in the DNS name of any router in the path. From the selected paths, we pick out the routers one-hop away (both a predecessor and a successor) from the router identified to be at the NAP and collect their IP addresses. This gives us a collection of IP addresses for routers that could potentially be used as destinations to measure paths passing through NAPs.

However, we still have to ensure that the paths do in fact traverse the NAP. For this, we run traceroutes from each of our PlanetLab sources to each of the predecessor and successor IP addresses identified above. For each PlanetLab source, we record the subset of these IP addresses whose traceroute indicates a path through the corresponding NAP. The resulting collection of IP addresses is used as a destination set for the PlanetLab source.

2.3 Bottleneck Identification Tool – *BFind*

Next, we need a tool that we can run at the chosen sources that will measure the bottleneck link along the selected paths. We define the *bottleneck* as the link in the path where the available bandwidth (*i.e.,* left-over capacity) to a TCP flow is the minimum. Notice that a particular link being a bottleneck does not necessarily imply that the link is heavily utilized or congested. In addition, we would like the tool to report the available bandwidth, latency and location (i.e. IP addresses of endpoints) of the bottleneck along a path. In this section, we describe the design and operation of our bottleneck identification tool – *BFind*.

2.3.1 BFind Design

BFind's design is motivated by TCP's property of gradually filling up the available capacity based on feedback from the network. First, BFind obtains the propagation delay of each hop to the destination. For each hop along the path, the minimum of the (non-negative) measured delays along the hop is used as an estimate for the propagation delay on the hop [1]. The minimum is taken over delay samples from 5 traceroutes.

After this step, BFind starts a process that sends UDP traffic at a low sending rate (2 Mbps) to the destination. A trace process also starts running concurrently with the UDP process. The trace process repeatedly runs traceroutes to the destination. The hop-by-hop delays obtained by each of these traceroutes are combined with the raw propagation delay information (computed initially) to obtain rough estimates of the queue lengths on the path. The trace process concludes that the queue on a particular hop is *potentially* increasing if across 3 consecutive measurements, the queuing delay on the hop is at least as large as the maximum of 5ms and 20% of the raw propagation delay on the hop. This information, computed for each hop by the trace process, is constantly accessible to the UDP process. The UDP process uses this information (at the completion of each traceroute) to adjust its sending rate as described below.

If the feedback from the trace process indicates no increase in the queues along any hop, the UDP process increases its rate by 200 Kbps (the rate change occurs once per feedback event, *i.e.*, per traceroute). Essentially, BFind emulates the increase behavior of TCP, albeit more aggressively, while probing for available bandwidth. If, on the other hand, the trace process reports an increased delay on any hop(s), BFind flags the hop as being a potential bottleneck and the traceroutes continue monitoring the queues. In addition, the UDP process keeps the sending rate steady at the current value until one of the following things happen: (1) The hop continues to be flagged by BFind over *consecutive* measurements by the trace process and a threshold number (15) of such observations are made for the hop. (2) The hop has been flagged a threshold number of times in total (50). (3) BFind has run for a pre-defined maximum amount of total time (180 seconds). (4) The trace process reports that there is no queue build-up on *any* hop implying that the increasing queues were only a transient occurrence.

In the first two cases, BFind quits and identifies the hop responsible for the tool quitting as being the bottleneck. In the third case, BFind quits without providing any reliable conclusion about bottlenecks along the path. In the fourth case, BFind continues to increase its sending rate at a steady pace in search of the bottleneck.

If the trace process observes that the queues on the first 1-3 hops from the source are building, it quits immediately, to avoid flooding the local network (The first 3 hops almost always encompass all links along the path that belong to the source stub network). Also, we limit the maximum send rate of BFind to 50Mbps to make sure that we do not use too much of the local area network capacity at the PlanetLab sites. Hence, we only identify bottlenecks with < 50Mbps of available capacity. If BFind quits due to these exceptional conditions, it does not report any bottlenecks.

By its very nature, BFind not only identifies the bottleneck link in a path, but also estimates the available capacity at the bottleneck equal to the send rate just before the tool quit (upon identifying the bottleneck reliably). For paths on which no bottlenecks have been identified, BFind outputs a lower bound on the available capacity.

Notice that in several respects, the operation of BFind is similar to TCP Vegas's [3] rate-based congestion control. However, our sending rate modification is different than Vegas for two reasons. First, we actually wanted to ensure that the bottleneck link experiences a reasonable amount of queuing in order to come to a definitive conclusion. Therefore, BFind needs to be more aggressive than Vegas. Second, the feedback loop of the trace process is much slower than Vegas. As a result, BFind lacks tight transmit control to use Vegas' more gradual increase/decrease behavior.

One obvious drawback with this design is that BFind is a relatively heavy-weight tool that sends a large amount of data. This makes it difficult to find a large number of sites willing to host such experiments. BFind is not suitable for continuous monitoring of available bandwidth, but rather for short duration measurements.

Since BFind may induce losses at the bottleneck, it is possible that other congestion controlled traffic may react and slow down. This may cause the queuing delays to vanish and BFind to possibly ramp up its transmission speed causing BFind to predict higher than the capacity really available to TCP. As a result, the available bandwidth reported by BFind is likely to be higher than the throughput that would be achieved by a TCP flow on the same path. In other words, BFind may report something between the TCP fair share rate on the path and the raw capacity of the path. In addition, we do not expect the loss of BFind's UDP probe packets to affect the results except in the unlikely case of persistent or pathological losses.

2.3.2 BFind Operation: An Example

Figure 2 shows examples of the operation of BFind. In Figure 2(a), BFind is run between `planet1.scs.cs.nyu.edu` (NYU) and `r1-srp5-0.cst.hcvlny.cv.net` (Cable Vision Corp, AS6128, tier-3). As BFind ramps up its transmission rate, the delay of hop 6 link between `at-bb4-nyc-0-0-0-OC3.appliedtheory.net` and `jfk3-core5-s3-7.atlas.algx.net`) begins to increase. BFind freezes its sending rate as the delay on this hop increases persistently. Finally, BFind identifies this hop as bottleneck with about 26Mbps of available capacity. This link also had a raw latency of under 0.5ms. The maximum queuing delay observed on

[1] If the difference in the delay to two consecutive routers along a path is negative, then the delay for the corresponding hop is assumed to be zero

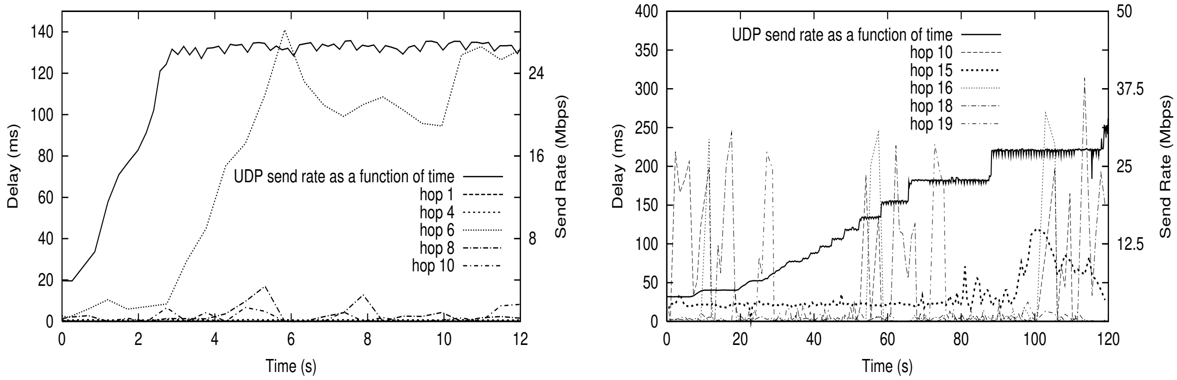


Figure 2: The operation of BFind: In (a), BFind identifies hop 6 as the bottleneck. In (b), BFind identifies hop 15 as the bottleneck, although this could potentially be a false positive.

this bottleneck link was about 140ms.

Figure 2(b) presents a potential false-positive. Running between `planetlab1.lcs.mit.edu` (MIT) and `Amsterdam1.ripe.net` (RIPE, tier-2), BFind observes the delays on various hops along the path increasing on a short time-scale causing BFind to freeze its UDP send rate quite often. The delay on hop 15 increases reasonably steadily starting at around 80 secs. This steady increase causes BFind to conclude that hop 15 was the bottleneck. However, it is possible that, similar to the other hops, this congestion was transient too, as indicated by a dip in the delay on hop 15 after 100secs.

