Receiver-Based Path Capacity Estimation for TCP

Christoph Barz, Matthias Frank, Peter Martini, Markus Pilz

Institute of Computer Science IV, University of Bonn, Germany
{barz, matthew, martini, pilz}@cs.uni-bonn.de

Abstract: This paper presents a new TCP-based packet pair measurement approach for network path capacity estimation. It addresses some drawbacks other approaches still bear. In contrast to these, our approach uses an unmodified TCP sender and utilizes techniques to identify probing packets that were less influenced by cross traffic. These techniques are applied at the receiver side only. The concept is verified by simulation and an outline of the applicability for real-life measurements in the Internet is presented.

1 Introduction

Available bandwidth and path capacity estimation techniques have been important subjects for research and still leave room for improvements. This paper concentrates on end to end path capacity estimation which is also a basis for many available bandwidth estimation techniques.

The knowledge of the maximum capacity of a given network path, i.e. the minimum capacity of the individual links, is useful for network management, engineering purposes, dynamic server selection, selection of paths in overlay networks and in heterogeneous access networks with different characteristics, e.g. Multi Radio Networks. Given the heterogeneity of independent interconnected networks like the Internet, the estimation of both capacity and available bandwidth become an end-to-end and hence decentralized task.

Our new approach of TCP-based packet pair capacity estimation makes use of the concepts that have been well explored and discussed in the literature, but also tackles some of the drawbacks that other approaches still bear. This paper presents the basic ideas and an exemplary verification of the concept performed by simulation analysis. The main distinction to known approaches is that our concept is receiver-based and does not require any modifications at the sender. In particular, our concept supports scenarios where the client is the receiver of a saturated downstream transmission from a server.

The rest of this paper is structured as follows: Section 2 presents an overview of the related work. Section 3 introduces new mechanisms for TCP-based packet pair measurements and also addresses details of their technical realization. Section 4 focuses on an evaluation of these approaches based on simulation. Section 5 comprehensively discusses our work on real-life measurements, concludes our work and outlines our future research.

2 Related Work

The basic principle of packet dispersion techniques was first described by Jacobson and Karels in 1988 [JK88]. In the context of TCP's congestion control they observed a "self-clocking effect", i.e. the spacing of TCP data packets on the slowest link of the path is preserved on the way to the receiver and by the acknowledgment packets (ACKs) which are triggered. To the best knowledge of the authors, the term "packet pair" was first mentioned in 1991 by Keshav [Ke91] in a control-theoretic approach to packet dispersion technologies. While also considering cross traffic effects, strong assumptions were made on the router behavior. In 1996 Carter and Crovella [CC96] introduced a tool for path capacity estimation called "bprobe" that is based on the ICMP ECHO mechanism and takes into account cross traffic, queuing delays, packet drops, and downstream congestion with weaker assumptions on router behavior. A new filtering technology is used based on the effect of cross traffic on different packet pair sizes to prevent an underestimation of path capacity. Using bprobe they present a survey of path capacities from their source to different WWW servers. Hindered by the use of the ICMP ECHO mechanism, no measurements from the WWW servers to their source were presented. In 1997 Paxson [Pa97] introduced a framework which was able to perform packet pair measurements by analyzing "real world" TCP traces between network sites. While his receiver-based approach allowed unidirectional path capacity estimation, both the TCP sender and receiver trace were required. Paxson's "packet bunch modes" were the first approach to take a multi-modal distribution of bandwidth estimates into account, but it was strongly based on heuristics. In 2001 Dovrolis, Ramanathan and Moore showed in [DRM01] that path capacity may not be the global mode of packet pair measurements. They classified the effects of cross traffic as Sub-Capacity Dispersion Range (SCDR) and Post-Narrow Capacity Modes (PNCM) and used a distribution of packet trains – the Asymptotic Dispersion Range (ADR) – to identify the path capacity mode. While refining the analysis of multi-modal path capacity estimation, again only a heuristic approach is given.

