Towards a Generic Classification and Evaluation Scheme for Bandwidth Measurement Tools

Fatih Abut
Department of Communication Systems
Fraunhofer Institute for Communication
Information Processing and Ergonomics Wachtberg-Werthhoven
Germany
faith.abut@fkie.fraunhofer.de

ABSTRACT: *In this paper, as a result of our extensive analysis of several existing measurement techniques and tools, we describe the main characteristics, accuracy and performance evaluation metrics as well as robustness assessment criteria of bandwidth measurement tools and propose a generic classification and evaluation scheme consisting of three major dimensions: classification dimension, dimension of accuracy and performance assessment, and dimension of robustness assessment. To the best of our knowledge,by proposing this generic scheme we make the first attempt in order to simplify and standardize the prospective classification and evaluation of bandwidth measurement tools.*

Keywords: Bandwidth Measurement Tool Classification, Band-Width Evaluation Metrics, Robustness Criteria

Received: 17 June 2012, Revised 31 July 2012, Accepted 6 August 2012

1. Introduction

The knowledge of bandwidth in computer networks can be useful in various applications. Some popular examples where bandwidth measurements can be valuable are validation of service level agreements, video/audio stream adaptation, tcp congestion control optimization, network route selection, dynamic server selection for downloads, peer-to-peer host selection, traffic engineering and detection of congested or underutilized links.

Resulting from these motivations, a plethora of bandwidth measurement tools have been developed in recent years and still, several new tools are currently being published. Our study in this research field revealed that there are currently over 70 different tools measuring bandwidth-related metrics. Figure 1 exemplifies the evolution of bandwidth measurement tools over the last several years.

One problem resulting from this plethora of tools is that they show a wide spectrum of different assumptions and characteristics, such as the achievable accuracy, measurement time needed and probing overhead caused, intrusiveness, ability to measure asymmetric, wireless or high-speed links and to work in uncooperative environments, to name just a few. Underlying models, metric definitions as well as measurement methodologies also differ. Consequently, to choose one among these several tools which suits best to a researcher's or network administrator's needs is a difficult and costly task, since before the decision of a tool, each of its relevant characteristics and evaluation metrics should be worked out.