As Figure 2(b) shows, we cannot entirely rule out the possibility of false-positives in our analysis. But we do believe that our choices of the set thresholds for BFind, chosen empirically after experimenting with various combinations while looking for minimal error in estimation, would keep the overall number of false positives reasonably low. Notice that false negatives might occur in BFind only when the path being explored was very free of congestion during the run, while being persistently overloaded at other times. Given that BFind runs for at least 30secs, and sometimes up to 150secs, we think that false negatives are unlikely.

2.3.3 BFind Validation

In this section we present the results from a limited set of experiments to evaluate the available bandwidth estimation and the bottleneck location estimation accuracies of BFind. To validate the available bandwidth estimate produced by BFind, we compare it against Pathload [13], a widely-used available bandwidth measurement tool. Pathload estimates the range of available bandwidth on the path between two given nodes. Since measurements are taken at either end of the path, control is necessary at both end-hosts.

To validate the bottleneck location estimation of BFind, we compare it with Pipechar [21], which operates similarly to tools like pathchar [12] and pchar [18]. Pipechar outputs the path characteristics from a given node to any arbitrary node in the Internet. For each hop on the path, Pipechar computes the raw capacity of the link, as well as an estimate of the available bandwidth and link utilization. We consider the hop identified as having the least available bandwidth to be the bottleneck link output by Pipechar and compare it with the link identified by BFind. We also compare the available bandwidth estimates output by BFind and Pipechar.

For these experiments, we perform transfers from a machine located at a commercial data center in Chicago, IL to a large collection of destinations. Some of these destinations are nodes in the PlanetLab infrastructure and hence we have control over both ends of the path when probing these destinations. The other destinations are randomly picked from the set of 68 addresses we probe (summarized in Table 2). In probing the path to the latter destinations, we do not have control over the destination end of the path. In total, we probe 30 destinations.

A small sample of the results of our tests are presented in Table 3. These samples are chosen to represent the three coarse grained classes of the bandwidth available on the paths we probe – high (>40Mbps, the first two destinations), low (<10Mbps, the next three destinations) and moderate (the last destination) [2]. In the table, the first three machines belong to the PlanetLab infrastructure. The fourth machine is located in Pittsburgh and attached via AT&T. The source is a host located in a Chicago area data center. In all cases, whenever a bottleneck was found by any tool, the corresponding hop number is shown in parentheses. Note that since BFind limits its maximum sending rate it cannot identify bottlenecks with a higher available capacity as shown by the probes to the first two destinations. In this case, BFind was further constrained to a maximum of 40Mbps at the data center. In the second case, the 180secs maximum execution time was insufficient for BFind to probe beyond 20Mbps[3]. From these results, it is apparent that the output of BFind is reasonably consistent with the outputs of Pathload and Pipechar – both in terms of available bandwidth as well as the location of the bottleneck link. We observe similar consistency in the outputs across all the other destinations we probe.

We also performed an initial cross-validation of our approach by checking if PlanetLab sources in a given metro area, attached to the same upstream provider, identify the same bottleneck links when probing destination IP addresses selected in Section 2.2. For example, in the Los Angeles metro area, we found that the sources at UCSD, UCLA, UCSB, and ISI all identify similar bottlenecks in paths to the destinations in all cases where: (1) the bottlenecks are not located in their access network (CalREN2) and (2) the paths are identical beyond the access network.

We also implemented BFind in the NS-2 network simulator for additional validation. We ran several tests to understand its probing behavior, particularly with regard to issues such as operation in the presence of multiple bottlenecks, interaction with competing TCP traffic, and comparison to TCP behavior. These results are

[2] About 20 of the destinations we probed had a very high available bandwidth. Of the remaining, 9 had very low available bandwidth. The remaining destination had moderate available bandwidth.

[3] In $> 97\%$ of the paths we probed, BFind completed well before 180s, either because a bottleneck was found or because the limit on the send rate was reached.

Destination Node	Path length	Pathload Report	Pipechar Report	BFind Report
CMU-PL	14	58.1 - 107.2Mbps	82.4Mbps	>39.1Mbps
Princeton-PL	12	91.3 - 96.8Mbps	94.5Mbps	>20.5Mbps
KU-PL	15	8.23 - 8.87Mbps	5.21Mbps (hop 12)	9.88Mbps (hop 12)
Pittsburgh-node	14	4.17 - 5.21Mbps	4.32Mbps (hop 11)	8.34Mbps (hop 11)
www.fnsi.net	11	N/A	8.2Mbps (hop 10)	8.43Mbps (hop 10)
www.i1.net	11	N/A	19.21Mbps (hop 7)	32.91Mbps (hop 8)

Table 3: BFind validation results: Statistics for the comparison between BFind, Pathload and Pipechar

described in the Appendix.

2.4 Metrics of Interest

Based on the results of BFind, we report the bandwidth and latency of the bottlenecks we discover. In addition to these metrics, we post-process the tool's output to report on the ownership and location of Internet bottlenecks. Such a categorization helps identify what parts of the Internet may constrain high-bandwidth flows and what parts to avoid in the search for good performance. We describe this categorization in greater detail below.

In our analysis, we first classify bottlenecks according to *ownership*. According to this high level classification, bottlenecks can be described as either those within carrier ISPs, which we further classify by the tier of the owning ISP, or those between carrier ISPs, which we further classify according to the tiers of the ISPs at each end of the bottleneck. In order to characterize each link in our measurements according to these categories, we use a variety of available utilities. We identify the AS owning the endpoint of any particular link using the whois servers from RADB [27] and RIPE [28] routing registries. In addition, we use the results of [33] to categorize these ASes into tiers.

Our second classification is based on the latency of the bottleneck links. We classify bottlenecks according to three different levels of latency – low latency ($<$ 5ms), medium (between 5 and 15ms) and high ($>$ 15ms). Within each level, we identify bottlenecks that are within ISPs and those that are between carrier ISPs.

For paths to the NAPs, we classify the path into three categories – those that do not have a bottleneck (as reported by BFind), those that have a bottleneck at the NAP, and those that have a bottleneck elsewhere. Again, we are only interested in non-access bottlenecks.

For each category in the classification scheme described above, we present a cumulative distribution function of the available capacity of the bottlenecks of the particular category.

2.5 A Subjective Critique

We describe some possible shortcomings of our approach here. To approximate the measurement of "typical" paths, we choose what we believe to be a representative set of network paths. While the set of paths is not exhaustive, we believe that they are diverse in their location and network connectivity. However, as the sources for our measurements are dominated by PlanetLab's academic hosts, there may be some hidden biases in their connectivity. For example, they may all have Internet2 connections which are uncommon elsewhere. This particular bias does not affect our measurements since our destinations are not academic sites (and hence the paths do not pass over Internet2). However, our test nodes are relatively USA-centric (only 3 international sources and 7 destinations) and may not measure international network connectivity well.

Routing could also have a significant impact on our measurements. If routes change frequently, it becomes difficult for the BFind tool to saturate a path and detect a bottleneck. Similarly, if an AS uses multipath routing, BFind's UDP probe traffic and its traceroutes may take different paths through the network. As a result, BFind may not detect any queuing delays nor, hence, any bottleneck despite saturating the network with traffic. If either of these situations occurred, traceroutes along the tested path would likely reveal multiple possible routes. However, despite our continuous sampling of the path with traceroute during a BFind test, we did not observe either of these routing problems occurring frequently. This is consistent with recent results showing that most Internet paths tend to be stable, even on an hours timescale [37].

The processing time taken by routers to generate traceroute ICMP responses can impact our measurement of queuing delay and, therefore, bottlenecks in the network. Many researchers have noted that ICMP error processing, typically done in the router "slow" processing path, takes much longer than packet forwarding. In addition, some routers pace their ICMP processing in order to avoid being overwhelmed. Either of these could cause the delays reported by traceroute to be artificially inflated. However, recent work [9] has shown that slow path/fast path differences should not affect traffic measurement tools in practice since the typically observed ICMP processing delays are on the order of 1-2 ms, well within the timescales we need for accurate bottleneck detection.

