

One Way Delay Trend Detection for Available Bandwidth Measurement

Alexander Chobanyan, Matt Mutka, Zhiwei Cen
Department of Computer Science and Engineering
Michigan State University
48824, East Lansing, MI
Phone: 517-353-9731
Email: {chobany1, mutka, cenzhiwe}@cse.msu.edu

Ning Xi
Department of Electrical and Computer Engineering
Michigan State University
48824, East Lansing, MI
Phone: 517-432-1925
Email: xin@egr.msu.edu

Abstract

Network measurements, Statistics, Available Bandwidth. Available bandwidth (AB) defined as a minimum spare capacity of links constituting a network path is an important QoS characteristic of the path. We propose to improve a whole range of “probe-rate” AB measurement tools that send sequences of measurement packets (called “trains”) across the network path. If transmission times of packets in the train, called One Way Delays (OWD), show an increasing trend as packet sequence number in the train increases, then AB is believed to be lower than the rate at which the train was sent. In contrast, an absence of a trend indicates that AB is higher than the rate of the train.

We propose an algorithm for efficient OWD trend detection and compare it to widely used OWD trend detection tests. Our experiments clearly show that our proposed method significantly outperforms tests used in present “probe-rate” AB measurement tools.

One Way Delay Trend Detection for Available Bandwidth Measurement

Alexander Chobanyan, Matt Mutka, Zhiwei Cen
Department of Computer Science and Engineering
Michigan State University
Email: {chobany1, mutka, cenzhiwe}@cse.msu.edu

Ning Xi
Department of Electrical and Computer Engineering
Michigan State University
Email: xin@egr.msu.edu

Abstract—Network measurements, Statistics, Available Bandwidth. Available bandwidth (AB) defined as a minimum spare capacity of links constituting a network path is an important QoS characteristic of the path. We propose to improve a whole range of “probe-rate” AB measurement tools that send sequences of measurement packets (called “trains”) across the network path. If transmission times of packets in the train, called One Way Delays (OWD), show an increasing trend as packet sequence number in the train increases, then AB is believed to be lower than the rate at which the train was sent. In contrast, an absence of a trend indicates that AB is higher than the rate of the train.

We propose an algorithm for efficient OWD trend detection and compare it to widely used OWD trend detection tests. Our experiments clearly show that our proposed method significantly outperforms tests used in present “probe-rate” AB measurement tools.

I. INTRODUCTION

Estimating the quality of a network path has always been an important problem for QoS real-time network applications. Available end-to-end bandwidth represents a maximal rate at which a sender can transmit data over the path without significantly affecting existing cross-traffic and therefore is an important QoS characteristic of the path. Rapidly evolving network real-time applications give rise to transmitting over the Internet new media types such as temperature, haptic data, and others. The increasing complexity of a set of data-streams that can be used by current real-time applications in turn poses a need to improve network resource allocation and management for supporting a real-time task execution. Being able to accurately and quickly measure spare capacity (available bandwidth) of an end-to-end network path has a crucial meaning for efficient QoS real-time task support.

Measurement of available bandwidth is a complicated task since cross-traffic in the network path dynamically changes thus causing change of the available bandwidth. The best that present measurement techniques can do is to measure the *mean* available bandwidth averaged over a certain period of time τ . Decreasing the measurement time τ is an important issue because the assumption that available bandwidth does not change over a smaller period of time is weaker and more realistic rather than the assumption that the bandwidth does not change over larger periods.

Presently two types of approaches to measurement of available bandwidth exist. Probe-gap models [1], [2] send a

pair of packets and make further inference about available bandwidth based on the measurement of a gap between packets at the receiver’s side. Probe-rate models [3]–[5] send sequences or “trains” of packets at different rates assuming that a sequence sent at a rate higher than the available bandwidth causes noticeable positive trend in times that packets in the sequence need for travelling from sender to receiver. These times are often called one-way-delays (OWDs). Then the available bandwidth at that time can be found by a simple binary search. Probe-gap approaches are less intrusive than probe-rate approaches. However, probe-gap approaches suffer from the so called “interrupt coalescence / context switch (IC/CS)” effect [6], which significantly distorts measurements of packets inter-arrival times. Probe-rate models are affected by IC/CS phenomenon as well but not as much as probe-gap models. Probe-rate models send a larger number of packets and therefore an IC/CS effect can at least be clearly observed and to some extent considered by a model deployed in the probe-rate measurement tool.