Since then extensions to the original packet pair approach have been suggested that take into account additional information to identify packet pairs less influenced by cross traffic. In 2003 ZiXuan et al. [Zi03] proposed a scheme called "packet triplet" where three equally sized packets are sent back to back, interpreted as two dependant packet pairs. In 2004 Karpoor et al. [Ka04] introduced a path capacity measurement tool called "CapProbe" that introduces integrated delay measurements for filtering purposes. Recently, Chen et al. extended this active measurement approach to passive sender-based measurements using TCP or TCP-Friendly Rate Control [Ch04].

Also in 2004 Jiang and Dovrolis [JD04] presented another approach to combine passive packet pair measurements with TCP. Based on the self-clocking effect described by Jacobson and Karels in 1988 [JK88] and the delayed ACK mechanism – implemented in many TCP stacks today – they showed that every TCP sender sends about 50% of all packets as packet pairs, triggered by delayed ACKs. This effect can be exploited for path capacity estimation. Jiang and Dovrolis used this observation to analyze network traces of routers with aggregated TCP traffic.

An overview of bandwidth estimation which also includes techniques for available bandwidth estimation may be found in [PD03]. The impact of the packet size of probing packets on measurement accuracy is discussed in [PV02].

3 Path Capacity Estimation at TCP Receivers

Our TCP-based capacity estimation approach is similar to the self-clocking effect of TCP [JK88] and avoids the drawbacks of the ICMP ECHO mechanism [CC96]. Being receiver-based and founded on TCP our approach can benefit from the same advantages as Paxson's measurement framework, but only a receiver trace is observed and no additional deamons at other sites are necessary. This brings us closer to one of our main goals: The analysis of downlink capacity from real FTP and WWW servers at sites all over the Internet. Like [JD04], our work is also facilitated by the delayed ACK mechanism of TCP which causes a saturated TCP sender in equilibrium to transmit two equally sized TCP packets with a low separation and thus a high potential bandwidth. Unlike their approach, our work does not focus on strictly passive measurements of aggregated TCP traffic at routers but on optimizing receiver-based measurements (cf. section 3.1). Monitoring traffic at the TCP receiver can help to exploit additional information that may not be available at intermediate routers. Furthermore, full control of the TCP receiver behavior gives tight control of the packets generated by the sender. Receiver control also allows to combine our new approach with recent developments like packet triplets and the CapProbe mechanism to filter out measurements influenced by cross traffic (cf. section 3.2).

3.1 Triggering Packet Pairs

TCP uses a sliding window mechanism to realize flow control. A TCP sender reacts to an acknowledgment of data by shifting its window according to the bytes acknowledged. The delayed ACK mechanism described in RFC 2581 will lead to a self clocking effect described in [JD04] (cf. Figure 1a). As shown in [JD04], without cross traffic this leads to a typical inter-packet arrival time of *L/C* for all TCP packets with packet size L and path capacity C at the receiver. With cross traffic injected at link K upstream of the bottleneck link, non back to back packets may suffer from an increased interference window ($2L/C\text{-}L/C_K \geq L/C_K$), leading to an increased SCDR. Therefore it is crucial to identify packets triggered by the same ACK, i.e. packet pairs. Since Jiang and Dovrolis were not able to do so in a strictly passive environment they focused on the estimation of "pretrace capacity C_P" at intermediate routers.

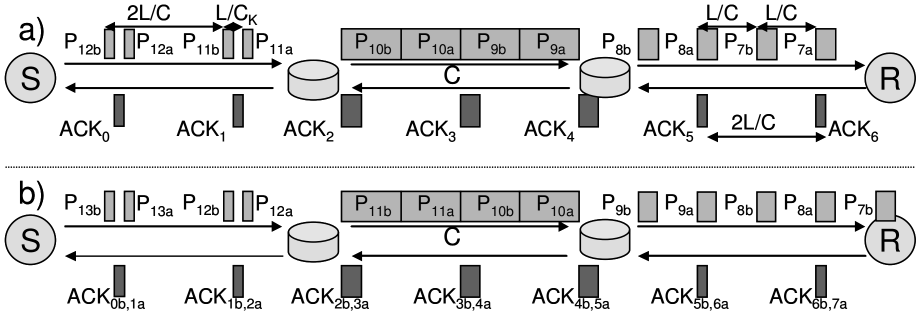

Figure 1: Self clocking of TCP with delayed ACKs with an even a) and odd b) number of data packets from sender S to receiver R

