Measuring Bandwidth *Signatures* of Network Paths

Mradula Neginhal, Khaled Harfoush, and Harry Perros[1]

Department of Computer Science
North Carolina State University
Raleigh, NC 27695
`[msneginh, harfoush, hp]@cs.ncsu.edu` [*]

Abstract. In this paper, we propose a practical and efficient technique, *Forecaster*, to estimate (1) the end-to-end available bandwidth, and (2) the speed of the most congested (tight) link along an Internet path. Forecaster is practical since it does not assume any a priori knowledge about the measured path, does not make any simplifying assumptions about the nature of cross-traffic, does not assume the ability to capture accurate packet dispersions or packet queueing delays, and does not try to preserve inter-packet spacing along path segments. It merely relies on a simple binary test to estimate whether each probe packet has queued in the network or not. Forecaster is efficient as it only requires two streams of probe packets that are sent end-to-end at rates that are much lower than the available bandwidth of the investigated path, thus avoiding path saturation. Theoretical analysis and experimental results validate the efficacy of the proposed technique.

1   Introduction

The ubiquity of computer networks, our increasing dependence on them, and the need to leverage their utilization, performance and economic value call for ways to measure their *characteristic* features to get a deeper understanding of their behavior. Network *bandwidth* is a key characteristic as it quantifies the data rate (throughput) that a network link or a network path can transfer. Measuring network bandwidth is useful for many Internet applications and protocols especially those involving high volume data transfer among others [4, 11, 27]. The bandwidth available to these applications directly affects their performance.

Research on bandwidth estimation has been quite popular over the last few years. Some researchers have targeted estimates of hop-by-hop *capacity* bandwidth (link speeds) [8, 6, 16]. Others have targeted the end-to-end capacity bandwidth of a network path, defined as the slowest link speed along that path [25, 5, 23, 20, 15]. The link with the slowest speed is referred to as the *narrow* link. The capacity bandwidth of the narrow link bounds the throughput that the path can transfer in the absence of any cross traffic. In the presence of cross-traffic,

---

the situation is different. A link with a higher link speed may be congested and the residual (*available*) bandwidth at this link may be even smaller than the residual bandwidth at the narrow link. In this case the available bandwidth at the most congested link (the one with the least residual bandwidth) will bound the path throughput. This link is referred to as the *tight* link. Tools to estimate the end-to-end available bandwidth of a network path, defined as the available bandwidth at the tight link, have been proposed in [7, 22, 9, 18, 19, 26, 14, 10].

There has also been recent interest in the capacity of the tight link as an important metric to help in the efficient estimation of the end-to-end available bandwidth [26, 7]. Assuming that the tight link of a path remains the same over some reasonably large time-scale, then an estimate of the capacity of the tight link helps track the available bandwidth of the path much more efficiently. To the best of our knowledge, the only technique to estimate the speed of the tight link has been proposed in [12]. The end-to-end available bandwidth of a path and the speed of its tight link thus completely identify the throughput of a network path, and we refer to this tuple for a network path as its bandwidth *signature*.

In this paper we propose a novel technique to estimate the bandwidth signature of a network path. Forecaster does not assume any a priori knowledge about the measured path, does not make any simplifying assumptions about the nature of cross-traffic, does not assume the ability to capture accurate packet dispersions or packet queueing delays, and does not try to preserve inter-packet spacing along path segments. Furthermore, it only requires two streams of probe packets that are sent end-to-end at rates that are much lower than the available bandwidth of the investigated path, thus avoiding path saturation. As a sample comparison, Envelope [12] examines each link on a path separately to obtain the path tight-link capacity estimate, and assumes that inter-packet spacing can be preserved along the path. Both problems are resolved in Forecaster. Forecaster also does not require information about the path capacity as in [26], does not overwhelm the path with probes at a rate as high as the available bandwidth of the path as in [2, 1], and does not require accurate measurements of packet dispersions as in [22]. The key idea is that the end-to-end path *utilization* can be estimated through a simple binary test that measures the fraction of probe packets that experienced queueing, *not* by measuring how much queueing they incurred. By sending two probing streams at different rates and measuring the corresponding path utilization, then the available bandwidth and the speed of the tight link can be *projected*.

