On the Accuracy of Active Capacity Estimation in the Internet

Simon Bauer, Janluka Janelidze, Benedikt Jaeger, Patrick Sattler, Patryk Brzoza, Georg Carle
Technical University of Munich (TUM), Department of Informatics
Chair for Network Architectures and Services, Garching b. München, Germany
{bauer | janelidze | jaeger | sattler | brzoza | carle}@net.in.tum.de

***Abstract*—Estimating the capacity of network paths is a frequently and versatilely used technique for network and flow analysis used by service providers and researchers to analyze available bandwidth, performance limitations of connections, or infrastructure deployments. While researchers evaluated different capacity estimation approaches in the early 2000s, there are no recent studies on the accuracy of estimates and capacity deployments in today's Internet.**

This paper is purposed to survey the accuracy of actively conducted capacity estimation in today's Internet. We implement passive packet pair dispersion-based capacity estimation according to the PPrate algorithm and conduct active measurements with TCP and ICMP traffic on controlled targets in the Internet and on public web servers to analyze the accuracy and stability of estimated capacities in the Internet.

Our study confirms the general accuracy of PPD-based measurements through the Internet while we observe and discuss impacts by interrupt coalescence, receive offloading, and ICMP rate limiting of middleboxes. Measurements to over 3500 web servers taken from the Alexa top 1M list indicate capacities of at least 1 Gbit/s for the majority of paths to measurement targets, while ICMP-based measurements frequently result in significant underestimation due to ICMP rate limiting.

***Index Terms*—Capacity, Packet-pair dispersion, Internet**

I. Introduction

Capacity is a metric of major importance for service and network providers applied for analysis of capacity deployments, measurements of available bandwidth, or root cause analysis of flow rates. Accordingly, capacity estimation is frequently addressed, resulting in different approaches and corresponding implementations. Packet-pair dispersion (PPD) is one of the most common capacity estimation approaches enabling single-ended, double-ended, and passive measurements, relying on inter-arrival times (IATs) between consecutively observed packets. However, capacity estimates might be falsified by interference with cross-traffic and queuing, as IATs of an analyzed flow potentially get compressed or expanded by cross-traffic packets.

Considering the evolution of the Internet, implying larger traffic volumes, increasing flow rates, and evolved queuing mechanisms, there are no recent studies on the accuracy of PPD-based capacity estimation in the Internet.

This paper pursues the question of how accurate PPD-based capacity estimation can be conducted in today's Internet, considering TCP download traffic and responses to bursts of ICMP Echo requests. To assess accuracy based on ground truth data, we conduct measurements in physical test environments and with controlled targets in the Internet. Further, we run measurements to public Internet targets to survey deployed capacities. In addition to the accuracy of estimates, we survey the trade-off between accuracy and intrusiveness, the consistency of estimates conducted on the same Internet path, and the responsiveness of routers to ICMP-based measurements.

The remainder of this paper is structured as follows: Section II provides background regarding capacity estimation followed by an overview of related work in Section III. Section IV introduces applied measurement approaches and setups. Implemented capacity estimation and the measurement framework used for hybrid Internet measurements are described in Section V. Results of measurements conducted in controlled test environments are presented in Section VI, while we analyze results of conducted Internet measurements in Section VII.

II. Background

A. Terminology

We refer to capacity as the minimum physical bandwidth of a network path, which corresponds to the maximum achievable transmission rate. Considering a network path $\mathcal{P}$ consisting of n hops h_i between the sender $\mathcal{S}$ and the receiver $\mathcal{R}$, i.e., $\mathcal{P} = (h_1, ..., h_n)$, we define C_i as the capacity of the path up to hop h_i. The capacity C of path $\mathcal{P}$ is then determined by the hop providing the smallest capacity C_i, i.e., $C_{\mathcal{P}} = \min\{C_1, ..., C_n\}$. We refer to the link with the smallest C_i as the narrow link of the path $\mathcal{P}$.

B. PPD-based Capacity Estimation

PPD-based capacity estimation assumes a packet pair (p_0, p_1) of packet size L to be sent consecutively without delay between each other. The inter-arrival time (IAT) between p_0 and p_1 measured at the receiver $\mathcal{R}$ is then referred to as dispersion. With the time interval between p_0 and p_1 at h_i referred to as Δ_i and $\Delta_1 = \frac{L}{C_1}$, $\Delta_i = \max(\Delta_{i-1}, \frac{L}{C_i})$. Accordingly, the narrow link determines the dispersion observed at $\mathcal{R}$ and allows to calculate the capacity of a path as $C_{\mathcal{P}} = \frac{L}{\Delta_{narrow\_link}}$, as illustrated in Figure 1.