To identify packet pairs it is neither sufficient for a TCP receiver to simply monitor the events of outgoing delayed ACKs at the receiver nor the acknowledged byte numbers as the flight size is not known and therefore the amount of TCP data packets on the path from the sender to the receiver could be even or odd. An odd number of data packets would lead to an ACK of packages from different packet pairs (cf. Figure 1b). As we have full control of the TCP receiver the receiver's advertised window can be constrained. With a small advertised receiver window the transmission window size at the sender will be identical to the advertised receiver window size most of the time. This means that the flight size is known to the receiver and thus the number of data packets between S and R. In addition, reducing the transmission window size and thus increasing the gap between consecutive packet pairs eases possible congestion, lessens the intrusiveness and makes consecutive packet pair measurements more independent. As an extreme example a receiver window size of two times the maximum segment size ensures that every flight consists of a single packet pair only but also reduces TCP's throughput and thus the number of packet pairs per time received. Changing the size of the receiver window can be accomplished at any time during a connection but will only take effect after new ACKs have reached the sender. Different probing packet sizes can be used to sidestep probing packet sizes with inaccurate measurements and to gain additional information on the path behavior. Our approach enables us to use different probing packet sizes. The Maximum Segment Size (MSS) Option defined in RFC 793 is used to signal the maximum segment size a receiver is willing to accept to the TCP sender during the TCP handshake only. This means that for capacity estimation with different probing packet sizes different TCP connections have to be used. It should be mentioned that the MSS is negotiated between the sender and receiver. A detailed examination of the implications of different probing packet sizes is out of the scope of this paper.

3.2 Enhanced Approaches

The packet triplet approach [Zi03] interprets three back to back packets as two correlated packet pairs. The deviation of the two dispersions is compared and measurements are discarded if their deviation exceeds a threshold. Packet triplets can also be generated when using our TCP-based measuring approach. Instead of acknowledging every second segment the receiver sends a cumulative ACK for every third segment. This is in conformance with RFC 2581 that states that at least for every second full-sized segment an ACK SHOULD be generated. The receiver's advertised window must be a multiple of three 3* MSS accordingly. We believe that there is a main drawback concerning the original approach which is reflected by their measurements. While reducing the SCDR caused by cross traffic packets between one of both packet pairs there is an increased chance of PNCMs in heavy load scenarios due to the queuing of the whole packet triplet at routers downstream of the bottleneck. The CapProbe technique [Ka04] combines path capacity estimation based on packet pair dispersion with RTT measurements to identify the influence of cross traffic. Since most CapProbe variants are sender-based, i.e. echoes are triggered at the receiver, the RTT for both packets of a packet pair can easily be measured at the sender. TCP Probe [Ch04], the TCP-based variant of CapProbe, is based on the assumption that for every data packet sent an ACK is triggered at the receiver. This is in contrast to the delayed ACK mechanism applied to most TCP receiver implementations today. This problem is circumvented by a TCP sender modification called “inverted packet pair”. Different from CapProbe our approach is based on receiver modifications and not on TCP sender modifications.