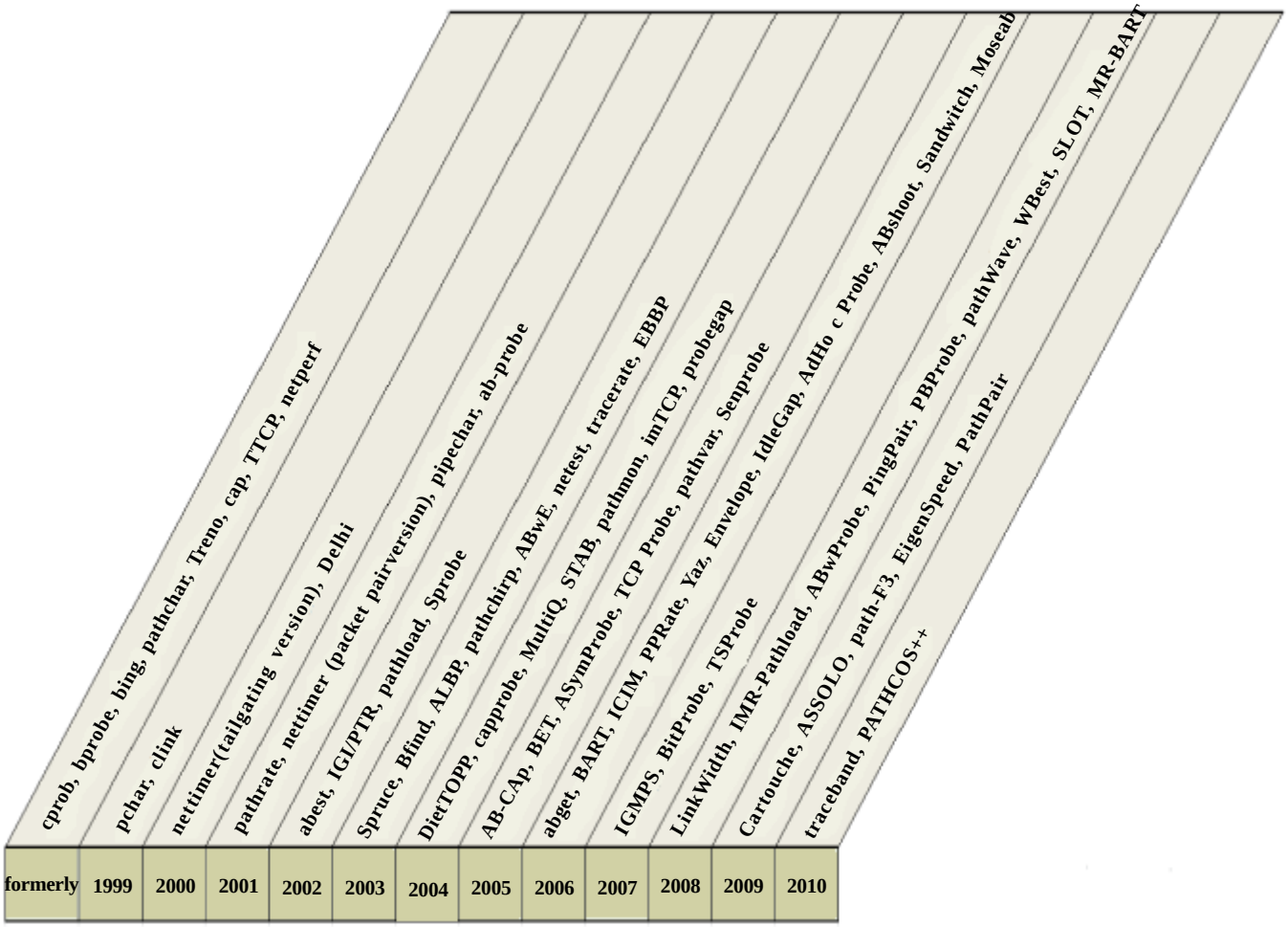

Figure 1. Evolution of bandwidth estimation tools (Each tool is assigned to its publication date)

Furthermore, as can be seen in the figure 1, starting from 1999, several tools are published annually and one can easily anticipate that this increasing trend will also continue for the next years. In order to assist prospective new tool developers (who are potentially unaware of several existing classification and evaluation criteria), a generic classification and evaluation scheme is needed in order to simplify and standardize the classification and evaluation of bandwidth measurement tools.

In this paper, as a result of our extensive analysis of several existing measurement techniques and tools, we de- scribe the main characteristics, accuracy and performance evaluation metrics as well as robustness criteria of bandwidth measurement tools and propose a generic classification and evaluation scheme consisting of three major dimensions:

• First, we introduce the **classification dimension** describing general and static characteristics of bandwidth measurement tools in order to fully classify a tool into this extensive research field. (e.g. metric to be measured, measurement methodology used and active vs. passive measurement)

• then, we continue with the **dimension of accuracy and performance assessment** describing the dynamic assessment criteria of the measurement tools which strongly depends on the testbed set-up, scenario and configuration used (e.g. measurement result accuracy and consistency, total measurement time required and the amount of measurement traffic generated)

• and finally, we discuss the **dimension of robustness assessment** containing the criteria about how the tools perform in various network environments (e.g. on wire- less, high speed or asymmetric links) under different conditions (e.g. in presence of cross traffic, route alter- nation or multichannel links).

We believe our scheme to be utilizable both for current and future use in different purposes. Fur current use, it allows new tool developers to easily classify their newly-developed tools into this extensive research area considering all essential classification and evaluation criteria. Moreover, it can serve as objective and extensive comparison scheme for further comparative analysis of different probing tools already implemented. As a future use, provided that our scheme will be actively used by several tool developers, it will shed light on several relevant tool parameters und thus support researchers and network administrators to make a decision about the more appropriated tool for their needs.

The rest of the paper is organized as follows. In section II, we discuss the classification dimension of our proposed scheme describing the main characteristics of bandwidth measurement tools. In section III, we continue with the description of the dimension of the accuracy and performance assessment. In section IV, we describe practical issues and difficulties that exist in different network environments impacting the accuracy and robustness of the bandwidth measurement tools and derive a robustness classification and assessment scheme. Finally, we conclude with an outlook in section V.

2. Classification Dimension

Existing bandwidth measurement tools mainly measure one of four related metrics: capacity, available bandwidth, achievable tcp/udp throughput and bulk transfer capacity (BTC). For definitions of these metrics we refer the interested readers to the numerous respective publications [1] [2].

The measurement of each metric is associated at least with one measurement technique. Representative examples of measurement techniques, among others, used by different measurement tools ranges from Packet Pair and Pathchar measuring the capacity to Probe Rate/Gap Model (PRM/ PGM) measuring available bandwidth to (parallel) tcp con- nections/emulations used for measuring achievable throughput and bulk transfer capacity (btc).

