Optimization of Probe Train Size for Available Bandwidth Estimation in High-speed Networks

Veronika Kirova[1], *Eduard* Siemens[1], *Dmitry* Kachan[1], *Oksana* Vasylenko[2] and *Kirill* Karpov[1]

[1]Anhalt University of Applied Sciences, Department of Electrical, Mechanical and Industrial Engineering, Bernburgerstr. 55, Köthen, Germany
[2]O. S. Popov Odessa National Academy of Telecommunications, Department of Higher Mathematics, Kovalska Str. 1, Odessa, Ukraine

Abstract. Available bandwidth parameter is a crucial characteristic in terms of networking and data transmission. The beforehand knowledge of its value and use of this parameter in various traffic engineering algorithms and QoS calculations is a key for high-efficient multigigabit data transport in nowadays networks. The challenge in available bandwidth estimations is not only in its accuracy and processing speed but also in the reduction of the amount of probe traffic injected into the network by keeping an adequate level of estimation accuracy. In this paper we extend existing active probing measurement algorithms for end-to-end available bandwidth estimation along with methods to reduce estimation times and amount of injected traffic while keeping measurement accuracy constant and even reducing the uncertainty of estimations. The main goal of this research was to detect a sufficient ratio of MTU, packet train size with the link capacity and available bandwidth (AvB) in up to 10 Gbps networks. In order to explore measurement accuracy under different conditions, a new tool for the AvB estimation named Kite2 has been developed and is presented in the paper. Comparative performance of AvB estimations using Kite2, Kite and Yaz is presented. Finally we calculate with statistical means dependency between the estimation error probability, measurement probing overhead and the measurement time.

1 Introduction

Accuracy of available bandwidth (AvB) estimation heavily relies on the method of packet probing and the mathematical assumptions concluded from the delay distribution of the probe packets [1].

Available bandwidth of the end-to-end path defines the instantly unused capacity of the path in a contrary to the overall capacity, which is defined by the physical parameters of the link, so AvB depends on the instant traffic load and has almost unpredictable character.

Since the traffic load in the link has a burst character, the task of defining spare capacity in a certain time is challenging, because a process of measurement is affected by continuous intervention by a diversity of cross traffic, context switching of transmission/reception processes, small packet buffers, interrupt coalescence mechanisms, etc. [2].

The knowledge of AvB is especially important for contemporary network protocols and applications, as it enables them to adapt the transmission data rate tightly to the abrupt changes in the communication channel, which are typical, for example, in SDN networks. Also fast and highly accurate AvB estimations lay the basis for efficient operations of SDNs with regard to re-routing and switching of high-priority paths through high-speed networks. Furthermore, non-intrusive AvB measurements are used for an embedding of machine learning and artificial intelligence algorithms into applications for data exchange. Hereby the precise estimation of instantly available bandwidth serves as main input parameter for machine learning algorithms of transport protocols.

It is worth noting that active measurement methodologies rely on the emission of probe traffic in the network and afterwards processing of the impact of intermediate network devices on the injected traffic: loss rate, delays, jitters, packet reordering, etc. [3]. Consequently, as active measurement tools raise traffic load in the network, one of its significant flaws is related to the intrusion caused by the measurement traffic which can result in a QoS deterioration, and thus provide biased available bandwidth measuring results. Also, the massive network occupation by active probes intermediate devices can falsely identify as DoS attacks or flooding. It can lead to the probe traffic rejection and cause the measurement process failure. The level of disturbance by active probes is then one of the key features which must be taken into account by the active probing tool design.

One of the approaches to reduce the bias caused by the active probing is based on the minimization of the samples number in its sending train sizes. On the other hand, if an active measurement tool operates with a small probing amount, it provides shorter execution times, which are significant for the highly dynamic traffic of the most networks. However, it also increases the risk to underestimate the available resources.

In this paper we evaluate the accuracy of active measurement algorithms of AvB estimations and optimize the packet sizes in probe trains by finding a trade-off between the amount data injected by active probing, estimation times and suitable accuracy of AvB estimation. In order to investigate measurement performance under the different probing intensity, a new tool called **Kite2** is proposed here. It extends the capabilities of **Kite** which was used for AvB estimations before [4]. With the presented improvements, Kite2 achieves faster convergence of AvB measurements and a significantly lower uncertainty of the estimations than Kite. The algorithms used in Kite2 along with its performance results are also described in this paper. Finally the limitations of measurements with Kite2, Kite and the older tool **Yaz** [5] are shown in section 4.2.