Researchers introduced different tools exploiting the advantages of PPD-based capacity estimation, like the independence of the used L4 protocol, single-ended measurements, or utterly passive capacity estimation conducted on captured network

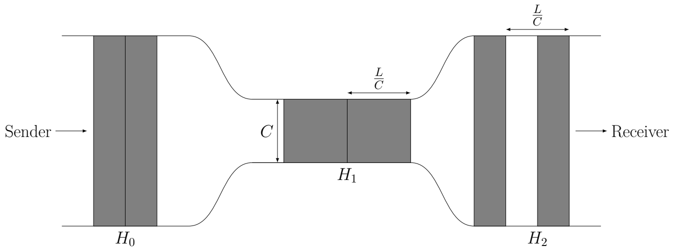


Fig. 1: PPD-based approach to capacity estimation.

traffic. PPD-based capacity estimation is sensitive to interference with cross-traffic and corresponding queuing effects. In particular, interference with cross-traffic potentially implies either compression or expansion of IATs between packet pairs implying falsified estimates.

In addition to PPD-based capacity estimation, variable packet size (VPS) probing is a frequently considered approach to capacity estimation [1], [2]. VPS-based approaches enable estimating capacities for single hops of a network path achieved by actively probing the capacity of each hop on the path. However, VPS-based estimates require active measurements and are sensitive to delays by store-and-forward devices [3].

As scalable measurements with reduced intrusiveness are of major interest for future research projects and PPD-based estimation is applicable more flexibly, we focus on PPD-based capacity estimation in this paper.

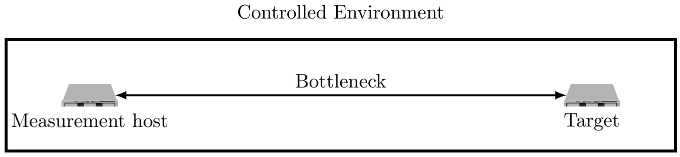
III. Related Work

Different capacity estimation tools were introduced and compared throughout the years. Dovrolis et al. [4] introduce the active and double-ended tool *pathrate*, relying on packet pair dispersion, packet train dispersion, and probing with different packet sizes. Thereby, *pathrate* focuses on accuracy instead of scalability or estimation duration [5]. Other active, PPD-based tools like *AsymProbe* [6] or *SProbe* [7] only require single-ended deployments, which enable measurements in uncooperative environments. Thereby named tools are purposed to enable estimates on asymmetric network paths, respectively, to reduce estimation duration. The tools *CapProbe* [8] and *PBProbe* [9] rely on combining PPD measurements with delay measurements to reduce cross-traffic impacts. Other tools, like *clink* [2] and *pathchar* [10], conduct active measurements with VPS probing.

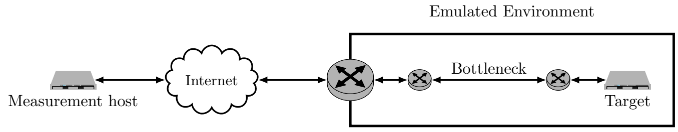
Regarding passive estimation based on PPD, the tools *nettimer* [11], *PPrate* [12], and *TraceProbe* [13], were introduced. A comparison of *PPrate* to *pathrate* shows comparable accuracy [12]. Another passive capacity estimation tool is *MultiQ* [14], based on so-called equally-spaced mode gaps. In addition to estimating narrow link capacity, *MultiQ* determines tight links, i.e., links providing minimum available bandwidth on a path. A comparison of *MultiQ* to *pathrate* and *nettimer* reveals comparable results between *MultiQ* and *pathrate*, while *MultiQ* is shown to be more accurate than *nettimer* [14]. Another study conducts a comparison of the three passive tools *MultiQ*, *nettimer*, and *PPrate* to the active tool *pathrate* based on PlanetLab paths and traces captured at an ADSL access network [15]. The study reveals *PPrate* to be the most accurate passive tool in most cases. Further, researchers introduce capacity estimation with a focus on wireless networks [16] and integration to the data plane to provide timely estimates for an active connection [17].

Capacity estimation is applied in the context of TCP root cause analysis, relying on capacity to detect unshared bottleneck limitations [18]–[20]. Other publications survey measurements of bandwidth-related characteristics of Internet paths and connections, such as available bandwidth [21]–[27].

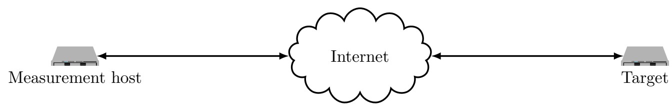
For our study, we surveyed the availability of listed capacity estimation tools to consider them for our measurements. We find that only *pathrate* is available and maintained [5]. While *clink* is also available, first measurements with the tool in controlled test environments did not result in any meaningful results. *PBProbe* is also available for download but did not compile on different tested operating systems. Other tools either provide broken references to corresponding sources or were not published.



(a) Measurement setup in controlled environments.



(b) Setup for measurements with hybrid Internet paths.



(c) Setup for measurements with public target servers.

Fig. 2: Considered measurement setups.

IV. Approach

This paper is purposed to survey characteristics of PPD-based capacity estimation in the Internet, like accuracy, robustness, and required intrusiveness of active measurements. We refer to accuracy as the deviation of an estimate to the actual capacity, while robustness describes the consistency of several estimates conducted on the same path, assuming constant capacity and varying conditions on the network path. Further, we refer to intrusiveness as the amount of data, respectively the number of packets, used for an estimate.