A metric can be measured on the entire path between two end-hosts (i.e. at the end-to-end scope), on a subpath consisting of a number of consecutive links of an end-to-end path, or hop-by-hop. Tools measuring end-to-end metrics are able to observe bottlenecks visible at the end-to-end-scope (e.g. CapProbe [3] and iperf [4]). Tools measuring subpath-specific metrics can estimate characteristics on links not visible at end-to-end scope and allow identifying a portion of the end-to-end path containing the bottleneck link (e.g. cartouche [5]).

Finally, hop-by-hop measurement tools allow measuring a metric for each hop along the path (e.g. pathchar [6], pchar [7] and clink [8]).

A tool can measure the metric actively by injecting additional measurement data into the measurement path or passively by monitoring the real traffic at an appropriate observation point without perturbing the network traffic (e.g. nettimer [9]). Some passive tools can also infer link charac-teristics from the analysis of a packet trace of TCP connections collected earlier (e.g. PBProbe [10] and MultiQ [11]).

Both active and passive tools have their advantages and disadvantages. The most significant advantage of passive tools is that they do not cause additional traffic and thus don't perturb the existing traffic on the path. Thus, passive tools are quite convenient for large-scale studies of Internet path characteristics. However, they have no control over the traffic pattern and duration. Consequently, existing traffic might not be suitable for the tool to produce an accurate estimate. Unlike a passive algorithm, an active tool can control the size and transmission times of its packets and produce a suitable probing stream leading to more accurate results, but this is done at the expense of the additionally caused probe traffic overhead.

One and the same metric can be measured at different layers of the TCP/IP model. A layer-2 link can normally transfer data at a constant bit rate, which is also called the nominal bandwidth of that link. However, from the sender's point of view, this nominal bandwidth cannot be completely used for the raw data transmission, since each layer in the TCP/IP model adds its own header to the data received from the upper layer. Thus, this overhead caused by adding layer-specific header information has a reducing effect on the nominal bandwidth obtainable at layer 2. Usually, active tools perform the measurement at IP layer since at layer 2, the exact amount of overhead of protocols such as ATM, PPPoE or PPPoA that carry the higher layer packets is unknown. Contrary, passive tools can take the measurement at layer 2 by simply capturing the incoming traffic. For example, the passive version of nettimer attempts to measure the capacity metric at the link layer (i.e. in case of 10BaseT Ethernet, reference value is 10 Mb/s) whereas pathrate[16] measures it at IP layer (reference value is 9.75 Mb/s [1]).

Some tools assume that the links are symmetric along the measurement path. However, recent deployment of ADSL lines, cable

Classification Table of Bandwidth Measurement Tools	
Name of the Tool	
Classification Categories and Criteria	**Possible Inputs**
Basic Classification Criteria	
Metric(s) to measure	Available bandwidth, capacity, achievable tcp/udp throughput, btc ...
End-to-end, subpath or hop-by-hop metric	End-to-end, subpath, hop-by-hop
Measurement technique(s)	Packet Pair, Pathchar, PRM, PGM, ...
Active / Passive	Active / Passive
Works in uncooperative environments	Yes / No
Works on asymmetric links	Yes / No
Target network environment the tool is designed for	High-Speed Links, Wireless Adhoc/ Infrastruce/Sensor Networks, ADSL ...
Classification Criteria related to Tool Implementation	
Author(s) of the tool	Author(s) of the tool
Version of tool developed	Version of tool developed
Publication title, date and download link of the tool	Footnote reference
Implemented in simulation or in a tool	Tool / Simulation
Protocol used	ICMP / UDP / TCP ...
Platform needed	Linux, Windows, NS-Nam, Qualnet ...
Privileges required to run the tool	User / Root
IPv6 support	Yes / No
Type of licence	Open source / Freeware / Commercial
Classification of the Tool's Measurement Results	
Type of reported result	Single point / Convergence range
Layer at which the metric is measured	Link layer / IP layer / Transport layer
Additional metrics reported	Latency, queuing delay, rtt, ...

Table 1. Classification table of bandwidth measurement tools

modems and satellite links is becoming more and more popular and rapidly changing this assumption. Being able to detect and measure asymmetrical links is a very desirable capability of a tool.

Tools can be classified as single-end or both-end tools. We classify a tool as a single-end tool, if it runs only at one host on the

entire path or a both-end-tool if it requires access to both ends. Single-end tools have the merit that the measurement software is only deployed locally on the measurement host and thus increasing the applicability of the tool significantly. Examples for single-end tools are SProbe[12], abget[13] and bprobe[14]. Single-end tools are very flexible since they enable measurements of paths from the source to any arbitrary destination. Furthermore, single-end tools don't need any synchronization between the sender and receiver clocks.