2 Background and related work

Several software tools for the available bandwidth estimation were developed in recent years which are using different methods and models of estimation network metrics, such as *pathload* [6] and *pathChirp* [7] which are based on evaluation of one-way-delay properties changes while sending periodic streams of packets; algorithm IGI [8], which detects proportional correlation of initial packet gaps changes and cross traffic in the path, thus detect a tight link; already mentioned Yaz which makes calculations based on difference of interpacket intervals on receiver and sender side, *Abing* [9] which evaluates packet pair dispersion; *ASSOLO* [10] which is based on enhanced algorithm of *pathChirp* and increase the accuracy of measurement by increasing of the probing stream density; *ttcp* [11] which measures achievable TCP throughput; *pathrate* [12] which is based on packet pairs dispersion method; *Netperf* that uses large TCP transfer [9]; and others which were described in papers like [13].

A summary of existing tools and comparison of their performance is given in a few papers like [14] and [15]. It is important to distinguish between different network metrics which are measured by these tools: link capacity, available bandwidth or achievable TCP throughput as each of these parameters define different characteristics.

Despite the fact of such amount of related investigations on AvB estimations, authors of this paper found only a few research related to applicability of these tools in the high-speed networking of 10 Gbps. A proposed methodology can be found in [4] and a pilot investigation in real networks in [16].

3 Testbed topology

The results of the paper have been collected from experiments performed on a testbed described in *Figure 1*.

The laboratory environment of 10Gbps network consists of equipment:

- 2 instances of Netropy 10G WAN Emulators, which emulate WAN links using variable inferences like packet loss ratio up to 100%, delays of up to 100s, diversified distributions of packet corruption and duplication laws;

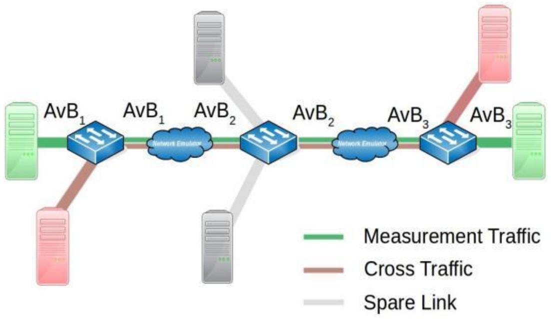


Fig 1. Setup of a measurement testbed.

- 2 instances of end systems one on which is sender with 12 x Intel(R) Xeon(R) CPUX5690 3.47 GHz processors, 32 GB RAM and equipped with 10G Chelsio NIC (driver cxgb4-2.0.0-ko), another one is receiver with 6 x Intel(R) Xeon(R) CPU X5690 3.47GHz processors, 42GB of RAM and 10G Chelsio NIC. Both machines run Linux distribution – Ubuntu 16.04 system with the 4.4.0-97-low-latency kernel.
- 10GE network switches Summit x650 from Extreme Networks were used as network devices.

In order to emulate fluidity of AvB under real world, one of the network emulators cyclically has been changed its capacity with such steps: 2 Gbps, 7 Gbps, 10 Gbps during the whole measurement.

Also, measurements have been performed on servers with different NICs - Intel (ixgbe-4.2.1-k) with an eye to confirm results tendency.

The RTT on the link between sender and receiver without emulation is equal to 250 µs. Additional emulated delay was set to a fixed value of 20 ms.

4 Measurement tool description

In terms of this research a new tool named Kite2 has been developed. It borrows its name from Kite application [4] estimation methodology of which had formed the basis of the developed tool.

Approach of Kite2 is based on the Probe Rate Model (PRM) of the active probing measurement and basically uses analysis of inter-packet interval deviation on the receiver side (IpIr) in order to detect whether the sending rate of probe packets meets the available bandwidth limits of the link between the sender and the receiver.

4.1. Kite2 workflow

The test application was developed with performance and simplicity in mind. It is written in C++ programming language using HighPerTimer library for precise time measurements [17], has a client-server architecture and supports 3 probing modes:

- Sequential (train mode);
- Batch (packet pairs);
- Packet pairs inverted.

