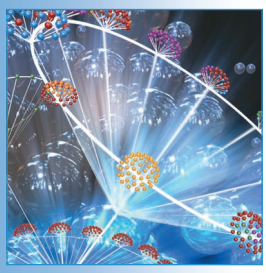

Locating Available Bandwidth Bottlenecks

The Spatio-Temporal Available Bandwidth estimator (STAB), a new edge-based probing tool, locates thin links — those links with less available bandwidth than all the links preceding them — on end-to-end network paths. By localizing thin links, STAB facilitates network operations and troubleshooting, provides insight into what causes network congestion, and aids network-aware applications. The tool uses special *chirp-probing trains*, featuring an exponential flight pattern of packets, which have the advantage of employing few packets while giving an accurate estimate of available bandwidth.

Vinay J. Ribeiro, Rudolf H. Riedi, and Richard G. Baraniuk
Rice University

Knowledge of a network's critical internal properties – available bandwidth on end-to-end paths or the location of any congested links, for example – greatly enhances various network applications. Unfortunately, obtaining this information directly from Internet routers is nearly impossible due to the Internet's decentralized nature, which discourages information sharing. Even if these routers were inclined to disseminate information, they couldn't spare scarce CPU resources without negatively affecting their own packet-forwarding performance. Edge-based measurements are therefore the best option for inferring critical internal properties. By injecting packets into the network, probing tools can estimate significant properties solely from the probe packets' end-to-end delays.

In this article, we present an edge-based probing tool designed to locate *thin links* – those with less available bandwidth than all the links preceding them on the end-to-end path. Thin-link localization gives us insight into what causes network congestion and suggests ways of circumventing it. Intuition suggests that congestion normally occurs at poorly provisioned links or at the very edge of the network,[1] but the truth is still unknown. Thin-link localization also augments applications as diverse as grid computing, overlay networking, server selection, and service-level agreement verification, all of which benefit from knowing whether paths share common congested links.[2] Finally, real-time information about a thin link's location aids network operations managers in various operational tasks such as troubleshooting and adjusting traffic routes. We term thin-link localization in space and over time as *spatio-temporal available bandwidth estimation.*

By using intelligently spaced probe-packet sequences and leveraging the science of network queuing, our edge-based probing tool (called STAB) performs spa-

 Published by the IEEE Computer Society 1089-7801/04/$20.00

tio-temporal available-bandwidth estimation. STAB's three key ingredients are *chirp-probe trains*, which feature exponential flight patterns of packets; *self-induced congestion*, an available-bandwidth estimation technique that temporarily congests the network by increasing the probing bit rate; and *packet tailgating*, a probing concept that allows estimation of spatially local network properties. In particular, chirp-probe trains are significantly more efficient than other suggested probing schemes (such as trains with equally spaced packets). They use very few packets yet provide accurate estimates of available bandwidth, thus keeping the probe-traffic load low.[3]

We've successfully tested STAB with both simulations and experiments over the Internet; visit www.spin.rice.edu/Software/STAB/ for a free open-source version of the tool.

Probing for Available Bandwidth

An Internet link's available bandwidth is the difference between its maximum transmission bandwidth and its average traffic load: a 100 Megabit-per-second (Mbps) link transmitting 12 Mbps of traffic has an available bandwidth of 88 Mbps. An end-to-end network path's available bandwidth is the minimum available bandwidth of its constituent links. A path's available bandwidth is closely related to the bit rate a new TCP connection can achieve by using the path. However, the two are not identical because other factors, including the path's round-trip delay, end-host system constraints, and the number of competing TCP connections, influence throughput.

The self-induced congestion principle provides an effective technique for estimating a path's available bandwidth. This principle relies on the fact that routers buffer incoming packets in queues before transmitting them via links. Figure 1 illustrates this well: if the incoming packet bit rate exceeds the outgoing link's transmission rate, packets fill up the corresponding queue and thus face queuing delays.

According to the principle of self-induced congestion, if we inject probe packets into a path at a faster bit rate than the available bandwidth, then the path's queues will congest, leading to increasing delays. The path's queues won't congest, however, if the probing bit rate is less than the available bandwidth. Therefore, we can estimate the available bandwidth simply by varying the injected probing bit rate to identify the minimum rate at which we start to see increasing packet-queuing delays.

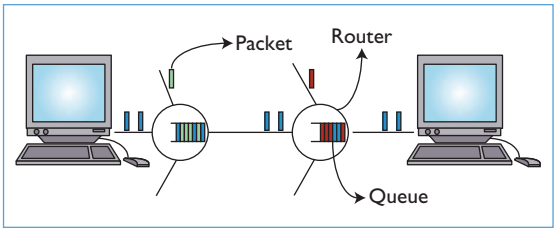

Figure 1. Packet buffering. If the incoming packet bit rate exceeds the outgoing link's transmission rate, routers buffer the packets in queues.