Address allocation may also skew our results. We rely on using the address reported by routers in their response to traceroute probes to determine their ownership. However, in some peering arrangements, a router owned by an ISP is allocated an address from the peer ISP's address space to make configuration convenient. In such situations, our link classification may erroneously identify the incorrect link (by one hop) as a the peering link between the ISPs. However, we believe that the common use of point-to-point links in private peering situations and separate address allocations used in public exchanges (these both eliminate the above problem) reduce the occurrence of this problem significantly.

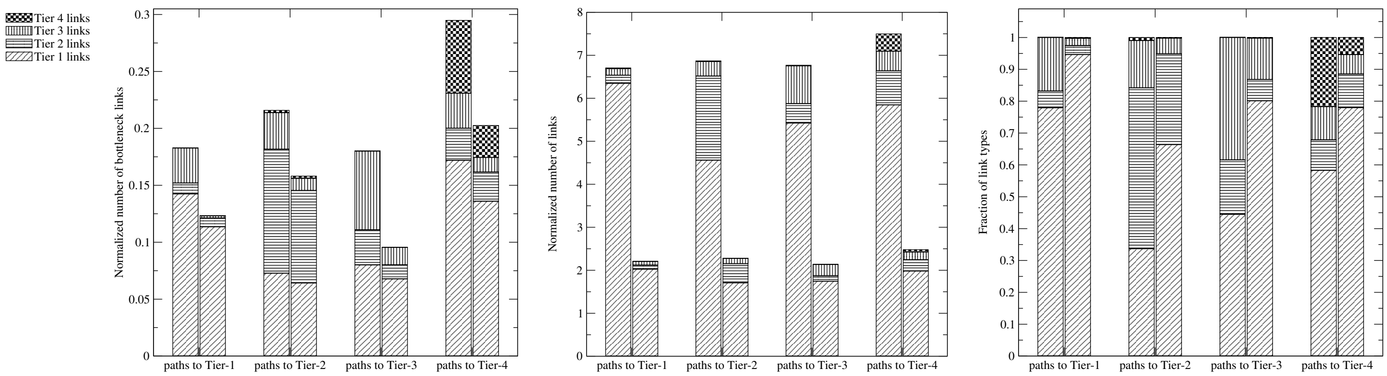
Finally, we note that our results represent an empirical snapshot of non-access Internet bottlenecks. That is, we focus on collecting observations from a large number of paths, rather than taking repeated measurements of a few paths over an extended period. While our approach provides a wider view of the characteristics and locations of bottlenecks, we cannot judge, for example, how stable or persistent the locations are. A longer-term characterization of bottlenecks is, hence, a natural extension to our work.

3. RESULTS

Over a period of 5 weekdays, we ran our BFind tool between our chosen source and destination sites. The experiments were conducted between 9am and 5pm EST on weekdays. These tests identified a large number (889) of non-access bottleneck links along many (2028) paths. As described in Section 2, our post-processing tools categorize these network links and bottlenecks in a variety of ways. In this section, we describe the properties of these paths and bottleneck in these different categories.

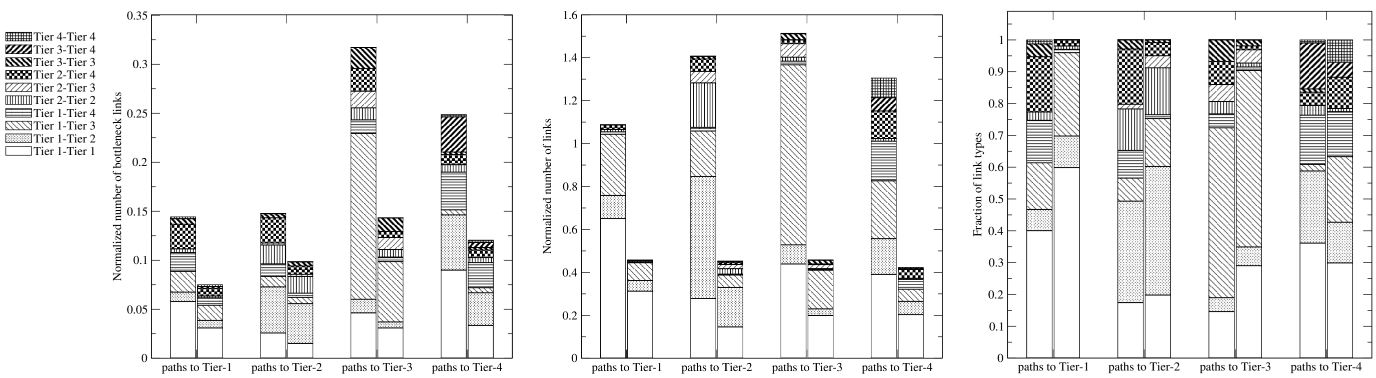
3.1 Path Properties

As described in Section 2, our results are based on observations made on paths between the PlanetLab sites and ISPs at different



(a) intra-ISP bottlenecks (b) All intra-ISP links (c) Relative proportions: bottleneck links vs. all links

Figure 3: Relative prevalence of intra-ISP bottlenecks: Graph (a) shows the average number of bottlenecks of each kind appearing inside carrier ISPs, classified by path type. The graph in (b) shows the total number of links (bottleneck or not) of each kind appearing in all the paths we considered. In (a) and (b), the left bar shows the overall average number of links, while the right shows the average number of unique links. Graph (c) shows the relative fraction of intra-ISP bottlenecks links of various types (left bar) and the average path composition of all links (right bar).



(a) Bottlenecks at peering links (b) All peering links (c) Relative proportions: bottlenecks vs. all links

Figure 4: Relative prevalence of peering bottlenecks: Graph (a) shows the average number of bottlenecks of each kind appearing between carrier ISPs, classified by path type. The graph in (b) shows the total number of links (bottleneck or not) of each kind appearing in all the paths we considered. In (a) and (b), the left bar shows the overall average number of links, while the right shows the average number of unique links. Graph (c) shows relative fraction of peering bottlenecks of various types (left bar) and the average path composition for all links (right bar).

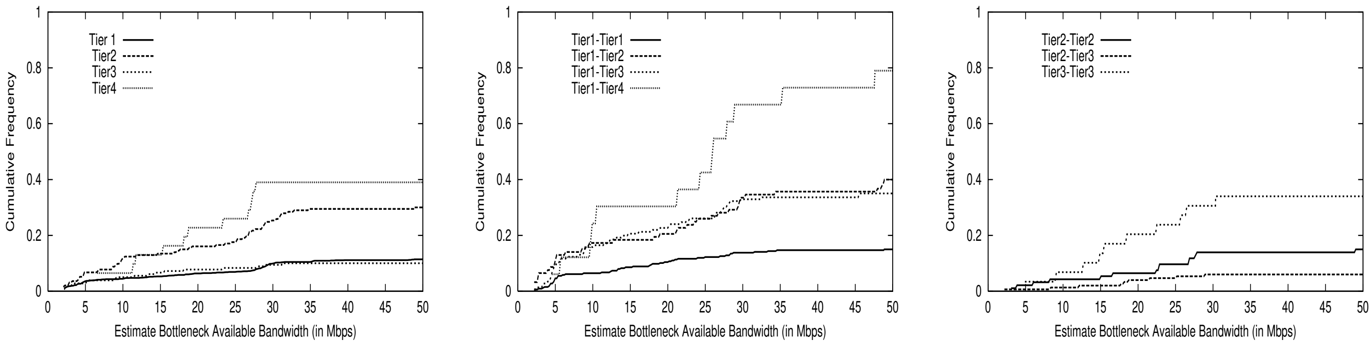
tiers in the Internet hierarchy. Before describing the results on bottleneck links, it is useful to consider some important overall characteristics of these paths.

The graphs in Figures 3(b) and 4(b) summarize overall features of paths from PlanetLab sites, classified by paths to ISPs of a particular tier. On the y-axis, we plot the normalized number of links, *i.e.*, the total number of links encountered of each type divided by the total number of paths in each class. Each path class has a pair of bars. The left bars in the graphs show the overall average properties of the paths. The right bars in the graphs show the average number of *unique* links that each path class adds to our measurements. This number is significantly less, by a factor of 2 or 3, than the actual link counts. This is because links near the sources and destinations are probed by many paths (and are counted repeatedly). Such links can bias our measurements since they may appear as bottlenecks for many paths. Therefore, we also present information about unique links instead of describing only average path properties.