In this work we propose to use a new metric for OWD trend detection of packets in a train together with a new method for dealing with the IC/CS effect. We compare our results with Pathload [3], which to our knowledge is the only reliably working tool that considers the IC/CS phenomenon. We demonstrate that our metric for trend detection and our way of dealing with interrupt coalescence and context switch effects provide much better OWD trend detection accuracy. It can also tolerate processor utilization of a receiving machine better and therefore can improve not only Pathload but also a wide-range of probe-rate oriented available bandwidth measurement tools that use the trend detection algorithm introduced by Pathload’s authors. In particular, with our approach in most cases we need to send only up to three trains while achieving the same accuracy as Pathload when it sends 6 trains. This allows a decrease in the measurement time without sacrificing accuracy.

II. BACKGROUND

Significant work that has been done in mean available bandwidth estimation is well reflected by Strauss, et al [1] and Hu, et al [7]. In short, deployed bandwidth estimation models can be divided into probe gap (or packet dispersion) and probe rate classes. Probe gap models used in Spruce [1]

and Delphi [8] analyze delay between probe packets and make further inference about available bandwidth based on that delay whereas probe rate models utilize trains of packets sent at different rates.

The probe rate approach is believed to be more intrusive than the probe gap approach [1]. At the same time as was shown in [9], Pathload measurements do not significantly affect round trip time of cross-traffic data packets and do not significantly change available bandwidth. There exist other lightweight probe rate measurement techniques [4], [5] that are even less intrusive than Pathload. They, however, use the same basic metrics as Pathload for trend-detection in a particular train of packets.

Both probe rate and probe gap measurement approaches suffer from known interrupt coalescence / context switch (IC/CS) effects [6]. If interrupt coalescence is enabled at the receiver's network adapter, then the adapter waits for a group of packets to arrive and only after that issues an interrupt request. Then the entire group is processed with one interrupt. The direct IC impact on network measurements is that packets are not time-stamped by a receiver after they really arrived but with a variable delay. Some may wait for an interrupt request for up to hundreds of microseconds, which is a significant influence on network measurements. Context switch effects have the same impact. If a processor is utilized by another job, then a certain random time-period is passed after a packet's arrival before it is processed and time-stamped by a receiver.

It is fairly complicated to consider the IC/CS influence when only pairs of packets are sent. To our knowledge there are no known probe-gap (packet dispersion) based available bandwidth measurements that consider the IC/CS effect. Pathload is the only tool from the probe-rate approaches that partially addresses the IC/CS issue. We therefore here-and-after focus our attention on Pathload as the most stable and accurate end-to-end available bandwidth measurement tool to our knowledge.

Pathload checks the time that a system needs for delivering packets from network adapter to upper layers. All packets that are received within this time-frame are assigned to one group or burst. Then in a packet train Pathload counts only one packet out of each burst and discards the rest of the packets.

After IC/CS related filtering, Pathload divides the remaining N packets into $\Gamma \approx \sqrt{N}$ groups and takes the median from each group. The remaining packets are discarded as well. Finally two metrics are computed based on remaining Γ measurements. The Pairwise Comparison Test (PCT) metric is evaluated as $S_{PCT} = \frac{\sum_{k=2}^{\Gamma} I(M_k > M_{k-1})}{\Gamma - 1}$ where I is an indicator function evaluated to one when the median of the k -th group M_k is larger than the median of the $k-1$ st group and to zero otherwise. As it follows by definition if a train has an increasing trend then the PCT metric approaches 1. If there is no trend, then the PCT metric should be close to zero. Pairwise Difference Test (PDT) metric is defined as $S_{PDT} = \frac{M_{\Gamma} - M_1}{\sum_{k=2}^{\Gamma} |M_k - M_{k-1}|}$ If the train does not have an increasing OWD trend then S_{PDT} should vary around zero,

whereas in a presence of an increasing trend it should be close to one. The PDT metric was introduced for detection of a trend with a significant start-to-end OWD difference. Finally, two tests together decide what to report for a given train of packets.