State-of-the-Art Tools

Several of today's available-bandwidth estimation tools are based on the self-induced congestion approach — examples include Trains of Packet Pairs (TOPP),[4] Initial Gap Increasing (IGI; http://gs274.sp.cs.cmu.edu/www/igi/),[5] Pathload (www.cc.gatech.edu/fac/Constantinos.Dovrolis/pathload.html),[6] Network Test (netest; http://dsd.lbl.gov/NCS/netest/),[7] and pathChirp (www.spin.rice.edu/Software/pathChirp/).[3] However, each of these tools provides only available bandwidth estimates for the end-to-end path; none provide spatio-temporal information such as thin-link locations.

These tools differ from each other in both the type of probe schemes and the algorithms they use. We call a group of several closely spaced probe packets a *packet train*; a packet train of just two packets is a *packet pair*. We define the *probing bit rate* between two consecutive probe packets as the ratio of packet size (in bits) to packet interspacing.

TOPP and IGI probe the network using packet pairs of different interspacings. The probing bit rate at which packet interspacing at the receiver host begins to exceed that at the sending host (due to self-induced congestion) gives the available bandwidth. Pathload uses packet trains of equally spaced packets, meaning the probing bit rate within a single train remains constant. Based on whether the end-to-end packet delays with a train increase, Pathload adaptively changes the bit rate from one packet train to the next and then converges to the available bandwidth using a binary search algorithm.

STAB uses pathChirp's algorithm for available-bandwidth estimation. In a chirp-probe train as shown in Figure 2a (next page), the spacing between successive packets decreases exponentially according to a spread factor γ. Denoting the first packet interspacing as T, the subsequent packet interspacings equal T/γ, T/γ^2, T/γ^3, and so on. By using a few probe packets, a chirp can thus

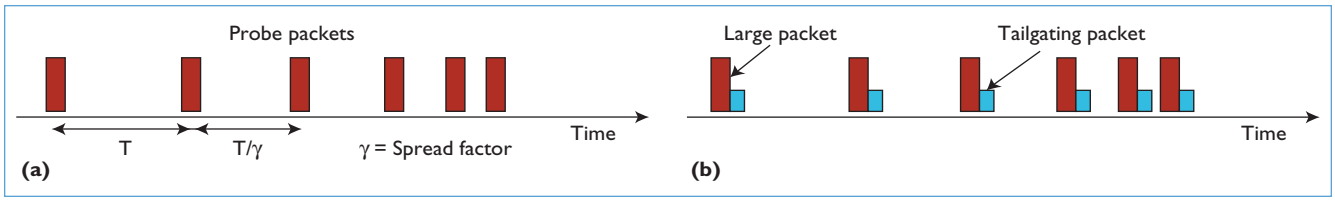

Figure 2. Probe packets. (a) The chirp-packet train's exponential flight pattern enables efficient available-bandwidth estimation. (b) With a packet-tailgating chirp train, we replace each packet with a large packet followed closely by a smaller one.

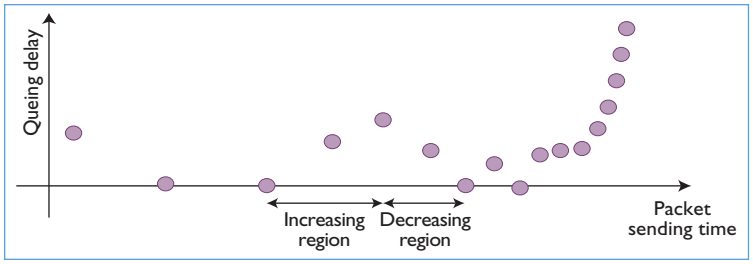

Figure 3. Queuing-delay signature. The profile of a chirp-packet queuing delay, also called its signature, *consists of several regions of delay increase and decrease.*

sweep through a wide range of probing rates, enabling a quick estimate of the available bandwidth based on self-induced congestion.

PathChirp analyzes the chirp-packet queuing delay's profile — also called the *queuing-delay signature* — to estimate the end-to-end path's available bandwidth. A typical signature shows a few regions with increasing queuing delays; these typically occur when probe packets encounter temporarily congested queues, as Figure 3 shows. A signature also contains regions with decreasing packet-queuing delays, which occur when probe packets encounter emptying queues.

Based on the self-induced congestion principle, pathChirp assigns an instantaneous available-bandwidth estimate to each region. The time average of such estimates from a chirp signature's different regions gives a single available-bandwidth estimate for that chirp, which we call the *per-chirp* estimate. We smooth these per-chirp estimates over time by using a moving average window to obtain the available bandwidth's final time-varying estimate.