Note that Figure 3 shows intra-ISP links while Figure 4 shows peering links. Characteristics of the entire paths are evident by examining the two together. For example, Figure 3(b) shows that the average path between a PlanetLab site and one of the tier-2 destinations traversed about 4.5 links inside tier-1 ISPs, 2.0 tier-2 ISP links, and 0.5 tier-3 links. Figure 4(b), which illustrates the location of the peering links, shows that these same paths also traversed about 0.25 tier-1 to tier-1 peering links, 0.75 tier-1 to tier-2 links, 0.2 tier-1 to tier-3 links, 0.2 tier-2 to tier-2 links, and a small number of other peering links. The total average path length of paths to tier-2 ISPs, then, is the sum of these two bars, i.e. $7 + 1.4 = 8.4$ hops. Similar bars for tier-1, tier-3 and tier-4 destinations show the breakdown for those paths. One clear trend is that the total path length for lower tier destinations is longer. The tier-1 average length is 7.8 hops, tier-2 is 8.3, tier-3 is 8.3 and tier-4 is 8.8. Another important feature is the number of different link types that make up typical paths in each class. As expected from the definition of the tiers, we see a much greater diversity (*i.e.,* hops from different tiers) in the paths to lower tier destinations. For example, paths to tier-4 destinations contain a significant proportion of all types of peering and intra-ISP links.

3.2 Locations of Bottlenecks

Figures 3(a) and 4(a) describe the different types of bottleneck links found on paths to different tier destinations. Recall that BFind identifies either one, or zero, bottleneck links on each path. The left bars in the graphs show the probability that the identified bottleneck link is of a particular type, based on our observations. For exam-



(a) intra-ISP (b) Tier-1 peering (c) Lower tier peering

Figure 5: Available capacity at bottleneck links: Graph (a) corresponds to bottlenecks within ISPs. Graphs (b) and (c) show the distribution of available capacity for bottlenecks in peering links involving Tier1 ISPs, and those in peering links not involving Tier1 ISPs, respectively. We do not show the distributions for bottleneck links between tiers 2 and 4 and those between tiers 3 and 4 since they were very small in number.

ple, from Figure 3(a), we see that the bottleneck links on paths to tier-2 networks consist of links inside tier-1 ISPs 7% of the time, tier-2 links 11% of the time, and tier 3 links 3% of the time (bottlenecks within tier-4 ISPs appear only in 0.2% of the cases). From Figure 4(a), we see that various types of peering links account for bottlenecks in tier-2 paths nearly 15% of the time, with tier-1 to tier-2 links appearing as the most likely among all types of peering bottleneck links. These two graphs together indicate that approximately 36% of tier-2 paths we measured had a bottleneck that we were able to identify. The other 64% appear to have bottlenecks with an available capacity greater than 50Mbps.

Figures 3(c) and 4(c) show the breakdown of links averaged across each type of path, for intra-ISP and peering links, respectively. Comparing the heights of components in the left and right bars gives an indication of the prevalence of the corresponding type of bottleneck link (left bar), relative to its overall appearance in the paths (right bar). From Figure 3(c), it first appears that lower-tier intra-ISP links are path bottlenecks in much greater proportion than their appearance in the paths. For example, Figure 3(c) shows that tier-3 links make up 17% of the bottlenecks to tier-1 destinations, but account for only about 2% of the links in these paths.

Note, however, that the right bars in Figure 3(a) show the number of unique bottlenecks links that we observed. Considering the first set of left and right bars (*i.e.*, all vs. unique bottlenecks for paths to tier-1 destinations) in Figure 3(a), we notice that there is a significant difference in the proportion of tier-3 bottleneck links. Upon further examination, we discovered that some of the PlanetLab sites were connected to the Internet via a tier-3 ISP. A few of these ISPs were bottlenecks for many of the paths leaving the associated PlanetLab site. More generally, though, we see in Figure 3(c) that lower-tier intra-ISP links seem to be bottlenecks more frequently than we would expect based on the appearance of these links in the paths.

A similar examination of Figure 4(c) reveals several details about the properties of bottlenecks at peering links. Figure 4(c) shows that tier-1- tier-1 peering links are bottlenecks less frequently than might be expected, given their proportion in the overall paths. Also, peering links to or from tier-2, tier-3 or tier-4 ISPs are bottlenecks more frequently than expected. For example, compare the proportion of tier-2 to tier-4 peering bottlenecks with the proportion of these links in the corresponding overall path length (e.g., 17% vs. 2% for paths to tier-1, and 17% vs. 4% for paths to tier-2).

Looking at Figures 3(a) and 4(a) together, we can observe some additional properties of bottleneck links. For example, total path lengths are around 8–9 hops (adding the heights of the bars in Figures 3(b) and 4(b)), of which only 1–1.5 hops are links between different ISPs. However, bottlenecks for these paths seem to be equally split between intra-ISP links and peering links (comparing the overall height of the bars in Figures 3(a) and 4(a)). This suggests that if there is a bottleneck link on a path, it is equally likely to be either in the interior of an ISP or between ISPs. Given that the number of peering links traversed is much smaller, however, the likelihood that the bottleneck is actually at one of the peering links is higher. But the fact that the bottleneck on any path is equally likely to lie either inside an ISP or between ISPs is surprising.

Another important trend is that the percentage of paths with an identified bottleneck link grows as we consider paths to lower-tier destinations. About 32.5% of the paths to tier-1 destinations have bottlenecks. For paths to tiers 2, 3, and 4, the percentages are 36%, 50%, and 54%, respectively. Note that while paths to tier-3 appear to have fewer intra-ISP bottlenecks than paths to tier-2, this may be because the peering links traversed on tier-3 paths introduce a greater constraint on available bandwidth.

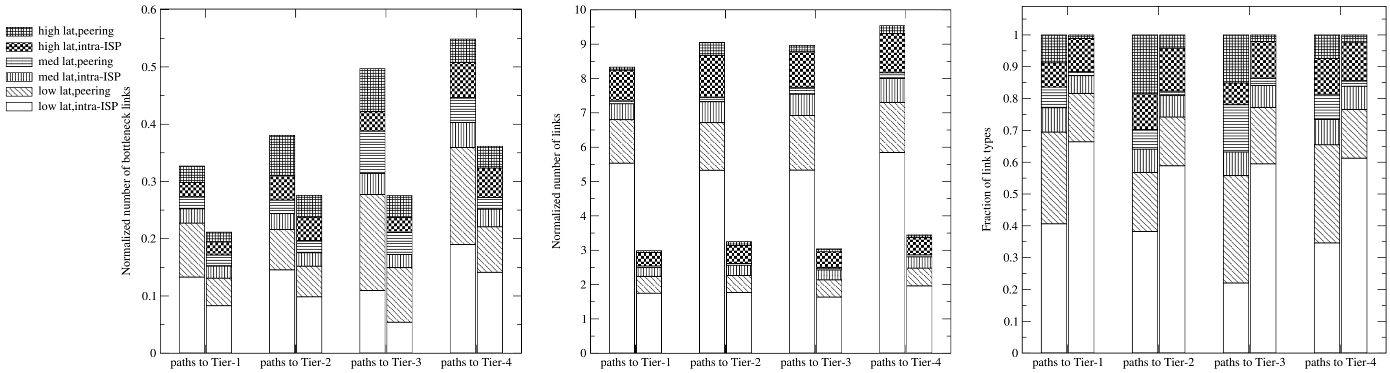
3.3 Bandwidth Characterization of Bottlenecks

In the previous section, we described the location and relative prevalence of observed bottleneck links, without detailing the nature of these bottlenecks. Here, we analyze the available bandwidth at these bottlenecks, as identified using BFind.

The graphs in Figure 5 illustrate the distribution of available bandwidth of bottleneck links observed in different parts of the network. Each graph has several curves, corresponding to different types of intra-ISP and peering links. Note that the CDFs do not go to 100% because many of the paths we traversed had more than 50 Mbps of available bandwidth. Recall that BFind is limited to measuring bottlenecks of at most 50 Mbps due to first hop network limitations. Hence we did not explore the nature of the bandwidth distribution above 50 Mbps.