To estimate capacities, we implement a tool according to the *PPrate* algorithm. We choose *PPrate* as it is assumed to be robust regarding outlying dispersion pairs, allows flexible traffic generation for active measurements, and enables large-scale passive measurements. As *PPrate* is a passive tool, traffic has to be generated separately to conduct active Internet measurements. For our measurements, we consider generating TCP traffic by requesting files via HTTP and sending bursts of ICMP Echo requests to capture corresponding Echo replies. For our study, we consider three measurement setups described in the following.

a) Evaluation in Controlled Environments: First, we conduct measurements in controlled test environments to validate implemented measurement tools, as shown in Figure 2a. Further, such measurements are motivated by surveying potential performance limitations of the used tools considering capacities of $1\,\mathrm{Gbit/s}$, respectively $10\,\mathrm{Gbit/s}$.

b) Measurements on Hybrid Internet Paths: Second, we conduct measurements on paths through the Internet with controlled endpoints and configured first, respectively, last hop capacity, as illustrated in Figure 2b. These hybrid setups allow measurements with ground truth data to assess accuracy, while analyzed traffic is expected to be affected by varying conditions on the Internet paths. For our measurements, we use paths between four different virtual nodes hosted in data centers in Munich (MUC), Helsinki (HEL), San Francisco (SF), and Singapore (SG), resulting in a total of 12 paths considering both directions between all nodes.

c) Measurements with Public Internet Targets: Third, we conduct measurements with public and uncontrolled Internet servers to survey capacity deployments in the Internet as shown in Figure 2c. As conducting measurements with uncontrolled targets requires traffic generation without access to the server side, this approach requires crawling for available files to conduct TCP-based downloads. Traffic generation with ICMP Echo requests remains straightforward, assuming that targets reply reliably.

d) Ethical Considerations: Active measurements on public infrastructure like the Internet require responsible measurement practices. The IP address of the measurement host used for crawling and measurements with public targets can be resolved to the website of our research group. We maintain an abuse contact email and react quickly to all requests. We use a custom HTTP user agent to be identifiable as a research group during crawling target files and conducting downloads.

V. Implementation

This section introduces implemented capacity estimation and the framework used for ground truth measurements and for conducting Internet measurements. Implemented capacity estimation according to the *PPrate* algorithm is available as free and open source [28].

A. Capacity Estimation

a) Passive Capacity Estimation based on the PPrate Algorithm: To the best of our knowledge, no PPD-based capacity estimation tool is publicly available nowadays. Therefore, we implement a tool according to the *PPrate* algorithm introduced by En-Najjary and Urvoy-Keller [12] in Python. *PPrate* is a passive capacity estimation tool applying filtering and analysis of inter-arrival times (IATs) between packets. Our tool expects a traffic capture or a list of inter-arrival times and a connection's maximum segment size (MSS) as input. En-Najjary and Urvoy-Keller propose to filter outlying IATs to prevent impacts by cross-traffic interference, the application layer, or other throughput limitations like flow control. In particular, *PPrate* filters IATs larger than $min(P_{75}+IQR, P_{95})$ and smaller than $P_{25}-IQR$ with IQR as the inter-quartile range and P_x as the x-th percentile of all observed IATs.

The algorithm determines a histogram of capacity probes calculated based on observed IATs δt and the MSS by the formula $C = \frac{MSS}{\delta t}$. The histogram is generated based on the configured capacity resolution, i.e., the used bin width for the histogram. As proposed by En-Najjary and Urvoy-Keller, we choose a bin width of $5\,\%$ of the IQR of observed IATs [12]. Afterward, the histogram is analyzed for modes. If a single mode is detected, it directly determines the final capacity estimate. If there are several modes, the *PPrate* algorithm tries to determine the so-called Asymptotic Dispersion Rate (ADR) by eliminating less significant modes by calculating the dispersion between trains of packets, i.e., between sequences of n packets. Thereby, n is increased until the capacity probe histogram shows a single mode corresponding to the ADR. Then, capacity is estimated by the first mode larger than the ADR in the capacity probe histogram of single packet pairs.

Based on observations during our measurements, we extend the *PPrate* algorithm by two features. First is an option to skip the second phase of mode determination and to directly choose the most significant bin of the capacity probe histogram as an estimate. Second, our implementation supports the use of actual packet sizes instead of the MSS of a connection. Using observed packet sizes is purposed to counteract the effects of receive offloading, which implies the composition of several TCP packets to a single packet by the network interface card (NIC) and, accordingly, larger packet sizes observed by the operating system. We determine packet sizes as the Ethernet frame size.

b) Traffic Generation for Active ICMP Measurements: While active, single-ended, and TCP-based capacity measurements require files to be downloaded to generate traffic, the usage of ICMP traffic implies more flexibility regarding measurement targets. Further, relying on ICMP Echo requests enables conducting measurements in a traceroute-like manner, i.e., estimating capacity from the measurement host to each hop on the network path. Therefore, we are interested in the suitability of ICMP traffic, i.e., ICMP Echo requests and corresponding Echo replies, for PPD-based capacity estimation.