PathChirp time-stamps packets at both the sender and receiver hosts, which lets it compute the packets' queuing delays. PathChirp requires only the relative increase or decrease of these delays within a chirp train and not the absolute value of the queuing delays: hence it doesn't require synchronized clocks at the end hosts and can tolerate a reasonable amount of clock skew.[3]

Thin-Link Localization

Our scheme of using end-to-end probing to locate thin links combines self-induced congestion with the concept of packet tailgating.[8] The latter helps us estimate the available bandwidth of segments in an end-to-end path, which in turn helps us find the thin links. We henceforth number links according to their distance from the sender.

Packet tailgating employs several pairs of closely spaced packets; the first packet in each pair is large but has a small time-to-live (TTL) header field of m, whereas the second (tailgating) packet is small but has a large TTL field. Because each router along the path decrements a packet's TTL field by one and discards the packet if it has a TTL of zero, the first packet in each tailgating pair vanishes after link m; the second packet proceeds to the receiver. A chirp in which each probe packet is replaced with a tailgating packet pair as in Figure 2b is called a *packet-tailgating chirp*.

STAB uses packet-tailgating chirps to estimate *subpath available bandwidth*. The subpath available bandwidth up to link m is the minimum available bandwidth among the path's first m links; this bandwidth is a nonincreasing function of m — it decreases at all thin link locations but stays constant between two consecutive thin-link locations. To locate the thin links, all we have to do is find those values of m at which we see a decrease in subpath available bandwidth. The last thin link is obviously the one with the least available bandwidth on the entire path — in other words, the *tight* link.

Packet-tailgating chirps provide a simple scheme for estimating subpath available bandwidth up to link m. Assume we have time stamps that indicate when the probe packets arrive at link m. By replacing the receiver time stamps in pathChirp's algorithm with these time stamps, we get the subpath available bandwidth up to link m.

Although we can't obtain time stamps for packet arrivals at link m for arbitrary m, we can closely approximate them with the small tailgating packets' receiver time stamps. According to the self-induced congestion principle, probe packets face increasing

Related Work in Thin-Link Localization

A recent study used a tool called *BFind* to locate a path's tight link.[1] BFind essentially induces network congestion by continuously transmitting UDP traffic; it then determines the tight link's location from traceroute round-trip times. Another tool, *Treno*, uses UDP packets with limited time-to-live (TTL) fields and router Internet Control Message Protocol (ICMP) echo responses to locate tight links.[2] Both tools have the drawback of introducing excessively large probe traffic loads on the network, which can potentially disrupt existing network traffic. Accordingly, we didn't test or compare them to STAB in our Internet experiments.

The *pipechar* tool provides estimates of raw transmission bandwidth and available bandwidth at each link on an end-to-end path (www-didc.lbl.gov/NCS/). To the best of our knowledge, pipechar's algorithm has not yet been published. Compared to STAB, it has the advantage of not requiring receiver host cooperation to run; it has the disadvantage of requiring routers to respond with ICMP packets when they receive packets with TTL decremented to zero, a feature on routers that administrators sometimes disable. It also requires superuser privileges (the ability to run programs as the root user) at the sender host. In the main text, we compare pipechar to STAB in Internet experiments.

Another tool currently being developed is *pathneck*, which locates thin links by injecting back-to-back bursts of packets into the network.[3] Compared to STAB, pathneck has the same advantages and disadvantages as pipechar.

References

1. A. Akella, S. Seshan, and A. Shaikh, "An Empirical Evaluation of Wide-Area Internet Bottlenecks," *Proc. Internet Measurement Conf.* (IMC), ACM Press, 2003, pp. 101–114.
2. M. Mathis and J. Mahdavi, "Diagnosing Internet Congestion with a Transport Layer Performance Tool," *Proc. Internet Society 6th Ann. INET Conf.*, 1996; www.isoc.org/isoc/whatis/conferences/inet/96/proceedings/d3/d3_2.htm.
3. N. Hu et al., "Locating Internet Bottlenecks: Algorithms, Measurements, and Implications," *Proc. Ann. Conf. Assoc. Computing Machinery's Special Interest Group on Data Comm.* (ACM SIGCOMM), ACM Press, 2004.

queuing delays from congestion only if the probing bit rate exceeds the available bandwidth. Because the large packets vanish after link m, the chirp-probing bit rate decreases drastically after m. As a result, the chirp consisting of only small packets has a probing bit rate that's too low to induce congestion or much queuing delay after link m. Consequently the small packets go through to the receiver, and their interspacing at link m remains more-or-less unperturbed.

STAB initially determines the number of links along the path by incrementing successive probe packets' TTL starting from one. Packets with TTLs smaller than the number of links are dropped along the path due to TTL expiration; others make it to the receiver. The smallest TTL of all the packets to reach the destination thus gives the number of links. STAB then sends out tailgating chirps and varies the large packets' TTLs in successive chirps to estimate the subpath available bandwidth up to link m for different values of m. Finally, the tool determines the probability that link m is a thin link as the fraction of time within a specified window during which the subpath available bandwidth up to link $m - 1$ is greater than that up to link m by a multiplicative factor α. The last link with a high probability of being a thin link is most likely the tight link of the entire end-to-end path. We choose $\alpha = 1.2$ in our experiments.

