Empirical Study of the Impact of Sampling Timescales and Strategies on Measurement of Available Bandwidth*

Alok Shriram and **Jasleen Kaur**

University of North Carolina at Chapel Hill

Abstract - Several tools have been designed for measuring end-to-end available bandwidth (AB) of a path by injecting probe traffic on the path and inferring AB based on the end-to-end delays observed. While the algorithmic aspects of designing probe streams and inferring AB have received considerable attention, most tool designs have ignored an important aspect of measuring AB—that of the sampling timescales and strategies used. In this paper, we address this issue by studying the impact of the measurement time-scale, tool run-time, sampling strategy, and sampling intensity, on the accuracy, variability, and predictability of the estimated AB. We passively analyze link-level packet traces collected from 15 Internet links (8 different locations). Our analysis intentionally ignores the tool-specific algorithmic aspects of designing probe streams and inferring AB—our conclusions, therefore, are applicable to a wide variety of AB estimation tools. We use our analysis to derive several guidelines for tool design.

1 Introduction

Several applications—including media streaming, overlay routing, and bulk-transfers of large datasets—can benefit from the knowledge of the end-to-end available bandwidth (AB) on a network path. Several tools—henceforth, referred to as AB estimation tools (ABETs)—that measure the end-to-end AB on a given network path have been designed in the recent past [6, 5, 12, 14, 16]. These tools typically operate by first injecting probe traffic into the network path, and then by observing the one-way delays experienced by the probe packets. These delay estimates are then used to infer the end-to-end AB. Existing ABET designs focus primarily on, and differ most significantly in, the construction of probe streams and in the logic used to estimate AB from the observed delays. For instance, Pathload [5] uses the notion of a *stream*, which is a set of packets sent at a constant rate, while Pathchirp [14] uses the notion of *chirps*, which are exponentially-spaced trains of packets. Most tool designs, however, seem to ignore three central temporal quantities related to measurement of the AB process—that of the *tool run-time*, the *measurement time-scale*, and the *sampling intensity and strategies*. In particular, not much is known about the impact of these quantities on the accuracy, variability, and predictability of the measured AB.

In this paper, we address this issue by investigating three key questions: *(i) how do the sampling strategy, intensity, and timescales affect the accuracy of the AB estimates? (ii) how do tool run-times and measurement-timescales impact the variability of the measured AB? And (iii) how stable is the available bandwidth in the post-measurement period?* In order to answer these questions, we passively analyze link-level packet traces collected from 15 different Internet links and derive several guidelines for ABET design. Our analysis is independent of the design parameters of existing ABETs—our findings are, therefore, applicable to most tools.

In what follows, we outline our objectives and approach in Section 2. Sections 3, 4, and 5 present our analysis results. We summarize our conclusions in Section 6.

2 Formulation

AB on a link is defined as [5]:

$$AB[t_1, t_2] = C - \frac{B(t_1, t_2)}{(t_2 - t_1)} \tag{1}$$

* This research was supported in part by NSF CAREER grant CNS-0347814, a UNC Junior Faculty Development Award, and NSF RI grant EIA-0303590.

where $AB[t_1, t_2]$ is the AB of the link over a given time interval $[t_1, t_2]$, C is the transmission capacity of the link, and $B(t_1, t_2)$ is the amount of traffic transmitted on the link during $[t_1, t_2]$.[1] The end-to-end AB of a path is defined as the minimum of the AB of its constituent links [5].

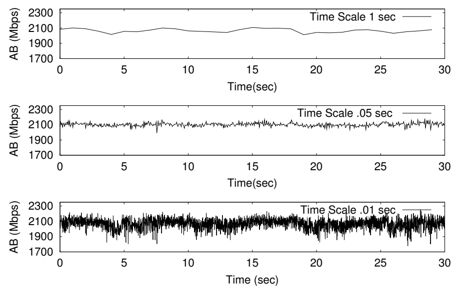

Fig. 1. AB Observed at Different Time-scales

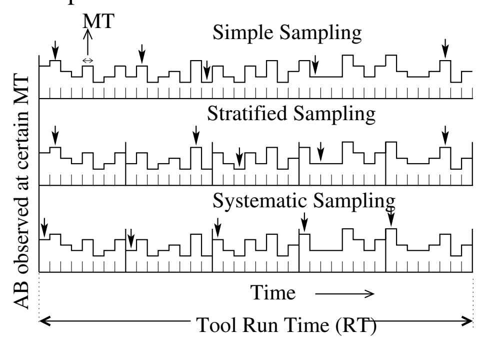

Fig. 2. Sampling Strategies

Several tools (ABETs) have been proposed recently for actively probing for the end-to-end AB on a given network path [6, 5, 9, 12, 14, 16]. These tools operate by injecting specially-designed streams of probe packets onto the path, observing the end-to-end delays experienced by the probe packets, and estimating the end-to-end AB from the observations. Each tool typically injects and observes several such probe streams before converging to an AB estimate. Fig 2 illustrates this approach—the length of individual probe streams determines the timescale at which the AB process is observed; arrows depict the times at which an ABET sends a probe stream.