These modes are described below. The measurement process is comprises of short send sessions (probing), according to specified probing mode:

- sequential, when the probe samples are sent with a constant interval, and the receiving side measures interval between each successive packets, called "train mode" [18] *(Figure 2)*;

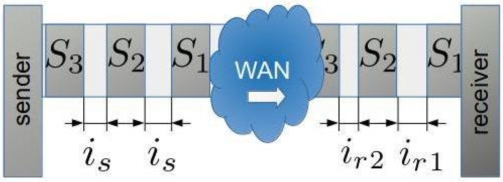


Fig 2. Train probing mode measurement.

- batch, when two consecutive samples are sent with the shortest possible interval and after that pair of packet, a pause of a specific inter-packet interval is inserted. It results in coupled packets. Receiver measures then intervals between pairs, which is defined as inter-send packet interval on the sending side, e.g. between 2nd and 3rd, 4th and 5th packets. This operational mode is called "packet pairs" (*Figure 3*).

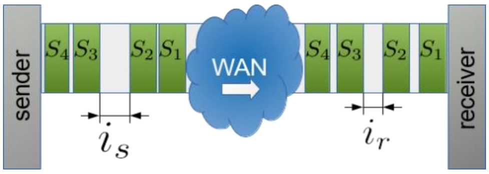


Fig 3. Pair probing mode measurement.

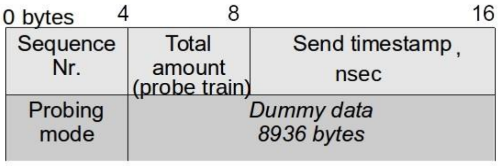
- "Packet pairs inverted" mode means the same operation as "packet pairs" one, but the receiver instead measures interval within a pair, which corresponds to the shortest possible send interval, e.g. between 1 and 2, 3 and 4 packets. This mode was developed in a contradiction to the previous ones in order to analyze the efficiency of well-known modes and a new one in the high-speed network environment.

In order to be able to investigate contemporary high capacity links, it is necessary to have a reliable time source and precise time measuring functions to keep a constant interval between probing samples while sending ones and to obtain accurate results during their reception. Linux API does not provide the sufficient accuracy [19], suitable for 1 Gbps and above networks, from this point HighPerTimer library provides up to 20 nanoseconds accuracy for time acquisition and waiting operations, thus allowing to investigate 10 Gbps and 40 Gbps networks, provided by FILA at Anhalt University of Applied Sciences.

Kite2 has an option to specify sending rate or approximate utilization of the bandwidth of a link, as the main goal of Kite2 is to implement an effective approach of bandwidth estimation with the smallest possible impact on the channel. The idea behind this estimation, as described above, is based on the observation of inter-packet interval deviations between sending and receiving sides.

However, despite the method simplicity there are pitfalls introduced by different features of network devices, such as undetermined buffering methods of intermediate network devices or interrupt coalescence on a host running Kite2 which is an objective of the current research.

Under the cover, client and server exchange UDP messages. The first one sends probing samples, structure of which is shown in *Figure 4*, with the defined interval.



0 bytes 4 8 16

Sequence Nr.	Total amount (probe train)	Send timestamp , nsec
Probing mode	*Dummy data* *8936 bytes*	

Fig 4. Kite2 sample structure (UDP payload).

The interval can be specified by a user as the transmission data rate option. Server waits for the first sample and then begins a probing session estimation, initiated by this first sample. During the single probe, server gathers samples calculating required intervals obtained from timestamps. On the sending side the sending time stamp is set by the Kite2 in the application level while on the receiving side the time stamp is set by the kernel on reception of the data packet. The sender keeps most recent samples according to measurement type: adjacent messages for “train” or "packet pairs inverted" modes, odd for “packet pairs”. Two packets are enough for current probing modes.

When all issued probes are received, the server calculates an average inter-packet interval and reception data rate based on this average. In case of packet losses receiver waits for 3 sec and prepares a report message to provide the results to the client application. This result calculations are done after the probes session for reduction of calculation load in order to obtain accurate timestamps.