Validation through Simulations

We use the *double Web farm* topology depicted in Figure 4 for our STAB simulations. (The Web farm is based on a topology provided along with the ns-2 source code, www.isi.edu/nsnam/ns.) Each Web farm consists of 420 clients downloading data from 40 servers over a bottleneck link of 20 Mbps; all other links in the Web farm have 40 Mbps full-duplex bandwidth. Each Web session consists of a client downloading 250 pages from a server. By choosing the page size from a heavy-tailed Pareto distribution, we ensure that the generated traffic has the bursty "fractal" nature ubiquitously present in Internet traffic.[9] We exponentially distribute the interarrival times between page downloads.

We set each Web farm's bottleneck link utilization by starting an appropriate number of Web sessions; by starting 200 Web sessions, for

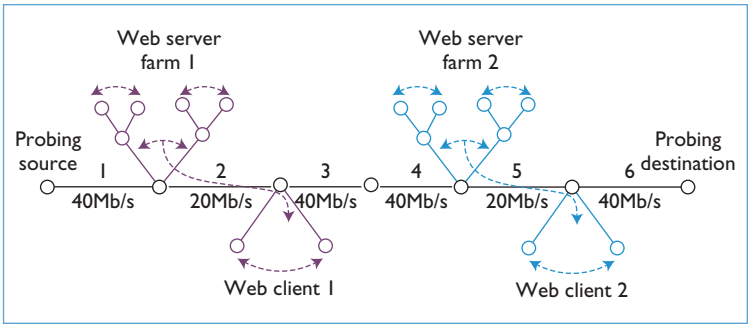

Figure 4. Double Web-farm topology. Web clients download data from servers, and congest links 2 and 5, which become the path's thin links.

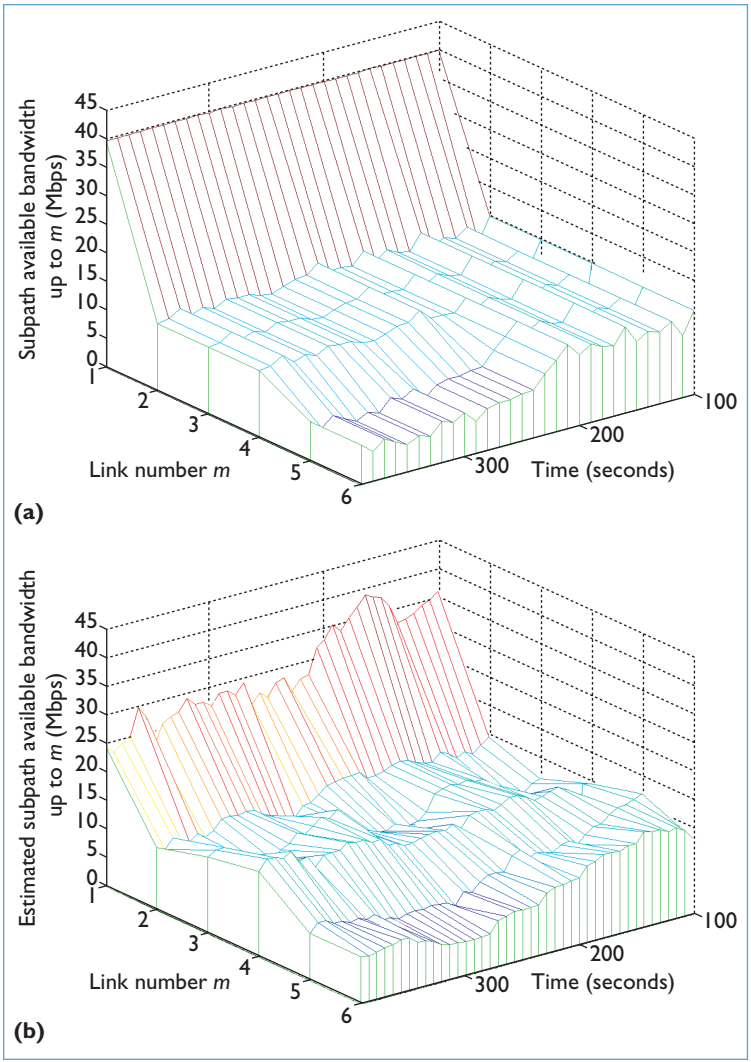

Figure 5. Subpath available bandwidth. The (a) actual and (b) STAB estimates during a simulation with the topology depicted in Figure 4. STAB's estimates track the actual subpath available bandwidth very well, including the dip at link 5 after time t *= 200 sec. The plot's colors at any point represent height, with the blue end of the spectrum representing small heights and the red showing larger heights.*

example, we set the bottleneck utilization to 5 Mbps, by starting 400 Web sessions, we set the bottleneck utilization to 10 Mbps, and so on. From Figure 4, we see that the STAB probes travel across both Web farm bottlenecks before reaching their destinations. All sources, including STAB's, use 1,000-byte packets, which are comparable in size to typical large Internet packets. We set the average probing load to 300 Kbps in all simulations, which is less than 1.5 percent of the raw bandwidth (maximum data-transfer rate) of all the path's links.

