Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2020) 28: 1803 – 1823

doi:10.3906/elk-1906-1

FACEST: feedback-assisted estimation of end-to-end capacity in IP-based communication networks

Fatih ABUT*
Department of Computer Engineering, Faculty of Engineering, Çukurova University, Adana, Turkey

Received: 01.06.2019 • **Accepted/Published Online:** 20.03.2020 • **Final Version:** 29.07.2020

Abstract: The end-to-end capacity, defined as the maximal transmission rate of the weakest link on the entire path between two end hosts, plays an important role in efficient network design and management. Although various capacity estimation tools have been proposed in the literature, there is still uncertainty in their accuracy and reliability when they are used in today's IP-based communication networks. The main reason for this is that all current capacity estimation tools only yield a potential candidate for an acceptable estimate, without being aware of its reliability level. In this study, we propose a new feedback-assisted end-to-end capacity estimation (FACEST) procedure that not only produces a candidate for a potentially acceptable estimate but also improves and categorizes its reliability level. Particularly, FACEST follows an ensemble estimation approach which meaningfully utilizes the correlation among the estimates produced by 3 independent capacity estimation tools; namely pathrate, DietTOPP and PBProbe. Through the correlation of 3 individual estimates, additional information about their reliability level is gained and, if necessary, the experiment is iteratively repeated with different sets of measurement parameter values until the required level of estimation accuracy is achieved, or in the worst case a kernel density estimator is applied on the collected experiment results. The proposed ensemble estimation approach has been implemented in a tool called FACEST, the performance of which has experimentally been evaluated on a three-hop testbed using a variety of tests with several scenarios and degrees of cross-traffic. For comparison purposes, individual experiments with pathrate, DietTOPP and PBProbe as well as with other alternative hybrid estimation tool from literature have also been conducted. The results reveal that FACEST outperforms individual and other hybrid capacity estimation tools and yields up to 18.29% lower estimation errors along with additional consistent information about the reliability level of the produced estimates.

Key words: Capacity estimation, ensemble estimation, testbed, network measurement

1. Introduction

The area of bandwidth estimation research attracts researchers' attention for many years, and still developing new estimation techniques steadily gains an increasing popularity. In principle, in bandwidth estimation research there exist 3 major metrics: capacity, available bandwidth and throughput. The capacity states the maximum number of bits per time unit a network link can theoretically transfer. The available bandwidth of a network link is defined as the average residual capacity of that link in a given time period. Finally, throughput, in turn, can be categorized in achievable throughput or bulk transfer capacity (BTC). Achievable throughput is the maximum number of bits per time unit that a link can provide to an application, given the current utilization, the transport protocol and operating system used, and the end-host performance capabilities. In contrast to achievable throughput, which can be measured using different transport protocols and multiple

*Correspondence: fabut@cu.edu.tr

parallel connections, BTC is a TCP specific metric exhibiting the maximal throughput attainable by a single TCP connection. Each of these metrics can be estimated either on the entire path between 2 end-hosts (i.e. at the end-to-end scope) or hop-by-hop. For a formal and detailed definition of these metrics, the interested reader is referred to the respective literature [1–3].

Knowledge and monitoring of bandwidth-related metrics are of great interest for both network operators and end-users as they play a significant role in efficient network management and operation. Acquiring information about capacity can e.g. be useful in validating service level agreements, detecting and bypassing bottleneck links, and performing network tomography to track and visualize Internet topologies. Similarly, knowledge and monitoring of available bandwidth and achievable throughput/BTC can help in diagnosing congested or underutilized links, detecting denial of service attacks, applying admission control policies at massively-accessed content servers, and optimizing network route selection and congestion control mechanisms for reliable transport protocols (e.g. for TCP) [4, 5].

Given the variety of bandwidth-related metrics and motivations, a plethora of techniques and tools for estimating the capacity, available bandwidth and achievable throughput/BTC has been developed in the last 2 decades. Figure 1 shows an overview of some major bandwidth estimation techniques and tools. Packet pair and packet train techniques estimate the end-to-end capacity; packet cartouche technique estimates the bottleneck capacity on a subpath segment consisting of a number of consecutive links of an end-to-end path; variable packet size and packet tailgating techniques estimate the per hop capacities; equally spaced mode gaps technique estimates the capacities of multiple congested links along a path; probe gap model and probe rate model estimate the end-to-end available bandwidth; and finally TCP connection and emulation techniques measure the achievable throughput and BTC. Each estimation technique is represented by several various tool implementations. They show a wide spectrum of different characteristics, such as, among other things, whether they perform the measurement actively or passively, their ability to measure asymmetric links, and the type of their deployment, i.e. whether they are run on one or both end hosts of the path under measure.

In contrast to available bandwidth and throughput metrics, the capacity is independent of the current utilization on the measurement path and does not vary over time. The not time-varying property relatively simplifies the estimation of capacity over the other 2 metrics as there is no compelling constraint on measurement duration and overhead. However, despite these beneficial properties and the significant previous research on estimating the end-to-end capacity, up to now we are still rather far from having reliability-aware accurate estimation procedures. The broad survey revealed that the inaccuracies and unreliabilities in estimating the end-to-end capacity are caused by a variety of different challenges/flaw sources that negatively affect the robust working of an estimation procedure. One of the most common reasons leading to inaccurate and unreliable estimates is that real measurement paths always contain cross-traffic which often disturbs the time gaps of carefully scheduled probing packets. To make the estimation tools cross-traffic resistant, several techniques have been proposed including confidence intervals, kernel density estimators and lower/upper bound filtering techniques. Unfortunately, there is no standard statistical approach that always leads to correct capacity estimates. The main reason making the deal with the cross-traffic difficult is that there exist several types of cross-traffic with different behaviors (e.g. deterministic cross-traffic rates, bursty cross-traffic or cross-traffic obeying to a particular distribution like exponential, Poisson or Pareto distribution) that interfere with an estimation procedure in different ways. In addition to cross-traffic induced disturbances, there are also several other challenges such as interrupt coalescing [6], limited system timer resolution [7], route alternations and multi-channel links [8], traffic shapers [9] or network components working with non-FIFO queuing disciplines [10],