Existing tools focus primarily on, and differ in the design choices they make along two dimensions: (i) the structure of a probe stream—for instance, while Abing [12] and Spruce [16] rely on using a *packet-pair* as a probe stream, Pathload [5] and PathChirp [14] rely on sending a *packet train* (uniformly and exponentially-spaced, respectively) in each probe stream—and (ii) the inference logic used for estimating AB from the end-to-end delays observed by the probe stream. Unfortunately, most existing tool designs ignore the following additional, yet fundamental, aspects of measuring AB:[2]

Measurement Timescale: A critical parameter in the definition of AB in Eq (1) is the length, $(t_2 - t_1)$, of the time interval over which it is observed—we refer to this quantity as the *measurement timescale* (MT). Note that the MT of an ABET is given by the probe-stream length used by it. In Fig 1, we plot the time-series of AB, observed at three different timescales of $10ms$, $50ms$, and $1s$, during the *same* $30s$ observation period on the Abilene-IC0 link (described in Fig 3). Not surprisingly, we observe that the AB process and its variability can be quite different at different timescales. Consequently, any application that relies on an ABET would want the tool to measure AB at an MT relevant to the application domain. For instance, while a large-file-transfer application is likely to be interested in only the average AB obtainable at super-second timescales, a media-streaming application is likely to also be interested in knowing the small-timescale variations in AB. The MT is also likely to impact the AB estimation accuracy of an ABET [11].

Unfortunately, most existing ABETs do not explicitly select (or report) the MT used in AB estimation. Furthermore, the implicit choices of MT made by these tools can

[1] Note that the availability of a link-level packet trace allows easy computation of $B(t_1, t_2)$. If C is known, Eq (1) can be used to compute the exact value of $AB[t_1, t_2]$ from the packet trace.

[2] The importance of considering measurement timescales and durations has also been mentioned in [7]. However, the impact of these parameters on AB measurement has not been quantified.

only be roughly estimated, and are a function of the path transmission capacity and tool configuration parameters. Tools such as Spruce [16] and Abing [12], that rely on using a packet-pair as a probe stream, have a MT on the order of $12\mu s$, on a $1Gbps$ path—this corresponds to the separation between two back-to-back $1500B$ packets.[3] Tools such as Cprobe [3], PathChirp [14], and Pathload [5], that instead rely on using longer packet trains as probe streams, have a much larger MT—ranging from $10ms$ to several hundreds of ms on a $1Gbps$ path. The exact value of the MT for a probe stream depends on the size of the packet train and the rate at which it is sent—both of these factors are adaptive in Pathload and PathChirp. Iperf [13], which is a tool used primarily for diagnostic purposes, measures the maximum throughput that a TCP connection can attain[4]—the MT is the same as the total tool run-time.

In this paper, we study how the choice of MT by a tool impacts the accuracy, variability, and stability of the measured AB. We use four different values of MT, representative of existing tools, that differ by more than an order of magnitude: $10ms$, $50ms$, $100ms$, and $500ms$.

Sampling Strategy and Intensity: Given an observation timescale, the *AB process* consists of a series of back-to-back readings of AB observed within a given time interval. ABETs essentially only *sub*-sample this AB process—the sampling strategy and the fraction of the AB process sampled are likely to impact the accuracy of estimating the mean AB in a given time interval. For instance, a larger sampling rate is likely to result in better AB estimation accuracy; however, it would also incur greater network overhead.

Existing tools differ in the fraction of the AB process—henceforth, referred to as the *sampling intensity* (SI)—that they sample during the tool run-time. In our analysis, we vary this fraction from 0.1 to 0.9 ($10-90\%$ of the AB process gets sampled). We vary SI by simultaneously controlling the MT and the sampling rate (number of AB samples collected per second). SI is given by the product of MT and the sampling rate.

Given a sampling rate, existing tools also differ in their *sampling strategy*—the manner in which AB samples are selected from within a given time interval. We use the framework described in [4] to study three kinds of sampling strategies (see Fig 2): (i) *Simple sampling*, in which AB samples are selected randomly from within the given time interval; (ii) *Stratified sampling*, in which the time interval is divide into equi-sized units, and one sample is selected randomly from each unit; and (iii) *Systematic sampling*, in which the time interval is divided into equi-sized units and the first AB reading from each unit is used as a sample. Spruce uses simple sampling, while Pathchirp, Pathload, and Abing use systematic sampling.