One crucial aspect of active ICMP-based measurements is the generation of ICMP Echo requests, as the intervals between two requests imply an upper bound to estimated capacities. For instance, estimating a capacity of $1\,\mathrm{Gbit/s}$ requires sending ICMP Echo requests with intervals of at least $12\,\mu\mathrm{s}$ assuming packet sizes of $1500\,\mathrm{B}$, as $\delta t = \frac{L}{C} = \frac{1500\,\mathrm{B}}{1\,\mathrm{Gbit/s}} \approx 12\mu s$. Therefore, we implement multi-threaded ICMP traffic generation in C based on preloading Echo requests in memory to ensure

	1 Gbit/s		10 Gbit/s	
	IC enabled	IC disabled	IC enabled	IC disabled
TCP	$-3.36\,\%$	$-1.35\,\%$	$-6.59\,\%$	$1.43\,\%$
ICMP	$70.39\,\%$	$1.84\,\%$	$-76.69\,\%$	$-80.55\,\%$

TABLE I: Mean relative errors measured in physical setups.

minimal IATs between sent requests.

B. Framework for Internet Measurements

As ground truth data regarding Internet paths' capacity is not available in general, we implement a distributed framework based on a client-server architecture enabling configuring a path's first and last hop capacities. The client is responsible for measurement orchestration and traffic initiation, while the server component a) runs a publicly reachable web server providing a file for downloads and b) responds to ICMP Echo requests.

Both the client and the server component can set up *Mininet* [29] topologies, which are connected to the Internet. In particular, the client- and server-side topology consists of a router, two switches, and a host, which is used as the measurement target when the framework runs in server mode. The router runs in the root namespace of the used measurement host and acts as a gateway between external networks, like the Internet, and the Mininet topology. The link between the two switches is used to configure bottleneck capacities with *Mininet's TCLink* class. Address translation between the internal and external network is implemented with the *iptables* utility. In client mode, the framework establishes downloads to a specified server, sends trains of ICMP Echo requests, conducts a traceroute measurement before each estimation run, and captures traffic with *tcpdump*. For traceroutes, we rely on *dublin-traceroute* [30] providing detection of load balancing and NATing. When running the framework in server mode, a web server, implemented with *webfsd*, is set up on the host in the *Mininet* topology and provides a downloadable file of configurable size for TCP-based capacity measurements, respectively, replies to received ICMP Echo requests.

VI. Measurements in Controlled Test Environments

We conduct measurements in controlled physical test setups providing capacities of $1\,\mathrm{Gbit/s}$, respectively, $10\,\mathrm{Gbit/s}$, to assess the accuracy of estimates regarding high capacities. For our measurements, we use two pairs of commodity hardware servers. Each of the hosts of the first pair is equipped with an Intel Xeon D-1518 CPU providing four physical cores, $32\,\mathrm{GB}$ of memory, and an Intel I210 NIC. Both hosts are interconnected by a $1\,\mathrm{Gbit/s}$ link. The second pair's hosts are equipped with Intel Xeon E31230 CPUs providing four physical cores, $16\,\mathrm{GB}$ of memory, and Intel X540-AT2 NICs, interconnected by a $10\,\mathrm{Gbit/s}$ link. We run the distributed measurement framework, as described in Section V, without *Mininet* topologies. We repeat measurements ten times for each setup and calculate the mean relative error of all estimates, as shown in Table I.

For TCP-based measurements we observe signed mean relative errors of $-3.3\,\%$, respectively $-6.6\,\%$, indicating underestimation. In the case of ICMP-based measurements, we find significantly larger mean relative errors of $70.3\,\%$, respectively $-76.6\,\%$. We trace back measurement errors of ICMP-based estimates in the $1\,\mathrm{Gbit/s}$ setup to active interrupt coalescence (IC), which is enabled on the used NICs by default. IC is purposed to reduce the computational load in interrupt-based network stacks, by only performing interrupts when a certain number of packets was received or after a specific time interval. IC is known to have impacts on network measurements as surveyed by Salehin et al. [31] and impacts IATs between packets when time stamping of packets is done in software. However, TCP-based estimates do not suffer from a comparable significant impact of IC. We find that captured TCP packet sizes are significantly larger than the MSS. This observation can be explained by generic receive offloading (GRO), which implies received TCP packets being merged by the NIC, resulting in larger segments, before interrupting the operating system.