At the server side processes the result of probes calculating a running-average IpIr for probes, forms the report with the estimated AvB and send it to the sender. For each reported AvB value client recalculates the data rate for transmission and the number of samples to be sent:

$$R_{s_i} = 1.1 \cdot R_{r_{i-1}} \tag{1}$$

where R_s – send data rate, R_r – receive data rate, i – number of samples in train;

and the amount of samples:

$$N = \frac{c \cdot R_r}{8L} \tag{2}$$

where N – amount of samples for the probe, c is the maximal portion of occupied bandwidth (as a ratio), L – size of the packet.

The client sends probes repeatedly increasing send data rate on each iteration until there is no increment of receiving data rate in response at least 3 times to ensure it is a current boundary of a link and not a random measurement error. The last value of receiving data rate is assumed the available bandwidth.

4.2 AvB tools performance comparison

Performance and accuracy of estimations have been approved tests under two general conditions:

1. On a testbed of an emulated 10 Gbps link with the variable network parameters (capacity, delay and loss);
2. In an environment between two dedicated servers with the physical link capacity of 1Gbps and background network load of TCP traffic approx. 150 Mbps (measured by *iperf3* tool).

The application tests in emulated 10G Network have been performed with the duration of the 80s. During one measurement cycle of the 20s, 4 variations of capacity has been configured in order to emulate an instability of AvB in a real environment and two comparative tools are used:

- Yaz or calibrated *pathload* tool based on the *Pathload* algorithm of self-load periodic streams but uses only one probe packets train opposed to fleets of trains in *Pathload* evaluates inter-packet interval (and corresponding to it constant bit rate - CBR) for each iteration of measurement as an increased value of the previous inter-packet interval on the sender side by its difference with the inter-packet interval on the receiver side. Yaz terminates the process of measurements as soon as difference of inter-packet intervals gains predefined thresholds. The previous study of this tool shows high performance in networks up to 1 Gps. Thus, Yaz is used as a reference tool in this research.

- Kite uses the same technique but with enhanced features like automatic adjustment of a number of samples which allows using the tool in high-speed networks, so they so called Step Decreasing Factor – rule of increasing the probe send rate. The adjustment of the number of samples in a train reduces significantly the data overhead for the bandwidth estimation while the SDF offers an enhances algorithm of increase of probe data rate which leads to significantly faster convergence of estimated data rates in comparison to Yaz.

Emulated delay while test case was continually enlarged to the threshold of 0.5 ms. Results of experiments are shown in *Figure 5*.

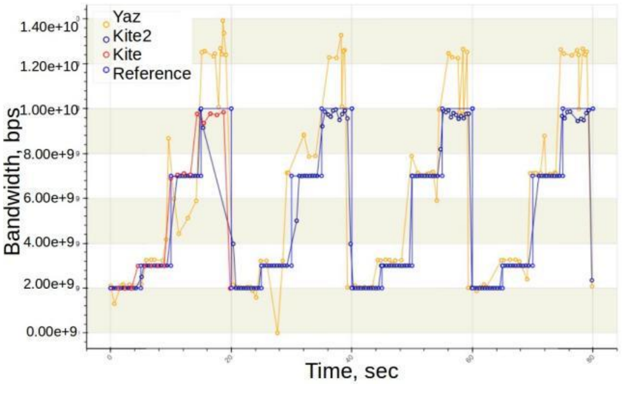


Fig 5. Accuracy of the AvB tools measurements in network with 0.5 ms of delay.

The light-blue graph corresponds to the reference bandwidth which is equal to the capacity in an idle link, the dark blue presents results of measurement by Kite2 tool, and results of experiments done by other two tool - Yaz and Kite are shown as red and yellow lines accordingly. Dots on each graph corresponds to the estimated values by each tool per one measurement iteration (so-called 'instant bandwidth value').

On the idle link when the capacity is limited only by emulators, all three tools show low overhead for the estimation of AvB up to 3 Gbps. With the increase of capacity overestimations of Yaz grow constantly, as well as the number of estimation outliers [20].

The most precise estimation for the range of 7-10 Gbps shows Kite2 tool, despite the fact that it is tended to overestimate the AvB at the high speeds. It might be related to the amount of occupied capacity by this tool while estimation. For the 10G link, it consumes approx. 1,6 Gbps for calculation each finite AvB value in a contradiction to 300 Mbps consumed by Kite tool.