Measurement Duration (Run-time): *Run-time* (RT) refers to the length of the time interval over which several samples of the AB process are collected, and used to infer properties of the AB process. In practical terms, the run-time is the total time taken by a tool from invocation to reporting an AB estimate. This includes the time taken to send several probe streams (each of which potentially returns one sample of AB), and converge on an AB estimate.

The most significant impact of run-time on AB measurement is in terms of its variability. For a given MT and SI, the longer is the tool run-time, the more variable

---

[3] Tools that rely on packet-pairs have been shown to be inaccurate, especially on high-speed paths [15]. This is conjectured to be so primarily because of the small MT—at such timescales, the AB process appears quite bursty. As a result, it is difficult to get reliable and stable AB estimates. We exclude such timescales from our analysis in this paper.

[4] It has been shown in [5] that TCP throughput is not an accurate measure of AB.

are likely to be the different AB samples collected. On the other hand, longer run-times are more likely to yield a *sufficient* number of samples for reliably estimating the mean as well as variability in the AB process.

RT (as well as MT) is also likely to affect the stability of the measured AB in the post-measurement periods. A longer run-time is likely to yield more reliable AB estimates, that are not subject to short-term traffic-load fluctuations, and are indicative of the AB that can be expected for some time.

Existing AB tools vary widely in their typical run-times—an recent evaluation study of AB tools reports the typical run-times of Abing, Spruce, Pathchirp, Iperf, and Pathload to be: 1, 2, 5, 10, and 20 seconds, respectively [15]. We use these values to study the impact of tool run-time on the variability as well as stability of the AB process during measurement and post-measurement periods, respectively.

We organize the issues raised above in the form of three main questions that are addressed in the rest of this paper: (i) How does the choice of sampling strategy, sampling intensity, MT and RT impact the *accuracy* of the estimated AB? (ii) How does the choice of MT and RT affect the *variability* of the measured AB? (iii) How *stable* is AB in the post-measurement periods? Our work represents, to the best of our knowledge, the *first* investigation of AB measurement along these dimensions.

2.1 Analysis Methodology

Trace	Link 0 Average Load (Mbps)	Link 1 Average Load (Mbps)
1 Gbps Links		
UNC	328.8	88.2
Leip	13.07	35.813
Cesca	228.2	245.9
SanDiego	68.01	39.3
2.5 Gbps Links		
Abilene IC	421.6	518.4
Abilene IK	320.7	585.8
MFN	349.1	608.1
Paix	107.2	n/a

Fig. 3. Data Sets

As mentioned before, existing ABETs focus primarily on the design of probe streams and an inference logic. In order to answer the above questions in a tool-independent manner, hence, we assume the existence of a perfect probing stream—referred to as an *Istream*—and a corresponding perfect inference logic, that can infer the sampled AB perfectly by analyzing the performance of an *Istream*. This assumption lets us study the impact of (currently) design-agnostic quantities—namely, run-time, measurement timescale, and sampling intensity and strategies—while isolating the analysis from the impact of design-dependent parameters. It also lets us adopt a *passive* trace-analysis based approach for answering the above questions, in which it is possible for us to compute the *ground truth*—as described before, the availability of a link-level packet trace gives us the ability to compute perfectly the AB process on the corresponding link at different timescales. We use the Coralreef [10] package for this processing.

Note that the use of link-level packet traces gives us access to the AB process of only a *single link*, and not the *end-to-end* AB process of a network path. Computing the latter passively would require access to the link-level packet traces of *all* the constituent links of a path—given the limited number of publicly-available packet traces, that is currently infeasible. Note, however, that in practice, analyzing just the link-level AB process may not be a significant limitation. This is because most end-to-end paths are expected to have at most a single bottleneck link, which is not likely to change during a tool run [5, 7, 12, 14, 16]—the AB process on such a bottleneck governs the end-to-end AB process [15].

We use link-level packet traces collected from 8 different locations (15 different bi-directional links). Fig 3 lists these traces. All links have gigabit or higher capacities—our results are, therefore, applicable to high-speed networks on which ABETs are expected to be increasingly deployed [17, 18]. Our traces are diverse in the link-locations, traffic loads, and user-communities represented. The UNC and Leip traces were collected, respectively, at the edges of the University of North Carolina and the University of Liepzig. The Abilene, MFN, Cesca, Paix, and San Diego traces were obtained from

CAIDA [1] and NLANR [2]. Due to space constraints, we present most of our analysis results for the UNC-0 and AbileneIC-1 traces, which capture the diversity of our observations. Most of the observations yielded by the other traces are similar to the UNC traces—we include results from other traces when this is not the case.

3 How does the way AB is sampled affect accuracy?

AB estimation tools necessarily *sub*-sample the AB process during their run-time. In what follows, we evaluate the impact of sampling strategies, intensity, timescale, and duration on the accuracy of the sampled AB. It is worth noting that a recent experimental study has shown that the accuracy of existing ABETs is no better than 10% on high-speed paths [15]. In this section, consequently, we consider any inaccuracy smaller than this value as insignificant.