Measurement tools able to work in uncooperative environ- ments are all based on the same principle. They send special packets eliciting acknowledgements or responses from the receiver side. Consequently, the measurement packets traverse the path twice, both in the forward and backward direction. However, this principle also entails the additional problem that cross traffic can affect the measurement both in the forward and reverse path. This is one of the main reasons why the single-end tools are usually less accurate than the both-end tools.

Both-end tools require the cooperation of both the source and the destination so as to their applicability is limited in just a few paths where the user has access at both the sender and the receiver. However, measuring in cooperative environments avoids the queuing in reverse paths leading to more accurate results. It should be noted that every both-end tool is able to measure asymmetric links by exchanging the sender and receiver components of that tool and applying the same measurement technique to the reverse path.

The protocol used by a tool is an another important aspect since they can also limit the applicability of the tools. ICMP packets, e.g., are often blocked by firewalls, rate-limited or handled differently than normal network traffic (e.g. due to fast path / slow path processing modes in the routers[24]). Furthermore, the choose of the protocol determines whether a tool needs the administrative rights required to run the tool, e.g. ICMP on Linux/Unix usually require root privileges, whereas UDP and TCP protocols also work with normal user privileges.

They are two kinds of how the tools report their measurement results. The usual way is reporting the final estimation in form of a single point. Probing results of some tools, however, converges to an estimation range, either because they only can approximate the metric (e.g. pathload[15]) or they only intend to report a lower and upper bound (e.g. pathrate).

Some tools are specifically designed to work in particular target network environments. This is because each network environment poses several different challenges to measurement tools that should be taken into consideration. For example, in wireless (adhoc/ infrastructure/sensor) networks, a tool must consider the rapidly varying channel and network conditions and link rate adaption techniques, whereas tools designed to work on high-speed links should cope with high- speed link related problems such as interrupt coalescence and limited system timer resolution (more details in section IV). Similarly, tools like DSLProbe[17], addresses the challenges of asymmetric links making it possible to measure the bandwidth in both directions of the path. Further examples for target network environments with different challenges are measurements on peer-to-peer paths (e.g. EigenSpeed[18]), wireless sensor networks (e.g. SenProbe[19]) or MPLS net- works (e.g. MABE[20]). Thus, we define the target network environment on which a tool is designed to work as a further classification criterion.

Although most measurement techniques are currently implemented in a tool and can be used on real Internet/network paths, they are also realizations of them which have been initially implemented only in simulation. Implementation of the basic measurement methodology in simulation, as a first stage, have the main merit, that it allows to avoid practical issues and difficulties such as route changes, multichannel links, timestamping inaccuracies and low system timer resolutions that can distort the measurement results. It enables to test the basic methodology in a controlled and reproducible manner. Thus, we further differentiate whether a measurement technique is implemented in a tool or only in simulation.

Further basic criteria used in our classification scheme are the name of the tool developer, the version of the tool developed, download link or homepage of the tool, publication title and date of the corresponding tool paper (if available), the license type of the tool (open source / freeware / commercial), ipv6 support, additional results except bandwidth-related metrics the tool also reports, operating system and privileges required to run the tool. Table 1 shows our proposed classification scheme along with its possible inputs.

3. Dimension of Accuracy and Performance Assessment

The measurement accuracy is one of the most significant assessment criteria of a tool. It describes how close the achieved measurement result comes to the actual (true) value. To determine the accuracy of a tool, someone typically repeats the measurement process with

the same tool several times to compute the average measured value. Provided that each measurement is performed under the same conditions (including the same cross traffic load), the average measured value will reflect the measurement accuracy of that tool.

The measurement accuracy of a tool will be evaluated in terms of its measurement error. Let be x^m the measurement result achievedby a tool and *ref_value* the actual value of the metric to measure. Then, the measurement error e_m of a tool can be calculated as

$$e_m = |(x^m - ref\_value) / ref\_value| \times 100\% \quad (1)$$

According to (1), smaller measurement error leads to a more accurate result.