The PCT and PDT tests represent a basic mechanism of trend detection in Pathload. In general, the number of packets in a train, the train packet size, and the bandwidth-rate search mechanism may vary among approaches. For example, Man, et al [5] introduce a light-weight train-based tool that is less intrusive than Pathload on one hand but sacrifices accuracy on the other hand. Nishikawa, et al [4] propose to evaluate the approximate distribution of available bandwidth as of a random quantity. Basic metrics in these works remained, however, the same as they were introduced first in Pathload.

III. MAIN CHALLENGES OF IC/CS ELIMINATION AND TREND DETECTION IN PATHLOAD

Jain and Dovrolis have shown in [9] that Pathload demonstrates reasonably good performance on paths that have tight and narrow link in the middle of the path. For such cases Pathload's error was around 5% of available bandwidth value when this value belongs to the range 70-80Mbps and around 10% of available bandwidth value when the value is smaller than 10Mbps.

In such cases, however, the slow link, which lies in the middle of the path, stretches packets, whereas fast links contract packets. In other words, suppose that packets are sent with a certain fixed delay. When they pass from a faster link to slower link the delay between packet arrivals becomes smaller or is eliminated. In contrast, when packets pass from a slow link to a fast link, then the delay between packets increases. Therefore by having a slow link in the middle of the path the authors are guaranteed that there will be a significant delay between packet arrivals at the end which first, makes trend detection easier and second, makes measurements less vulnerable to the IC/CS effect.

We believe that another "friendly" factor in the experimental setup was a good match of a length of a packet train to the path across which the train was sent. More precisely, when a train was sent at rates higher than the available bandwidth at that time, routers in the middle of the path were experiencing instantaneous increase of packet queues but at the same time were not dropping any significant portion of packets. That experimental design allows to avoid addressing the problem of packet loss.

Ubik, et al [10] performed an extensive analysis of Pathload's operation for various networks, values of available bandwidth and values of a cross-traffic. Tables in their work clearly show that the performance of Pathload is far from ideal.

Even when experiencing packet loss, Pathload still may sometimes report available bandwidth from a correct range despite the high error rate of trend detection at the level of a particular train. Pathload creators recognized the challenge of packet losses and introduced an artificial way of partially avoiding this problem. They set up two levels of packet losses.

If the percentage of losses in at least one train in a set of trains sent at some fixed rate is higher than HIGH LOSS RATE then the rate is automatically believed to be higher than the available bandwidth. If losses in more than 50% of trains sent at a fixed rate exceed MEDIUM LOSS RATE, then the rate is believed to be higher than the available bandwidth as well. These two levels of percentages are initially set in Pathload as 15% and 7%, respectively.

The artificial way described above of dealing with packet losses has its own shortcomings. First, HIGH and MEDIUM loss rate parameters have to be tuned. Second, packets may be lost because of a processor utilization at a receiving computer. As a consequence of Pathload’s method of dealing with packet losses, Pathload fails to report correct bandwidth even when processor utilization at a receiving machine reaches 5%. This can be considered as a significant limitation of a tool for a long-lasting measurements given that a receiver machine may be running under any of a variety of non-real-time operating systems.