3.1 Does the choice of sampling strategy impact accuracy of the sampled AB?

We consider the three kinds of sampling strategies—simple, stratified, and systematic—described in Section 2. For a given choice of MT, SI, and RT, we analyze each packet-trace as follows: (i) we translate the trace into a corresponding AB process observed at the timescale MT; (ii) we divide the AB process into segments of time-length RT each (Fig 2 depicts one such segment); (iii) for each segment i: (a) we compute the average, AB^i_{avg}, of the AB process observed within that segment; (b) we sub-sample the AB process according to the three sampling strategies—simple, stratified, and systematic (see Fig 2)—and compute the averages of the samples as: AB^i_{sim}, AB^i_{stra}, and AB^i_{sys}, respectively; (c) we compute the *sampling inaccuracies* for the segment as: $|AB^i_{avg} - AB^i_{sim}|$, $|AB^i_{avg} - AB^i_{stra}|$, and $|AB^i_{avg} - AB^i_{sys}|$, respectively; and (iv) we compute the cumulative distribution (CDF) of these three inaccuracy metrics, over all segments in the trace.

Strategy	UNC-0			Abilene-IC1			Abilene-IK1		
	5 %	50 %	95 %	5 %	50 %	95 %	5 %	50 %	95 %
Simple	0.04	0.38	1.10	0.12	1.3	3.91	0.13	1.45	4.64
Systematic	0.08	0.9	3.14	0.62	6.21	19.0	0.64	6.84	29.38
Stratified	0.79	3.12	5.57	0.59	6.74	19.23	0.72	7.49	29.78

Fig. 4. Sampling Strategy vs. Inaccuracy (Mbps)

Fig 4 lists the 5%, 50%, and 95% of the inaccuracies for the three sampling strategies, observed within the UNC-0, Abilene-IC1, and Abilene-IK1 traces, with MT = 10 ms, SI = 0.7, and RT = 10s. We observe that the median inaccuracy in measuring AB is smaller with simple sampling (within 1.5 Mbps and 0.4 Mbps for the Abilene and UNC traces, respectively) than with systematic or stratified sampling (7 Mbps and 3 Mbps for the Abilene and UNC traces, respectively). A similar trend is visible for the 95% values of the computed inaccuracies. However, for *all* traces analyzed, we find that even the 95% values of the inaccuracies lie within 10% of the link AB—this is close to the resolution accuracy of existing ABETs. Thus, it may be fair to conclude that *although simple strategy is likely to yield better sampling accuracy, the inaccuracies of systematic and stratified sampling are not significant for current tools*. Since most existing ABETs rely on systematic sampling, we use it in all of our subsequent analysis.

3.2 How does probe-stream duration impact the accuracy of estimated AB?

The duration of individual probe streams transmitted within the run-time of a tool determine its MT—indeed, each probe stream samples the AB process for this amount of time. In order to assess the impact of MT on a tool's accuracy, we analyze each trace as follows. Using systematic sampling, for a given RT, SI, and MT , we compute the CDF of the sampling inaccuracy $|AB^i_{avg} - AB^i_{sys}|$ observed over all segments i within the trace, exactly as described in Section 3.1. Using an RT of 10s, we compute the above CDFs for MT of 10 ms and 100 ms, and SI of 0.1, 0.5, and 0.9. Figs 5 and 6 plot these CDFs for the Abilene-IC1 and UNC-0 traces, respectively. As expected, we observe that for a given MT, increasing the SI improves the accuracy of the sampled AB. We also observe that for a given SI, MTs that differ by even an order of magnitude have a negligible impact on the sampling accuracy. Thus, while SI impacts the measurement accuracy significantly, MT does not.

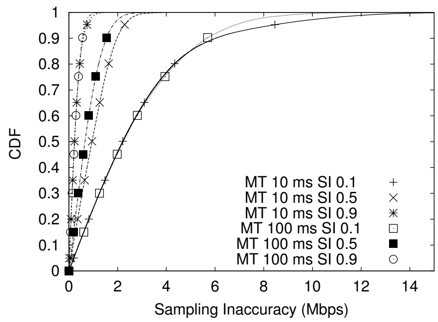

Fig. 5. UNC

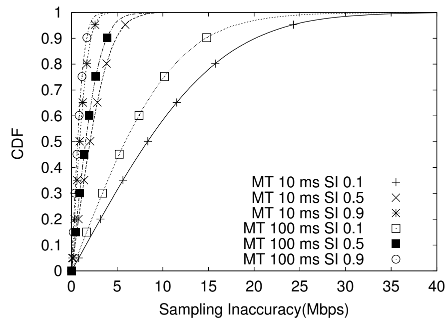

Fig. 6. Abilene

The above observations have the following implications for ABET design: (i) *The same sampling accuracy may be attained by a tool by either using a few long probe-streams, or several short probe-streams (as long as both result in the same SI).* The latter may be useful for applications that benefit from the timely-availability of an initial AB estimate, even if its only roughly accurate. The first few probe streams are likely to yield such a rough estimate quickly, while the later probes make the estimate robust. This flexibility may not be available if longer, fewer probe streams are used. (ii) *Any application-specific MT may be used for sampling, without impacting the measurement accuracy significantly, as long as an inversely proportional number of samples are collected at that timescale* (thus, maintaining the same SI).[5]