Assuming that real Internet paths almost always have cross traffic from other sources, possibly with different load levels and types, results obtained from repeated measure- ments on such paths will be found to vary. This follows from the fact that most measurement techniques including Packet Pair, Pathchar and Probe Gap/Rate Model are all based on delay measurements which can easily be distorted by the presence of cross traffic. Consequently, the results will be not the same. However, it is expected that they will be similar, with variations within the acceptable limits. To express the grade of those variations of a tool's measurement results, we introduce a new metric, called consistency, which indicates the closeness of agreement between results of successive repeated measurements on the same path. Full consistency in measurement results can only be achieved when they are no any other biases during the measure- ment process or the measurement technique is capable to fully eliminate them. As proposed in [21], the consistency of a tool's measurement results c_m can be calculated as

$$c_m = \left[1 - \sum_{i=1}^{n} |x^i - \bar{x}| \Big/ (n \times \bar{x})\right] \times 100\% \quad (2)$$

Accuracy and Performance Assessment Table of Bandwidth Measurement Tools									
Evaluation Criteria	**Name of the Tool**								**Input Units**
Measurement error									%
Measurement consistency									%
Total measurement time required									sec. or min.
Amount of probe traffic generated									KB or MB
Intrusiveness									KB or MB per sec.
	0%	10%	20%	...	90%	<avg. pkts/sec.>	<shape> <scale>	...	
	Amount of constant CT generated in %					Poisson-like CT	Pareto-like CT	...	
	Type of cross traffic (CT) with its type-specific parameter(s)								

Table 2. Accuracy and performance assessment table of bandwidth measurement tools

where n is the number of measurements performed, x^i is the result of the i -th measurement with $1 \le i \le n$, and $\bar{x}$ is the average of all n measurement results. According to (2), bigger c_m means the repeated measurements of a certain tool basically yield the same result. In summary, a measurement tool can be seen as reliable, if its measurement results have both little measurement error and large consistency.

Ideally, the estimation time required to complete the mea- surement process should be very short, as metrics like available bandwidth and throughput (and even capacity in case of route alternation) can vary over time during the measurement process. Fast estimation would also allow further applications the "*online*" usage of the measurement tool to estimate the metric required. Moreover, since the rate of a wireless link can vary dynamically and rapidly due to changes in interference or distance, timely knowledge of bandwidth in wireless networks is of critical importance.

For large-scale deployment and use of active tools and for having less effect on the network, it is important that they generate low

amount of probe traffic. We quantify the probe traffic overhead as the amount of probe data totally sent, expressed in KB or MB /s.

Another essential property derived from the two previous parameters is the intrusiveness of an active tool. A tool is called intrusive if it generates significant traffic load such that it causes significant delay and losses in the cross traffic packets by overfilling the queue of the bottleneck link of the path, otherwise the tool is called non-intrusive. Ideally, tools should be non-intrusive so that they do not disturb the ongoing applications traffic in the network. However, because all active measurement tools are based on injecting probing traffic into the measurement path, they all are intrusive to some degree. Although there is no fixed definition of how to quantify intrusiveness, we propose to measure it as the average amount of probing traffic injected into the measurement path per second, expressed in KB or MB /s.

Consider that both the estimation time required and amount of traffic generated by an active tool can depend on various factors like the number of existing hops along a path, the round trip time a probe packet requires to traverse the path, cross traffic load/type available on the measurement path, the link/path speed to which a tool attempts to converge and/or tool-specific adjustable options like packet size or customized number of measurements. Moreover, note that existing cross traffic on a measurement path not only affects the estimation time and traffic overhead of a tool but also its accuracy and consistency. Thus,

Robustness Classification & Assessment Table of Bandwidth Measurement Tools	
Name of the Tool	
Robustness Criteria	**Class**
Robustness to cross traffic of different types	
Robustness to asymmetric links	
Robustness to multi-path-diversity / route alternation	
Robustness to multi-channel links	
Robustness to non-FIFO queues	
Robustness to traffic shaping nodes	
Robustness to context switch	
Robustness to clock skew problem	
Robustness to multiple bottlenecks	
Robustness to interrupt coalescence mode of the NIC's used	
Robustness to OS's limited system timer resolution	
Robustness to limited system I/O throughput of the measurement end-hosts	

Table 3. Robustness classification and assessment table of bandwidth measurement tools. For each criterion, the tool to be evaluated should be assigned in one of the following classes: robust, robust-aware or not robust. Tools classified as robust may be quantified/assessed more precisely by tool developers.

for each tool, all accuracy and performance metrics listed in table 2 should be evaluated in terms of cross traffic type and load under different conditions.

Finally, we would notice that due to dynamic nature of the assessment criteria of a tool, the assessment of all those criteria is only valid in the specific testbed set-up, scenario and configuration used, so they can't be generalized.

4. Dimension of Robustness Assessment

Ideally, bandwidth measurement tools should work robust in the variety of network environments under different conditions: few or many hops from source to destination, empty, moderately or highly congested links, one or several channels per link, wired, wireless, high-speed or asymmetric links and different queuing disciplines. In this section, we describe the current practical issues and difficulties in the field of bandwidth measurement that should be taken into consideration in the design and implementation phase of a tool and then propose a tool robustness classification and assessment scheme.