Figure 5 depicts the actual subpath available bandwidth up to link m for different intermediate links m and their variation over time. In the first half of the simulation — that is, up to time t = 200 sec — only the first Web farm generates traffic. As a result, link 2 is the path's tight link; consequently, the available bandwidth plot flattens out after link 2 at any time prior to t = 200 sec. We observe that the end-to-end path's available bandwidth is about 15 Mbps at this point in the simulation.

In the second half of the simulation, both Web farms generate traffic. Because the second Web farm generates more than the first, link 5 now becomes the tight link. Observe from Figure 5a that the available bandwidth plot dips at link 5 after time t = 200 sec. We see that the path's available bandwidth is about 5 Mbps at this point in the simulation.

From Figure 5b, we see that STAB accurately estimates subpath available bandwidth. We compute this bandwidth up to link m at any time instant by averaging the estimate of available bandwidth from the past 20 chirps that have large packets' TTL set to m. Observe that prior to time t = 200 sec, the estimates flatten after link 2, but after t = 200 sec, the estimates dip at link 5 due to traffic from the second Web farm.

By comparing both halves of Figure 5, we see that STAB underestimates the first link's available bandwidth by a small amount. This is explained by the fact that subsequent links on a path have a minor influence on chirps consisting of only small tailgating packets, something we neglected in our earlier discussions. Because STAB requires a minimum of 20 chirps in this experiment to form estimates of subpath available bandwidth, it generates these estimates only after time t = 100 sec. We therefore don't plot information prior to this time instant in Figure 5.

Plots such as Figure 5b can prove very useful for optimizing network performance. By choosing an alternate route that bypasses the tight link (link 5) after time t = 200 sec, for example, the receiver can potentially download data from the sender over a path with three times the available bandwidth. If the alternate route also bypasses the first thin link (link 2), it can potentially have eight times the current path's available bandwidth. In practice, we can get alternate routes through multihoming (having more than one connection to the public Internet), by using overlay networks (virtual networks that sit atop the Internet, usually consisting of hosts at the network edge), or with the help of mirror sites (multiple locations from which to download the same data).

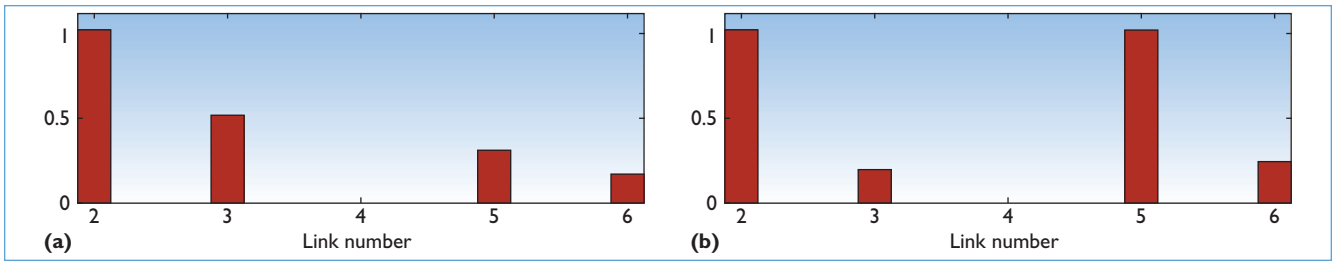

Figure 6. Finding thin links. We compute the probability of different links being thin at time instants (a) t = 180 sec and (b) t = 360 sec from the subpath available bandwidth in Figure 5b. At t = 180 sec, only link 2 has a high probability of being thin; at t = 360 sec, both links 2 and 5 have high probabilities of being thin.

Using subpath-available bandwidth estimates, we can compute the probability that different links on the path are thin. We start by computing the probabilities at any time instant using estimates of the subpath available bandwidth in the past 100 sec. Recall that a link m qualifies as a thin link if it has less available bandwidth than all preceding links, and that the thin link farthest away from the source is the entire path's tight link. Figure 6a plots the probability of different links being thin links at time instant t = 180 sec. We see that link 2 is almost certainly a thin link whereas the other links have low probabilities. This strongly suggests that link 2 is the path's last thin link, which means it's the tight link. Figure 6b plots the probability of different links being thin at time instant t = 360 sec. Here, both links 2 and 5 are almost certainly thin links. Clearly, at this time instant, link 5 is most likely the path's tight link.