3.3 What is the marginal gain in increasing sampling intensity?

The observations made above indicate that the sampling intensity has a significant impact on the accuracy of the sampled AB; we next examine this impact quantitatively. For this, for each trace, we compute the CDF of the sampling inaccuracy, $|AB^i_{avg} - AB^i_{sys}|$ for a given choice of MT, SI, and RT, as described before. Fig 7 plots the 95% of sampling inaccuracy observed with different values of SI, with an MT of 10 ms and an RT of 10 s, for several traces. As expected, we find that increasing the SI decreases the sampling inaccuracy—however, the marginal improvement in sampling accuracy decreases with increasing SI. In particular, *an ABET is unlikely to improve its sampling accuracy significantly beyond a sampling intensity of* 30%—maintaining a low SI can help the ABET reduce the network overhead of AB estimation.

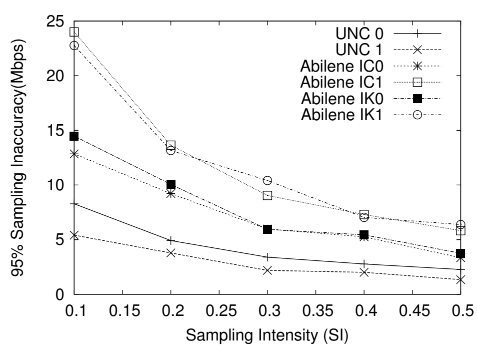

Fig. 7. Sampling Accuracy vs. SI

3.4 How does RT impact accuracy?

Finally, we evaluate the impact of the tool run-time on its sampling accuracy. For each trace, we compute CDFs of the sampling inaccuracy $|AB^i_{avg} - AB^i_{sys}|$ as described

[5] It is important to note that our analysis assumes a *perfect* probe stream and inference logic. In practice, it has been shown that due to the interaction between probe packets and cross-traffic, the use of smaller MTs results in high estimation bias [11]. However, in high-speed networks and at MTs of $10ms$ or higher, as is considered in this paper, the bias is negligible.

before. For MT = 10 ms and SI = 0.5, Fig 9 plots the 95% value from the CDFs, as a function of RT. We find that as RT increases, the sampling inaccuracy decreases. This is to be expected, as a larger RT yields a larger number of AB samples for a given SI—we find, however, that the marginal improvement in sampling accuracy reduces with increasing RT. In particular, *an ABET is unlikely to improve its sampling accuracy significantly beyond an RT of 5 s.*

RT (s)	SI	UNC-0 5 %	UNC-0 50 %	UNC-0 95 %	Abilene-IC1 5 %	Abilene-IC1 50 %	Abilene-IC1 95 %	Cesca-0 5 %	Cesca-0 50 %	Cesca-0 95 %
1	0.5	0.154	1.660	4.7	0.321	3.4	10.29	0.159	1.725	5.06
2	0.4	0.1457	1.579	4.62	0.291	3.16	9.50	0.144	1.57	4.613
10	0.2	0.155	1.66	4.91	0.25	2.714	9.215	0.133	1.467	4.18
20	0.1	0.186	1.895	8.2	0.291	3.12	9.97	0.148	1.64	4.7

Fig. 8. RT vs. Inaccuracy (Mbps)

Observe that increasing RT or SI has a positive impact on the sampling accuracy. However, increasing either of these also results in a proportional increase in the total probe traffic introduced into the network. We next ask: *does any one of these two parameters represent a better trade-off between the sampling accuracy and network overhead?* Fig 8 lists the 5%, 50%, and 95% values of the inaccuracy CDFs, computed with an MT of 10 ms, for several combinations of (RT, SI): (20s, 0.1), (10s, 0.2), (2s, 0.4), and (1s, 0.5). We find that for a given trace, the sampling inaccuracies are similar for the combinations of: (1s, 0.5) and (20s, 0.1), as well as for: (2s, 0.4) and (10s, 0.2). This observation has two implications. First, it suggests that an ABET can achieve similar AB estimation accuracy by sampling more intensely within a shorter run-time. In particular, this cautions against an unqualified claim made in [7] that a tool with a longer run-time is likely to yield more accurate AB estimates—our analysis indicates that this is true only if the sampling intensity is held constant. In reality, *it is thus possible to design a faster tool without sacrificing estimation accuracy, by simply increasing the sampling intensity of the tool*. Second, note that in order to maintain the same accuracy, the relative increase in SI is larger than the relative reduction in run-time. Thus, *a single invocation of a faster tool that achieves similar accuracy, is likely to insert more probe-traffic into the network*.