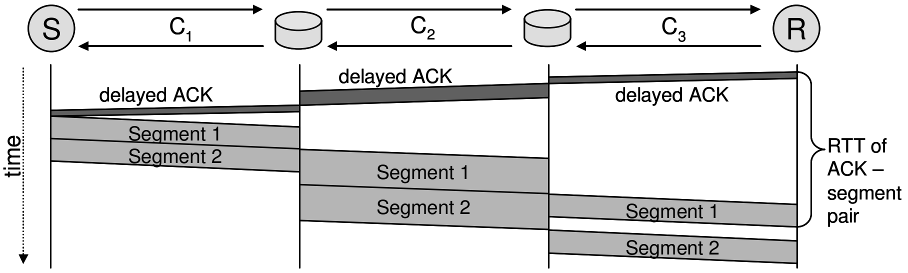

Figure 2: RTT of ACK - Segment Pair

We see three alternatives in combining the delay information with our approach: The first way is to use sender timestamps. Clock synchronization is not required as we are not interested in absolute delays. Unfortunately, our experiments showed an inadequate resolution of ~10ms when using the TCP timestamp option (RFC 1323). An additional, more accurate timestamp cannot be applied at the sender due to our restriction to receiver modification only. The second approach is to measure the RTT of each delayed ACK and the first segment it triggers (cf. Figure 2). The third approach is to send two separate ACKs, each acknowledging one segment, back to back at the time the cumulative delayed ACK would be sent and thus producing two RTT measurements. However, as the second ACK packet will not instantaneously trigger the second data packet – the data packets will usually be larger than the ACKs – the second RTT will not contain additional information. Instead cross traffic at the ACK stream might disturb measurements. As a consequence we focus on the second approach. This adapted filtering technique has little impact on the reduction of the SCDR but may significantly reduce PNCMs.

4 Simulation

The following section presents an application and validation of the proposed techniques by means of simulation with ns-2. After an introduction to the simulation scenario the distribution of packet separation at the receiver of an unmodified TCP stream is compared to our combined receiver-based path capacity estimation approach for TCP which is discussed in sections 3.1 and 3.2. In addition, a separate analysis of the enhancement approaches packet triplet and delay filtering is performed. Due to space limitations we must dispense with a more detailed analysis. In the following we will refer to each measurement as “sample”.

4.1 Simulation Scenarios

In the context of packet pair simulations two predominant scenario types are used to analyze capacity estimation techniques in the presence of cross traffic: path persistent cross traffic and one-hop-persistent cross traffic scenarios, cf. [DRM01], [Zi03] and [Ka04]. In this paper the focus is on one-hop-persistent scenarios as they yielded the most interesting effects in terms of cross traffic interaction.

The chosen one-hop-persistent scenario is sketched in Figure 3. The probing packets are sent from S to R via the routers R_i and have to pass 6 hops with capacities of 10, 7.5, 5.5, 4, 6 and 8 Mbit/s. The cross traffic between each two routers is generated by 4 TCP streams (dotted line). Following [Fr03], the cross traffic packet sizes are 50 bytes, 570 bytes, 820 bytes and 1500 bytes at the IP level.

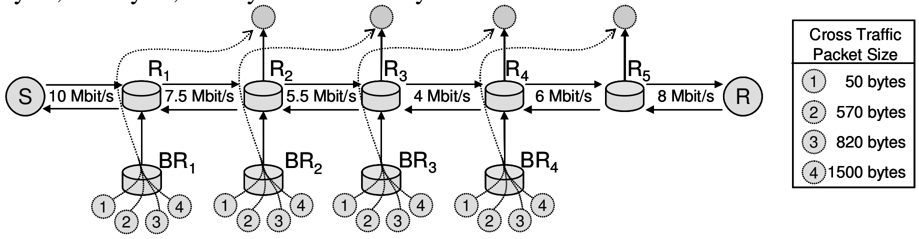

Figure 3: One-hop-persistent Simulation Scenario

In order to load each hop between two routers with a certain percentage of cross traffic, the link between a router and a border router (e.g. the link between R_1 and BR_1 in Figure 3) is limited correspondingly. It should be mentioned that [DRM01] and [Ka04] use cross traffic sources with a Pareto distribution. The motivation is to model long-range dependent cross traffic. As Pareto sources have on and off times, this can lead to phases with less cross traffic. Therefore, saturated TCP sources were preferred for cross traffic. Heavy fluctuations within the TCP streams are avoided by using RED queues [FJ93]. This prevents the scenario from phases without cross traffic. Furthermore, the probing stream starts after the cross traffic sources have tuned in.