The rest of this paper is organized as follows: In Section 2 we provide the theory behind Forecaster and show how to estimate the path utilization. In Section 3 we use the theory furnished in Section 2 to devise the Forecaster algorithm to estimate bandwidth signatures. In Section 4 we validate the efficacy of the proposed technique through simulations. We finally conclude in Section 5.

2 Model

We model each link as a *queue*, and make use of concepts from basic queueing theory to estimate the *utilization* of a network path consisting of a sequence of links (queues). Note that an estimate of a link/path utilization does *not* by itself *tell* about its available bandwidth. For example, a 100 Mbps link with utilization of 0.5 has more available bandwidth than a 10 Mbps link with similar utilization. Still our estimates of the utilization will help in estimating bandwidth signatures as we elaborate in section 3.

2.1 One-Hop Path

In a queueing system consisting of a single queue, i, the utilization, ρ_i, of the system is expressed as

$$\rho_i = 1 - \pi_i^0$$

where π_i^0 is the probability that there are no packets in the queue. This equation is generic and does not make any assumptions about the nature of the cross-traffic. If additional probe packets, transmitted at a rate r bps traverse this queue, then the effective utilization, $\rho_i(r)$, can be expressed as

$$\rho_i(r) = \min\left(1, \rho_i + \frac{r}{C_i}\right) \tag{1}$$

where C_i is the processing speed (the capacity of the modeled link). Notice that $\rho_i(r)$ is a linear function of r, bounded by $\rho_i(r) = 1$, assuming that the *raw* link utilization, ρ_i, is stable over the probing period.

2.2 Multi-Hop Path

Consider a network path consisting of a sequence of H links modeled as H successive queues. Assuming that the utilization of successive queues are *uncorrelated*, then the end-to-end utilization of the system, ρ, can be expressed as

$$\rho = 1 - \prod_{1 \leq i \leq H} (1 - \rho_i) \tag{2}$$

However, the correlation between successive queues is expected to happen as they may be traversed by same flow packets. As shown in [21], correlation only delays convergence and does not lead to divergence. In other words, Equation 2 holds when the system is observed over a larger time-scale. In the experiments in Section 4, we test the performance of Forecaster as correlation exists.

The end-to-end utilization of the system, $\rho(r)$, when probing at a rate of r is used can be expressed as

$$\rho(r) = \min\left(1, 1 - \prod_{1 \leq i \leq H}\left(1 - \left(\rho_i + \frac{r}{C_i}\right)\right)\right) \tag{3}$$

$\rho(r)$ is a non-linear function of r bounded by $\rho(r) = 1$ and can be expressed as a function of degree H of the form

$$\rho(r) = \min\left(1, \sum_{i=1}^{H} c_i r^i\right) \tag{4}$$

where c_i is the i-th coefficient. Simple manipulation of Equation 3 reveals that

$$c_i = \begin{cases} 1 - \displaystyle\prod_{1 \leq k \leq H}(1 - \rho_k) & i = 0 \\[3em] (-1)^{i+1} \displaystyle\sum_{\substack{1 \leq k_1, k_2, \ldots k_i \leq H \\ k_1 \neq k_2 \ldots \neq k_i}} \frac{\displaystyle\prod_{\substack{1 \leq j \leq H \\ j \neq k_1 \neq k_2 \ldots \neq k_i}}(1 - \rho_j)}{C_{k_1} C_{k_2} \ldots C_{k_i}} & 1 \leq i < H \\[3em] (-1)^{H+1} / \displaystyle\prod_{1 \leq k \leq H} C_k & i = H \end{cases}$$

Notice that the first coefficient, c_0, is the end-to-end utilization of the path without any induced probe traffic, ρ. The above equations highlight the fact that $|c_i| \gg |c_{i+1}|$ since the numerator of each constant term, c_i, is relatively small and the denominator dramatically increases as i increases. To visualize the differences in the magnitude of the c_i values, consider a path comprised of 5 links, $H = 5$, with per-link capacities C_i and utilization ρ_i where $i = 1 \ldots 5$. Let the links' capacities be $C_1 = 100$Mbps, $C_2 = 1$Gbps, $C_3 = 100$Mbps, $C_4 = 10$Mbps, and $C_5 = 1$Mbps. Also, let the utilization of the links be $\rho_1 = 0.4$, $\rho_2 = 0.2$, $\rho_3 = 0.3$, $\rho_4 = 0.3$ and $\rho_5 = 0.1$. Substituting these values in Equation 3, $\rho(r) = 0.78 + 2.7 X 10^{-7} r - 4.2 X 10^{-14} r^2 + 1.15 X 10^{-21} r^3 - 9.3 X 10^{-30} r^4 + 10^{-38} r^5$. Thus, $c_0 = 0.78, c_1 = 2.7 X 10^{-7}, c_2 = 4.2 X 10^{-14}, c_3 = 1.15 X 10^{-21}, c_4 = 9.3 X 10^{-30}$ and $c_5 = 10^{-38}$. As values of c_i become negligible for larger i, Equation 4 can be approximated by a function of a degree lower than H. A *first-order approximation* of $\rho(r)$ is of the form