Repeating measurements with disabled interrupt coalescence results in decreased mean relative errors for both kinds of generated traffic in the $1\,\mathrm{Gbit/s}$ setup, where ICMP-based relative errors improve significantly down to a mean of $1.8\,\%$. Measurements with ICMP traffic in the $10\,\mathrm{Gbit/s}$ setup still show significant relative errors with a mean of $-80.5\,\%$. This observation can be traced back to the implemented ICMP load generation, as discussed in Section V. Estimating a capacity of $10\,\mathrm{Gbit/s}$ requires IATs to be smaller than 1.2 μs. However, we observe that ICMP packets are generated with a mode around 5.5 μs in the $10\,\mathrm{Gbit/s}$ setup, while $C = \frac{1500\,\mathrm{B}}{5.5\mu s} \approx 2.2\,\mathrm{Gbit/s}$ implying significant underestimation.

VII. Internet Measurements

As described in Section IV, we conduct Internet measurements with controlled and uncontrolled measurement targets to assess the accuracy of capacity estimates and to survey capacity deployments in the Internet.

A. Measurements on Hybrid Internet Paths

For measurements with two controlled endpoints, we set up the measurement framework in client and server mode on the virtual measurement hosts introduced in Section IV. The narrow link is configured at the first hop on the path from the server to the client, while data packets, respectively ICMP Echo replies, are sent from the server, pass the emulated narrow link, the Internet path, and get captured at the client. We conduct downloads of a $10\,\mathrm{MB}$ file for TCP-based measurements and generate bursts of 2000 Echo requests with a packet size of $1500\,\mathrm{B}$ for ICMP-based measurements. For each direction of the six paths between the four measurement hosts, we conduct measurements for 48 hours. We configure 9 different capacities from $10\,\mathrm{Mbit/s}$ up to $200\,\mathrm{Mbit/s}$ and conduct seven measurements back-to-back for each capacity

Path	TCP				TCP (adapted algo.)				ICMP			
	$C \leq 150\,\mathrm{Mbit/s}$		$C \leq 200\,\mathrm{Mbit/s}$		$C \leq 150\,\mathrm{Mbit/s}$		$C \leq 200\,\mathrm{Mbit/s}$		$C \leq 150\,\mathrm{Mbit/s}$		$C \leq 200\,\mathrm{Mbit/s}$	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median
All	6.60	5.76	44.80	5.20	-1.47	-0.86	35.81	-0.92	521.89	-0.34	$> 1K$	0.04
All \ SG	7.36	6.51	7.21	6.50	-1.15	-0.85	-1.50	-0.92	-0.61	-0.70	994.55	-0.58
HEL ↔ MUC	6.02	5.43	5.78	5.42	-0.78	-0.76	-1.15	-0.79	-0.74	-0.79	481.05	-0.75
SF ↔ MUC	7.11	7.12	6.66	7.12	-0.84	-0.80	-1.08	-0.87	0.01	-0.39	$> 1K$	-0.02
SF ↔ HEL	9.07	7.55	9.30	7.55	-1.85	-1.34	-2.28	-1.59	-1.05	-0.77	750.46	-0.69
MUC ↔ SG	5.20	2.75	156.40	2.08	-1.59	-0.95	150.99	-0.97	438.53	1.08	$> 1K$	2.70
SF ↔ SG	5.20	2.20	85.08	1.26	-1.03	-0.84	75.00	-0.89	$> 1K$	2.93	$> 1K$	5.66
HEL ↔ SG	6.78	6.11	45.06	3.60	-3.43	-0.87	33.33	-0.93	788.44	0.12	$> 1K$	0.73

TABLE II: Mean and median of relative errors in % for capacities estimated during hybrid measurements.

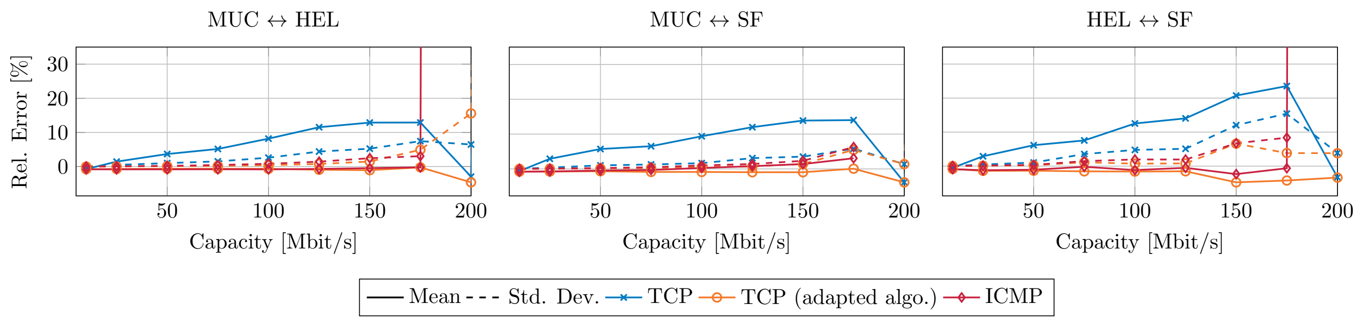


Fig. 3: Relative error of estimates measured on hybrid paths.

before the next capacity is configured. This procedure is repeated for 48 hours, resulting in over 60 iterations, each consisting of seven estimates per capacity for both kinds of traffic.