4.2 Basic TCP Probing Stream vs. Modified Receiver

Simulations were performed with an unmodified TCP probing stream (cf. Figure 4 left), and a TCP probing stream with a reduced transmission window size to identify back to back packets, packet triplet and delay information enhancements to identify cross traffic influence (cf. Figure 4 right). For the unmodified TCP probing stream the transmission window is only limited by the congestion window and for all enhanced probing streams the congestion window is limited to 3*MSS to support packet triplets. The simulation time in all experiments is the same, resulting in significantly fewer samples with a transmission window size of 3*MSS. Note that when using packet triplets the transmission window size must be a multiple of 3*MSS but should not be too large (cf. section 3.1). The probing packet size is 200 bytes and the links are 60% loaded with cross traffic. All bandwidth estimate histograms have a bin size of 100 kbit/s.

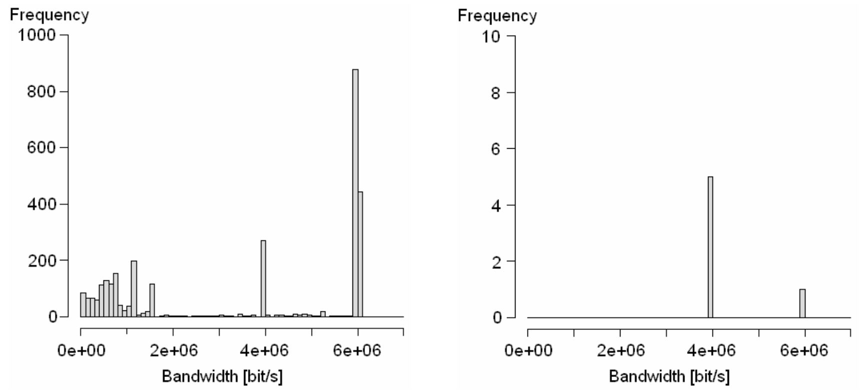

Figure 4: TCP Estimates at the Receiver with standard TCP parameters (left) and Packet Triplet Estimates with low Delay Filtering (right)

Regarding the bandwidth estimation histogram of the unmodified TCP probing stream (cf. Figure 4 left) a bottleneck of 4 Mbit/s can be identified as a local mode in the histogram, but the global mode is at 6 Mbit/s (PNCM). In contrast, the results of the probing stream with all section 3.2 enhancements show the global mode at the bottleneck capacity of 4 Mbit/s and no SCDR.

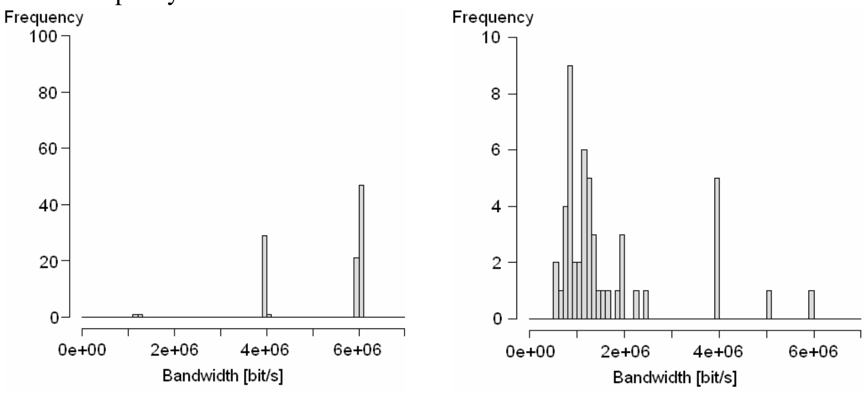

Figure 5: Packet Triplet Filtering (left) Delay Filtering (right)

Figure 5 shows results of our approach combined with only one enhancement – packet triplet filtering (left) and delay filtering (right) in contrast to using both enhancements as in Figure 4 (right) – to visualize the different filtering effects. An inspection of the packet triplet dispersions identified the bottleneck capacity and the PNCMs while the samples with the lowest RTTs determine the bottleneck capacity and an SCDR. Thus the combined enhanced approach results in a superior filtering and path capacity estimation.