In the next two sections, we evaluate the impact of MT and RT on the variability and stability of the AB process. For the analysis in the rest of this paper, we assume a sampling intensity of 1 (the AB process is observed completely by an ABET).

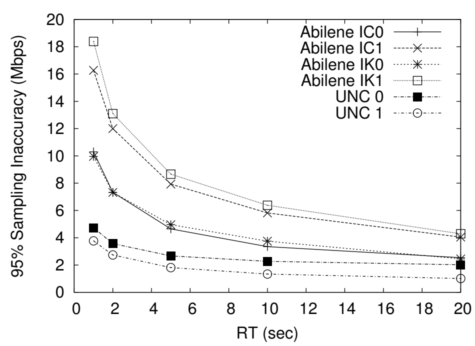

Fig. 9. Run-Time vs. Accuracy

4 How does duration of measurement affect AB variability?

Recall from Fig 1 that the AB process can exhibit low-to-high variability, depending on the timescale at which it is observed. Furthermore, the longer is the tool run-time, the greater is the opportunity to witness variability in the corresponding AB process. To quantify these effects, we next evaluate the impact of MT and RT on AB variability. Our objective is to find the set of timescales and durations that characterize an AB process with low variability.

What is a predictable measure of variability? The importance of reporting the variability in AB, in addition to its average, has been recognized recently—a new variant of Pathload reports variability in the form of the maximum and minimum AB observed during the tool's run [8]. In this section, we first address the issue of *what metric is appropriate for characterizing AB variability as a function of MT and RT?* In particular, we investigate whether for a given value of MT and RT, the *standard-deviation*—which is likely to be more robust to outliers—is a more predictable metric than the *range* metric described above.

RT (s)	MT (ms)	AB_{range} (Mbps)			AB_{std} (Mbps)		
		5 %	50 %	95 %	5 %	50 %	95 %
1	10	215.01	288.7	390.1	44.6	56.9	73.8
	50	54.4	81.3	126.6	14.4	20.7	31.3
	100	31.6	56.7	96.8	9.5	16.9	28.8
5	10	293.5	367.2	479.4	47.7	58.7	73.1
	50	88.5	117.6	176.8	17.1	22	32.6
	100	69.7	103.5	154.3	13.7	21.2	33.3
20	10	355.5	436.7	571.4	51.1	60.4	73
	50	117.5	153.4	238.8	19.1	23.6	37.4
	100	107	144.5	212.9	17.8	25.2	36.7

Table 1. Abilene: AB variability metrics

RT (s)	MT (ms)	AB_{range} (Mbps)			AB_{std} (Mbps)		
		5 %	50 %	95 %	5 %	50 %	95 %
1	10	118.6	150.9	193.3	22.3	26.5	31.1
	50	34.5	50.7	71.3	9.1	12.7	17.1
	100	17.6	29.2	45.6	5.4	8.8	13.1
5	10	160.3	188.1	231.7	23.9	26.9	30.0
	50	57.6	71.3	93.2	11.1	13.4	16.4
	100	34.6	45.7	64.5	7.5	9.7	12.8
20	10	190.5	218.8	267.3	24.8	27.1	29.5
	50	74.2	87.6	113.9	12.3	13.8	16.3
	100	48.4	59	84.8	8.7	10.2	13

Table 2. UNC: AB variability metrics

We analyze each trace as follows: (i) we compute the AB process at MT, and divide it into segments of time-length RT each,as described in Section 3.1; (ii) for each segment i, we compute the range, AB^i_{range}, and the standard deviation, AB^i_{std}, of the AB values observed in that segment; (iii) we compute the CDFs of the AB^i_{range} and AB^i_{std}, as observed over all segments within the trace. Tables 1 and 2 list the 5%, 50%, and 95% values of the AB_{range} and AB_{std} CDFs, for the Abilene-IC1 and UNC-0 traces, respectively. We observe that for any given MT and RT, the difference between the 95% and 5% values of AB_{range} is much larger than that of AB_{std}. This implies that for a given combination of MT and RT, the latter is a more predictable metric of variability.