Figure 5(a) shows the bottleneck speeds we observed on intra-ISP links. The tier-1 and tier-3 ISP links appear to have a clear advantage in terms of bottleneck bandwidth over tier-2 ISP bottlenecks. The fact that the tier-3 bottlenecks we identified offer higher available capacity than tier-2 bottlenecks was a surprising result. Links in tier-4 ISPs, on the other hand, exhibit the most limited available bandwidth distribution as expected.

In Figures 5(b) and (c) we consider the distribution of bottleneck bandwidth on peering links. Tier-1 to tier-1 peering links are the least constrained, indicating that links between the largest network providers are better provisioned when compared to links between lower-tier networks. Again, we find, surprisingly, that tier-2 and



(a) Bottleneck Links (b) All Links (c) Relative proportions: bottleneck links vs. all links

Figure 6: Relative prevalence of bottlenecks of various latencies: Graph (a) shows the average number of bottlenecks of the three classes of latencies further classified into those occurring between ISPs and those occurring inside ISPs. Graph (b) shows the actual number of links (bottleneck or not) of each kind appearing in all the paths. Graph (c) shows the relative fraction of bottleneck links of various latency types (left bar) and the average path composition of all links (right bar).

tier-3 links exhibit very similar characteristics, in their peering links to tier-1 networks (Figure 5(b)). Also, peering links between tier-2 and tier-3 are not significantly different than tier-2 to tier-2 links (Figure 5(b)). We do see, however, that bottleneck peering links involving networks low in the hierarchy provide significantly less available capacity, as expected. This is clearly illustrated in the bandwidth distributions for tier-1 to tier-4, and tier-3 to tier-3 links.

3.4 Latency Characterization of Bottlenecks

In this section, we analyze the latency of bottlenecks, in particular exploring the correlation between high-latency links and their relative likelihood of being bottlenecks. Figure 6 is similar to Figures 3 and 4, except that rather than classifying links on each type of path by their location, we separate them into latency classes (and whether they are peering or intra-ISP links). Low latency links have a measured latency, ℓ, of $\ell < 5$ ms, as determined by the minimum observed round-trip time. Medium latency and high latency links have minimum round trip times of $5 \leq \ell \leq 15$ and $\ell \geq 15$ ms, respectively. Though this is clearly a rough classification, we chose these classes to correspond to links at a PoP, links connecting smaller cities to larger PoPs, and long-haul links.

Figure 6(b) shows the overall latency characteristics of the paths. For example, paths to tier-2 destinations have an average of 5.3 low-latency intra-ISP, 1.4 low latency peering, 0.6 medium latency intra-ISP, 0.1 medium latency peering, 1.2 high latency intra-ISP, and 0.4 high latency peering links. In general, all path types have a high proportion of low-latency hops (both intra-ISP and peering) and high-latency intra-ISP hops. The latter is indicative of a single long-haul link on average in most of the paths we measured. While high latency peering links would seem unlikely, they do occur in practice. For example, one of the PlanetLab sites uses an ISP that does not have a PoP within its city. As a result, the link between the site and its ISP, which is characterized as a peering link, has a latency that exceeds 15ms.

In Figure 6(c) we illustrate the prevalence of bottlenecks according to their latency. We can observe that high-latency peering links are much more likely to be bottlenecks than their appearance in the paths would indicate. In observed paths to tier-2 destinations, for example, these links are 18.5% of all bottlenecks, yet they account for only 4% of the links. This suggests that whenever a high-latency peering link is encountered in a path, it is very likely to be a bottleneck. High latency intra-ISP links, on the other hand, are not overly likely to be bottlenecks (*e.g.*, 11% of bottlenecks, and 13.5% of overall hops on paths to tier-2).

In general, Figure 6 suggests that peering links have a higher likelihood of being bottlenecks, consistent with our earlier results. This holds for low, medium, and high-latency peering links. For example, very few paths have any medium latency peering links, yet they account for a significant proportion of bottlenecks in all types of paths. Also, low-latency peering links on paths to the lower tiers (*i.e.*, tier-3 and tier-4) have a particularly high likelihood of being bottlenecks, when compared to paths to tier-1 and tier-2 destinations. Recall from Figures 5(b) and (c) that these lower-tier peering bottlenecks also have much less available bandwidth.

3.5 Bottlenecks at Public Exchange Points

As mentioned in Section 2, one of our goals was to explore the common perception that public exchanges are usually network choke points, to be avoided whenever possible. Using the procedure outlined in Section 2.2.2, we identified a large number of paths passing through public exchanges, and applied BFind to identify any bottlenecks along these paths.

As indicated in Figure 7(a), we tested 466 paths through public exchange points. Of the measured paths, 170 (36.5%) had a bottleneck link. Of these, only 70 bottlenecks (15% overall) were at the exchange point. This is in contrast to the expectation that many exchange point bottlenecks would be identified on such paths. It is interesting to consider, however, that the probability that the bottleneck link is located at the exchange is about 41% ($= 70/170$). In contrast, Figures 3(a) and 4(a) do not show any other type of link (intra-ISP or peering) responsible for a larger percentage of bottlenecks.[4] This observation suggests that if there is a bottleneck on a path through a public exchange point, it is likely to be at the exchange itself nearly half of the time.

4. DISCUSSION

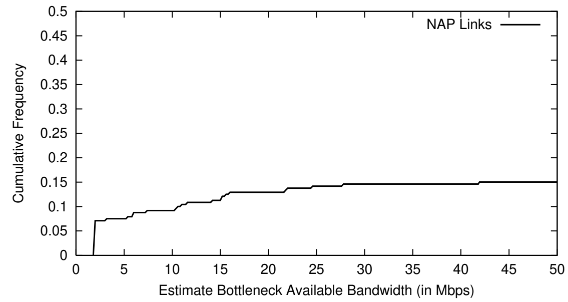
Our study, while to some degree confirms conventional wisdom about the location of Internet bottlenecks, yields a number of interesting and unexpected findings about the characteristics of these links. For example, we find a substantial number of bottleneck links *within* carrier ISPs. In addition, we also observed that low latency links, whether within ISPs or between them, can also constrain available bandwidth with a small, yet, significant probability.

Furthermore, our observations can provide some guidance when considering other related issues such as choosing an access provider,

[4] However, in Figure 4(a), bottlenecks between tiers 1 and 3 in paths to tier-3 destinations are comparable to bottlenecks at exchange points in this respect.

#Paths to exchange points	466
#Paths with non-access bottlenecks	170
#Bottlenecks at exchange point	70

(a) Relative prevalence



(b) Available bandwidth distribution

Figure 7: Bottlenecks in paths to exchange points: Table (a) on the left shows the relative prevalence of bottleneck links at the exchange points. Figure (b) shows the distribution of the available capacity for bottleneck links at the exchange points.

optimizing routes through the network, or analyzing performance implications of bottlenecks in practice. In this section we discuss some of these issues in the context of our empirical findings.

4.1 Providers and Provisioning

Our measurements show that there is a clear performance advantage to using a tier-1 provider. Our results also show that small regional providers, exemplified by the tier-4 ASes in our study, have relatively low-speed connectivity to their upstream carrier, irrespective of the upstream carrier's size. In addition, their networks often exhibit bottlenecks (as we define them). This may be considered a reflection of the impact of economics on network provisioning if we assume that carriers lower in the AS hierarchy are less inclined to overprovision their networks if their typical customer traffic volume does not thus far require it. As a result, there is a clear disadvantage to using a tier-4 provider for high-speed connectivity. However, the tradeoffs between tier-2 and tier-3 networks are much less clear.

Paths to tier-3 destinations had a larger percentage of bottleneck links than tier-2 paths. Despite this, we also observed that tier-2 and tier-3 bottlenecks show similar characteristics in terms of available capacity, with tier-3 bottlenecks (both intra-AS and peering links) performing slightly better in some cases. This might be explained if we conjecture that tier-2 ASes, by virtue of their higher degree of reachability, carry a larger volume of traffic relative to their capacity, when compared with tier-3 ASes. Extending this hypothesis, we might conclude that if a stub network desires reasonably wide connectivity, then choosing a tier-3 provider might be a beneficial choice, both economically and in terms of performance, assuming that connectivity to tier-3 providers is less expensive.