$$\rho(r) \approx \min(1, c_0 + c_1 r) \tag{5}$$

Higher-order approximations are also possible but the gain in accuracy does not justify the extra complexity and overhead as evident from the experimental results in Section 4. Equation 5 generalizes our estimate of the utilization, $\rho(r)$ for a path of arbitrary number of hops, where $c_0 = \rho$ as expressed in Equation 2, and

$$c_1 = \sum_{1 \leq k \leq H} \frac{\prod_{\substack{1 \leq j \leq H \\ j \neq k}}(1 - \rho_j)}{C_k}$$

The similarity between Equations 1 and 5 suggests that the first order approximation of the utilization, $\rho(r)$, for a multi-hop path can be interpreted as

the utilization over a single link with raw utilization of c_0 and link speed of $1/c_1$, assuming that the utilization of the links in the path are stable over the probing period.

2.3 Estimating a Path Utilization ($\rho(r)$)

Forecaster relies on estimates of $\rho(r)$. As discussed in sections 2.1 and 2.2, $\rho(r)$ is the probability of having a probe packet, from a stream of packets transmitted at a rate r over the investigated path, queue in any of the path queues. To obtain an estimate of $\rho(r)$ we send probes from one end of the path towards the other end with exponential inter-departure times at an average rate of r to attain the well-known Poisson Arrival See Time Average (PASTA) property [24]. According to the PASTA property, the probe packets arriving in the queueing system will sample the system queues, on average, as an outside observer would at an arbitrary point in time.

   $\rho(r)$ is estimated as the fraction of probe packets that experienced queueing in the system. Probe packets are time-stamped at both ends and the time difference is used to distinguish probe packets that experienced any queueing from those that did not. Let T^* be the minimum experienced time difference. Assuming that T^* corresponds to packets that did not experience any queueing, then the fraction of probe packets leading to time differences larger than T^* corresponds to $\rho(r)$. This assumption however may not hold especially if the path is highly utilized or if the number of probe packets in the stream is small. Modeling the probing process as a *geometric* distribution with probability $p = 1 - \rho$ of a probe packet not experiencing any queueing delay, the expected number of probes in order to identify the minimum time difference is $(1-p)/p = \rho/(1-\rho)$. As ρ gets larger, the number of probes needed to identify the minimum time difference needs to be larger. By inspecting this equation, one can see that one probe is expected to be enough if $\rho \leq 0.5$ and around 99 probes are needed when $\rho = 0.99$. Obviously, If $\rho = 1$, then all probes will experience queueing delay. This back-of-the-envelope analysis suggests that the number of probe packets needed per stream does not have to be very large.

3   Estimating Bandwidth Signatures

We next show how to use a path utilization, $\rho(r)$, to estimate the path available bandwidth, A, and the speed of the tight link, C.

   The key idea is that when the probing rate, r, becomes equal to the path available bandwidth, A, then $\rho(r)$ reaches its bound, $\rho(r) = 1$. In other words, the following first-order approximation equation holds

$$1 \approx c_0 + A c_1$$

; i.e.