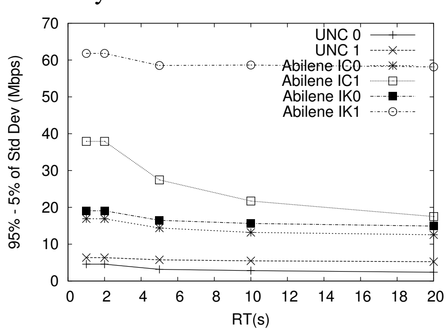

Fig. 10. AB_{std} predictability vs. RT

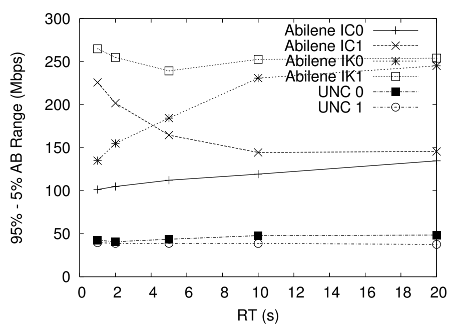

Fig. 11. AB_{range} predictability vs. RT

Furthermore, we find that the predictability of AB_{std} improves with increase in RT, whereas the predictability of AB_{range} does not. This is illustrated in Figs 10 and 11, that plot the difference between the 95% and 5% values of these two metrics respectively, as a function of RT and when MT = 10ms. We observe that the difference decreases with RT for the AB_{std} metric, but exhibits no such trend for the AB_{range} metric. This implies that the *standard-deviation is a better choice to use for characterizing AB variability*. Furthermore, *tools with longer run-times are likely to report more robust variability estimates.*

How does RT impact AB variability? From Tables 1 and 2, we also observe that for a given MT, as the RT increases, the median value (as well as other percentiles) of AB_{std} also increases. The relative increase in the variability, however, is small. This suggests that *tools with longer run-times are likely to report only slightly higher values of AB variability*.

How does MT impact AB variability? For any given RT, Tables 1 and 2 indicate that as MT increases, AB_{std} reduces. The reduction in variability is most significant at smaller timescales. For instance, at an MT of $10ms$, AB_{std} can be as high as $100Mbps$ (maximum observed value) for the Abilene-IC1 trace. At an MT of $50ms$ or higher, AB_{std} lies within $40Mbps$ for all traces (including Abilene-IC1). This latter value corresponds to less than 2% of the link capacity, which is within the resolution accuracy of all existing ABETs [15]. This implies that *in order to sample an AB process that does not exhibit significant variability, ABETs should sample it at timescales of* $50ms$ *or higher*. In particular, *the results of ABETs that rely on using* packet-pairs *instead of longer* packet-trains *are likely to be significantly impacted by AB variability*.

5 How stable is AB?

Applications that rely on ABETs necessarily use an AB estimate only *after* the measurement has been made. Thus, they implicitly assume that the AB process does not change significantly during *post-run* periods. In order to study the validity of this assumption, we next study the stability of the AB process across several successive runs of an ABET.

RT (s)	N	Abilene (Mbps)			UNC (Mbps)		
		5 %	50 %	95 %	5 %	50 %	95 %
1	1	1.3	13.1	41.2	0.3	3.7	11.9
	5	15.4	36.6	75.4	3.9	9.2	19.3
	30	37.4	63.3	128.6	9.8	16.1	33.7
5	1	0.8	11	39.5	0.3	3	10.2
	5	10.8	30.6	95.1	3.5	8	18.9
	30	37.4	81	162.1	8.9	14.5	47.3
20	1	1.0	11.6	59.1	0.2	2.6	11.9
	5	16.1	43.4	120.8	2.3	7.7	22.6
	30	56.2	111	204.4	11.1	19.2	55

Fig. 12. Stability in AB

For a given RT and N, the number of successive tool runs examined, we analyze each trace as follows: (i) we compute the AB process with MT equal to RT, and divide it into segments of time-length RT each, as described in Section 3.1; we denote the single AB reading of the i^{th} segment by AB^i_{avg}; (ii) for the i^{th} segment, we compute the *post-run deviation* metric as: $PRD^i_N = \max_{j\in[1,N+1]}\{|AB^i_{avg} - AB^{i+j}_{avg}|\}$; (iii) we compute the CDF of PRD^i_N observed over all segments within the trace.

Fig 12 lists the 5%, 50%, and 95% of the observed PRD^i_N, for RT = $\{$1s, 5s, 20s$\}$ and N = $\{1, 5, 30\}$, for the Abilene-IC1 and UNC-0 traces. As expected, we find that as time elapses (N increases) after an AB measurement is conducted, the AB process deviates more from the measured value. We also find that, in general, PRD^i_N increases with increase in RT, although not significantly.