One of the most relevant robustness criteria of a measurement tool is its resistance to cross traffic since Internet paths almost always contain cross traffic. To enhance the robustness of tools to cross traffic, several techniques have been proposed including confidence intervals, kernel density estimator functions and lower/upper bound filtering techniques [9]. Unfortunately, there is no standard statistical approach that always leads to correct estimation. The main reason making the deal with the cross traffic difficult is that there exist several types of cross traffic (e.g. deterministic cross traffic with different loads or cross traffic obeying to a particular distribution like poisson or pareto distribution) causing different type-specific measurement errors.

In packet-switched networks, data packets belonging together can reach their destination over different paths. This could be caused, e.g., due to dynamic route alternation or load sharing. Route alternation is the property of a path between two hosts to change over time, usually between a small set of possibilities (e.g. in case of node failures or load balancing). Note that in case of route alternation, there is only one possible route that a router can take at a given time. Contrary to route alternation, load sharing can route the packets over two or more different interfaces at the same time. Assume that on a measurement path, during the probing process such a route alternation or load sharing occurs. Then, the probe traffic will be transmitted over different links/paths which potentially will suggest different link/path characteristics (including different bandwidth speeds) causing significant measurement inaccuracies. Therefore, a robust tool should characterize the measurement path in order to be flexible to bandwidth and route changes.

Along a path, a link can be multi-channeled which means that it is made up of a number of parallel channels. If a link of total capacity C is made up of k channels, the individual channels forward packets in parallel at a rate of C/k. In such a case, a tool may incorrectly tend to measure the bandwidth of a single channel, instead of the total bandwidth of that link.

In Internet, traffic shapers are often employed to control the volume/rate of the networking traffic in order to guarantee some QoS parameter like latency, bandwidth and avoid bursty traffic. In such a scenario, the measurement process and result of a tool may be affected if its probing rate is higher than the rate the traffic shaper allows. Moreover, in case of capacity measurement, the link on which traffic shaping will be performed will have two different capacity metrics, namely the unlimited raw capacity and the sustainable rate of the traffic shaper. Thus, if a tool's measurement methodology cannot overcome traffic shaping limitations, it should at least clearly define, which capacity metric it actually intends to measure for paths with traffic shaping nodes.

Several measurement methodologies and tools have the fundamental assumption that the bottleneck router uses FIFO-queuing, i.e. what comes in first is handled first. Other Non-FIFO queue processing techniques like Token Bucket Filter or prioritized queuing could distort the measurement process and thus should be detected.

Interrupt coalescing (IC) is a well known and proven technique for reducing CPU utilization when processing high packet arrival rates. Normally, a network interface card (NIC) without IC generates an interrupt for each incoming packet. This causes significant cpu load when packet arrival rate increases. By using IC, the workload for the host processor can be reduced significantly by grouping multiple packets, received in a short time interval, in a single interrupt. In this way, the number of interrupts to be generated will be reduced significantly. However, lower CPU utilization is done at the cost of increased network latency, since the frames are first buffered at the NIC before they are processed by the operating system (the host is not aware of the packet until the NIC generates an interrupt). Thus, the receiving timestamps for the packets will be distorted (in such a case, all incoming packets may have the same timestamp) which may lead to erroneous measurements.

Most measurement tools are based on sending probing packets at a certain transmission rate, i.e. they must send packets in regular intervals in order to perform a proper measurement (e.g. pathload, PBProbe, and abget). Consider that a tool needs to send packets at a transmission rate R with

$$R = \frac{packet\_size\ (s)}{time\ (t)} \tag{3}$$

i.e., every t time units, a packet of size s should be sent. Two different approaches can be taken in order to achieve the rate R. Firstly, a tool could perform busy waiting by continuously checking the system clock and send packets of size s every time when the clock reaches the corresponding value of t. The maximum rate R obtainable with this approach depends on the time which is required to perform the clock checking process. Though the busy waiting mechanism allows to achieve high transmission rates, it wastes a lot of CPU cycles affecting the efficient processing of tasks from other applications, especially if the measurement process lasts for several seconds (which is typically the case for the most tools).