$$A \approx \frac{(1 - c_0)}{c_1} \tag{6}$$

For a one-hop path (Section 2.1), Equation 6 maps to the popular Equation in [5]. In order to estimate A without having to actually probe the path at a rate r equal to A, thus avoiding the need to fill up the communication pipe, we need to find the values of c_0 and c_1. This could be achieved by sending two probing streams at two different rates r_1 and r_2 and measuring $\rho(r_1)$ and $\rho(r_2)$. From Equation 5, we have:

$$\rho(r_1) \approx c_0 + r_1 c_1 \tag{7}$$

$$\rho(r_2) \approx c_0 + r_2 c_1 \tag{8}$$

Solving Equations 7 and 8, we calculate c_0 and c_1. Thus

$$c_1 = \frac{\rho(r_2) - \rho(r_1)}{(r_2 - r_1)} \tag{9}$$

$$c_0 = \rho(r_1) - r_1 c_1 \tag{10}$$

Substituting the values of c_0 and c_1 in Equation 6 we estimate A. Figure 1 provides a schematic representation of the observed utilization, $\rho(r)$, using Equation 3 and neglecting the bound on $\rho(r)$, as we vary the probing rate, r, both for (A) the single-hop ($H = 1$) and (B) the multi-hops ($H = 3$) cases. Notice that there is only one r leading to $\rho(r) = 1$ in the $H = 1$ case, and that there are three different r values leading to $\rho(r) = 1$ in the $H = 3$ case. The r values leading to $\rho(r) = 1$ are basically the rates that are needed to fill each link. In practice, and as reflected in Equation 3, $\rho(r)$ is bounded by 1 and the smallest r value leading to $\rho(r) = 1$ is the available bandwidth in the tight link, A. Figure 1 also shows how the projection of the line connecting the points $(r_1, \rho(r_1))$ and $(r_2, \rho(r_2))$ in these figures till the point $(A, 1)$ reveals the value of A. The speed of the tight link, C, also results from a projection of the line in the other direction, towards $\rho = 0$, as shown in the figures. We note that $1/c_1$ is exactly C. Thus

$$C = \frac{1}{c_1} \tag{11}$$

Notice how the first order approximation leads to conservative estimates of A and C. The overall procedure is sketched in Algorithm 1.

Table 1. Common Internet links and their capacities

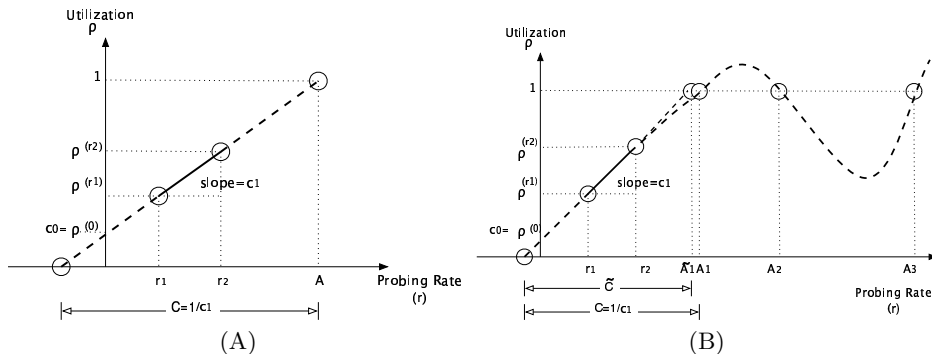| Link Type	Mbps	Link Type	Mbps
10BaseT	10.000	OC1	51.840
100BaseT	100.000	OC12	622.080
1000BaseT	1000.000	OC96	4976.000
		OC192	9953.280

Fig. 1. The observed utilization as we vary the probing rate for (A) a one link path, and (B) a multi-hop path.

Algorithm 1 Forecaster algorithm: Estimating the end-to-end available bandwidth and the capacity of the tight link of a network path

---

Measure $\rho(r_1)$ and $\rho(r_2)$ resulting from two probe sequences, at rates of r_1 and r_2, respectively
$c_1 = (\rho(r_2) - \rho(r_1))/(r_2 - r_1)$
$C = 1/c_1$
$c_0 = \rho(r_1) - r_1 c_1$
$A = (1 - c_0)/c_1$
return A, C

---

4   Experimental Results

We next validate our proposed approach using $ns - 2$ simulations [3].

4.1   Setup

The link speeds used in our simulations are picked from common standards – shown in Table 1. Cross-traffic packets are introduced following measurement findings in [17] and [13], with 60% of the cross-traffic for TCP traffic and the remaining 40% for UDP traffic. Cross-traffic packet sizes are distributed between 40 bytes, 576 bytes and 1500 bytes following the observed Internet trends. Some of the cross-traffic flows used are *hop-persistent* and some are *path-persistent*, that is some travel only along one hop and some travel along large segments of the investigated path. Path-persistent flows introduce larger correlation between the path queues, which we intentionally introduce to monitor its impact of on our bandwidth estimates, as discussed in Section 2. Probe packets are sized at 1500 bytes. Recall that two probe sequences at two different rates, r_1 and r_2 are used. We use 200 probe packets per sequence and pick r_1 and r_2 to induce a pronounced difference in the observed path utilization, $\rho(r_2) - \rho(r_1) \geq 0.1$, whenever possible. r_1 is picked to be a small rate (50kbps) and r_2 as $C_1/10$ where C_1 is the speed of the link connected to the probing host, which is typically known.