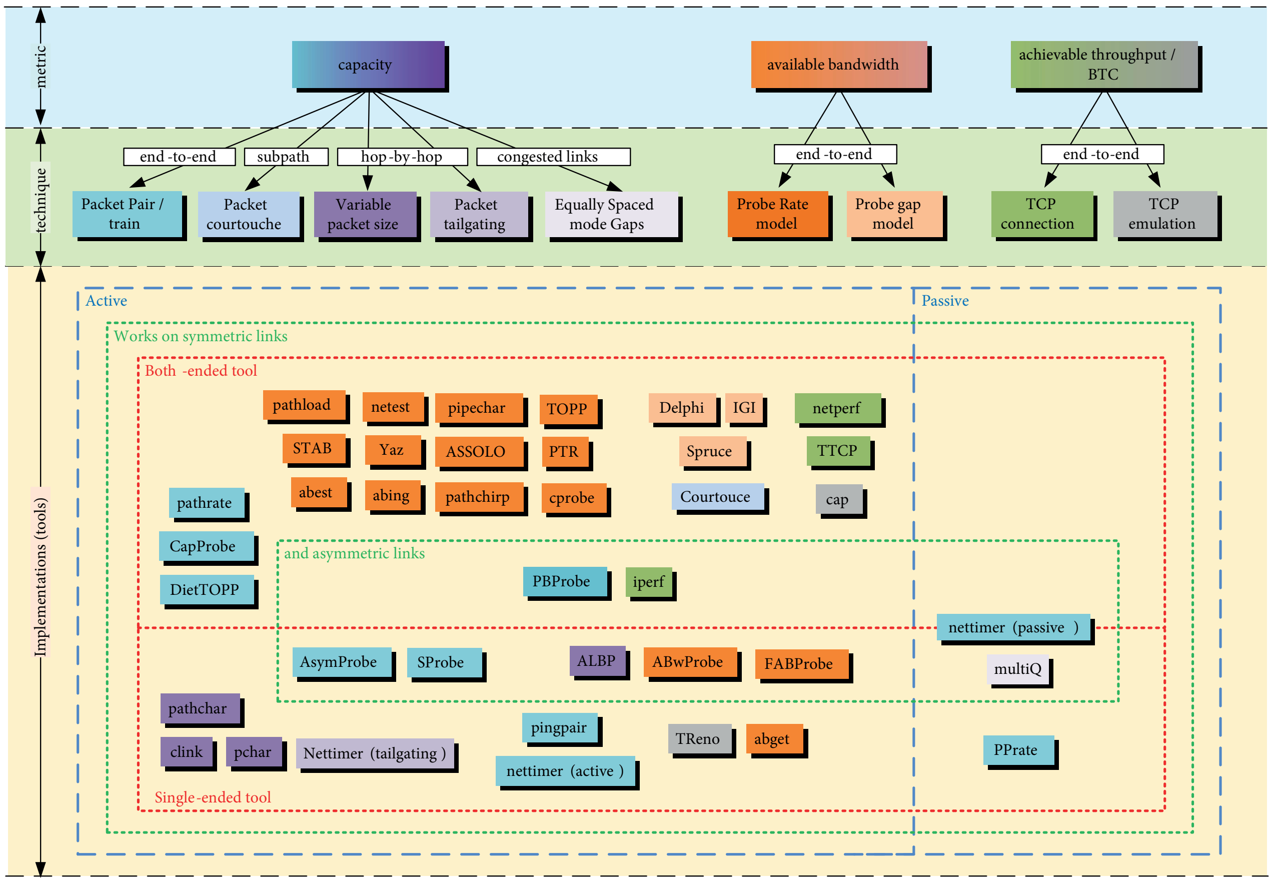


Figure 1. An overview: metrics, techniques and tools.

which make an accurate and reliable estimate of capacity with currently existing techniques a more challenging task. Such practical issues and difficulties negatively affect an estimation procedure and lead to highly unstable and inaccurate estimates, even if the estimation technique is theoretically validated and proved to achieve very accurate estimates.

The existence of so many challenges, practical issues and difficulties in today's network environments thus entails the requirement that an up-to-date capacity estimation procedure should not only yield a candidate for a potentially acceptable estimate, but also additional information about its reliability level. However, the currently existing capacity estimation techniques report their final result in form of an entirely unreliable estimate, without any supporting feedback about its trustworthiness. Consequently, such estimates will not be of significant relevance for most use cases since it cannot be simply distinguished whether the achieved result reflects an acceptable estimate or a highly inaccurate one. For that reason, it is very important to detect such inconveniences and repeat the estimation procedure with different sets of measurement settings instead of reporting a potentially inaccurate estimate, or in the worst case to provide the end-user with a low reliability indicator of the produced estimate.

This study proposes a new hybrid approach based on three individual tools, namely pathrate, DietTOPP and PBProbe, to improve the accuracy and reliability-awareness in estimating the end-to-end capacity in IP-based communication networks. The overall contributions of the paper can be summarized as follows:

- **Overview and classification of existing capacity estimation techniques and tools from related literature:** A comprehensive literature review is presented to survey and classify the major characteristics of existing capacity estimation techniques and tools. Moreover, the challenges, practical issues and difficulties faced by current capacity estimation techniques are outlined.

- **Design of a feedback-assisted and reliability-aware hybrid methodology, implemented in a tool called FACEST, for estimating the end-to-end capacity in wired IP-based communication networks:** The proposed hybrid estimation procedure differentiates itself from all other existing techniques in 2 aspects. First, it uses a feedback-assisted mechanism which leverages the correlation among the consensus properties of 3 individual tools obtained from the receiver to iteratively provide the sender with a feedback until the required level of estimation accuracy is achieved, or in the worst case a kernel density estimator is applied on the collected experiment results. Second, the approach not only produces a candidate for a potentially acceptable estimate but also assesses and categorizes its reliability level.

- **Experimental evaluation of the proposed methodology and a comparative analysis:** The performance of FACEST has experimentally been evaluated on a three-hop testbed using a variety of tests with several scenarios and degrees of cross-traffic. Moreover, comparative evaluations to individual and other hybrid estimation tools from literature have been performed to draw a conclusion about the accuracy and robustness totally gained.

The rest of the paper is organized as follows: Section 2 gives a brief introduction of related works. Section 3 presents the proposed feedback-assisted end-to-end capacity estimation procedure. Section 4 gives information on testbed and evaluation methodology. Section 5 presents the evaluation results and discussion. Finally, Section 6 concludes the paper along with some directions for future work.

2. Related works

In the related literature, in principle there exists 4 basic types of capacity estimation, namely the estimation of end-to-end capacity, the estimation of bottleneck capacity on a subpath segment consisting of a number of consecutive links, the estimation of capacities of each individual hop, and finally the estimation of capacities of multiple congested links along a measurement path. Depending on the type of capacity estimation, several various estimation techniques have been proposed over the past years, such as packet pairs and trains, packet twins, packet triplet, variable packet size, equally spaced model gaps, and other hybrid methodologies based on combinations of these individual estimation techniques.

The packet pair and train techniques estimate the end-to-end capacity of a path between 2 hosts. In other words, the maximum transmission rate of the weakest link on the entire path between 2 hosts is determined. A variety of individual end-to-end capacity estimation tools based on variations of packet pair and train techniques have been proposed including nettimer [11], PPrate [12], SProbe [13], pingpair [14], AsymProbe [15], CapProbe [16], pathrate [17], DietTOPP [18] and PBProbe [19]. Particularly, nettimer uses packet pairs to passively measure the end-to-end capacity along a path in real-time and in both directions, i.e. in upstream and downstream directions. Similarly, pathrate uses many packet pairs to uncover the multimodal bandwidth distribution whereby the challenge is to identify the local modes, and to select the mode that corresponds to the path capacity. SProbe is a fast, scalable and accurate measurement tool that works in uncooperative environments by using the properties of TCP. Another single-ended tool, i.e. pingpair, uses the classical packet dispersion technique, enhanced by a novel algorithm for the selection of the best measurement samples based on queueing delay estimation. AsymProbe, also a "sender only" tool, uses round-trip procedure to estimate capacities on popular asymmetric links (e.g. DSL and satellite links). CapProbe combines delay as well as dispersion measurements of packet pairs to filter out samples distorted by cross-traffic. DietTOPP measures the capacity and available bandwidth of a network path by using the measured dispersion of probe packet trains. PBProbe relies on the concept of "Packet Bulk" to adapt the number of probing packets in each sample in accordance to the dispersion measurement. Finally, PPrate passively extracts capacity information of a path from the packet trace of a TCP connection.

Similar to packet pair probing, the techniques of packet twins [20] and packet triplet [21] also focus on the estimation of the path capacity. The technique of packet twins exploits the different sizes of the twin probe packets, in addition to the traditional parameters like dispersion and delay. The technique of packet triplet, on the other hand, sends multiple 3 back-to-back packets to probe the path, different from packet pair and packet twins, which employ 2 back-to-back probing packets as one probing unit. Both packet twins and packet triplet have been evaluated and validated in simulation only. The variable packet size technique allows measuring the capacities of each individual hop in the path to the receiver. Three popular tools including pathchar [8], pchar [1] and clink [2] implement the VPS technique by associating the round-trip times with several packets of different sizes for inferring the per hop capacities. Finally, the equally spaced model gaps technique, implemented in a tool called MultiQ [22], infers the capacities of multiple congested links for a passively captured TCP flow. In more detail, MultiQ utilizes the equally-spaced mode gaps in TCP flows' packet interarrival time distributions to detect multiple bottleneck capacities in their relative order.