The way that Pathload deals with the IC/CS effect is not perfect as well. The time that the operating system needs for delivering packets to the upper layer cannot be precisely estimated. Pathload’s PCT/PDT metrics also leave a place for improvement. Figure 1 illustrates a typical example of a train misclassified by the PCT/PDT test. The train was sent at the rate which was 20Mbps higher than the available bandwidth. On the left hand-side are plotted one-way delays of 100 packets belonging to the train. We see that the plot has parts with negative trend and these parts are strictly “linear”, which is a direct consequence of the IC/CS effect. We can think of IC/CS as of an additional buffer at which packets spend certain time and which is periodically emptied. The last packet in the group before the “buffer” is emptied waits the least time whereas the one that came straight after the previous emptying waits the longest period of time. Also, it is known that all packets were released by the sender with some fixed unchanged delay. It follows therefore that all packets related to the strictly linear decreasing part of the plot belong to the same group. Now consider a subset of packets constituted by the last packet taken out of each strictly-linear decreasing part. On the left hand-side plot one can clearly observe that aforementioned set constitutes subsequence with clearly-revealed positive trend.

On the right hand-side of figure 1 is depicted a subsequence of points chosen by a Pathload IC/CS elimination procedure. One can see on that plot that at least some points of the chosen subsequence do not characterize packets that are least delayed by the IC/CS effect. Moreover the trend picture that is clearly observed even by a naked eye on left plot is not so evident on right plot. The information about the index of packets finally chosen by IC/CS elimination procedure is also not considered by Pathload.

Finally, let us apply the PCT test to what is left on the right plot. There are left only seven points and only half of those are bigger than their direct predecessors. As a final result for this real-life typical example Pathload received a value of 0.43

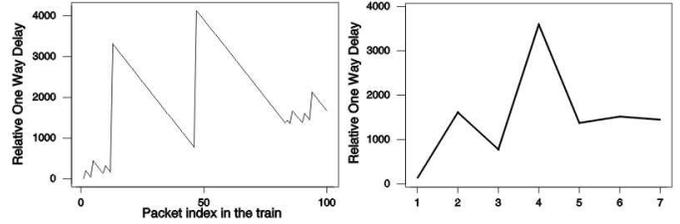


Fig. 1. Shortcomings of IC/CS elimination procedure in Pathload

for the PCT metric and 0.17 for the PDT metric, which are deep in the region classifying the train as the one without a trend.

The conclusion is that Pathload throws away significant part of measurement information at the IC/CS elimination stage as well as when dividing the rest of the data into groups and discarding everything except medians in each group. Also there is an absence of statistical motivation of the efficiency of the PCT/PDT test as well as absence of reasoning why the chosen thresholds for PCT and PDT metrics should do a reasonably good job for an arbitrarily taken network path. The example which we plotted in figure 1 occurred frequently in data that we collected at our network segment and therefore explains significant inaccuracy of the PCT/PDT test when the experimental setup is not as friendly as it was with a highly-revealed tight and narrow link in the middle of the path.

IV. IC/CS ELIMINATION PROCEDURE AND T-TEST

For IC/CS elimination we detect cases when OWDs of several packets in a row constitute a strictly linear decreasing trend. In other words, all packets with one way delays constituting a clearly revealed straight line with a negative slope are discarded except the very last one. We require such a group to have at least three packets in order to discard all packets in the group except the very last packet. It is important that at this stage that we keep information about indices of packets left for a further statistical analysis. This way of dealing with the IC/CS effect is more simple and effective than the one proposed in Pathload. We do not assume that the very last packet in the group is processed immediately after it arrives. We only say that it waits the shortest time for an interrupt among all packets in its group. Also we assume that the time that the last packet in a group waits for an interrupt is normally distributed and therefore methods of linear regression analysis can be further applied to chosen subsequence of packets for trend detection.

Under a linear regression model we plot a best-fit “least-square” line across data points. x -coordinates of data-points are indices of packets in a train that incorporate time flow. y -coordinates are corresponding to one way delays. A simple linear regression model gives a direct way [11] to compute slope coefficient $\hat{\beta}_1$ of the plotted line. It is known [11] that under linear regression model assumptions coefficient $\hat{\beta}_1$ follows a t-distribution with $n - 1$ degrees of freedom where n is a number of data-points available after IC/CS

elimination. The null hypothesis in the linear regression t-test is an assumption that the train does not contain any OWD trend or in other words the slope coefficient of a regression line is zero. Therefore the distribution of a random quantity $\hat{\beta}_1$ under null hypothesis is t-distribution centered around zero with $n - 1$ degrees of freedom. The error of the false rejection of null hypothesis is an error of classifying the train with no-trend as a train with a trend and is called a type I error. We refer to the error of classifying a train with a trend as a train without a trend as a type II error.

