Edge-based Inference and Control in the Internet

Aleksandar Z. Kuzmanovic

Department of Electrical and Computer Engineering

Rice University

Ph.D. Thesis Proposal

Abstract

Realizing new services on the Internet ultimately requires edge-based solutions for both deployability and scalability. I propose to design, implement, and evaluate a series of three edge-based algorithms and protocols for efficient inference and control of the Internet from its endpoints. The proposed solutions together form a new foundation for quality-of-service communication via a scalable edge-based architecture where the novel functionality is added strictly at either edge routers or end hosts. In particular, this thesis proposes techniques for multi-class service inference, active probing for available bandwidth, and end-point-based protection against Denial of Service (DoS) attacks.

The proposed multi-class service inference techniques reveal the sophisticated multi-class network components such as service disciplines and rate limiters using solely passive packet monitoring at the network edges. These inferences significantly enhance the network monitoring and service validation capabilities and provide vital information for making efficient use of resources. The proposed active probing scheme infers and utilizes only the available network bandwidth and aims to realize a low-priority service from the network endpoints, a functionality that would otherwise require a multi-priority or separate network. Finally, the end-point-based protection against DoS research aims to detect the main fragile points of TCP from the perspective of a DoS attacker. This would not just help to significantly improve the robustness of TCP in presence of DoS attacks but would indicate that the protection mechanisms against DoS should be implemented not only in the network core as conventionally done but also at the network edge.

1 Introduction

The Internet has evolved into a system of astonishing scale and complexity such that the development and deployment of advanced services on it has reached a crossroads: efforts to add new services in the network core have quickly encountered scalability problems, yet new services are in critical demand and must be rapidly and widely deployed.

Consequently, there is a quest to step back toward the original design principles of the Internet [1] and push the advanced functionality to the network edge while implementing minimum functionality at the network core. And while the design principles and constraints are left unchanged, the expectations from the future Internet significantly overcome a moderate single best-effort datagram service of the past: the users demand multiple traffic classes, service differentiation, and QoS guarantees. To achieve these goals, network researchers are challenged to devise sophisticated new algorithms or improve the existing ones, yet using solely measurements obtained at the network edge.

To implement a QoS or a transmission control from the network edge, it is essential to make solid *inferences* of both static and dynamic internal network properties. For example, to make successful capacity planning decisions, a network engineer needs estimates of both lower (or guaranteed) and upper service bounds. On the other hand, to perform a solid transmission control in a wide-area network that does not provide any explicit information about the network path, a transmission protocol should form its own estimates of current network conditions, and then to use them to adapt as efficiently as possible. A classic example of such estimation and control is how TCP infers the presence of congestion along an Internet path by observing packet losses, and either cuts its sending rate in the presence of congestion, or increases it in the absence.

In this thesis, I propose to design, implement, and evaluate a series of three edge-based algorithms and protocols for efficient inference and control of the Internet from its endpoints. The proposed solutions together form a new foundation for quality-of-service communication via a scalable edge-based architecture where the novel functionality is added strictly at either edge routers or end hosts. This work devises two novel algorithms that infer both important static and dynamic network properties. Also, it proposes improvements for TCP, the dominant end-point-based protocol of todays Internet, that should significantly enhance its immunity to

DoS attacks.

First, this thesis develops a framework for monitoring, validation, and inference of multi-class services. In particular, it shows how passive monitoring of system arrivals and departures can be used to detect if a class has a minimum guaranteed rate and/or a rate limiter. Moreover, if such elements exist, this work shows how to compute their maximum likelihood parameters. Beyond a single class, it also shows how inter-class relationships can be assessed. For example, this research devises tests which infer not only whether a service discipline is work-conserving or non-work-conserving, but also the relationship among classes, such as weighted fair or strict priority.

Second, this thesis devises TCP Low Priority (TCP-LP), an end-point protocol that achieves two-class service prioritization without any support from the network. The key observation is that end-to-end differentiation can be achieved by having different end-host applications employ different congestion control algorithms as dictated by their performance objectives. Since TCP is the dominant protocol for best-effort traffic, TCP-LP is designed to realize a low-priority service as compared to the existing best effort service. Namely, its objective is for TCP-LP flows to utilize the bandwidth left unused by TCP flows in a non-intrusive, or TCP-transparent, fashion.

Finally, this proposal analyzes TCP in presence of misbehaving flows, which are one of the major threats to adequate QoS. Conventional wisdom says that today's Internet is stable due to TCP and its congestion control mechanisms. The hypothesis of this research is that these mechanisms can be used as a tool for DoS attacks and the objective is to detect the main fragile points of TCP. The other conventional wisdom says that misbehaving flows are high bit-rate flows. In contrast, this thesis argues that this is not necessarily true and shows that it is only thin a line between non-intrusive inference techniques and DoS attacks.

2 Background

This section provides a brief review of inter-class resource sharing theory, which is a starting point upon which the multi-class inference techniques are developed. Next, it reviews the main TCP congestion control phases with emphasis on those which are of interest for available bandwidth inference and DoS research.