In the second approach, a measurement tool associates its action of sending probing packets with a system timer mechanism which is a recurring timeout process in an OS. Every time when this timer expires and a timeout occurs, the tool fires its probing packets. Consequently, creating a timeout event which sends packets of size s with timeout value as t allows to achieve the rate R. This approach avoids the problem of busy waiting, since the CPU is merely stressed if and only if a probing packet should be sent. Due to this significant advantage, almost all tools use this timing mechanism. Unfortunately, contrary to busy waiting mechanism, the maximum transmission rate obtainable using this approach is strongly limited by the insufficient system timer resolution. The minimum system timer resolution among the common operating systems is 1 µs. By considering the biggest size of 1500 Byte in classic Ethernet networks, only a transmission rate of $R = 1, 2$ Mbps can be achieved. To overcome this problem, the technique of packet trains is proposed. For further information, we refer the reader to the respective publications [22]. Note that a clock's resolution also affects the preciseness of the packet timestamps. The higher the clock's resolution is, the more accurate the timestamp on the packets.

An another challenge arises when a tool's measurement process gets interrupted by a context switch at the end hosts. Assume that a probing stream with a specified rate will be transmitted by an end host, i.e. the probe packets are sent out periodically every time unit. If during this transmission process a context switch occurs, the specified rate can't be sustained any more reducing the measurement accuracy. Thus, to avoid this problem, the transmission period of a tool's probing stream should be as short as possible and complete before a context switch interruption occurs.

Much measurement tools fail to accurately estimate high- speed network bandwidth since they do not take the capabilities of the measurement host system into account (e.g. host's memory, I/O bus speed etc.). If end system capabilities are involved, then the measurement will be of the end system throughput and will not indicate a correct assessment of network bandwidth. Thus, either the bandwidth measurement algorithm should not be dependent on end host performance or the tool should implement additional methods to determine if the end hosts are capable of performing a proper measurement [22].

Several both-end tools rely on the assumption of a synchronized clock between the endpoints. However, the clocks on different machines are usually not synchronized and the offset between two different clocks usually changes over time. To ensure reasonable measurement accuracy, a tool should be robust to this clock skew problem.

The measurement technique of a robust tool should remain valid in the presence of multiple bottleneck links on a path. For example, probe gap model based tools like spruce [23] assume that there is only a single bottleneck link on the measurement path. On the other side, they are also tools such as MultiQ which are not only robust to multiple bottlenecks, but also able to measure the bandwidth of multiple congested bottlenecks.

One important question resulting from this section discussion is how to classify/assess the robustness of a tool. A possible classification of the robustness of a tool to a particular issue (listed in table 3) could be coarsely assigned into one of the three groups: robust, robust-aware or not robust tools. We call a tool robust if it takes such an issue into consideration in its design and implementation and attempts to overcome it in order to give a reasonable accuracy (e.g. pathrate's robustness to interrupt coalescence issue using packet trains [16]). This type of robustness classification may be quantified/assessed by the tool's developer more precisely. We call a tool robust-aware if it is only able to detect such an inconvenience and aborts the measurement process instead of reporting a potentially inaccurate estimate (e.g. SProbe's robust-awareness to cross traffic by detecting its presence and aborting the measurement process [12]). Finally, we call a tool not robust if it does not consider / not

able to detect such an issue but still continues its measurement process giving potential inaccurate results (e.g. spruce's lack to detect and cope with multiple bottleneck links [23]).

5. Conclusion and Future Work

In this paper, we describe the most relevant characteristics, accuracy and performance metrics, and robustness assessment criteria of bandwidth measurement tools and propose a generic classification and evaluation scheme consisting of three major dimensions: classification dimension, dimension of accuracy and performance assessment and dimension of robustness assessment. To the best of our knowledge, by proposing this generic scheme we make the first attempt in order to simplify and standardize the prospective classification and evaluation of bandwidth measurement tools. For each of the three dimensions, we describe the major characteristics as well as accuracy and robustness assessment criteria. We believe our proposed scheme to be utilizable in different purposes such as allowing new tool developers to easily classify their newly-developed tools into this extensive research area considering all essential classification and evaluation criteria or serving as objective and extensive comparison scheme for further comparative analysis of different probing tools already implemented. Moreover, provided that our scheme will be actively used by several tool developers, they will significantly simplify the search and survey for suitable tools and assist researchers and network administrators in making a decision about the more appropriated tool for their needs.

As a future work, we intend to experimentally evaluate all the tools shown in figure 1. Our next goal will be to find the most reliable and robust tools applicable on real paths under realistic conditions and publish them in a further work using our proposed classification and evaluation scheme.

References

[1] Murray, M., Prasad, R. S., Dovrolis, C., Claffy, K. (2003). Bandwidth estimation: metrics, measurement techniques, and tools in IEEE Network, p. 27-35.

[2] Guojun Jin. Algorithms and Requirements for Measuring Network Bandwidth, technical report LBNL-48330, 2003.