In addition to individual capacity estimation techniques, there are also some studies which have aimed to efficiently combine several different and independent estimation techniques in a hybrid tool. For example, Kang et al. [23] proposed the envelope technique which estimates both the end-to-end capacity and available bandwidth over multi-hop paths by combining a multi-link recursive extension of unbiased single-hop estimators and a variation of the packet cartouche technique. Similarly, Pasztor et al. [24] proposed the technique of packet quartets which combines packet pair with a delay variation based model to estimate capacities over multiple links. Finally, Lin et al. [25] combined the variable packet size and packet tailgating techniques to estimate each hops link capacity in both directions. All these approaches are only implemented and evaluated in simulations under ideal conditions which do not reflect the given details of real network paths.

In the last years, a few real prototype implementations of hybrid approaches have also been proposed. Cong et al. [26] integrated packet pair and extended self-induced congestion principle in a tool named as pathwave to estimate both the end-to-end capacity and available bandwidth based on statistical signal processing theory. Chakravarty et al. [27] proposed linkwidth, a novel tool for estimating the end-to-end capacity and available bandwidth for IP paths. Particularly, linkwidth follows a hybrid approach consisting of both self-loading of periodic streams and train of packet pair techniques to estimate both the capacity and available bandwidth using single-ended control. Man et al. [28] combined packet pair and a modified version of self-loading of periodic streams in a passive tool named ImTCP to enable simultaneous estimation of both end-to-end capacity and available bandwidth in a TCP connection. Harfoush et al. [29] proposed the packet cartouche technique which integrates packet pair with packet tailgating to estimate the bottleneck capacity on a subpath segment consisting of a number of consecutive links of an end-to-end path. Packet cartouche can reveal a portion of the end-to-end path containing the bottleneck link and thus allows identifying characteristics on links which

[1]Mah BA (2001). Pchar [online]. Website http://www.employees.org/~bmah/Software/pchar/ [accessed 30.03.2020].
[2]Downey AB (1999). Clink [online]. Website http://allendowney.com/research/clink/ [accessed 30.03.2020].

are not visible at end-to-end scope. Finally, Lai et al. [30] used a deterministic model of packet delay to derive both the packet pair property of FIFO-queueing networks and a new technique called packet tailgating for actively measuring link capacities by end-to-end measurements.

However, instead of improving the reliability-awareness of individual capacity estimation techniques, all of the studies mentioned above have mainly used the potential of hybrid techniques for other motivations, e.g. for hop-by-hop measurements, simultaneous multi-metric estimations, quick real-time estimations or asymmetric link measurements. In contrast to all these works, this study aims to leverage the potential of hybrid techniques to design a feedback-assisted and reliability-aware estimation procedure that not only yields a candidate for a potentially acceptable capacity estimate but also improves and categorizes its reliability level.

3. Proposed feedback-assisted end-to-end capacity estimation procedure

The feedback-assisted end-to-end capacity estimation (FACEST) algorithm proposed in this study is a hybrid procedure consisting of 3 individual capacity estimation tools including pathrate, DietTOPP and PBProbe, the details of which are described in [17], [18] and [19], respectively. The FACEST algorithm aims to aggregate the consensus properties of pathrate, DietTOPP and PBProbe to produce reliability-aware and more accurate estimates of the end-to-end capacity. The necessary and sufficient condition for ensemble estimation to outperform its individual members is that the combined tools are accurate and diverse. Another critical decision when performing ensemble estimation is the aggregation technique used for combining the resulting estimates from the individual tools into a single decision. Our aggregation technique is based on performing majority voting or repeating the experiments with different sets of measurement parameter values based on a feedback retrieved from the receiver until a consensus is reached or in the worst case selecting the final estimate among all previously collected candidates using a kernel density estimator function.

The main steps of the FACEST algorithm can be summarized as follows. After completing the measurement processes of pathrate, DietTOPP and PBProbe, their final estimates are passed as input parameters to the ensemble capacity estimator. Three steps are followed until a final estimate agreed with at least two of the three individual tools is chosen. The first step involves the majority voting strategy where a consensus among the estimates reported by the 3 individual tools is to be reached. Obviously, the decision criterion for majority voting is fulfilled when at least 2 of the three tools point the same estimate, or a unique consensus among the 3 tools for the same estimate has been observed. In this study, the estimates of 2 tools are categorized as a consensus, when the difference between the 2 estimates is within 3% of the lower estimate. In case of unique consensus, the reliability level of the estimate is set to “high”, whereas in case of majority consensus, “medium” is used as the reliability indicator.

In the other case, where the decision criterion is not fulfilled, i.e. each of the 3 tools reports a different estimate, the algorithm proceeds with the second step. In the second step, measurement experiments are repeated with different sets of measurement parameter values including the length of packet trains of phase 1 and phase 2 for pathrate, the degree of the accuracy for DietTOPP, and the number of experiments for PBProbe. As also reported in the respective tool publications, the correct tuning of these particular parameters of the three tools highly influences the accuracy level of the final estimates. Table 1 shows the selected values of the 3 parameters to be evaluated by the 3 tools, if necessary. In total, 27 different combinations are evaluated to investigate whether an improvement in the estimated results can be achieved. The order of selecting parameter value combinations is arranged so that values leading to relatively shorter measurement duration are evaluated first. As in the first step, depending on the unique or majority consensus, either “high” or “medium” reliability indicator is assigned to the final estimate, respectively.

Table 1. Selected values of measurement parameters for pathrate, DietTOPP and PBProbe. Values marked with an asterisk represent the default values used in stage 1.

Tool	Measurement parameter	Value
Pathrate	Length of packet train for phase 1 (P1) and phase 2 (P2)	P1 = 500*, P2 = 250*
		P1 = 1000, P2 = 500
		P1 = 1500, P2 = 750
DietTOPP	Accuracy degree	5*
		28
		50
PBProbe	Experiment number	100*
		200
		300

If still no consensus is reached after completing the second step, the algorithm terminates with the third step, where the final estimate is chosen among all previously collected experiment results using a kernel density estimator (KDE) function. The rationale in using a KDE is that estimates influenced by cross-traffic will tend not to correlate with each other while the accurate estimates will correlate strongly with each other. This is because it is assumed that cross-traffic will have random packet sizes and will arrive randomly at the links along the path. The equation of utilized KDE function is given in Eq. (1) which calculates the density at a received capacity estimate x as

$$d\left(x\right) = \frac{1}{n}\sum_{i=1}^{n} K\left(\frac{x - x_i}{c * x}\right), \tag{1}$$

where c is the kernel width ratio (selected as 0.10), n is the number of points within $c*x$ of x, and x_i is the ith such point. The utilized kernel function is given in Eq. (2). This function gives greater weight to samples close to the point at which the density is to be estimated [11].

$$K\left(t\right) = \left\{ \begin{array}{ll} 1+t & t \leq 0 \\ 1-t & t > 0 \end{array} \right\}. \tag{2}$$

After applying the KDE function to each capacity sample, the one with the highest density is selected and given as the final estimate along with a "low" reliability indicator. The pseudo-code of FACEST is illustrated in Algorithm.

It is noteworthy that FACEST does not focus on estimating the end-to-end capacity over wireless links which require special estimation procedures with low overhead and resilience to rapidly changing conditions [31–33]. Instead, the focus of FACEST lies on estimating the end-to-end capacity with reliability awareness using an iterative feedback-assisted procedure, with design focus to work on wired IP-based network paths. Furthermore, FACEST is not designed to produce quick real time estimations (e.g. to support its plug-in usage for third-party applications). Rather, the design goal intends to conduct offline analysis of path characteristics. Accordingly, a defined estimation procedure may last several minutes and yield long term average capacity estimates. Finally, the design of the FACEST methodology assumes double-end controlled network paths, i.e. the prototype implementation consists of 2 cooperating components which are deployed on both end-hosts of the paths under measure. Design and development of an estimation procedure that can work on single-end controlled paths is not in the scope of this study.

```
Input: Individual estimates of pathrate, DietTOPP and PBProbe
Output: A new estimate produced by FACEST
    Retrieve the individual estimates of the three tools
    Apply the majority of voting principle on the triple of individual tool estimates
    if (consensus for an estimate is reached)
        then select the final estimates by "majority of votes" and calculate
        their arithmetic average
            if (unique consensus)
                then Reliability = HIGH;
            else //majority consensus
                Reliability = MEDIUM;
            end if
    else //no consensus
        repeat
            Start a new measurement iteration with the next values of measurement
            parameters, as illustrated in Table 1
            if (consensus for an estimate is reached)
                then select the final estimates by "majority of votes" and calculate
                their arithmetic average
                    if (unique consensus)
                        then Reliability = HIGH;
                    else //majority consensus
                        Reliability = MEDIUM;
                    end if
                    FACEST_READY = true;
            end if
        until all value combinations of measurement parameters have been evaluated
        and !FACEST_READY

    if (!FACEST_READY) //still no consensus after the second step
        Select the final estimate among all previously collected candidates using
        kernel density estimator
        Reliability = LOW;
    end if
```

Algorithm. Pseudo-code of the FACEST algorithm.