As we observe impacts by interrupt coalescence for single estimates of larger capacities, resulting in several magnitudes of overestimation, we further differentiate between two sets of estimates. A set of estimates for capacities smaller or equal than $150\,\mathrm{Mbit/s}$ and a set consisting of all estimates, i.e., for capacities smaller or equal than $200\,\mathrm{Mbit/s}$. Results for both sets are presented in Table II.

For capacities $\leq 150\,\mathrm{Mbit/s}$ we observe a mean relative error of $6.6\,\%$ for TCP-based measurements and $521.9\,\%$ for ICMP-based measurements considering all paths. Such larger errors for ICMP-based estimates and the difference between both kinds of traffic can be explained by receive offloading applied for TCP but not for ICMP. Mean relative errors for capacities $\leq 200\,\mathrm{Mbit/s}$ are even more significant as the impact of outliers due to interrupt coalescence increases. Further, measurements on paths including the measurement host in Singapore show larger errors than the other paths regarding larger capacities. Mean relative errors without such paths decrease drastically for ICMP, resulting in a mean relative error of $-0.61\,\%$. In general, medians of relative errors are more robust to outliers and, therefore, do not show such significant error rates as observed for the means of relative errors. However, we still observe median relative errors larger than $5\,\%$ for TCP-based estimates.

As we observe more significant errors for TCP-based measurements than ICMP-based measurements, we analyze the histograms of capacity probes for both kinds of traffic. We find that our implementation of the *PPrate* algorithm does not detect uni-modal distributions for the vast majority of TCP-based measurement runs and observe that the ADR already approximates the actual capacity. Accordingly, the first mode larger than the ADR implies overestimation. Therefore, we adapt our implementation to directly estimate capacity based on the bin in the capacity probe histogram containing the most samples. As shown in Table II the adapted version of *PPrate* significantly decreases means and medians of relative errors for TCP-based measurements. Further, we are interested in the stability of estimates and the corresponding deviations of relative errors. Figure 3 shows the mean relative error of estimates and the standard deviation for selected paths. Note that we filter outliers larger than 10K for plotting. We observe that relative errors for different capacities are quite stable for ICMP-based and TCP-based measurements analyzed with the adapted *PPRate* algorithm. The same applies to the corresponding standard deviation of errors, which is mostly smaller than $5\,\%$.

In addition to the accuracy of estimates in general, we are interested in the impact of the number of analyzed packets, respectively bytes, on estimates. Therefore, we analyze the so-called intrusiveness score I_x comparing the estimate after the first x captured packets, respectively, bytes, to the estimate considering all data, i.e., $I_x = \frac{est_x}{est_n}$. While packets are a suitable metric for ICMP-based measurements, bytes are more reasonable for TCP-based measurements, as packet numbers are skewed due to merged packets when receiver offloading is applied. We find that deviations of ICMP-based measurements converge quickly to the final estimate considering all 2000 packets, with a significant decrease of deviations between

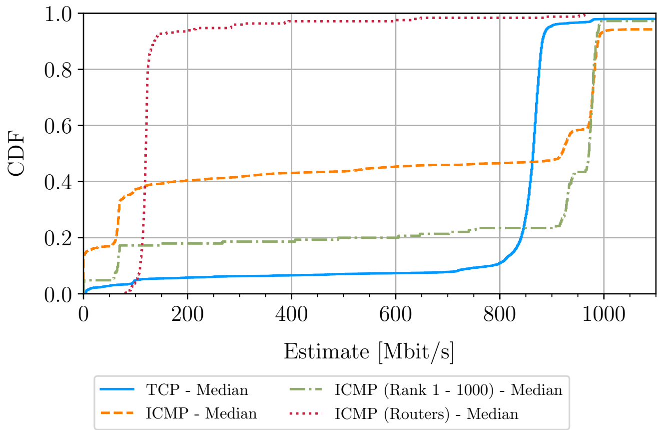


Fig. 4: Distribution of estimated capacities.

estimates after 100 and 200 packets. For TCP, we observe an ongoing decrease of I_x for all considered amounts of data, while outliers are reduced significantly during the first $1.5\,\mathrm{MB}$ Deviations to the final estimate remain smaller than $5\,\%$ for estimates considering at least $1.5\,\mathrm{MB}$ of captured data.

B. Blackbox Measurements

As measurements with ground truth data confirm accurate capacity estimation across Internet paths, we conduct measurements on web servers behind domains chosen from the Alexa Top 1 Million list [32] to survey capacity deployments and the stability of estimates. We crawl 15K sampled entries of the Alexa top list for static files to be downloaded for TCP-based estimations. We only consider files larger than $1\,\mathrm{MB}$ to ensure a sufficient amount of packets per download. Crawling results in over 3500 suitable target servers. Such servers are also used as targets for measurements based on ICMP Echo requests, while we generate 300 requests with a size of $1500\,\mathrm{B}$ for each measurement run. For each target, we conduct ten measurement runs per protocol. Measurements are conducted from a physical server hosted in a campus data center in central Europe, connected with a $1\,\mathrm{Gbit/s}$ up- and downlink to a national science network that connects to the Internet. The measurement host is equipped with an Intel Xeon E5-2630 CPU providing six physical cores at a clock frequency of $2.6\,\mathrm{GHz}$, $32\,\mathrm{GB}$ memory, and a Broadcom NetXtreme BCM5719 Gigabit NIC. We configure a lower limit of 1 packet per interrupt on the measurement host to avoid impacts by IC and enable generic receive offloading.