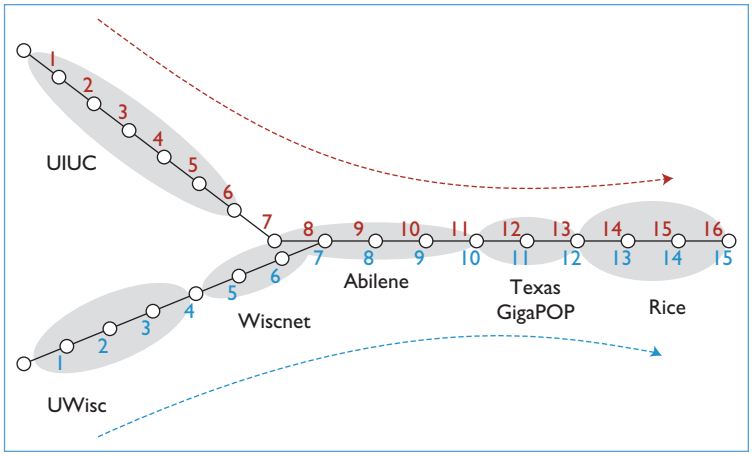

Figure 7. Localization topology. In our Internet experiment, we ran STAB simultaneously on two paths: one from the University of Wisconsin at Madison to Rice University, and the other from the University of Illinois at Urbana-Champaign to Rice.

Internet Experiments

To prove that STAB can locate the thin links in Internet paths, we ran STAB simultaneously on two paths: one from the University of Wisconsin at Madison (UWisc) to Rice University, and the other from the University of Illinois at Urbana-Champaign (UIUC) to Rice. Figure 7 shows that the two paths share eight common links. The results depicted correspond to a 30-minute experiment that began at 9 a.m. on Tuesday, 25 May 2004. STAB used an average probing load of 300 kbps in this experiment.

Figure 8 (next page) plots STAB's estimates of subpath available bandwidth over time for both paths. We computed the bandwidth up to link m at any time instant using the estimates of available bandwidth from the past 30 chirps (those that had large packet TTLs set to m). The plots revealed several interesting facts. Subpath available bandwidth estimates were almost always less than 100 Mbps, which we expect because the very first links of both paths were 100 Mbps Ethernet links. Next, notice how the subpath available bandwidth dips at links 13 in Figure 8a and 14 in Figure 8b, after which the plots flatten out. This strongly suggests that these links are the two paths' tight links. In fact, both correspond to the same 100 Mbps Fast Ethernet link within Rice University (see Figure 7). STAB's estimates for the two paths are consistent.

We can confirm Figure 8's results by plotting the probability of different links being thin. First, we compute the probabilities at any time instant using subpath available bandwidth over the past 3.5 minutes. From Figures 9 and 10, we see that at different time instants in the experiment, link 13 and link 14 are indeed the last links with a high probability of being thin for the UWisc-to-Rice and UIUC-to-Rice paths, respectively. These links are located close to the edge of the end-to-end path, supporting the intuition that congestion normally occurs at the network edge.