4. Testbed and evaluation methodology

All evaluation experiments of FACEST have been conducted on the laboratory testbed illustrated in Figure 2. The controlled testbed is based on general purpose PCs equipped with an Intel Core i5-7400 CPU and 8 GB RAM memory running Ubuntu GNU/Linux 18.04 (32 bit). The sender and receiver components of FACEST have been installed on sender and receiver hosts, respectively. For comparison purposes, the same end hosts were also provided with publicly available implementations of individual pathrate, DietTOPP and PBProbe tools, and another alternative hybrid end-to-end capacity estimation tool proposed in literature, i.e. pathwave. The end-to-end path being measured traverses 2 routers. Two other computers are linked together through a gigabit ethernet link and serve as routers. A traffic generator called distributed internet traffic generator (D-ITG) [34] was employed to generate the cross-traffic in order to reproduce different network load conditions. Different hosts are used for cross- and probe-traffic generators in order to avoid interferences that can be caused by CPU overloading or context switching. The sending and receiving hosts running the estimation tools and cross-traffic generators have been linked to the source and destination networks, respectively, via 2 D-LINK DGS-1100-06/ME gigabit switches to make the testbed more realistic. The ethtool Unix utility was used to adjust the bottleneck capacity of the gigabit link between router 1 and router 2 to 100 Mb/s according to the measurement scenario.

In total, 3 different cross-traffic scenarios were considered to evaluate the performance of FACEST. First, UDP cross-traffic has been generated at deterministic rates of 20, 40, 60, and 80 Mb/s. Second, UDP cross-traffic with bursty inter-departure time between packets and constant packet size (i.e. 1500 Bytes) has been generated. Particularly, the on and off period durations of bursty inter-departure time have been controlled by 2 random variables: the former is an exponential with average 100, while the latter is a Weibull with shape 10 and scale 100). Third, UDP cross-traffic characterized by interarrival times modeled as random variables of an exponential distribution (with a mean value of 10 ms), Poisson distribution (with a mean value of 10 ms) and Pareto distribution (with a mean value of 10 ms and shape factor a = 100) have been generated.

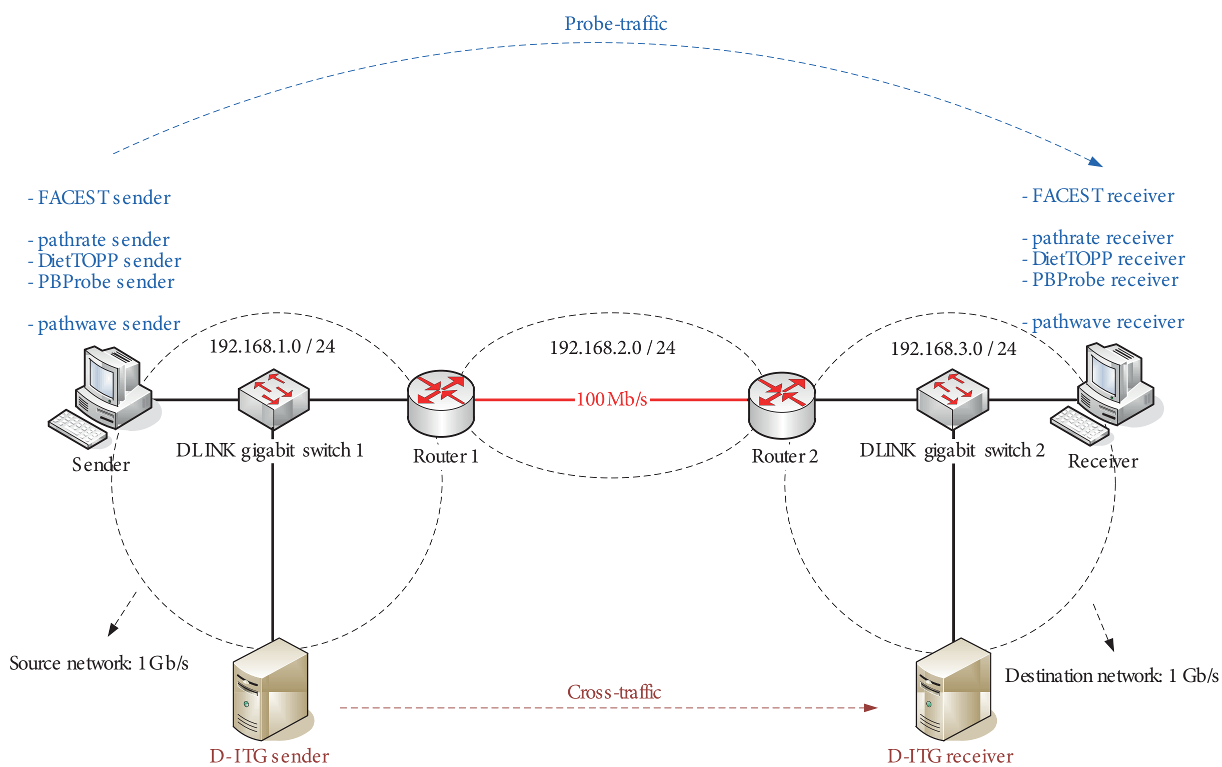


Figure 2. Testbed environment and its components.

The performance of FACEST and other tools has been evaluated in terms of their estimation error e_t. Particularly, let be x_t the estimate achieved by a tool t and *ref_value* the actual value of the end-to-end capacity to estimate. Then, the estimation error e_t of a tool t is calculated as

$$e_t = \left|\left(x_t - ref\_value\right)/ref\_value\right| \times 100\%. \tag{3}$$

According to Eq. (3), smaller estimation error leads to a more accurate result. The reference values for the capacities of fast and gigabit ethernet links on the IP layer correspond to approximately 97.5 Mb/s and 975.0 Mb/s, respectively.

5. Results and discussion

This section includes five subsections. In the first subsection, numerical results of FACEST and the 3 individual tools are illustrated. The discussion regarding the FACEST results are presented in the second subsection. In the third subsection, the results of FACEST are compared with the ones obtained by individually running pathrate, PBProbe and DietTOPP. In the fourth subsection, the results of FACEST are compared with the ones obtained by using pathwave, which in contrast to the rest of individual capacity estimation tools, falls into the same class of hybrid estimation tools as FACEST, and thus enable an additional direct comparison with the proposed approach. Finally, in the fifth subsection, the results and discussion for performance evaluation of FACEST on gigabit links are presented.

5.1. Results

Tables 2 and 3 present the estimation results of FACEST under deterministic cross-traffic rate scenarios, and cross-traffic scenarios characterized by bursty behavior and interarrival times modeled as random variables of various distributions, respectively. For comparison purposes, Tables 4 and 5 show the estimation errors of the 3 individual tools under the same cross-traffic scenarios.

Table 2. Estimation results provided by FACEST, in the presence of cross-traffic with deterministic rates of 20, 40, 60, and 80 Mb/s.