An interesting data point is that of $N = 1$. We find that in any pair of neighboring tool-runs, the AB does not change by more than $12Mbps$ or $40Mbps$ for the UNC-0 and Abilene-IC1 traces, respectively. In fact, we find that for *all* of the traces analyzed, the AB measured in a pair of back-to-back tool runs does not differ by more than 4%, most of the time. Since the accuracy of existing ABETs is at best around 10% [15], this value lies within the resolution accuracy of existing tools. This observation is relevant for the design of ABETs for applications that need to continuously monitor the AB on a path by running an ABET repeatedly. In particular, consider the case when such applications use Pathload-like ABETs, that spent a considerable portion of their run-time in arriving at a *coarse* estimate of AB , and then work on fine-tuning that estimate. Such ABETs could exploit the fact that *AB does not change significantly between neighboring tool runs*, and use the result of the *last* tool-run as the *coarse* estimate of the current AB —this should speed-up the next tool-run, while also introducing much less probe traffic into the network.

6 Conclusions

Our analysis of several link traces yields several implications for design of ABETs.
Sampling-accuracy related: (i) A simple sampling strategy is likely to yield more sampling accuracy, although the gain over systematic and stratifies sampling lies within the resolution accuracy of current ABETs. (ii) A higher sampling intensity results in better sampling accuracy, although the gains are insignificant beyond an SI of 30%. (iii) The

choice of MT does not impact sampling accuracy significantly, as long as SI is maintained. In particular, the same sampling accuracy may be attained by a tool by either using a few long probe-streams, or several short probe-streams (both with the same SI). (iv) Tools with longer run-times are likely to achieve better sampling accuracy, although the gains are insignificant beyond a RT of 5s. Faster tools may, however, achieve similar accuracy by increasing their sampling intensities—this, however, results in the introduction of a larger amount of probe traffic into the network.
AB variability related: (v) ABETs should use the standard-deviation, as against the range, for reporting variability in the sampled AB. Also, tools with longer run-times are likely to report only slightly higher values of AB variability; however, the variability estimates are likely to be more robust. (vi) The AB process exhibits significant variability at MTs smaller than 50 ms. This corresponds to sending several packets within each probe stream (unlike ABETs that use packet-pairs).
AB stability related: (vii) The average AB does not change significantly across neighboring back-to-back tool runs. This observations can be exploited for applications that need to continuously monitor the AB on a path by running an ABET repeatedly.

References

1. Cooperative association for internet data analysis (caida) (https://oc48-data.caida.org:8447/data/).
2. National laboratory for applied networking research (nlanr) (http://pma.nlanr.net/special/).
3. R. Carter and M. Crovella. Measuring Bottleneck Link Speed in Packet-Switched Networks. Technical Report BUCS-TR-1996-006, Computer Science, Boston University, Mar 1996.
4. K.C. Claffy, G.C. Polyzos, and H. Braun. Application of sampling methodologies to network traffic characterization. In *Proceedings of ACM SIGCOMM*, pages 194–203, 1993.
5. C. Dovrolis and M. Jain. End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput. *IEEE/ACM Transactions in Networking*, Aug 2003.
6. N. Hu and P. Steenkiste. Evaluation and Characterization of Available Bandwidth Probing Techniques. *IEEE JSAC Internet and WWW Measurement, Mapping, and Modeling*, 2003.
7. M. Jain and C. Dovrolis. Ten fallacies and pitfalls on end-to-end available bandwidth estimation. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, Oct 2004.
8. M. Jain and C. Dovrolis. End-to-end estimation of the available bandwidth variation range. In *Proceedings of ACM Sigmetrics*, June 2005.
9. Guojun Jin. netest-2, 2004. http://www-didc.lbl.gov/NCS/netest.html.
10. K. Keys, D. Moore, R. Koga, E. Lagache, M. Tesch, and k Claffy. The architecture of CoralReef: an Internet traffic monitoring software suite. In *PAM*, Apr 2001.
11. X. Liu, K. Ravindran, B. Liu, and D. Loguinov. Single-hop probing asymptotics in available bandwidth estimation: A sample-path analysis, 2004.
12. J. Navratil. ABwE: A Practical Approach to Available Bandwidth. In *PAM*, 2003.
13. NLANR. Iperf v1.7.0, 2004. http://dast.nlanr.net/Projects/Iperf.
14. V. Ribeiro. pathChirp: Efficient Available Bandwidth Estimation for Network Path. In *PAM*, 2003.
15. A. Shriram, M. Murray, Y. Hyun, N. Brownlee, A. Broido, M. Fomenkov, and K.C. Claffy. Comparison of public end-to-end bandwidth estimation tools on high-speed links. In *PAM*, 2005.
16. J. Strauss, D. Katabi, and F. Kaashoek. A Measurement Study of Available Bandwidth Estimation Tools. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, Miami, Florida, Oct 2003.
17. TeraGrid. Teragrid. http://www.teragrid.org/.
18. Richard Wolski. Dynamically forecasting network performance using the network weather service. *Cluster Computing*, 1(1):119–132, 1998.