4.2 Test Cases

Table 2. Simulated Path Setups.

	L_1	L_2	L_3	L_4
Scenario I	100BaseT	OC12	10BaseT	-
Scenario II	1000BaseT	OC192	100BaseT	-
Scenario III	OC1	OC96	10BaseT	-
Scenario IV	1000BaseT	OC96	OC12	100BaseT

We simulate four different path setups (scenarios), each consisting of either three or four links of different link capacities as shown in Table 2. In all scenarios we set up links of higher capacities in the middle of the simulated path and links of lower capacities at the edge, which is generally the case in the Internet. However, we test the performance of our technique when the links in the middle are the tight links.

For each scenario, we experiment with four different cases, which differ in the utilization induced by cross-traffic on each link. Consequently, different cases may differ in the available bandwidth, the speed and the location of the tight link. The test cases have been formulated such that

- Case 1: Base case – No competing cross-traffic on any of the links.
- Case 2: The narrow link is the tight link.
- Case 3: The narrow link is not the tight link.
- Case 4: The high speed link in the middle of the network path is the tight link. This case is typically the hardest to estimate since a tight high-speed link implies that its utilization needs to be very large, at least 90% in all our cases. It has been shown in [10] that as the utilization of the tight link increases, so does the variation in the average available bandwidth, the metric that we are measuring.

Cases 2,3, and 4 are designed to stress-test our estimation methodology in heavily loaded network paths. Case 4 is not very common in the Internet but is designed to pinpoint the limitations of Forecaster. Tables 3, 4, 5, and 6 summarize the utilization of the links and highlight the narrow and tight links in all cases. An important observation that is evident from our analysis and results is that the order of the links does not change the results. For example, in all investigated cases, the narrow link is the last hop on the path. Making the narrow link as the first hop does not change our results or conclusions.

4.3 Performance Metrics

We use two performance metrics namely: (1) ϵ^A, which is the error in the estimation of the available bandwidth and (2) ϵ^C, which is the error in the estimation

of the capacity of the tight link. These metrics are defined as follows:

$$\epsilon^A = \frac{A - \overline{A}}{A} \tag{12}$$

where A is the actual end-to-end available bandwidth of the network path and $\overline{A}$ is the estimated value.

$$\epsilon^C = \frac{C - \overline{C}}{C} \tag{13}$$

where C is the actual capacity bandwidth of the tight link on the network path and $\overline{C}$ is the estimated value.

Table 3. Accuracy of A and C for Considered Test Cases of Scenario I.

Scenario I					
Case	ρ_1	ρ_2	ρ_3	ρ_4	Narrow Link	Tight Link
1	0.0	0.0	0.0	-	L_3	L_3
2	0.3	0.4	0.2	-	L_3	L_3
3	0.95	0.2	0.01	-	L_3	L_1
4	0.2	0.99	0.01	-	L_3	L_2
Case	A (Mbps)	$\overline{A}(Mbps)$	ϵ^A	C (Mbps)	$\overline{C}$ (Mbps)	ϵ^C
1	10	10.14	-0.014	10	10.16	-0.016
2	8	8.14	-0.017	10	24.3	-1.43
3	5	5.09	-0.018	100	127.15	-0.271
4	6.22	6.12	+0.016	622.08	722.03	+0.16

Table 4. Accuracy of A and C for Considered Test Cases of Scenario II.