4.2 Network Under-utilization

More than 50% of the paths we probed seemed to have an available capacity close to 40-50 Mbps or maybe more. This is true across most non-access links irrespective of their type. We hypothesize from this that large portions of the network are potentially under-utilized on average, confirming what many large ISPs report about the utilization of their backbone networks. However, the fact that this holds even for providers of smaller size (*e.g.* tier-3) as well as for most peering links and even links at NAPs, seems surprising.

This observation about under-utilization, coupled with our results about the existence of potential hot-spots with low available bandwidth, opens the following key question – Is it possible to avoid these bottlenecks by leveraging existing routing protocols? While there has been considerable work on load-sensitive routing of traffic within an AS, little is known about how to extend this across ASes. We plan to explore this path in the future.

4.3 Route Optimization

It is sometimes suggested that a large proportion of the peering links between large carrier ISPs (tier-1) could emerge as bottlenecks, due to the lack of economic incentive to provision these links and the large volume of traffic carried over them. However, our measurements seem to suggest otherwise. We believe that this could imply that either the peering links are in fact quite well provisioned, or that a smaller portion of the entire Internet traffic traverses these links than what might be expected intuitively.

While it is difficult to discern the exact cause for this lack of bottlenecks, it may have important implications for the design of systems or choice of routes. For example, purchasing bandwidth from two different tier-1 ISPs may be significantly better from a performance perspective than buying twice as much bandwidth from a single tier-1 ISP.[5] In fact, it might be more economical to purchase from one ISP. Similarly, a shorter route to a destination that passed through a tier-1 to tier-1 peering link might be better than a longer route that stays within a single, lower-tier provider.

5. RELATED WORK

Several earlier research efforts have shared our high-level goal of measuring and characterizing wide-area network performance. This past work can be roughly divided into two areas: 1) measurement studies of the Internet, and 2) novel algorithms and tools for measuring Internet properties. In this section we review several recent representative efforts from each of these categories.

5.1 Measurement Studies

Typically, measurement studies to characterize performance in the Internet have taken two forms: 1) some, such as [23, 36, 19, 32], use active probing to evaluate the end-to-end properties of Internet paths and, 2) other studies, such as [2, 35] have used passive monitoring or packet traces of Internet flows to observe their performance in the Internet.

In [23] multiple TCP bulk transfers between pairs of measurement end-points are monitored to show evidence of significant packet re-ordering, correlated packet losses, and frequent delay variations on small scales. The authors also describe the distribution of bottleneck capacities observed in the transfers. The study by Savage *et al.* used latency and loss measurements between network endpoints to compare the quality of direct and indirect paths between nodes [32]. The authors note that the performance gains come from avoiding congestion and using shorter latency paths. Using active measurements in the NIMI [25] infrastructure, Zhang *et al.*

[5]Of course, it might be useful for reliability purposes.

Non-access bottlenecks are equally likely to be links within ISPs or peering links between ISPs
The likelihood of a bottleneck increases on paths to lower tier ISPs
Interior and peering bottlenecks in tier-2 and tier-3 ISPs exhibit very similar available capacity
Internal links in lower tier ISPs appear as bottlenecks with greater frequency than their overall presence in typical paths
Bottlenecks appeared in only 15% of the paths traversing public exchanges, but when a bottleneck is found on such paths, the likelihood of it being at the exchange is more than 40%
All paths have a high proportion of low-latency links (interior and peering) and roughly one high-latency interior link

Table 4: Summary of key observations

study the constancy of Internet paths in terms of delay, loss, and throughput [36]. For each notion of constancy, they observed that all three properties were steady on at least a minute's timescale. Finally, a recent study of delay and jitter across several large backbone providers aimed to classify paths according to their suitability for latency-sensitive applications [19]. The authors found that most paths exhibited very little delay variation, but very few consistently experienced no loss. In comparison with these efforts, our work has a few key differences. First, rather than exploring true end-to-end paths, our measurement paths are intended to probe the non-access part of the Internet, *i.e.,* the part responsible for carrying data between end networks. Second, we measure *which* part of the network may limit the performance of end-to-end paths.

In [2], the authors study packet-level traces to and from a very large collection of end-hosts, and observe a a wide degree of performance variation, as characterized by the observed TCP throughput. With a similar set of goals, Zhang *et al.* analyze packet traces to understand the distribution of Internet flow rates and the causes thereof [35]. They find that network congestion and TCP receiver window limitations often constrain the observed throughput. In this paper, our aim is not to characterize what performance end-hosts *typically* achieve and what constrains the typical performance. Instead, we focus on *well-connected* and unconstrained end-points (*e.g.,* no receiver window limitations) and comment on how ISP connectivity constrains the performance seen by such end-points.

5.2 Measurement Tools

The development of algorithms and tools to estimate the bandwidth characteristics of Internet paths continues to be an active research area (see [6] for a more complete list). Tools like *bprobe* [5], Nettimer [17], and PBM [23] use packet-pair like mechanisms to measure the *raw bottleneck capacity* along a path. Other tools like *clink* [7], *pathchar* [12], *pchar* [18], and *pipechar* [10], characterize hop-by-hop delay, raw capacity, and loss properties of Internet paths by observing the transmission behavior of different sized packets. A different set of tools, well-represented by *pathload* [14], focus on the *available capacity* on a path. These tools, unlike BFind, require control over both the end-points of the measurement. Finally, the *TReno* tool [20] follows an approach most similar to ours, using UDP packets to measure available bulk transfer capacity. It sends hop-limited UDP packets toward the destination, and emulates TCP congestion control by using sequence numbers contained in the ICMP error responses. TReno probes each hop along a path in turn for available capacity. Therefore, when used to identify bottlenecks along a path, TReno will likely consume ICMP processing resources for every probe packet at each router being probed as it progresses hop-by-hop. As a result, for high-speed links, TReno is likely to be more intrusive than our tool.

In addition to available bandwidth, link loss and delay are often performance metrics of interest. Recent work by Bu, *et al.* describes algorithms that infer and estimate loss rates and delay distributions on links in a network using multicast trees [4].

In this paper we develop a mechanism that measures the available capacity on the path between a controlled end-host and an arbitrary host in the Internet. In addition, we identify the portion of the network responsible for the bottleneck. Our tool uses an admittedly heavyweight approach in the amount of bandwidth it consumes.

6. SUMMARY

This goal of this paper was to explore the following fundamental issue: if end networks upgrade their access speeds, which portions of the rest of the Internet are likely to become hot-spots? To answer this question, we performed a large set of diverse measurements of typical paths traversed in the Internet. We identified non-access bottlenecks along these paths and studied their key characteristics such as location and prevalence (links within ISPs vs. between ISPs), latency (long-haul vs. local), and available capacity. Table 4 summarizes some of our key observations.

The results from our measurements mostly support conventional wisdom by quantifying the key characteristics of non-access bottlenecks. However, some of our key conclusions show trends in the prevalence of non-access bottlenecks that are unexpected. For example, our measurements show that the bottleneck on any path is roughly equally likely to be either a peering link or a link inside an ISP. We also quantify the likelihood that paths through public exchange points have bottlenecks appearing in the exchange.

In addition, our measurements quantify the relative performance benefits offered by ISPs belonging to different tiers in the AS hierarchy. Interestingly, we find that there is no significant difference between ISPs in tiers 2 and 3 in this respect. As expected, we find that tier-1 ISPs offer the best performance and tier-4 ISPs contain the most bottlenecks.

In summary, we believe that our work provides key insights into how the future network should evolve on two fronts. Firstly, our results can be used by ISPs to help them evaluate their providers and peers. Secondly, the observations from our work can also prove helpful to stub networks in picking suitable upstream providers.

Acknowledgment

We are very grateful to Kang-Won Lee, Jennifer Rexford, Albert Greenberg, Brad Karp and Prashant Pradhan for their valuable suggestions on this paper. We also thank our anonymous reviewers for their detailed feedback.

7. REFERENCES

[1] D. Andersen, H. Balakrishnan, M. Kaashoek, and R. Morris. Resilient Overlay Networks. In *Proceedings of the 18th Symposium on Operating System Principles*, Banff, Canada, October 2001.