For each train of packets we obtain p-value [11] of the t-test as our final quantity for judging trend presence. Low p-value should be considered as an indicator of a trend presence, whereas high values should be considered in favor of accepting the null hypothesis. We choose the decision threshold for a p-value to be equal to 0.01. The threshold set in the p-value space has a statistical motivation behind it in contrast to empirically chosen thresholds for PCT/PDT metrics. It means that even before beginning an experiment we want to have the misclassification probability for trains which really do not have the OWD trend (i.e., trains which were sent at the rate lower than the available bandwidth) to be approximately equal to 0.01.

A consideration of how much information is still available after IC/CS elimination for further analysis can be viewed as one more argument towards choosing p-value as a final quantity of interest. With a decreasing of number of points remaining after IC/CS elimination, and corresponding decrease of degrees of freedom n t-distribution becomes more and more “stretched-out”. This means that for a fixed value of $\hat{\beta}_1$ p-value will be larger and larger with decreasing of amount of available data-points. Correspondingly, in order to have a sufficient basis for rejecting the null hypothesis we will need a much higher value of $\hat{\beta}_1$ compared to the case when more data points are available for the analysis.

Another advantage of our scheme of IC/CS elimination and trend detection is to deal better with packet losses. The influence of packet loss to trend detection is very significant. When routers in the path discard significant number of packets, they may “recover” from “instantaneous” overflowing of their queues. Therefore trend detection becomes available only when considering a “piece-wise” pattern for each piece that is constituted by a contiguous chunks of received packets. The effect of packet-loss is well-illustrated by figure 2. We can clearly see that even after elimination of IC/CS parts the whole picture does not give an impression that there is a trend whereas every contiguously received chunk does give such an impression. In such cases Pathload reports a high percentage of “uncleared” trains because Pathload’s scheme throws away too much information and therefore it frequently does not have any reserve left for further splitting a train into a number of sub-trains. Our scheme can afford such splitting and therefore we report much lower “unclear” trains, which results in a significant reduction of the number of trains that are required to be sent in order to achieve the desired detection accuracy. We report a train as “unclear” only when we have finally less

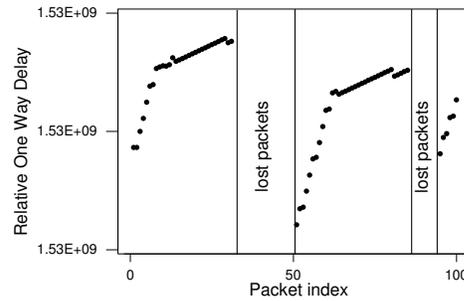


Fig. 2. OWD pattern at the presence of packet loss. Train was sent at rate 20 Mbps higher than available bandwidth

than 4 data-points left. If we have a number of reported p-values related to a number of sub-trains then we decide about the trend in the whole train based on the majority-vote rule in sub-trains. With decreasing the number of trains required to be sent, the measurement time also is significantly reduced. This gives our approach a higher degree of flexibility in available-bandwidth related planning, which is of a high importance for present real-time network applications.

V. EXPERIMENTAL EVALUATION

The goal for our experiment was to compare trend-detection efficiency of our approach with Pathload’s PCT/PDT test. We experimented within our laboratory network, which consists of three 100Mbps capacity segments. The sender and receiver were placed in the first and third segments, while the path from the first to the third segment was lying through the second segment. We wanted to have both Pathload’s PCT/PDT test and our approach to have absolutely equal conditions. Therefore we modified an original version of a Pathload by adding a number of printouts that were outputting to the separate file the OWD times measured by Pathload. Also for each train we outputted values of the PCT/PDT metric together with Pathload’s final decision about the train. Then we processed the same train data by our code, which eliminated the IC/CS effect, conducted t-test, outputted p-values, and finally made a decision about the trend presence.