Constant cross-traffic rate (Mb/s)	Estimated end-to-end capacity (Mb/s)	Estimation error (%)	Reliability level	Decision stage
-	97.11	0.42	High	Stage 1: Unique consensus Pathrate: P1 = 500, P2 = 250 DietTOPP: Accuracy = 5 PBProbe: Experiment number = 100
20	96.24	1.31	Medium	Stage 1: Majority consensus DietTOPP: Accuracy = 5 PBProbe: Experiment number = 200
40	98.19	0.69	Medium	Stage 2: Majority consensus Pathrate: P1 = 500, P2 = 250 DietTOPP: Accuracy = 28
60	95.16	2.42	Medium	Stage 2: Majority consensus Pathrate: P1 = 1500, P2 = 750 DietTOPP: Accuracy = 5
80	93.18	4.45	Low	Stage 3: Kernel density estimation

Table 3. Estimation results provided by FACEST, in the presence of cross-traffic characterized by bursty behavior and interarrival times modeled as random variables of exponential, Poisson and Pareto distributions.

Cross-traffic scenario	Estimated end-to-end capacity (Mb/s)	Estimation error (%)	Reliability level	Decision stage
Bursty	96.99	0.54	Medium	Stage 2: Majority consensus Pathrate: P1 = 500, P2 = 250 DietTOPP: Accuracy = 5
Exponential	97.34	0.18	Medium	Stage 2: Majority consensus Pathrate: P1 = 1000, P2 = 500 DietTOPP: Accuracy = 50
Poisson	97.74	0.22	Medium	Stage 2: Majority consensus Pathrate: P1 = 500, P2 = 250 DietTOPP: Accuracy = 28
Pareto	97.82	0.30	Medium	Stage 2: Majority consensus DietTOPP: Accuracy = 28 PBProbe: Experiment number = 300

Table 4. Estimation results provided by pathrate, DietTOPP and PBProbe, in the presence of cross-traffic with deterministic rates of 20, 40, 60, and 80 Mb/s.

Tool	Constant cross-traffic rate (Mb/s)	Estimated end-to-end capacity (Mb/s)	Estimation error (%)
Pathrate	-	97.50	0.02
	20	102.50	5.10
	40	96.00	1.55
	60	122.00	25.10
	80	48.50	50.26
DietTOPP	-	96.32	1.23
	20	95.58	1.98
	40	93.53	4.09
	60	93.58	4.04
	80	90.32	7.38
PBProbe	-	97.51	0.01
	20	82.44	15.46
	40	81.97	15.94
	60	82.08	15.83
	80	81.69	16.23

5.2. Discussion

Based on the results shown in Tables 2 and 3, it is seen that FACEST, in general, provides very low estimation errors, even in the presence of intense cross-traffic scenarios. Particularly, the average estimation errors produced by FACEST under deterministic cross-traffic rate scenarios change between 0.42% and 4.45%. Similarly, the average estimation errors produced by FACEST under bursty and distribution function-based cross-traffic

Table 5. Estimation results provided by pathrate, DietTOPP and PBProbe, in the presence of cross-traffic characterized by bursty behavior and interarrival times modeled as random variables of exponential, Poisson and Pareto distributions.

Tool	Cross-traffic scenario (Mb/s)	Estimated end-to-end capacity (Mb/s)	Estimation error (%)
Pathrate	Bursty	101.50	4.08
	Exponential	101.00	3.56
	Poisson	98.00	0.49
	Pareto	107.00	9.72
DietTOPP	Bursty	94.49	3.10
	Exponential	94.19	3.41
	Poisson	96.01	1.54
	Pareto	94.19	3.41
PBProbe	Bursty	102.61	5.21
	Exponential	106.82	9.53
	Poisson	108.63	11.39
	Pareto	100.05	2.59

scenarios vary between 0.18% and 0.54%. Furthermore, it is seen that on the average estimation errors of FACEST produced under bursty and distribution function-based cross-traffic scenarios are lower than under deterministic cross-traffic rate scenarios. Particularly, the average estimation errors of FACEST under deterministic cross-traffic rate and bursty/distribution function-based scenarios are 2.21% and 0.23%, respectively.

Increasing the deterministic cross-traffic rate to 60 Mb/s and 80 Mb/s, in general, leads to a parallel increase in estimation errors of FACEST. Similar observations regarding the detrimental effects of higher cross-traffic rates on estimation accuracy also apply for the estimates obtained by individually running pathrate, DietTOPP and PBProbe. However, in contrast to the results of individual tools, whose estimation errors range between 4.04% and 50.26%, the estimation errors of FACEST still remain within limits of acceptable accuracy, reaching in the worst case an estimation error of 4.45%.

The results obtained by FACEST show that the decision of a single estimate, marked with a *high* reliability, has been made by unique consensus voting (i.e. in case of ideal measurement scenario with no cross-traffic); 7 estimates, marked with *medium* reliability, have been made by majority consensus voting (i.e. in case of measurement scenarios with deterministic cross-traffic rates equal to 20 Mb/s, 40 Mb/s, 60 Mb/s; and bursty, exponential, Poisson and Pareto cross-traffic); and finally, only a single estimate (i.e. in case of measurement scenario with deterministic cross-traffic rate = 80 Mb/s), marked with *low* reliability, has been decided by applying the KDE function to the collected experiment results. Out of 9 measurement scenarios, 2 scenarios have been decided in stage 1 (i.e. the three tools were executed with their default settings and a consensus could be reached); 6 scenarios have been decided in stage 2 (i.e. various combinations of measurement parameter values were evaluated, and a consensus could be reached); and only a single scenario has been decided in stage 3 (i.e. via kernel density estimation).

5.3. Comparing FACEST with individual estimation tools

In this subsection, the average results of FACEST are compared with the average ones of pathrate, DietTOPP and PBProbe, and it is shown that the performance gain obtained by using FACEST compared to individual tools on estimation of end-to-end capacity is statistically significant.

Table 6 shows the average estimation errors of FACEST, pathrate, PBProbe and DietTOPP under different cross-traffic scenarios. The 3 individual tools were started with their default configuration (i.e. pathrate: P1 = 1000, P2 = 500; DietTOPP: Accuracy degree = 5; PBProbe: Experiment number = 100). It is seen that FACEST consistently provides the lowest estimation errors, regardless of under which cross-traffic scenario the experiments have been conducted. The average estimation error obtained by using FACEST considering all cross-traffic scenarios is 1.26%. The second-best performance in terms of estimation accuracy has been obtained by DietTOPP which gives an average estimation error of 3.62%. The third best performance was exhibited by PBProbe with an average estimation error of 11.52%. The relatively highest average estimation error was generated by pathrate with 12.48%.

Table 6. Comparing the average estimation errors (%) of FACEST with the average ones of individual estimation tools under different cross-traffic scenarios.

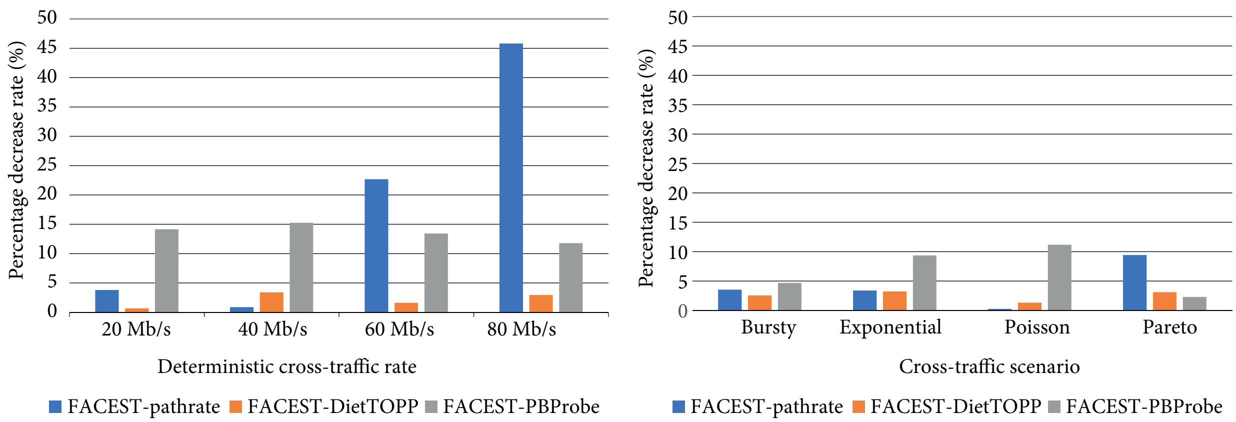
	FACEST	Pathrate	DietTOPP	PBProbe
Deterministic cross-traffic rate scenarios	2.21	20.50	4.37	15.86
Bursty cross-traffic scenarios	0.54	4.08	3.10	5.21
Distribution function-based cross-traffic scenarios	0.23	4.59	2.79	7.84