[2] H. Balakrishnan, S. Seshan, M. Stemm, and R. H. Katz. Analyzing stability in wide-area network performance. In *Proceedings of ACM SIGMETRICS*, Seattle, WA, June 1997.

[3] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson. TCP Vegas: New Techniques for Congestion Detection and Avoidance. In *Proceedings of the SIGCOMM '94 Symposium on Communications Architectures and Protocols*, August 1994.

[4] T. Bu, N. Duffield, F. L. Presti, and D. Towsley. Network tomography on general topologies. In *Proceedings of ACM SIGMETRICS*, Marina Del Ray, CA, June 2002.

[5] R. L. Carter and M. E. Crovella. Measuring bottleneck link speed in packet-switched networks. *Performance Evaluation*, 27–28:297–318, October 1996.

[6] Cooperative Association for Internet Data Analysis (CAIDA). Internet tools taxonomy. `http://www.caida.org/tools/taxonomy/`, October 2002.

[7] A. Downey. Using pathchar to estimate internet link characteristics. In *Proceedings of ACM SIGCOMM*, Cambridge, MA, August 1999.

[8] L. Gao. On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking*, 9(6), December 2001.

[9] R. Govindan and V. Paxson. Estimating router ICMP generation delays. In *Proceedings of Passive and Active Measurement Workshop (PAM)*, Fort Collins, CO, 2002.

[10] J. Guojun, G. Yang, B. R. Crowley, and D. A. Agarwal. Network characterization service (NCS). In *Proceedings of IEEE International Symposium on High Performance Distributed Computing (HPDC)*, San Francisco, CA, August 2001.

[11] U. Hengartner, S. Moon, R. Mortier, and C. Diot. Detection and analysis of routing loops in packet traces. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop (IMW)*, November 2002.

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

[13] M. Jain and C. Dovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. In *Proceedings of ACM SIGCOMM*, Pittsburgh, PA, August 2002.

[14] M. Jain and C. Dovrolis. Pathload: A measurement tool for end-to-end available bandwidth. In *Proceedings of Passive and Active Measurement Workshop (PAM)*, Fort Collins, CO, March 2002.

[15] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley. Measurement and classification of out-of-sequence packets in a tier-1 IP backbone. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop (IMW)*, November 2002.

[16] C. Labovitz, A. Ahuja, and F. Jahanian. Experimental study of Internet stability and backbone failures. In *Proceedings of IEEE International Symposium on Fault-Tolerant Computing (FTCS)*, Madison, WI, June 1999.

[17] K. Lai and M. Baker. Nettimer: A tool for measuring bottleneck link bandwidth. In *Proceedings of USENIX Symposium on Internet Technologies and Systems*, March 2001.

[18] B. A. Mah. *pchar*: A tool for measuring internet path characteristics. `http://www.employees.org/~bmah/Software/pchar/`, June 2000.

[19] A. P. Markopoulou, F. A. Tobagi, and M. J. Karam. Assessment of VoIP quality over Internet backbones. In *Proceedings of IEEE INFOCOM'02*, New York, NY, June 2002.

[20] M. Mathis and J. Mahdavi. Diagnosing Internet Congestion with a Transport Layer Performance Tool . In *Proc. INET '96*, Montreal, Canada, June 1996. `http://www.isoc.org/inet96/proceedings/`.

[21] Network Characterization Service: Netest and Pipechar. http://www-didc.lbl.gov/pipechar, 1999.

[22] ns-2 Network Simulator. http://www.isi.edu/nsnam/ns/, 2000.

[23] V. Paxson. End-to-end internet packet dynamics. *Proceedings of the SIGCOMM '97 Symposium on Communications Architectures and Protocols*, pages 139–152, September 1997.

[24] V. Paxson. End-to-end routing behavior in the internet. *IEEE/ACM Transactions on Networking*, 5(5):601–615, October 1997.

[25] V. Paxson, A. Adams, and M. Mathis. Experiences with NIMI. In *Proceedings of Passive and Active Measurement Workshop (PAM)*, Hamilton, New Zealand, April 2000.

[26] PlanetLab. `http://www.planet-lab.org`, 2002.

[27] RADB whois Server. `whois.radb.net`.

[28] RIPE whois Service. `whois.ripe.net`.

[29] BGP Tables from the University of Oregon RouteViews Project. `http://moat.nlanr.net/AS/data`.

[30] University of Oregon, RouteViews Project. `http://www.routeviews.org`.

[31] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, J. Voelker, and J. Zahorjan. Detour: a case for informed internet routing and transport. *IEEE Micro*, volume 19 no. 1:50–59, January 1999.

[32] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson. The end-to-end effects of internet path selection. In *Proceedings of the SIGCOMM '99 Symposium on Communications Architectures and Protocols*, pages 289–299, 1999.

[33] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the Internet hierarchy from multiple vantage points. In *Proceedings of IEEE INFOCOM*, June 2002.

[34] Traceroute.org. `http://www.traceroute.org`.

[35] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker. On the characteristics and origins of Internet flow rates. In *Proceedings of ACM SIGCOMM*, Pittsburgh, PA, August 2002.

[36] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker. On the constancy of Internet path properties. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop (IMW)*, November 2001.

[37] Y. Zhang, V. Paxson, and S. Shenker. The stationarity of internet path properties: Routing, loss, and throughput. Technical report, ICSI Center for Internet Research, May 2000.

APPENDIX

BFind Validation: Simulation Results

In Section 2.3.3, we presented a small set of results from wide-area experiments that compare the performance of BFind against similar tools. Our results showed that the output of BFind is consistent with other tools on the paths probed. In this section, we extend

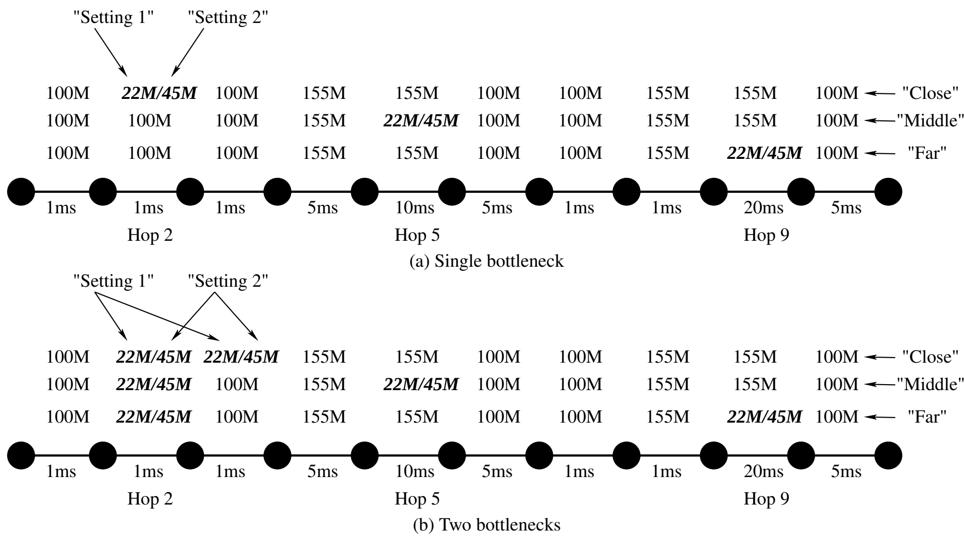


Figure 8: Topology used in our NS simulations. The topologies are explained in detail below. "M" stands for Mbps. The first row corresponds to location of the bottleneck link being "close", the second corresponds to "middle" and the third to "far".

Location	Capacity of bottleneck link							
	Capacity = 22Mbps ("Setting 1")				Capacity = 45Mbps ("Setting 2")			
	BFind Output	Time	Available BW (BFind)	TCP Throughput	BFind Output	Time	Available BW (BFind)	TCP Throughput
Close ($2nd$ hop)	2	17.1	5.8	4.6	2	26.1	8.2	20.53
Middle ($5th$ hop)	5	20.6	6.2	5.12	5	51.1	15.8	23.1
Far ($9th$ hop)	9	19.6	6.2	4.5	9	57.1	20.2	24.09

Table 5: The bandwidth-probing performance of BFind. The table shows, for each of the six configurations of the topology in Figure 8(a), the output obtained from BFind and its comparison with a TCP flow on the bottleneck hop.

the validation results with simulation experiments in NS-2 [22] to address the following issues about BFind:

1. How accurately, and quickly, can BFind estimate the location of bottlenecks? Does the capacity of the bottleneck links or their location on the path impact the speed or accuracy? How does the presence of multiple bottlenecks affect the detection?
2. How does the bandwidth probing behavior of BFind compare with that of a TCP flow across the bottleneck link? Is BFind more or less aggressive than TCP?
3. How does BFind compete with long-lived TCP cross traffic while probing for available bandwidth at a bottleneck link (given that the bottleneck faced by the competing TCP flows is different from that faced by BFind)?