[3] Kapoor, P., Chen, L., Lao, L., Gerla, M., Sanadidi, M. (2004). CapProbe: A Simple and Accurate Capacity Estimation Technique, *In*: Proc. of ACM SIGCOMM.

[4] Tirumala, A., Cottrell, L., Dunigan, T. Measuring End-to-end Bandwidth with Iperf using Web100, *In*: Proc. of PAM, 2003.

[5] Harfoush, K., Kestavros, A., Byers, J. W. (2009). Measuring Capacity Bandwidth of Targeted Path Segments, *In*: Proc. of IEEE/ACM Transactions on Networking, Feb, p. 80-92.

[6] Downey, A. B. (1999). Using Pathchar to Estimate Internet Link Characteristics, *In*: Proc. of ACM SIGCOMM, Sept. p. 222-223.

[7] Mah, B. A. (1999). pchar, http://www.caida.org/tools/utilities/others/-pathchar/.

[8] Downey, A. B. (1999). Clink, http://rocky.wellesley.edu/downey/clink/.

[9] Lai, K., Baker, M. Nettimer. (2001). A Tool for Measuring Bottleneck Link Bandwidth, *In*: Proc. of USITS, p.123-134.

[10] Chen, L., Sun, T., Wang, B., Sanadidi, M. Y., Gerla, M. (2008). PBProbe: A Capacity Estimation Tool for High Speed Networks, Computer Communications, 31 (17) 3883-3893, Nov.

[11] Katti, S., Katabi, D., Blake, C., Kohler, E., Strauss, J. (2004). MultiQ: automated detection of multiple bottleneck capacities along a path, in Proc. of IMC, p. 245-250.

[12] Saroiu, S., Gummadi, P. K., Gribble, S. D. (2002). Sprobe: A fast technique for measuring bottleneck bandwidth in uncooperative environments, *In*: Proc. of IEEE INFOCOM, New York, NY, USA, Jun.

[13] Antoniades, D., Athanatos, M., Papadogiannakis, A., Markatos, E., Keromytis, A. D. (2006). Available bandwidth measurement as simple as running wget, *In*: Proc. of Passive and Active Measurement Conference (PAM), Adelaide, Australia, Mar.

[14] Carter, R., Crovella, M. (1996). Measuring Bottleneck Link Speed in Packet-Switched Networks, *In*: Proc. of ACM PERFORMANCE, Oct.

[15] Jain, M., Dovrolis, C. (2002). Pathload: A measurement tool for end-to-end available bandwidth, *In*: Proc. of Passive and Active Measurements (PAM) Workshop, p. 14 - 25.

[16] Dovrolis, C., Ramanathan, P., Moore, D. (2001). What do Packet Dispersion Techniques Measure?, *In*: Proc. of IEEE Infocom, Apr. p. 905-914.

[17] Croce, D., En-Najjary, T., Urvoy-Keller, G., Biersack, E. (2008). Capacity estimation of ADSL links, *In*: Proc. of ACM CoNEXT, Madrid, Spain.

[18] Snader, R., Borisov, N. (2009). EigenSpeed: Secure Peer-to-Peer Bandwidth Evaluation, *In*: Proc. of Eighth Int'l Workshop Peer-To-Peer Systems (IPTPS '09), Apr.

[19] Sun, T., Chen, L. -J., Yang, G., Sanadidi, M. Y., Gerla, M. (2005). SenProbe: Path Capacity Estimation in Wireless Sensor Networks, the Third International Workshop on Measurement, Modeling, and Performance Analysis of Wireless Sensor Networks, San Diego, USA.

[20] Anjali, T., Scoglio, C., Chen, L., Akyildiz, I., Smith, J., Sciuto, A. (2002). MABE: A New Method for Available Bandwidth Estimation in an MPLS Network, *In*: Proc. of IEEE Networks, Atlanta, USA, Aug.

[21] Li, W., B. Zeng, B., Zhang, D., Yang, I. (2008). Performance Evaluation of End-to-End Path Capacity Measurement Tools in a Controlled Environment, *In*: Proc. of GP Cp. 222-231.

[22] Jin, G., Tierney, B. (2003). System Capability Effects on Algorithms for Network Bandwidth Measurement, *In*: Proc. of the Internet Measurement Conference, Miami, Florida, Oct.

[23] Strauss, J., Katabi, D., Kaashoek, F. (2003). A measurement study of available bandwidth estimation tools, *In*: Proc. of ACM IMC, Oct.

[24] Govindan, R., Paxson, V. (2002). Estimating router ICMP generation Delays, *In*: Proc. of Passive and Active Measurements (PAM), Mar.