During the first part of the experiment it was known with a high degree of confidence that the background traffic in all three segments is very low (around 1 Mbps) and does not exceed 2-3Mbps at any period of time of the length of 200 milliseconds. For that part of the experiments we knew therefore that all trains sent at the rate lower than 95Mbps should be reported as trains without trend. Also we knew that all three links capacities were 100Mbps and therefore all trains that were sent at rates 98 Mbps and higher should be reported as trains with a trend. This experimental setup did not contain any tight/narrow link in the middle of the path. We run Pathload continuously for the period of approximately 7 minutes and analyzed in total 4650 trains. Then we imposed a slight utilization to the processor of a receiver and repeated the experiment. Figure 3 shows how the performance of the PCT/PDT test deployed in a Pathload is compared with the performance of our approach when no processor utilization was imposed at the measurement machine responsible for

Rate at which trains were sent	Classified as 0-95 Mbps (without trend)		Classified as more than 98 Mbps (with trend)		Unclear trains		Trend detection accuracy for a fleet consisting out of 6 trains		N of trains that t-test needs to achieve 6-train accuracy of PCT/PDT test	N of trains required to achieve at least 0.99 accuracy		Total number of analyzed trains
	PCT/PDT	t-test	PCT/PDT	t-test	PCT/PDT	t-test	PCT/PDT	t-test		PCT/PDT	t-test	
15-35			0.000	0.014	0.500	0.014	0.985	1.000	3	7	3	72
35-55			0.560	0.023	0.233	0.000	0.312	1.000	1	>100	3	564
55-75			0.002	0.040	0.093	0.000	1.000	0.999	9	3	3	420
75-95			0.002	0.054	0.098	0.006	1.000	0.997	11	3	3	1296
> 98	0.083	0.026			0.296	0.000	0.971	1.000	1	9	3	2226

Fig. 3. Comparison of PCT/PDT test and t-test performances on nearly empty network and zero receiver's processor utilization

Rate at which trains were sent	Classified as 0-95 Mbps (without trend)		Classified as more than 98 Mbps (with trend)		Unclear trains		Trend detection accuracy for a fleet consisting out of 6 trains		N of trains that t-test needs to achieve 6-train accuracy of PCT/PDT test	N of trains required to achieve at least 0.99 accuracy		Total number of analyzed trains
	PCT/PDT	t-test	PCT/PDT	t-test	PCT/PDT	t-test	PCT/PDT	t-test		PCT/PDT	t-test	
15-35			0.441	0.010	0.190	0.055	0.508	1	1	>100	3	199
35-55			0.493	0.008	0.152	0.063	0.412	1	1	>100	3	1186
55-75			0.063	0.010	0.077	0.074	0.994	1	3	5	3	1070
75-95			0.033	0.011	0.053	0.069	0.999	1	5	4	3	1926
> 98	0.081	0.061			0.302	0.046	0.973	0.996	3	9	5	710

Fig. 4. Comparison of PCT/PDT test and t-test performances on nearly empty network and 5 – 7% receiver's processor utilization

Rate at which trains were sent	Classified as less than 35 Mbps (without trend)		Classified as more than 35 Mbps (with trend)		Unclear trains		Trend detection accuracy for a fleet consisting out of 6 trains		N of trains required to achieve at least 0.99 accuracy		Total number of analyzed trains
	PCT/PDT	t-test	PCT/PDT	t-test	PCT/PDT	t-test	PCT/PDT	t-test	PCT/PDT	t-test	
35-55			0.005	0.922	0.453	0.000	0.000	0.992	>100	5	592
55-75			0.021	0.761	0.958	0.000	0.004	0.853	>100	17	426
75-95			0.325	0.958	0.394	0.000	0.157	0.999	>100	3	335
> 98	0.276	0.082			0.305	0.001	0.766	0.991	38	5	1601

Fig. 5. Comparison of PCT/PDT test and t-test performances on heavy loaded network and zero receiver's processor utilization