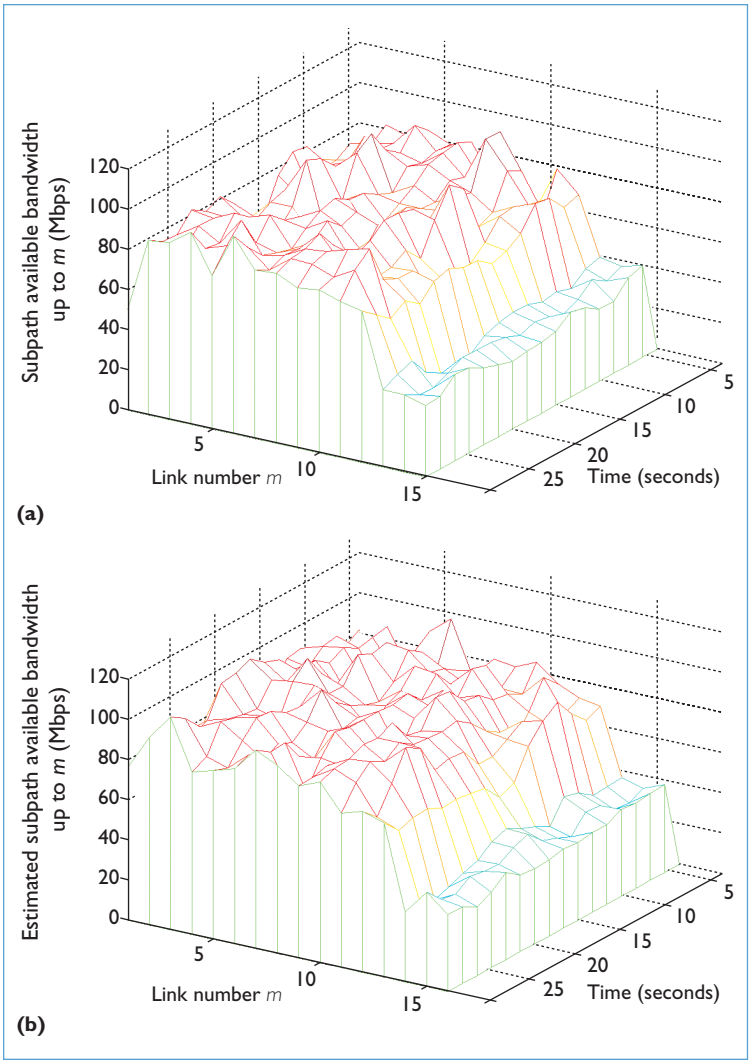

Figure 8. STAB estimates. From the topology depicted in Figure 6, we see the subpath available bandwidth for the two networks: (a) the University of Wisconsin/Madison-to-Rice University, and (b) the University of Illinois/Urbana-Champaign-to-Rice. We see a steep drop at (a) link 13 and (b) link 14, after which the plots flatten out indicating tight links.

We used the Multirouter Traffic Grapher (MRTG; http://people.ee.ethz.ch/~oetiker/webtools/mrtg) tool to get data from all links in the Abilene, Texas-GigaPOP, and Rice University networks belonging to the two paths — except for two OC-12 layer-2 links within the Texas-GigaPOP. These are two of the four layer-2 links that comprise link 12, which is a layer-3 link, of the UWisc-to-Rice path in Figure 7. Among all the links from which we have MRTG data, link 13 of the UWisc-to-Rice path has the least available bandwidth, roughly 80 Mbps. STAB underestimated its available bandwidth to be approximately 50 Mbps (see Figure 8); understanding the causes of this underestimation is part of our ongoing work.

Finally, we compared STAB to the pipechar tool (see the "Related Work in Thin-Link Localization" sidebar). We ran pipechar twice to locate the tight link on the UWisc-to-Rice path immediately after concluding our experiment with STAB. Pipechar estimated that link 12 had the least available bandwidth on the path, slightly less than the available bandwidth of link 13. In the two runs we did, pipechar estimated the available bandwidth to be 45.8 Mbps and 59.4 Mbps at link 12, and 59.4 Mbps and 61.2 Mbps at link 13. Pipechar's estimates for link 13's available bandwidth corroborated STAB's available-bandwidth estimates for the same link, but we can't verify its estimates for link 12 because of incomplete MRTG data.

Conclusions

We plan to enhance STAB, which currently locates thin links only on a single end-to-end network path, to provide detailed maps of the Internet by combining it with network tomography.[10] Network tomography transfers probe packets between multiple sender and receiver hosts to determine various internal properties of the network. This is akin to medical imaging tomography (such as CAT scans), where X-rays or some other form of radiation is sent through a patient from different angles and the results are combined to obtain a detailed 3D internal picture of the patient.

Adapting STAB for use in wireless networks is also an important item on our agenda. Tools based on self-induced congestion, such as STAB, implicitly assume that the network delay of packets is mainly caused by queue build-ups at congested routers. Although this may be true in wired networks, wireless networks encounter other factors such as poor channel quality and interference from neighboring wireless computers, which can introduce significant packet delays and hence can't be ignored. Wireless network probing is a nascent research area with several unexplored and challenging problems.

Acknowledgments

US National Science Foundation grants ANI-0099148 and ANI-0338856, US Department of Energy SciDAC grant DE-FC02-01ER25462, DARPA grant F30602-00-2-0557, and the Texas Instruments Leadership University program supported this work. We thank Jennifer Hou (University of Illinois, Urbana-Champaign) and Robert Nowak (University of Wisconsin) for facilitating STAB experiments at their respective organizations.

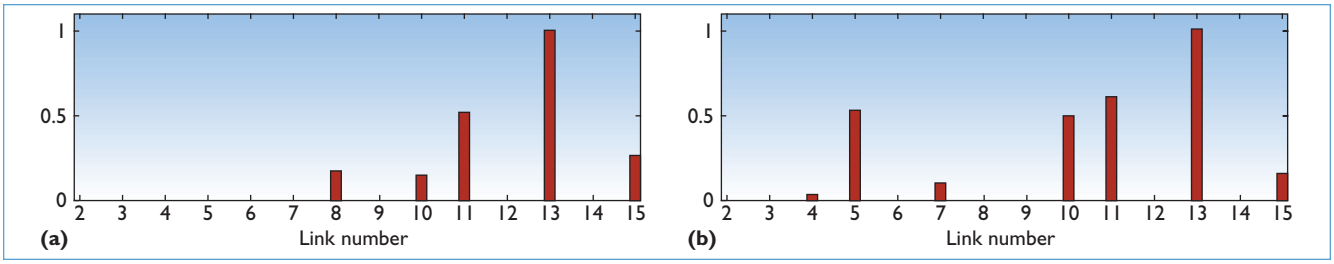

Figure 9. Thin-link probabilities. On the University of Wisconsin/Madison-to-Rice University path, we see that link 13 is the last link with a high probability of being thin at time instants (a) $t = 10$ *min and (b)* $t = 20$ *min.*