It should be mentioned that for lower cross traffic loads and different packet sizes even a basic TCP probing stream can obtain a global mode at 4 Mbit/s in this scenario. The packet size of 200 bytes and a cross traffic load of 10% to 20% result in a path capacity estimate of 4 Mbit/s by means of a global mode.

5 Conclusions and Further Work

The goal of the presented path capacity estimation techniques is to offer a way for end-to-end measurements within heterogeneous networks like the Internet without a distributed measurement framework. The basic approach to estimate the path capacity as presented in section 3.1 can be realized with packet sniffers, tools to retrieve files via FTP or HTTP and TCP receiver parameter tuning. Experiments showed that many peers accept a suggested MSS parameter. In order to use the proposed packet triplet technique, the delayed ACK mechanism at the receiver has to be modified. This has been realized by a lightweight user-space TCP receiver implementation on a Linux system. This allows for triggering triplets at all senders.

The results obtained by simulation showed that the techniques are able to identify bottleneck capacity even on highly loaded paths. Based on the prototypical implementation mentioned above, a further investigation via “real world” measurements is the next step, including an analysis of the potential bandwidth of WWW servers under high load conditions. Although first results are available, their validity in terms of system and network configuration is open and a more detailed simulation study will follow.

Bibliography

[CC96] Carter, R.; Crovella, M.: Measuring bottleneck link speed in packet-switched networks. Performance Evaluation, Vol. 27-8: 297-318, October 1996

[Ch04] Chen et al.: CapProbe based Passive Capacity Estimation. Technical Report, UCLA Computer Science Department, Los Angeles, USA, 2004

[DRM01] Dovrolis, C.; Ramanathan, P.; Moore, D.: What do packet dispersion techniques measure? In Proc. of IEEE INFOCOM: 905-914, Anchorage, Alaska, April 2001

[FJ93] Floyd, S.; Jacobson, V.: Random Early Detection gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking V.1 N.4, August 1993

[Fr03] Fraleigh et al.: Packet-level Traffic Measurement from the Sprint IP Backbone, IEEE Network Magazine, November 2003

[JD04] Jiang, H.; Dovrolis, C.: The effect of flow capacities on the burstiness of aggregate traffic. In Proc. of PAM 2004: 93-102, Antibes Juan-les-Pins, France, April 2004

[JK88] Jacobson, V.; Karels, M: Congestion Avoidance and Control. In Proc. of ACM SIGCOMM, Stanford, August 1988

[Ka04] Kapoor et al.: CapProbe: A Simple and Accurate Capacity Estimation Technique. In Proc. of ACM SIGCOMM, Portland, Oregon, USA, August 2004

[Ke91] Keshav, S.: A Control-Theoretic Approach to Flow Control. In Proc. of ACM SIGCOMM: 3-15, Zurich, September 1991

[Pa97] Paxson, V.: End-to-End Internet Packet Dynamics. In Proc. of ACM SIGCOMM: 139-152, Cannes, France, 1997

[PD03] Prasad, R.; Dovrolis, C.: Bandwidth Estimation: Metrics, Measurement Techniques, and Tools. IEEE Network, November/December 2003

[PV02] Pásztor A, Veitch D: The Packet Size Dependence of Packet Pair Like Methods. IEEE/IFIP International Workshop on Quality of Service (IWQoS), 2002

[Zi03] ZiXuan et al.: Packet Triplet: An enhanced packet pair probing for path capacity estimation. In Proc. of Network Research Workshop, Rep. of Korea, Aug. 2003