Estimation error during the experiments is shown in *Figure 6*. Bars of the graphs corresponds to a minimal and maximal estimated error value, green and red areas show lower and upper quartiles of the mean. As it is represented by the boxplots for each examined tool, the lowest range of error values corresponds to Kite and Kite2 measurement with a tendency of underestimation up to 100 Mbps for the 10G network. Peak outliers of Yaz measurements in *Figure 6* are detected in the wide range of the boxplot, which spreads 400 Mbps of overestimation.

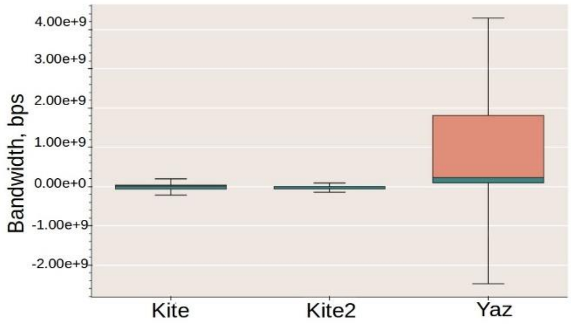


Fig 6. Estimation error of the AvB tools measurements in network with 0.5 ms delay.

In order to track the changes of measurement accuracy in the deterioration of conditions by the network, 1% constant Packet Losses has been configured on the link. Results of experiments are shown in *Figure 7*.

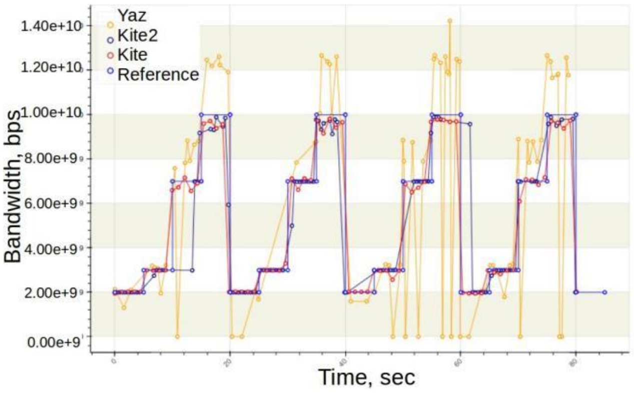


Fig 7. Estimation error in network with 1% losses.

Kite and Kite2 present the error scatter shown in *Figure 8*.

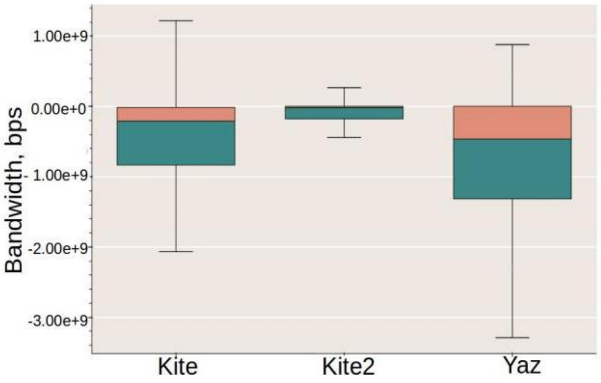


Fig 8. Estimation error of the AvB tools in a network with 1% of losses.

A tendency of AvB underestimation observed in all investigated tools and varies from 100Mbps for Kite2 to 1.3Gbps for Yaz.

Results of a measurement in environment are shown in *Figure 9*. Test have been performed on machines which run Web and synchronization operations and accumulate a valid network load by clients' requests and data transmission. A physical link at each instance has a capacity of 1 Gbps, RTT value corresponds to 10 ms.

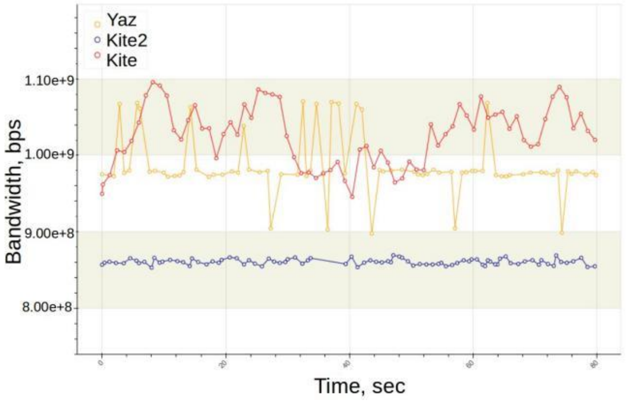


Fig 9. Accuracy of the AvB measurements in a real network.