Difference between rate at which trains were sent and available bandwidth	Classification error		Unclear trains		N of trains that t-test needs to achieve 6-train accuracy of PCT/PDT	N of trains required to achieve at least 0.99 accuracy		Total number of analyzed
	PCT/PDT	t-test	PCT/PDT	t-test		PCT/PDT	t-test	
15-35	0.122	0.008	0.464	0.000	1	16	1	1023
35-55	0.054	0.000	0.299	0.002	1	7	1	711
55-75	0.020	0.003	0.344	0.001	1	6	1	751
75-95	0.443	0.292	0.397	0.001	1	>100	29	741
> 98	0.476	0.065	0.346	0.001	1	>100	5	664

Fig. 6. Comparison of PCT/PDT test and t-test performances on average loaded network (around 50Mbps $\pm 10Mbps$ cross-traffic) and zero receiver's processor utilization

receiving packets. Figure 4 shows detection performance comparison when processor utilization was 5 – 7%. Tables are structured as follows. In the second bolded column are shown type II errors for PCT/PDT test and t-test, respectively. In particular, there is shown the portion of trains that were incorrectly classified as trains without a trend when the trend was really present. Note the this portion has been computed “conditionally” i.e., as a portion out of total number of trains that were sent in the specified rate-range and were not assigned to “uncleared” train set. In the third bolded column is shown type I error for multiple rate ranges. Let us remember that we have set a decision threshold for p-value equal to 0.01. We see in all tables that experimentally evaluated type I error is not that far from “theoretically-predicted” value 0.01, which can be considered to some extent as a verification of model's applicability. “Unclear trains” column is followed by

a column showing the number of trains that the t-test based tool needs to send in order to achieve accuracy of PCT/PDT test sending 6 trains at a given rate. Naturally, if a number in this column is greater than 6 then PCT/PDT test demonstrates better performance for a corresponding rate and vice-versa otherwise.

For the case when there was no any extra processor utilization, t-test performed better for all trains sent at the rate higher than 98Mbps as well in the range 15-55Mbps. Although we have slightly higher error rate for a range 55-95Mbps than PCT/PDT test has, still both approaches need to send three trains for achieving 99% accuracy. Difference in performances for the mentioned range therefore can be considered as not significant whereas for other rates we significantly outperform PCT/PDT test.

Figure 4 clearly shows that the PCT/PDT performs worse

when operating even under very small on-going processor utilization. In particular, the PCT/PDT test fails to correctly classify trains sent at the rate-range 15-55Mbps. Its error approaches 0.5 and sometimes even exceeds this value, which makes the rate classification procedure either unacceptably long or not applicable at all. At the same time in the global scale Pathload was unable to report even once available bandwidth from a correct range when processor utilization was imposed. For example, the maximal reported value of available bandwidth out of twenty runs was 70Mbps when the channel was almost free and processor utilization was 3.5%. The responsibility of this failure is Pathload's method to deal with packet losses by means of HIGH/MEDIUM LOSS RATE variables, which was described in the background section.

During the second part of our experiment it was known with a high degree of confidence that the total cross-traffic was always exceeding the level of 65Mbps. In other words, we were guaranteed that if trains that were sent at rate 35Mbps or higher are classified as those with trend, then this classification is correct. There was no processor utilization imposed to a measurement machine. The goal of the experiment was to check how efficient t-test and PCT/PDT test detect trend in trains that are sent at the rate lower than the capacity but higher than the available bandwidth of the network path.

Figure 5 shows results of t-test and PCT/PDT test accuracy for trains sent at the rate higher than the available bandwidth. Instead of type I error in the third bolded column we show detection accuracy for a specified range of rates. Figure 5 shows that the PCT/PDT test completely fails to detect trend when suffering packet losses and non-friendly experimental setup. We also note that despite this failure at the level of trains, Pathload still always reported values of the available bandwidth from the correct range because of the artificially implemented packet-loss mechanism. In this case values reported by PCT/PDT test simply were ignored by a Pathload and it adjusted it's rate solely based on HIGH and MEDIUM LOSS RATE variables. We believe, however, that an available bandwidth measurement tool can be significantly improved after excluding an artificial mechanism of dealing with packet losses and respectively relying solely on a train-level reported statistic.