Scenario II					
Case	ρ_1	ρ_2	ρ_3	ρ_4	Narrow Link	Tight Link
1	0.0	0.0	0.0	-	L_3	L_3
2	0.2	0.1	0.5	-	L_3	L_3
3	0.92	0.1	0.01	-	L_3	L_1
4	0.1	0.99	0.0	-	L_3	L_2
Case	A (Mbps)	$\overline{A}(Mbps)$	ϵ^A	C (Mbps)	$\overline{C}$ (Mbps)	ϵ^C
1	100	100.37	-0.004	100	100.39	-0.004
2	50	52.72	-0.049	100	148.48	-0.484
3	80	72.1	+0.098	1000	994.13	+0.006
4	89.6	93.05	-0.038	9953	8481.9	+0.148

4.4 Results

In tables 3, 4, 5, and 6 we report on the accuracy of our estimates of A and C for each simulated case. It is clear that the $\overline{A}$ estimates are accurate especially in cases 1, 2 and 3. In case 4 , the heavy cross-traffic rate on the high speed tight link restrains Forecaster's accuracy, and arguably, would stress any bandwidth estimation technique. In such case, the estimation error, ϵ^A, can be as high as 20% (Scenario III). The accuracy of the tight link capacity estimates, $\overline{C}$, is a different story. The error, ϵ^A, is much higher in some cases, reaching 143% in one case (Case 2, Scenario I).

In general, the errors in Forecaster's estimates may be due to (1) the correlation between successive queues, (2) the large utilization preventing the identification of the end-to-end no-queueing delay, and/or (3) the approximation in the $\rho(r)$ equation for the multi-hop case (Equation 5). While errors due to (1) and (2) can be resolved by increasing the number of probe packets per stream, errors due to (3) cannot. However, errors in capacity estimates can in general be reduced by noting that link capacities typically have standard link speeds and are typically not assigned arbitrary values. For example, by matching the estimate $\overline{C} = 24.3 Mbps$ of Case 2, Scenario I to the closest standard link speeds in Table 1, one can see that the closest link speed is 10BaseT, the correct link speed. In fact, matching the $\overline{C}$ of *all* considered cases to Table 1 leads to the correct tight link bandwidth estimates.

Table 5. Accuracy of A and C for Considered Test Cases of Scenario III.

Scenario III						
Case	ρ_1	ρ_2	ρ_3	ρ_4	Narrow Link	Tight Link
1	0.0	0.0	0.0	-	L_3	L_3
2	0.5	0.1	0.2	-	L_3	L_3
3	0.1	0.2	0.7	-	L_3	L_1
4	0.0	0.99	0.1	-	L_3	L_2
Case	A (Mbps)	$\overline{A}(Mbps)$	ϵ^A	C (Mbps)	$\overline{C}$ (Mbps)	ϵ^C
1	51.84	52.37	-0.010	51.84	52.42	-0.011
2	25.92	25.54	+0.015	51.84	55.14	-0.064
3	30	29.14	-0.028	100	102.34	+0.0214
4	50.2	39.92	+0.204	4976	3360.14	+0.324

5 Conclusions and Future Work

We introduced Forecaster, a promising bandwidth-estimation tool. Simulation results reveal that Forecaster estimates the available bandwidth and the speed of the tight link with reasonable accuracy in most reasonable scenarios. We also introduced a technique to correct deviating tight link capacity estimates.

Table 6. Accuracy of A and C for Considered Test Cases of Scenario IV.

Scenario IV						
Case	ρ_1	ρ_2	ρ_3	ρ_4	Narrow Link	Tight Link
1	0.0	0.0	0.0	0.0	L_4	L_4
2	0.2	0.1	0.1	0.3	L_4	L_4
3	0.95	0.1	0.1	0.01	L_4	L_1
4	0.2	0.99	0.2	0.01	L_4	L_2
Case	A (Mbps)	$\overline{A}(Mbps)$	ϵ^A	C (Mbps)	$\overline{C}$ (Mbps)	ϵ^C
1	100	100.33	-0.003	100	100.33	-0.003
2	70	71.1	-0.015	100	127.22	-0.272
3	50	51.73	-0.035	1000	1276.65	-0.277
4	49.76	43.15	-0.133	4976	5350.6	-0.075

We are currently testing Forecaster in a controlled lab setup before Internet deployment. Another goal that we intend to pursue is to integrate Forecaster in a rate-adaptive congestion control protocol.