Our simulations in this section are meant to validate the soundness of the methodology used by BFind and show that BFind does not yield incorrect results. However, this is not a substitute for additional wide-area experimental validation.

To address the above issues, we ported BFind to NS-2. We setup path topologies shown in Figure 8(a) and (b).[6] In either figure, the path contains 10 hops (the delays used are those observed on traceroutes from a machine in CMU and `www.amazon.com`, where the hop-by-hop delays are computed as mentioned in Section 2.3.1). The capacity of the non-bottleneck links in the path is shown in normal-faced font. For example, in either topology, the capacity of link 1 is 100Mbps. In Figure 8(a), there is exactly one bottleneck in the path. To test the probing behavior of BFind we vary the location of this bottleneck link along the path (between "close", "middle" and "far", as explained below), as well as its raw capacity (between 22Mbps - referred to as "Setting 1" - and, 45Mbps - referred to as "Setting 2" shown in italicized bold font). In Figure 8(a), when the location of the bottleneck link is "close", hop 2 is the bottleneck link; when the location is "middle", hop 5 is the bottleneck; and when the location is "far", hop 9 is the bottleneck. Therefore, Figure 8(a) pictorially summarizes 6 different experiments with a single bottleneck link on the path – 3 different bottleneck "Locations", each with 2 different bandwidth "Settings".

The topology in Figure 8(b), on the other hand, has two similar bottleneck links. In "Setting 1", both links have an identical capacity of 22Mbps; in "Setting 2", they have an identical capacity of 45Mbps. When the "Location" of the bottlenecks is "close", hops 2 and 3 are chosen to be the identical bottleneck links; when it is "middle", hops 2 and 5 are the bottlenecks; and when it is "far", hops 2 and 9 are the bottlenecks.

In either topology, unless otherwise specified, there is cross traffic between neighboring routers. The cross traffic consists of 25 HTTP sessions in NS-2, each configured with 25 maximum connections. In addition, the cross traffic also includes 25 constant rate UDP flows with default parameters as set in NS-2. Cross traffic on the reverse path between neighboring routers is also similar. Notice that the cross traffic on all hops is similar, in the average sense.

In Table 5, we show the performance of BFind on the topology in Figure 8(a). In this Figure, for each of the six experiments summarized in Figure 8(a), we show if BFind correctly identifies the appropriate bottleneck (for example, the bottleneck corresponding to location "close" should be hop 2), the time taken until detection, and the available bandwidth reported by BFind. In addition, we also report the average throughput of a TCP connection whose end-points are routers at either end of the bottleneck link. For example, when location is "middle", the TCP flow runs from router-4

[6] We chose not to experiment with more complicated topologies since BFind probes only along a single path. As a result, all other nodes and links in the topology become auxiliary.

Location	Capacity of bottleneck link			
	Capacity = 22Mbps ("Setting 1")		Capacity = 45Mbps ("Setting 2")	
	BFind Output	Time	BFind Output	Time
Close ($2nd$ and $3rd$ hops)	2	17.1	2	17.1
Middle ($2nd$ and $5th$ hops)	5	22.1	2	29.6
Far ($2nd$ and $9th$ hops)	9	17.1	2	38.6

Table 6: Performance of BFind in the presence of two similar bottlenecks. The table shows the hops identified by BFind as being the bottleneck in each of the six configurations in Figure 8(b), and the time taken to reach the conclusion.

Location	Capacity of bottleneck link			
	Capacity = 22Mbps ("Setting 1")		Capacity = 45Mbps ("Setting 2")	
	BFind Output	Time	BFind Output	Time
Close ($2nd$ and $3rd$ hops)	3	20.6	2	46.6
Middle ($2nd$ and $5th$ hops)	2	17.1	2	20.1
Far ($2nd$ and $9th$ hops)	2	17.1	2	27.6

Table 7: Performance of BFind in the presence of two slightly different bottlenecks. The table shows the hops identified by BFind as being the bottleneck in each of the six configurations in Figure 8(b) when the bandwidth of one of the hops on the path is chosen to be slightly higher than that of the other.

to router-5, where the underlying path has a base-RTT of 20ms. Also, the TCP connection runs under the *exact* same conditions of cross traffic as BFind. When running the TCP connection, obviously, BFind *is not* run in parallel.

The results presented in Table 5 show that: (1) BFind accurately determines bottleneck links for both capacity values. When the capacity of the bottleneck link is higher, the time taken by BFind is not necessarily worse. (2) The throughput probed by BFind is roughly similar to that achieved by the TCP connection. When the capacity of the bottleneck link is low, BFind probes somewhat more aggressively than TCP; however, when the capacity is higher, BFind does not probe as aggressively.

In Table 6, we show the results for the performance of BFind in the presence of two, very similar, bottlenecks along a path (the topology in Figure 8(b)). The results show that BFind identifies one of the two links as being a bottleneck. However, the output is non-deterministic. For example, BFind identifies, hops 2, 5 and 9 as being the bottlenecks in "Setting 1" across different "location"s. To further investigate BFind's ability to detect bottlenecks where multiples may exist along a path, we slightly modified the topology in Figure 8(b) as follows: instead of having two identical bottlenecks along the path, we deliberately set one of them to a *slightly* higher capacity. In "Setting 1", the second bottleneck link in each case – hop 3 (location being "close"), 5 (location being "middle") and 9 (location being "far") – had a capacity of 25Mbps (instead of 22Mbps previously). In "Setting 2", the capacity of the second link was chosen to be 50Mbps. The results for these experiments are shown in Table 7. In almost all cases, BFind correctly identifies hop 2 as the bottleneck link, despite almost similar capacity of the second potential bottleneck along the hop. Also, the time taken for detection is not necessarily worse.

We also show results demonstrating the interaction of BFind with competing long-lived TCP traffic. For these simulations, we use the single bottleneck topology in Figure 8(a), with a location of "mid" (i.e., bottleneck is hop 5). We eliminated cross traffic along hop 5 and, instead, started N long-lived TCP flows between router-4 and router-5 such that the total bandwidth achieved by the TCP flows is *at most* 10Mbps at any point of time. We then started BFind on router 0 to probe for the available capacity on the bottleneck link, hop 5. Notice that in "Setting 1", BFind should report an available capacity of at most 12Mbps (since the raw capacity is 22Mbps), while in "Setting 2", it should report at most 35Mbps. In Figure 9, we plot the available bandwidth reported by BFind in either setting, as a function of the number of TCP flows, N. In "Setting 2", the bandwidth reported by BFind is always lower than 35Mbps,

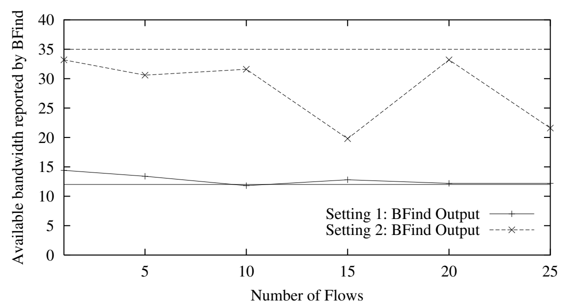


Figure 9: BFind interaction with competing long-lived TCP flows. The figures plots the available bandwidth reported by BFind for the topology in Settings 1 and 2, when competing long-lived TCP flows on the bottleneck hops are constrained to at most 10Mbps.

indicating that BFind does not have undesirable interactions with competing TCP traffic. In "Setting 1", the bandwidth from BFind is almost exactly 12Mbps as long as $N \geq 5$, again, reinforcing the fact that BFind competes fairly with long-lived TCP traffic (though in the RTT-proportional fairness sense, BFind is unfair).