TCP-based estimates per target show mostly means and medians between $800\,\mathrm{Mbit/s}$ and $900\,\mathrm{Mbit/s}$, while less than $10\,\%$ of targets show significantly smaller estimated capacity, as shown in Figure 4 for observed medians. For ICMP-based measurements, we find significant outliers due to interrupt coalescence, despite the configuration of the measurement host, resulting in means larger than downlink capacity. In addition, we observe larger shares of median capacity estimates smaller than $100\,\mathrm{Mbit/s}$, indicating underestimation due to artifacts like ICMP rate limiting by the measurement targets, as TCP-based measurements on the same path result in larger estimates. Considering only targets chosen from rank one up to 1000 of the Alexa top list, we observe larger shares of targets responding to ICMP Echo requests and larger shares of ICMP-based estimates approximating downlink capacity, implying differences in the handling of ICMP Echo requests by targets. Regarding the stability of estimates, we observe median absolute deviations (MAD) mostly smaller than $50\,\mathrm{Mbit/s}$ for TCP-based estimates, which also applies to ICMP-based estimates approximating downlink capacity.

One intention behind surveying capacity estimates based on ICMP Echo requests is conducting measurements to intermediate hops on the path to a certain target. To survey the suitability of routers as targets for active capacity measurements, we conduct three ICMP-based measurement runs to over 250 routers observed during traceroutes to targets chosen from the first 2000 entries of the Alexa top list. We receive responses from 215 routers, while 52 do not respond to ICMP Echo requests. Median estimates per router indicate a significant underestimation of several hundreds of $\mathrm{Mbit/s}$ for most targets, as shown in Figure 4, resulting in a cluster of median estimates around $100\,\mathrm{Mbit/s}$. Such observation can be traced back to ICMP rate limiting, which was observed in the Internet before [33], [34].

VIII. Conclusion

In this paper, we surveyed the accuracy of PPD-based capacity estimation in the Internet. We implemented capacity estimation according to the *PPrate* algorithm and conducted active measurements by initiating TCP downloads and generating bursts of ICMP Echo requests.

Conducting active measurements on hybrid Internet paths providing ground truth data confirms the general accuracy of PPD-based measurements in the Internet, while we observe large relative errors for ICMP-based measurements affected by interrupt coalescence. In the case of TCP traffic, the impact of interrupt coalescence on estimates is mitigated by receiver offloading. We conduct Internet measurements to over 3500 public web servers to study capacity estimates and their characteristics. Conducted TCP measurements show dominating shares of estimates approximating path capacities of at least $1\,\mathrm{Gbit/s}$. For nearly $50\,\%$ of measurement targets, we find significant differences between TCP- and ICMP-based estimates implying underestimation of several hundreds of $\mathrm{Mbit/s}$ for ICMP-based measurements, while remaining ICMP-based estimates are close to $1\,\mathrm{Gbit/s}$. Further, we observe that routers are no suitable measurement target due to ICMP rate limiting increasing inter-arrival times.

Acknowledgments

This work has been supported by the German Federal Ministry of Education and Research, project 6G-ANNA (16KISK107), project PRIMEnet (16KIS1370), ACE-SUPPRA, and the European Union's Horizon 2020 research and innovation programme under grant agreement ID 101008468 (SLICES-SC).

REFERENCES

[1] A. B. Downey, "Using pathchar to estimate internet link characteristics," *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4, 1999.

[2] ——, "Clink: a tool for estimating internet link characteristics," 1999.

[3] R. S. Prasad, C. Dovrolis, and B. A. Mah, "The effect of layer-2 store-and-forward devices on per-hop capacity estimation," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, vol. 3. IEEE, 2003.

[4] C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet dispersion techniques measure?" in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. IEEE, 2001.

[5] C. Dovrolis and R. Prasad, "Readme for pathrate-2.4.1," 2017, last accessed on 2022/08/04. [Online]. Available: https://github.com/brucespang/pathrate

[6] L.-J. Chen, T. Sun, G. Yang, M. Y. Sanadidi, and M. Gerla, "End-to-end asymmetric link capacity estimation," in *NETWORKING 2005. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*. Springer Berlin Heidelberg, 2005.

[7] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "Sprobe: A fast technique for measuring bottleneck bandwidth in uncooperative environments," in *IEEE INFOCOM*, 2002.

[8] R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M. Y. Sanadidi, "CapProbe: A Simple and Accurate Capacity Estimation Technique," in *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '04. ACM, 2004.