References

1. Bbftp. http://doc.in2p3.fr/bbftp/
2. Iperf. http://dast.nlanr.net/Projects/Iperf/
3. The network simulator-ns-2. http://www.isi.edu/nsnam/ns/
4. Andersen, D., Balakrishnan, H., Kaashoek, F., Morris, R.: Resilient Overlay Networks. Symposium on Operating Systems Principles, (2001) 131–145
5. Dovrolis, C., Ramanathan, P., Moore, D.: Packet-Dispersion Techniques and a Capacity Estimation Methodology. IEEE/ACM Transactions on Networking (TON), **12** (2004) 963–977
6. Downey, A.: Clink: A tool for estimating internet link characteristics. http://www.rocky.wellesley.edu/downey/clink/
7. Hu, N., Steenkiste, P.: Evaluation and characterization of available bandwidth and probing techniques. IEEE JSAC Special Issue in Internet and WWW Measurement, Mapping, and Modeling, **21** (2003) 879–894
8. Jacobson, V.: pathchar - A Tool to Infer Characteristics of Internet Paths. ftp://ee.lbl.gov/pathchar
9. Jain, M., Dovrolis, C.: End-to-End Available Bandwidth : Measurement Methodology, Dynamics, and Relation with TCP Throughput. IEEE/ACM Transactions on Networking, **11** (2003) 537–549
10. Jain, M., Dovrolis, C.: End-to-End Estimation of the Available Bandwidth Variation Range. ACM Sigmetrics, (2005)
11. Jannotti, J., Gifford, D., Johnson, K., Kaashoek, F., O'Toole, J.: Overcast: Reliable Multicasting with an Overlay Network. Symposium on Operating System Design and Implementation, (2000) 197–212
12. Kang, S., Liu, X., Bhati, A., Loguinov, D.: On Estimating Tight-Link Bandwidth Characteristics over Multi-Hop Paths. 26th IEEE International Conference on Distributed Computing Systems, (2006) 55–65

13. Fomenkov, M., Keys, K., Moore, D., Claffy, K.: Longitudinal study of internet traffic in 1998-2003. http://www.caida.org/publications/papers/2003/nlanr/nlanr_overview.pdf

14. Kiwior, D., Kingston, J., Spratt, A.: PATHMON, A Methodology for determining Available Bandwidth over an Unknown Network. IEEE Sarnoff Symposium on Advances in Wired and Wireless Communications, (2004)

15. Lai, K., Baker, M.: Nettimer: A tool for measuring bottleneck link bandwidth. USENIX Symposium on Internet Technologies and Systems, (2001) 123–134

16. Mah, B.: pchar: A Tool for Measuring Internet Path Characteristics, (2001). http://www.kitchenlab.org/www/bmah/Software/pchar/

17. McCreary, S., Claffy, K.: Trends in Wide Area IP Traffic Patterns - A View from Ames Internet Exchange. 13th ITC Specialist Seminar on Internet Traffic Measurement and Modelling, (2000)

18. Melander, B., Bjorkman, M., Gunningberg, P.: A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks. IEEE Global Internet Symposium, (2000)

19. Melander, B., Bjorkman, M., Gunningberg, P.: Regression-Based Available Bandwidth Measurements. International Symposium on Performance Evaluation of Computer and Telecommunications Systems, (2002)

20. Paxson, V.: End-to-End Internet Packet Dynamics. IEEE/ACM Transactions on Networking (TON), **7** (1999) 277–292

21. Presti, F., Duffield, N., Horowitz, J., Towsley, D.: Multicast-based inference of network-internal delay distributions. IEEE/ACM Transactions on Networking, **10** (2002) 961–775

22. Ribeiro, V., Riedi, R., Baraniuk, R., Navratil, J., Cottrell, L.: PathChirp: Efficient Available Bandwidth Estimation for Network Paths. Proceedings of The Conference on Passive and Active Measurements, (2003)

23. Carter, R., Crovella, M.: Measuring Bottleneck Link Speed in Packet-Switched Networks. *Performance Evaluation*, **27** (1996) 297–318

24. Wolff, R.: Poisson Arrivals See Time Average. *Operations Research*, **30** (1982) 223–231

25. Keshav, S.: A Control-Theoretic Approach to Flow Control. ACM Sigcomm, (1991)

26. Strauss, J., Katabi, D., Kaashoek, F.: A Measurement Study of Available Bandwidth Estimation Tools. ACM/USENIX Internet Measurement Conference (IMC), (2003)

27. Zhu,Y., Dovrolis, C., Ammar, M.: Dynamic Overlay Routing Based on Available Bandwidth Estimation: A Simulation Study. *Computer Networks*, **50** (2006) 742–762