Deviation of measurement results is shown in *Figure 10* and represents stability of processed values by Kite2 tended to underestimate 150 Mbps of AvB at the first glance.

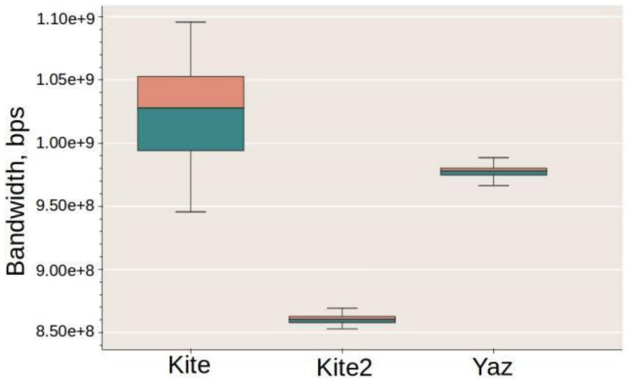


Fig 10. Estimation error of the AvB tools in a real network.

In order to determine an actual AvB *iprerf3* tool which is based on link congestion has been used. Its results during of 1 min operation with a send data rate speed of 1 Gbps showed that AvB value is within 740 Mbps to 875 Mbps on the investigated link with the background bursty web traffic.

Results of the tests show competitiveness of Kite2 measurement tool and its potential to investigate high-speed networks.

5 Experimental research

This section describes the correlation between the measurement accuracy and the probing packet size. The threshold of a network load for a defined measurement accuracy has been chosen experimentally.

5.1 Test scenarios

Measurements have been performed on the emulated network testbed with spare link and constant AvB equal to its capacity of 10 Gbps. Duration of each test corresponded to 360 s and had been repeated 10 times.

In addition, each test was executed with an eye for two Maximal Transmission Unit (MTU) size values: 1500 bytes (the most common value in Internet) and 9000 bytes (common in 10 Gbps and above networks) which corresponds to 1472 bytes of data in UDP packet payload and 8972 bytes accordingly, further referred to as payload. The steps of ratio variation for the network occupation by probing samples are presented in a *Table 1*.

Table 1. Number of estimated results (N) and bandwidth usage ratio (p).

p	0.015	0.025	0.035	0.05	0.065	0.095	0.15	0.35
N	773	438	250	159	150	120	122	89

An observed result has been taken with the payload size of 8972, while tests with the lower value have been rejected due to the high error rate (more that 60% for the 1472 bytes). Therefore, the research was focused on higher MTU size, besides, for bigger payload the measurement performance increased and processing time was reduced to 0.75 s in average but causes more significant network load, though.

5.2 Statistical analysis of the test results

With the help of statistical software for analyzing of empirical data *SPSS* [21] we define the correlation values for observations: measurement errors for available bandwidth estimation (*X*), the ratio of bandwidth utilization by measurement probes (*Y*), the amount of estimation results (*Z*).

Correlation coefficients corresponds to such values:

- **Pearson's**

$$r_{xy} = \frac{(\overline{xy} - \overline{x} \cdot \overline{y})}{S_x \cdot S_y} \approx -0.396\,, \qquad (3)$$

where:

- n is the sample size;
- x_i, y_i are the single samples indexed with i;
- $\overline{x}, \overline{y}, \overline{xy}$ are the sample mean;
- S_x, S_y are the standard deviations.

- **Spearman's**

$$r_s = 1 - \frac{6\sum_{i=1}^{n} d_i^2 + T_X + T_Y}{\mathrm{n} \cdot (\mathrm{n}^2 - 1)} \approx -0.671 \qquad (4)$$

where:

- $d_i = rg(X_i) - rg(Y_i)$ is the difference between the two ranks of each observation;
- n is the number of observations;
- T_X, T_Y – corrections of related ranks.

Because the value of relation between *X* and *Y* is evaluated on the data set, thus its statistical significance should be tested, i.e. estimation of possibility to expand the obtained results over population universe.