[9] L.-J. Chen, T. Sun, B.-C. Wang, M. Sanadidi, and M. Gerla, "PBProbe: A capacity estimation tool for high speed networks," *Computer Communications*, vol. 31, no. 17, 2008.

[10] V. Jacobson, "Pathchar: A tool to infer characteristics of Internet paths," 1997.

[11] K. Lai and M. Baker, "Nettimer: A Tool for Measuring Bottleneck Link Bandwidth." in *USITS*, vol. 1, 2001.

[12] T. En-Najjary and G. Urvoy-Keller, "PPrate: A Passive Capacity Estimation Tool," in *End-to-End Monitoring Techniques and Services, 2006 4th IEEE/IFIP Workshop on*. IEEE, 2006.

[13] C. Marcondes, C. Palazzi, M. Y. Sanadidi, M. Gerla, M. Martinello, and M. T. Torres, "Regenerating tcp dynamics from traces path characteristics," in *2007 3rd International Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities*, 2007.

[14] S. Katti, D. Katabi, C. Blake, E. Kohler, and J. Strauss, "MultiQ: Automated detection of multiple bottleneck capacities along a path," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. ACM, 2004.

[15] T. En-Najjary and G. Urvoy-Keller, "Passive capacity estimation: Comparison of existing tools," in *2008 International Symposium on Performance Evaluation of Computer and Telecommunication Systems*. IEEE, 2008.

[16] A. Jindal, K. Psounis, and M. Liu, "Capest: A measurement-based approach to estimating link capacity in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 12, 2011.

[17] N. S. Kagami, R. I. T. da Costa Filho, and L. P. Gaspary, "Capest: Offloading network capacity and available bandwidth estimation to programmable data planes," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, 2019.

[18] M. Siekkinen, G. Urvoy-Keller, E. W. Biersack, and D. Collange, "A root cause analysis toolkit for tcp," *Computer Networks*, vol. 52, no. 9, 2008.

[19] S. Bauer, K. Holzinger, B. Jaeger, P. Emmerich, and G. Carle, "Online Monitoring of TCP Throughput Limitations," in *2020 IEEE/IFIP Network Operations and Management Symposium (NOMS 2020)*, 2020.

[20] S. Bauer, F. Wiedner, B. Jaeger, P. Emmerich, and G. Carle, "Scalable TCP Throughput Limitation Monitoring," in *2021 IEEE/IFIP Symposium on Integrated Network and Service Management (IM 2021)*, 2021.

[21] M. Jain and C. Dovrolis, "Pathload: A measurement tool for end-to-end available bandwidth," in *In Proceedings of Passive and Active Measurements (PAM) Workshop*. Citeseer, 2002.

[22] ——, "End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput," *IEEE/ACM Transactions on Networking (TON)*, vol. 11, no. 4, 2003.

[23] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy, "Bandwidth Estimation: Metrics, Measurement Techniques, and Tools," *IEEE network*, vol. 17, no. 6, 2003.

[24] J. Sommers, P. Barford, and W. Willinger, "A proposed framework for calibration of available bandwidth estimation tools," in *11th IEEE Symposium on Computers and Communications (ISCC'06)*. IEEE, 2006.

[25] E. Goldoni and M. Schivi, "End-to-end available bandwidth estimation tools, an experimental comparison," in *Traffic Monitoring and Analysis*. Springer Berlin Heidelberg, 2010.

[26] A. K. Paul, A. Tachibana, and T. Hasegawa, "Next: New enhanced available bandwidth measurement technique, algorithm and evaluation," in *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*. IEEE, 2014.

[27] ——, "Next-fit: Available bandwidth measurement over 4g/lte networks– a curve-fitting approach," in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 2016.

[28] S. Bauer, J. Janelidze, B. Jaeger, P. Sattler, and P. Brzoza, "Code and measurement data," https://github.com/bauersi/PyPPrate, 2022.

[29] R. L. S. De Oliveira, C. M. Schweitzer, A. A. Shinoda, and L. R. Prete, "Using Mininet for Emulation and Prototyping Software-Defined Networks," in *2014 IEEE Colombian Conference on Communications and Computing (COLCOM)*. IEEE, 2014.

[30] A. Barbeiro, "Visualizing multipath networks with dublin traceroute," *Presentation at the MOCA Italian hacker camp*, 2016.

[31] K. M. Salehin, V. Sahasrabudhe, and R. Rojas-Cessa, "Remote measurement of interrupt-coalescence latency of internet hosts," in *2017 IFIP Networking Conference (IFIP Networking) and Workshops*. IEEE, 2017.

[32] "Alexa. top 1m sites." http://s3.dualstack.us-east-1.amazonaws.com/alexa-static/top-1m.csv.zip; Last accessed: 15 Sep 2022.

[33] R. Ravaioli, G. Urvoy-Keller, and C. Barakat, "Characterizing icmp rate limitation on routers," in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015.

[34] H. Guo and J. Heidemann, "Detecting icmp rate limiting in the internet," in *International Conference on Passive and Active Network Measurement*. Springer, 2018.