According to the results given in Table 6, the following comments can be made:

- Compared with the average estimation errors of pathrate, the average estimation errors of FACEST are 18.29%, 3.53%, and 4.35% lower for deterministic cross-traffic rate scenarios, bursty cross-traffic scenarios, and cross-traffic scenarios characterized by interarrival times modeled as random variables of various distributions, respectively.
- Compared with the average estimation errors of DietTOPP, the average estimation errors of FACEST are 2.15%, 2.56%, and 2.55% lower for deterministic cross-traffic rate scenarios, bursty cross-traffic scenarios, and cross-traffic scenarios characterized by interarrival times modeled as random variables of various distributions, respectively.
- Compared with the average estimation errors of PBProbe, the average estimation errors of FACEST are 13.65%, 4.67%, and 7.60% lower for deterministic cross-traffic rate scenarios, bursty cross-traffic scenarios, and cross-traffic scenarios characterized by interarrival times modeled as random variables of various distributions, respectively.

Figure 3 represents the average percentage decrease rates in estimation errors of end-to-end capacity estimation for FACEST compared to estimation errors obtained by pathrate, DietTOPP and PBProbe under different cross-traffic scenarios.

The statistical significance of the performance gain obtained by using FACEST compared to individual feature selectors on estimation of end-to-end capacity has been determined using the well-known Wilcoxon



(a) Deterministic cross-traffic rate scenarios.

(b) Bursty and distribution function-based cross-traffic scenarios.

Figure 3. Percentage decrease rates in estimation errors of end-to-end capacity estimation for FACEST compared to estimation errors obtained by pathrate, DietTOPP and PBProbe under different cross-traffic scenarios.

signed-rank test. More specifically, the test has been applied on the average estimation errors of the 3 pairs including (FACEST, pathrate), (FACEST, DietTOPP), and (FACEST, PBProbe) produced under various experiment scenarios. The sample size of the test case equals to 27 ($n = 27$), and the two-sided level of significance, i.e. α, is set to 0.05. The test statistic for the Wilcoxon signed-rank test is W, defined as the smaller of $W+$ and W-, which are the sums of the positive and negative ranks, respectively. It is to be checked whether the observed test statistic W supports the null or research hypothesis. This check is performed using the critical value of W, which can be found using a pre-defined and well-known table of critical values. The calculated value of W equals 5, and the critical value of W for $n = 27$ at $\alpha = 0.05$ is 92. Since W is less than the critical value, the null hypothesis is rejected and it can be concluded that the performance gain obtained by using FACEST compared to individual estimation tools is statistically significant at $\alpha = 0.05$ for estimation of the end-to-end capacity.

5.4. Comparing FACEST with another hybrid estimation tool

After an exhaustive search of the related literature, we were able to find 3 major studies that proposed alternative hybrid tools for estimating the end-to-end capacity, i.e. pathwave [26], linkwidth [27] and ImTCP [28]. However, to the best of our knowledge, the source codes of linkwidth and ImTCP tools have never been publicly released so that the performance of FACEST could only be compared against the one achieved by using the pathwave tool.

There are 3 major differences between FACEST and pathwave. First, in contrast to pathwave, which aims to produce quick, real-time capacity estimates in only one step, FACEST follows a detailed three-step procedure, focusing on producing reliable-aware and accurate capacity estimates. Particularly, FACEST evaluates the individual tool estimates by taking into accounting (a) the majority voting principle; (b) the different settings regarding various measurement parameters of the individual tools; and finally (c) the selection of the final estimate among several independently collected measurement results using kernel density estimator. Secondly, FACEST is made up of 2 variations of packet train techniques (i.e. pathrate and DietTOPP) and a packet bulk technique (i.e. PBProbe); whereas pathwave incorporates 2 estimation techniques, namely the packet-pair dispersion technique and the self-induced congestion principle. Finally, in pathwave there is no weighting among the estimation techniques. In principle, FACEST also starts with no weighting among the techniques. However, in case the consensus for the final estimate cannot be decided by the first two steps, FACEST allows to dynamically prioritize an estimation tool the estimate of which provides the highest density in the last step.

Tables 7 and 8 show the estimation errors of FACEST and pathwave under various cross-traffic scenarios. The results reveal that compared with the average estimation errors of pathwave; the average estimation errors of FACEST are 2.19%, 4.15% and 2.31% lower for deterministic cross-traffic rate scenarios, bursty cross-traffic scenarios and cross-traffic scenarios characterized by interarrival times modeled as random variables of various distributions, respectively.

Table 7. Comparing the estimation results of FACEST with the ones of pathwave, in the presence of cross-traffic with deterministic rates of 20, 40, 60, and 80 Mb/s.

Tool	Constant cross-traffic rate (Mb/s)	Estimated end-to-end capacity (Mb/s)	Estimation error (%)
FACEST	-	97.11	0.42
	20	96.24	1.31
	40	98.19	0.68
	60	95.16	2.42
	80	93.18	4.45
Pathwave	-	100.46	3.01
	20	100.88	3.45
	40	101.28	3.85
	60	102.68	5.29
	80	102.43	5.03

Table 8. Comparing the estimation results of FACEST with the ones of pathwave, in the presence of cross-traffic characterized by bursty behavior and interarrival times modeled as random variables of exponential, Poisson and Pareto distributions.

Tool	Cross-traffic scenario (Mb/s)	Estimated end-to-end capacity (Mb/s)	Estimation error (%)
FACEST	Bursty	96.99	0.54
	Exponential	97.34	0.18
	Poisson	97.74	0.22
	Pareto	97.82	0.30
Pathwave	Bursty	92.94	4.69
	Exponential	99.74	2.27
	Poisson	100.75	3.31
	Pareto	99.53	2.06

Finally, FACEST and pathwave were also compared in terms of their measurement durations and traffic overhead needed to produce the final estimates. Obviously, due to its iterative and feedback-assisted ensemble nature, FACEST requires longer measurement time and produces higher overhead than pathwave. Particularly, the measurement duration required by FACEST mainly depends on pathrate taking relatively much longer time than DietTOPP and PBProbe, lasting approximately 2 and 53 min for the shortest (i.e. stage 1 decisions) and longest (i.e. stage 3 decisions) experiment scenarios, respectively. When pathwave is used to measure the same path, the measurement lasts about 3 s. Similarly, the total amount of measurement traffic generated by FACEST is approximately 29 MB and 190 MB for the shortest and longest experiment scenarios, respectively. In contrast, pathwave produces 364 KB of measurement traffic. Thus, it can be concluded that compared to pathwave, FACEST estimates the end-to-end capacity with higher accuracy and reliability-awareness, but at the expense of longer measurement duration and higher probing overhead. However, given the fact that capacity values of measurement paths do not change over time and the measurement is performed only once, measurement duration and traffic overhead of FACEST in such magnitudes can be neglected.

5.5. Evaluating the performance of FACEST on gigabit links

In this subsection, the performance of FACEST is evaluated on a path offering gigabit capacity. Particularly, the same testbed setup illustrated in Figure 2 along with the same cross-traffic scenarios was considered, with the 2 differences that the capacity of the link between router 1 and router 2 was increased to 1 Gb/s, and the rates of synthetically generated cross-traffic during the measurements were adapted to the higher path capacity.