Let's calculate the *t*-statistics for the correlation coefficients:

- Pearson's

$$t(X, Y) = r\frac{\sqrt{n-2}}{\sqrt{1-r^2}} \approx -19.76 \qquad (5)$$

- Spearman's

$$t(X, Y) = r\frac{\sqrt{n-2}}{\sqrt{1-r^2}} \approx -41.46 \qquad (6)$$

and compare both with a critical value t_{crit} :

$$t_{\alpha,l} = \begin{cases} 2.578 & for\ \alpha \leq 0.01 \\ 1.961 & for\ \alpha \leq 0.05 \end{cases} \qquad (7)$$

Since the absolute value of *t*-statistics are greater than critical ones $|t| > t_{0.01,l} = 2.578 > t_{0.05,l} = 1.961$ for $\alpha \leq 0.05$ and $\alpha \leq 0.01$ we can assume that the correlation coefficient is significant for $\alpha = 0.01$ level.

Hence, the relationship *exists* between measurement error of available bandwidth estimations (*X*) and the bandwidth occupation by probing train (*Y*) for significance value $\alpha \leq 0.01$. Pearson's correlation coefficient is equal *-0.396* which means the average inverse relation. I. e. while increasing the value of *X* feature, the value of *Y* one is decreasing. The corresponding results is possible to expand over all the data of the same type, because the value of correlation coefficient is significant for level $\alpha \leq 0.01$. Spearman's coefficient is equal to *-0.671* which confirms the average reverse relation between the mentioned feature sets. The test for significance of Spearman's coefficient also confirms that the results can be expanded over the general population for level $\alpha \leq 0.01$ ($|41.24| > 2.578_{0.01}$).

Correlation coefficients for *X* and *Z* according to *Equations (3, 4)*:

Pearson's $r_{xz} \approx 0.598$ and Spearman's $r_s \approx 0.669$.

To check statistical significance of correlation value between *X* and *Z* we provided calculation of *t*-statistics:

Pearson's $t(X, Y) \approx 34.18$ and Spearman's $t(X, Y) \approx 41.24$

and compared the results with a critical value t_{crit} according to *Equitation (7).*

Hence, the relationship *exists* between measurement errors of available bandwidth estimation (*X*) and the number of estimated results (*Z*) for significance value $\alpha \leq 0.01$. Pearson's correlation coefficient is equal *0.598* which means the average direct relation. I. e. while increasing the value of *X* feature, the value of *Z* one increasing as well. The corresponding results is possible to expand over all the data of the same type (the value of correlation coefficient is significant for level $\alpha \leq 0.01$. Spearman's coefficient is equal to *0.669* which confirms the average direct relation between the mentioned feature sets. The test for significance of Spearman's coefficient also confirms that the results can be expanded over the general population within $\alpha \leq 0.01$ ($|41.24| > 2.578_{0.01}$).

6 Conclusion and future work

The research presented here shows that there is a direct correlation between the probing size in available bandwidth estimation and measurement errors, as described in Section 5.2. It has been shown that the methods, implemented in Kite2 enables us to reduce the amount of samples by 20%, while the bandwidth estimation deviation was reduced to 100 Mbps for 10 Gbps links. Tests with payload sizes of 1472 and 8972 bytes have shown that MTU size essentially impact the estimation time. While estimation with smaller packets takes up to 6.5s measurements with jumbo frames need only 0.75s for stable results. Moreover, estimation with normal IP packet sizes is limited by 3.4 Gbps while with jumbo frames up 9.98 Gbps where estimated with sufficient accuracy. Statistical investigations on tests show that 5% probing load ratio gives up to 94% measurement accuracy in 10 Gbps networks with 20 ms of delay, compared to the 97% of accuracy when probing samples occupy quarter of the path's AvB. That will be taken into account in further experiments.

Another significant finding was, that the network delay does not affect the accuracy of estimations and processing time. As next research tasks we consider adjustment of probing size in high-speed networks. While in emulated networks the estimation accuracy was very accurate in real-world networks achieved results have shown much higher uncertainty. So algorithm adoptions for real-world networks have to be applied. Also some more sophisticated test scenarios with presence of heavy cross traffic must be elaborated and executed. Also, an algorithm for the sending rate adaptation to the changing network conditions could lead for further improvements.

Acknowledgement

This work has been funded by Volkswagen Foundation for trilateral partnership between scholars and scientists from Ukraine, Russia and Germany within the project CloudBDT: Algorithms and Methods for Big Data Transport in Cloud Environments.