In the third part of our experiment we artificially generated and monitored cross-traffic in order to obtain a clearer pattern of error percentage as a function of difference between rate at which a train was sent and available bandwidth. Unfortunately we were unable to capture all traffic on the network and were able to monitor traffic only with a certain error. In our setup generated traffic was fluctuating around 50 Mbps. Since there was no 100% confidence in cross-traffic measurements and in order to be on a safe side we monitored trains which were sent at rates either at least 15Mbps higher or 15Mbps lower than tentatively reported available bandwidth. Based on the setup of our experiment we were able with sufficiently high degree of confidence to assume that these train rates were either higher or lower than the available bandwidth. Figure 6 shows an output for this part of the test. Now we

have only one column showing classification errors. For lines where difference between train rate and available bandwidth is negative, the error represents type I error. For lines with positive difference between rate and available bandwidth, the error represents type II error. Figure 6 shows that our approach again outperforms PCT/PDT test for all shown ranges.

VI. CONCLUSION

We proposed a simple algorithm of dealing with the IC/CS effect together with a trend detection algorithm based on a simple linear regression model and p-value reported by t-test. We showed that our algorithm is more robust with respect to IC/CS effect and packet losses. We have chosen a Pathload for comparative analysis because to our knowledge it is the most stable and regularly updated public measurement tool that at the same time is the only tool that addresses the IC/CS phenomenon.

We conducted a series of real measurement experiments in our laboratory network where we were able to significantly influence and tentatively monitor existing cross-traffic and therefore possessed some verification information about available bandwidth behavior. These experiments clearly show better performance of our approach in comparison to PCT/PDT test deployed first in Pathload and later in other probe-rate measurement tools [4], [5].

As a result, with our scheme either the accuracy of any probe-rate based available bandwidth measurement tool can be significantly improved or the measurement time can be reduced at least twice while preserving the same measurement accuracy.

REFERENCES

- [1] J. Strauss, D. Katabi, and F. Kaashoek, "A measurement study of available bandwidth estimation tools," in *IMC*, 2003.
- [2] J. Navratil and R. L. Cottrell, "ABWE: a practical approach to available bandwidth estimation," in *Passive and Active Measurements Workshop*, 2003.
- [3] M. Jain and C. Dovrolis, "Pathload: a measurement tool for available bandwidth estimation," in *Passive and Active Measurements Workshop*, 2002.
- [4] H. Nishikawa, T. Asaka, and T. Takashi, "ABdis: Approach to estimate available bandwidth distribution using a multi-rate probe," in *International Conference on Communication and Broadband Networking (ICBN04)*, 2004.
- [5] C. L. T. Man, G. Hasegawa, and M. Murata, "A new available bandwidth measurement technique for service overlay networks," in *6th IFIP/IEEE International Conference on Management of Multimedia Networks and Services*, 2003.
- [6] R. Prasad, M. Jain, and C. Drovolis, "Effects of interrupt coalescence on network measurements," in *Passive and Active Measurements Workshop*, 2004.
- [7] N. Hu and P. Steenkiste, "Evaluation and characterization of available bandwidth techniques," in *IEEE JSAC Special Issue in Internet and WWW Measurement, Mapping, and Modeling*, 2003.
- [8] V. Ribeiro, M. Coates, R. Riedi, S. Sarvotham, and R. Baraniuk, "Multifractal cross traffic estimation," in *ITC specialist seminar on IP traffic Measurement*, 2000.
- [9] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with tcp throughput," in *ACM/IEEE Transactions on Networking*, 2003.
- [10] S. Ubik and A. Kral, "End-to-end bandwidth estimation tools," Tech. Rep. WPI-CS-TR-03-18, CESNET, November 2003.
- [11] J. A. Rice, *Mathematical Statistics and Data Analysis*. Belmont, California: Duxbury Press, 1995.