Tools for measuring high-speed link capacities, in general, have to cope with 2 major challenges, namely NIC's interrupt coalescing (IC) feature and OS's limited system timer resolution. IC is a well-known and proven technique for reducing CPU utilization when processing high packet arrival rates. Normally, a 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 (i.e. the host is not aware of the packet until the NIC generates an interrupt). Thus, the receiving timestamps for the packets sent by an estimation tool will be distorted (i.e. in such a case, all incoming packets may have the same timestamp) which may lead to erroneous estimations. Furthermore, most estimation tools are based on sending probing packets at a certain transmission rate, i.e. they must send packets in regular intervals to perform a proper measurement. To this end, an estimation 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 achieving the rate $R=s/t$. Unfortunately, the maximum transmission rate obtainable using this approach can also be limited by the insufficient system timer resolution which is 1 μs in most Linux-based OSs. Thus, tools to properly measure high-speed links should incorporate additional mechanisms, such as sending packet trains or bulks, to overcome or mitigate such issues. In the following, it is to be investigated how FACEST performs on gigabit paths under such challenging conditions.

Tables 9 and 10 present the estimation results obtained by applying FACEST on gigabit links under deterministic cross-traffic rate scenarios, and cross-traffic scenarios characterized by bursty behavior and inter-arrival times modeled as random variables of various distributions, respectively. According to these results, it is seen that FACEST also produces acceptable estimation errors when applied to paths offering gigabit capacities. Particularly, the average estimation errors produced by FACEST under deterministic cross-traffic rate scenarios vary from 2.67% to 12.16%. The average estimation errors produced by FACEST under bursty and distribution function-based cross-traffic scenarios change between 1.08% and 5.91%. Analogously to the results obtained by applying FACEST on Fast Ethernet bottleneck links, it is repeatedly confirmed that on the average estimation errors of FACEST produced under bursty and distribution function-based cross-traffic scenarios are lower than under deterministic cross-traffic rate scenarios. In more detail, the average estimation errors of FACEST under deterministic cross-traffic rate and bursty/distribution function-based scenarios are 6.06% and 3.41%, respectively.

Table 9. Estimation results obtained by applying FACEST on gigabit links, in the presence of cross-traffic with deterministic rates of 200, 400, 600, and 800 Mb/s.

Constant cross-traffic rate (Mb/s)	Estimated end-to-end capacity (Mb/s)	Estimation error (%)	Reliability level	Decision stage
-	1013.00	3.89	High	Stage 1: Unique consensus Pathrate: P1 = 500, P2 = 250 DietTOPP: Accuracy = 5 PBProbe: Experiment number = 100
200	948.89	2.67	Medium	Stage 2: Majority consensus Pathrate: P1 = 1000, P2 = 500 PBProbe: Experiment number = 300
400	1026.11	5.24	Medium	Stage 2: Majority consensus Pathrate: P1 = 1500, P2 = 750 DietTOPP: Accuracy = 50
600	1015.76	4.18	Low	Stage 3: Kernel density estimation
800	1093.56	12.16	Low	Stage 3: Kernel density estimation

Similar to previous experiments, the performance of FACEST on gigabit links has also been compared to the ones achieved by individual and hybrid tools. Due to space constraints, not all results can be presented in detail. However, to sum up the major observations, it is seen that FACEST either exhibits comparable performance or outperform the individual tools, regardless of the cross-traffic scenario being evaluated. Particularly, in scenarios with deterministic cross-traffic rates, the average estimation errors of FACEST are 9.26%, 4.68% and 8.29% lower than the ones of pathrate, DietTOPP and PBProbe, respectively. Similarly, in bursty and distribution function-based cross-traffic scenarios, the percentage decrement rates in estimation errors of FACEST are 5.66%, 10.29% and 6.39% compared to pathrate, DietTOPP and PBProbe, respectively. The same experiments and scenarios have also been conducted using pathwave. However, pathwave could often not produce an estimate, and even if so, it yielded highly unreliable and fluctuating estimates with unacceptable error rates. Thus, it has been excluded from the evaluations on gigabit path.

The results obtained by applying FACEST on gigabit path show that the decision of 2 estimates, marked with a *high* reliability, have been made by unique consensus voting; 5 estimates, marked with *medium* reliability, have been made by majority consensus voting; and finally, 2 estimates, marked with *low* reliability, have been decided by applying the KDE function to the collected experiment results. Out of 9 measurement scenarios, one scenario has been decided in stage 1; 6 scenarios have been decided in stage 2; and 2 scenarios have been decided in stage 3. It is noteworthy that although 2 final estimates produced by FACEST under heavy deterministic cross-traffic rate scenarios are marked with *low* reliability, these estimates can still be considered within limits

Table 10. Estimation results obtained by applying FACEST on gigabit links, in the presence of cross-traffic characterized by bursty behavior and interarrival times modeled as random variables of exponential, Poisson and Pareto distributions.

Cross-traffic scenario	Estimated end-to-end capacity (Mb/s)	Estimation error (%)	Reliability level	Decision stage
Bursty	1032.70	5.91	Medium	Stage 2: Majority consensus Pathrate: P1 = 1000, P2 = 500 PBProbe: Experiment number = 200
Exponential	1003.23	2.89	High	Stage 2: Unique consensus Pathrate: P1 = 500, P2 = 250 DietTOPP: Accuracy = 28 PBProbe: Experiment number = 200
Poisson	1011.55	3.74	Medium	Stage 2: Majority consensus Pathrate: P1 = 1000, P2 = 750 DietTOPP: Accuracy = 28
Pareto	964.43	1.08	Medium	Stage 2: Majority consensus DietTOPP: Accuracy = 50 PBProbe: Experiment number = 200

of acceptable accuracy.

Finally, the performance of FACEST on gigabit path has been evaluated in terms of measurement duration and traffic overhead. Particularly, pathrate and DietTOPP perform the measurements with fixed number of experiments, independent of the characteristics of the path under measure. Unlike both tools, the duration of the measurement with PBProbe can vary in length depending on the path capacity being estimated. More specifically, PBProbe adapts the number of probe packets in each sample. For paths with high bottleneck capacities, PBProbe increases the bulk length and sends several packets together. This, in turn, causes a slight increase in the duration and traffic volume generated by FACEST in the order of some additional seconds and MBs, respectively. Consequently, the duration and traffic overhead of FACEST measurements on gigabit links remain approximately in the same order of magnitude as in previous experiment cases conducted on Fast Ethernet links, lasting approximately 2 and 55 min for the shortest (i.e. stage 1 decisions) and longest (i.e. stage 3 decisions) experiment scenarios, respectively. Similarly, the total amount of measurement traffic generated by FACEST ranges from approximately 31 MB to 193 MB, depending on the number of experiments being performed.

6. Conclusion and future work

In this study, we proposed a new feedback-assisted end-to-end capacity estimation procedure that not only yields a candidate for a potentially acceptable estimate but also improves and categorizes its reliability level. Particularly, the proposed ensemble approach, implemented in a tool called FACEST, utilizes the correlation among the estimates given by three independent individual capacity estimation tools; namely pathrate, DietTOPP and PBProbe, to produce more accurate and reliability-aware estimates. Through the correlation of 3 individual estimates, additional information about their reliability level is gained and, if necessary, the experiment is iteratively repeated with different sets of relevant measurement parameter values until the required level of estimation accuracy is achieved, or in the worst case a kernel density estimator is applied to the collected experiment results. The performance of FACEST has experimentally been evaluated on a three-hop testbed using a variety of tests with several scenarios and degrees of cross-traffic. The achieved results show that the feedback-assisted nature of FACEST improves the accuracy and reliability of the produced final estimates. Particularly, compared with the estimation errors of individual tools, FACEST produced statistically significant lower estimation errors achieved under different cross-traffic scenarios. The performance of FACEST has also been compared with pathwave, another alternative hybrid estimation tool from literature which is constructed from the Packet Pair technique and self-induced congestion principle. Again, depending on the considered cross-traffic scenario, FACEST shows its advantages by achieving up to 4.15% performance gain compared to pathwave tool for estimation of end-to-end capacity. Furthermore, the feasibility and capability of FACEST in accurately estimating gigabit link capacities have also been validated.