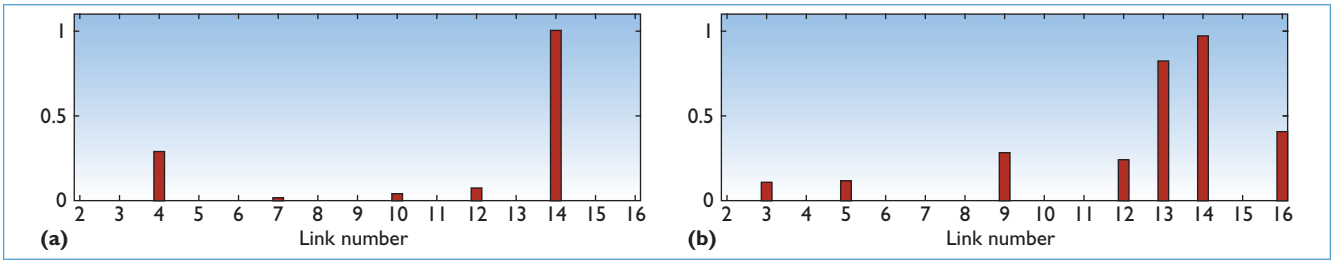

Figure 10. Thin-link probabilities. On the University of Illinois/Urbana-Champaign-to-Rice University path, we see that link 14 is the last link with a high probability of being thin at time instants (a) $t = 10$ *min and (b)* $t = 20$ *min.*

References

1. A. Akella, S. Seshan, and A. Shaikh, "An Empirical Evaluation of Wide-Area Internet Bottlenecks," *Proc. Internet Measurement Conf.*, ACM Press, 2003, pp. 101–114.
2. D. Rubenstein, J, Kurose, and D. Towsley, "Detecting Shared Congestion of Flows via End-to-End Measurement," *IEEE/ACM Trans. Networking*, vol. 10, no. 3, 2002, pp. 381–395.
3. V. Ribeiro et al., "pathChirp: Efficient Available Bandwidth Estimation for Network Paths," *Proc. Passive and Active Measurement Workshop*, 2003, http://moat.nlanr.net/PAM2003/PAM2003papers/3824.pdf.
4. B. Melander, M. Björkman, and P. Gunningberg, "A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks," *Proc. IEEE Globecom Global Internet Symp.*, IEEE CS Press, 2000, pp. 415–420.
5. N. Hu and P. Steenkiste, "Evaluation and Characterization of Available Bandwidth Probing Techniques," *IEEE J. Selected Areas in Comm. Special Issue on Internet and WWW Measurement, Mapping, and Modeling,* vol. 21, no. 6, 2003, pp. 879–894.
6. M. Jain and C. Dovrolis, "End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput," *IEEE/ACM Trans. Networking,* vol. 11, no. 4, 2003, pp. 537–549.
7. G. Jin and B. Tierney, "Netest: A Tool to Measure the Maximum Burst Size, Available Bandwidth, and Achievable Throughput," *Proc. Int'l Conf. Information Technology, Research and Education (ITRE)*, 2003; http://dsd.lbl.gov/DIDC/papers/netest-mbs.pdf.
8. K. Lai and M. Baker, "Measuring Link Bandwidth Using a Deterministic Model for Packet Delay," *Computer Comm. Rev.,* vol. 30, no. 4, 2000, pp. 283–294.
9. W. Leland et al., "On the Self-Similar Nature of Ethernet Traffic (Extended Version)," *IEEE/ACM Trans. Networking*, vol. 2, no. 1, 1994, pp. 1–15.
10. M. Coates et al., "Internet Tomography," *IEEE Signal Processing*, vol. 19, no. 3, 2002, pp. 47–65.

Vinay J. Ribeiro is a PhD candidate in the Department of Electrical and Computer Engineering at Rice University. His research interests include computer networking, signal processing, traffic analysis and modeling, queuing theory, network tomography, and wavelets. Contact him at vinay@rice.edu.

Rudolf H. Riedi is an assistant professor in the Department of Statistics at Rice University. His research interests focus on multiscale stochastics, which concentrate on the theory and application of multiscale models and analysis (such as self-similar processes and multifractals) to networking, economics, and turbulence. Riedi received his PhD from ETH Zürich. Contact him at riedi@rice.edu.

Richard G. Baraniuk is a professor in the Department of Electrical and Computer Engineering at Rice University. His research interests include network modeling and inference, sensor networks, and multiscale geometric analysis for high-dimensional signal processing. Baraniuk received his PhD from the University of Illinois at Urbana-Champaign. Contact him at richb@rice.edu.