References

1. R. Prasad, C. Dovrolis, M. Murray and K. Claffy, “Bandwidth estimation: metrics, measurement techniques, and tools,” IEEE Network, vol. 17, no. 6, 2003, pp. 27-35.
2. K. M. Salehin and R. Rojas-Cessa, "Method for measuring the packet processing time of Internet workstations with the detection of interrupt coalescence", 2016 IEEE 17th International Conference on High Performance Switching and Routing (HPSR), Yokohama, 2016, pp. 142-147.
3. N. Hu and P. Steenkiste, “Evaluation and characterization of 33 Areas in Communications, vol. 21, no. 6, 2003, pp. 879-894.
4. D. Kachan, E. Siemens and V. Shuvalov, "Available bandwidth measurement for 10 Gbps networks," 2015 International Siberian Conference on Control and Communications (SIBCON), Omsk, 2015, pp. 1-10.
5. J. Sommers, P. Barford, W. Willinger “A Proposed Framework for Calibration of Available Bandwidth Estimation Tools,” Proc. of the 11th IEEE Symposium on Computers and Communications, 2006, pp. 709-718.
6. M. Jain and C. Dovrolis, “Pathload: a measurement tool for end-to-end available bandwidth,” in Proceedings of the 3rd Passive and Active Measurements Workshop, 2002.
7. V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil and L. Cottrell, “pathChirp: efficient available bandwidth estimation for network paths,” in Proceedings of the 4th Passive and Active Measurements Workshop, 2003.
8. Ningning Hu and P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques," in IEEE Journal on Selected Areas in Communications, vol. 21, no. 6, pp. 879-894, Aug. 2003.
9. L. Angrisani, S. DAntonio, M. Vadursi and G. Ventre, “Performance comparison of different techniques for available bandwidth measurement in packet switched networks,” in VECIMS
10. 2003 - International Symposium on Virtual Environments, Human-Computer Interfaces, and Measurement Systems, 2003.
11. E. Goldoni, G. Rossi and A. Torelli, "Assolo, a New Method for Available Bandwidth Estimation," 2009 Fourth International Conference on Internet Monitoring and Protection, Venice/Mestre, 2009, pp. 130-136.
12. E. T. Saulnier and B. Bortscheller, "Data transfer bottlenecks over SPARC-based computer networks," Proceedings of 20th Conference on Local Computer Networks, Minneapolis, MN, 1995, pp. 289-295.
13. C. Dovrolis, P. Ramanathan and D. Moore, "Packet-dispersion techniques and a capacity-estimation methodology," in IEEE/ACM Transactions on Networking, vol. 12, no. 6, pp. 963-977, Dec. 2004.
14. Y Labit, Powezarski, N. Larrieu, “Evaluation of Active Measurement Tools for Bandwidth Estimation in Real Environment”, Workshop on End-to-End Monitoring Techniques and Services, IEEE 2005
15. A. A. Ali, F. Michaut and F. Lepage, “End-to-end available bandwidth measurement tools : a comparative evaluation of performances,” Proc. 4th International Workshop on Internet Performance,IPS-MoMe 2006.
16. C.M.Leong, T.S. Wei, A.T. Samsudin “Performance evaluation of optimized available bandwidth estimation tools in FTTH networks” Communications (APCC), 2012 18th Asia-Pacific Conference.
17. I. Fedotova, E. Siemens, and H. Hu, ”A high-precision time handling library,” J. of Comm. and Computer, vol. 10, pp. 1076-1086, 2013.
18. C. Dovrolis, P. Ramanathan, and D. Moore, “What do Packet Dispersion Techniques Measure?,” Proc. IEEE INFOCOM, Apr. 2001, pp. 905–14.
19. D. Kachan, E. Siemens, H. Hu. Tools for the high-accuracy time measurement in computer systems, (Russian) // 6th Industrial Scientific Conference «Information Society Technologies». Moscow, 2012
20. Kirova V., Korniichuk M. “Performance Comparison of PRM-based Available Bandwidth Estimation Tools in 10G Emulated Networks”, 72d scientific and technical conference of professors faculty, academics, postgraduate students and students, 2017.
21. “IBM SPSS Statistics v23.0 documentation.” [Online] Available: https://www.ibm.com/support/knowledgecenter/en/SSLVMB_23.0.0/spss/product_landing.html. [Accessed 04-Jan-2018].