As future activities, this study can be extended in multiple directions. Additional experiments on further testbeds under different scenarios, including multi-hop and multiple bottleneck conditions, can be conducted in order to further generalize the promising potential of FACEST over the performance of individual methodologies for estimation of end-to-end capacity. Further unevaluated individual end-to-end capacity estimation tools can be incorporated into the set of ensemble algorithms to further optimize the performance of the approach for this research field. Moreover, FACEST can also be extended by integrating accurate available bandwidth estimation tools that can act as lower bound for the capacity estimates. Finally, the scope of FACEST can be enhanced to also support measuring the capacities of wireless links while preserving the feedback-assisted and reliability-awareness features.

Acknowledgment

This work was partially supported by Research Projects Center of Adana Alparslan Türkeş Science and Technology University under grant no 18103001.

References

[1] Abut F. Through the diversity of bandwidth-related metrics, estimation techniques and tools: an overview. International Journal of Computer Network and Information Security 2018; 10 (8): 1-16.

[2] Prosad R, Davrolis C, Murray M, Claffy KC. Bandwidth estimation: metrics, measurement techniques, and tools. IEEE Network 2003; 17 (6): 27-35.

[3] Salcedo D, Guerrero CD, Martinez R. Available bandwidth estimation tools: metrics, approach and performance. International Journal of Communication Networks and Information Security 2018; 10 (8): 580-887.

[4] Chaudhari SS, Biradar RC. Survey of bandwidth estimation techniques in communication networks. Wireless Personal Communications 2015; 83: 1425-1476.

[5] Abut F, Leischner M. An experimental evaluation of tools for estimating bandwidth-related metrics. International Journal of Computer Network and Information Security 2018; 10 (7): 1-11.

[6] Prasad R, Jain M, Dovrolis C. Effects of interrupt coalescence on network measurements. In: Proceedings of Passive and Active Measurement; Antibes Juan-les-Pins, France; 2004. pp. 247-256.

[7] Jin G, Tierney B. System Capability Effects on Algorithms for Network Bandwidth Measurement. In: Proceedings of the Internet Measurement Conference; Miami, Florida, USA; 2003. pp. 27-38

[8] Downey BA, College C. Using pathchar to estimate Internet link characteristics. In: Proceedings of ACM SIGCOMM Conference; Cambridge, Massachusetts, USA; 1999. pp. 241-250.

[9] Kanuparthy P, Dovrolis C. ShaperProbe: end-to-end detection of ISP traffic shaping using active methods. In: Proceedings of the ACM SIGCOMM Conference; New York, USA; 2011. pp. 473-482.

[10] Kazantzidis M, Maggiorini D, Gerla M. Network independent available bandwidth sampling and measurement. Lecture Notes in Computer Science 2003; 2601: 117-130.

[11] Lai K, Baker M. Nettimer: A tool for measuring bottleneck link bandwidth. In: Proceedings of the 3rd conference on USENIX Symposium on Internet Technologies and Systems; Berkeley, CA, USA, 2001. pp. 1-12.

[12] En-Najjary T, Urvoy-Keller G. PPrate: a passive capacity estimation tool. In: Proceedings of 4th IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services; April Vancouver, Canada; 2006. pp. 82-89.

[13] Saroiu S, Gummadi PK, Gribble SD. Sprobe: A fast technique for measuring bottleneck bandwidth in uncooperative environments. In: Proceedings of the Computer and Communications Societies; New York, NY, USA; 2002. pp. 1-11.

[14] Di Pietro A, Ficara D, Giordano S, Oppedisano F, Procissi G. PingPair: A lightweight tool for measurement noise free path capacity estimation. In: Proceedings of IEEE International Conference on Communications; 2008; Beijing, 2008. pp. 1-5.

[15] Jiang W, Williams TF. Detecting and measuring asymmetric links in an IP network. Technical Report CUCS009-99. New York, NY, USA: Columbia University, 1999.

[16] Kapoor R, Chen LJ, Lao L, Gerla M, Sanadidi MY. CapProbe: a simple and accurate capacity estimation technique. In: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications; New York, NY, USA; 2004. pp. 67-78.

[17] Dovrolis C, Ramanathan P, Moore D. What do packet dispersion techniques measure? In: Proceedings of IEEE INFOCOM; Anchorage, AK, USA; 2001. pp. 905-914.

[18] Johnsson A, Melander B, Björkman M, Bjorkman M. DietTOPP: a first implementation and evaluation of a simplified bandwidth measurement method. In: Proceedings of Swedish National Computer Networking Workshop; Karlstad, Sweden; 2004. pp. 1-5.

[19] Chen LJ, Sun T, Wang BC, Sanadidi MY, Gerla M. PBProbe: a capacity estimation tool for high speed networks. Computer Communications Journal 2008; 31 (17): 3883-3893.

[20] Xu J. PacketTwins: a novel method for capacity estimation of a heavy-loaded path. Research Letters in Communications 2009; 2009: 1-4.

[21] Zou ZX, Lee BS, Fu CP, Song J. Packet triplet: a novel approach to estimate path capacity. IEEE Communications Letters 2005; 9 (12): 1076-1078.

[22] Katti S, Katabi D, Blake C, Kohler E, Strauss J. MultiQ: automated detection of multiple bottleneck capacities along a path. In: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement; Taormina, Sicily, Italy; 2004. pp. 245-250.

[23] Kang SR, Liuy X, Bhati A, Loguinov D. On estimating tight-link bandwidth characteristics over multi-hop paths. In: Proceedings of the International Conference on Distributed Computing Systems; Lisboa, Portugal; 2006. pp.1-20.

[24] Pásztor A, Veitch D. Active probing using packet quartets. In: Proceedings of the ACM SIGCOMM Workshop on Internet Measurement; New York, NY, USA; 2002. pp. 293-305.

[25] Lin Y, Wu H, Cheng S, Wang W, Wang C. Measuring asymmetric link bandwidths in Internet using a multi-packet delay model. In: Proceedings of IEEE International Conference on Communications; Anchorage, AK, USA; 2003. pp. 1601-1605.

[26] Cong L, Lu G, Chen Y, Deng B, Li X. pathWave: combined estimation of network link capacity and available bandwidth using statistical signal processing. In: Proceedings of IEEE International Conference on Networks; New Delhi, India; 2008. pp.1-6.

[27] Chakravarty, S, Stavrou, A, Keromytis, AD. LinkWidth: a method to measure link capacity and available bandwidth using single-end probes. Computer Science Department. Technical Report CUCS-002-08, Columbia University, 2008.

[28] Man CLT, Hasegawa G, Murata M. A merged inline measurement method for capacity and available bandwidth. In: Proceedings of Passive and Active Measurement; Boston, MA, USA; 2005. pp. 341-344.

[29] Harfoush K, Bestavros A, Byers J. Measuring bottleneck bandwidth of targeted path segments. In: Proceedings of IEEE INFOCOM; San Francisco, CA, USA; 2003. pp. 2079-2089.

[30] Lai K, Baker M: Measuring link bandwidths using a deterministic model of packet delay. In: Proceedings ACM SIGCOMM; Stockholm, Sweden; 2000. pp. 283-294.

[31] Yang T, Jin Y, Chen Y, Jin Y. RT-WABest: a novel end-to-end bandwidth estimation tool in IEEE 802.11 wireless network. International Journal of Distributed Sensor Networks 2017; 13 (2): 1-11.

[32] Aina F, Yousef S, Osanaiye O. RAAC: a bandwidth estimation technique for admission control in MANET. Electronics and Energetics 2019; 32 (3): 463-478.

[33] Nyambo B, Janssens G, Lamotte W. Bandwidth estimation in wireless mobile ad hoc networks. Journal of Ubiquitous Systems and Pervasive Networks 2015; 6: 19-26.

[34] Botta A, Dainotti A, Pescapè A. A tool for the generation of realistic network workload for emerging networking scenarios. Computer Networks 2012; 56 (15): 3531-3547.