2.1 Inter-class Resource Sharing

In [2], statistical admission control tests are developed for several multi-class schedulers. The key technique for exploiting inter-class resource sharing is to characterize a class' available service beyond its worst-case allocation. For example, in a Weighted Fair Queuing (WFQ) server, a class with weight ϕ_i receives service at rate no less than $\phi_i C$ whenever it is backlogged. However, due to statistically varying demands of other classes, the service received can be far greater than this lower bound. A statistical service envelope is a general characterization of the service received by a class over intervals of different length for which the class is continually backlogged. More important, the service envelopes reveal the internal network mechanisms used for service differentiation since they are functions of scheduler parameters and other class' input traffic. Thus, they are an ideal tool that is used to both detect the type of an unknown scheduler and to estimate its parameters.

2.2 TCP Congestion Control

Figure 1 shows a temporal view of the TCP/Reno congestion window behavior at different stages with points on the top indicating packet losses. Data transfer begins with the *slow-start* phase in which TCP increases its sending rate exponentially until it encounters the first loss or maximum window size. From this point on, TCP enters the *congestion-avoidance* phase and uses an additive-increase multiplicative-decrease policy to adapt to congestion. Losses are detected via either time-out from non-receipt of an acknowledgment, or by receipt of a triple-duplicate acknowledgment. If loss occurs and less than three duplicate ACKs are received, TCP reduces its congestion window to one segment and waits for a period of Retransmission Time Out (RTO), after which the packet is resent. The RTO parameter is dynamically updated based on the estimates of smoothed Round-Trip Time (RTT) and RTT variation. Allman and Paxson [3] show that TCP achieves near optimal performance if there exists a lower bound for the RTO of one second. Thus, it has been accepted [4] that whenever RTO is computed, if it is less than one second then it should be rounded up to one second. In the case that another time out occurs before successfully retransmitting the packet, TCP enters the *exponential-backoff* phase and doubles RTO until the packet is successfully acknowledged.

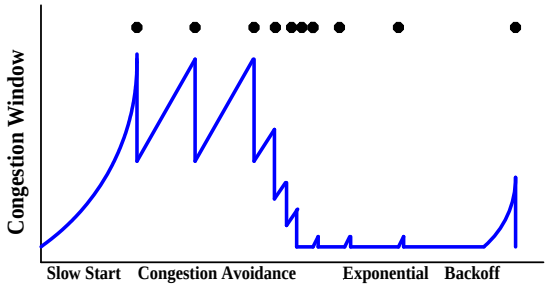

Figure 1: Behavior of TCP Congestion Control

3 Thesis Contributions

Our contributions are as follows.

I. Multi-Class Service Inference

A. Network Scenario [5]

This thesis devises an algorithm which infers the multi-class network components such as service disciplines and rate limiters using solely passive packet monitoring at the network edges. Using empirical arrival and service rates measured across multiple time scales, and using the theoretical service envelopes developed in [2], this work first devises hypothesis tests for determining the most likely service discipline among Earliest Deadline First (EDF), WFQ, and Strict Priority (SP)[1]. The hypothesis tests are performed over multiple time scales, and the final decision is obtained using majority rule over all time scales. After inferring the scheduler, the algorithm finds the Maximum Likelihood Estimate (MLE) of unknown scheduler parameters such as EDF delay bounds, WFQ weights, and relative SP class priorities. Finally, using the same methodology, we show how non-work-conserving elements are detected and parametrized.

This research shows that time scales play a key role in determining multi-class components. Short time scale measurements are crucial for detecting rate limiters while long time scale measurements best reveal “link sharing” rules and weights. Thus, a key aspect of this work’s contribution is that it treats phenomena occurring at

[1]The devised methodology can be applied to any other scheduler for which a statistical service envelope is derived.

different time scales in a uniform and methodical way. Simulation experiments in ns-2 show high accuracy in both inferring unknown schedulers and estimating their parameters.

B. QoS-enabled Web Server Scenario [6]

This thesis further extends the above algorithm to QoS-enabled web-server scenario. The key issue here is the high variability of service times due to factors such as different processing times, disk service times and variable file sizes. To overcome this problem, this work modifies the above inference algorithm such that it first assesses and statistically characterize the web-server's service variability, and then determine mutual relationships among classes. I evaluate the modified algorithm using simulator from [7] and find out that it achieves high accuracy provided that the variability of service times due to the above server-specific factors is not significantly larger than the service variability due to other class'workload.

II. TCP and Available Bandwidth Inference

A. TCP-LP: Design and Analysis [8]

This proposal develops TCP-LP, a novel distributed protocol for low priority data transfer whose objective is to utilize the bandwidth left unused by TCP flows. The methodology for developing TCP-LP is as follows. This work first develops a reference model to formalize the two design objectives: TCP-LP transparency to TCP, and (TCP-like) fairness among multiple TCP-LP flows competing to share excess bandwidth. To approximate the reference model from a distributed end-point protocol, TCP-LP employs two new mechanisms. First, in order to detect oncoming congestion prior to TCP flows, TCP-LP uses inferences of one-way packet delays as early indications of network congestion vs. packet losses used by TCP. TCP-LP's second mechanism is novel congestion avoidance policy which modifies the additive-increase multiplicative-decrease policy of TCP via the addition of an inference phase and use of a modified back-off policy.

I implement TCP-LP in ns-2 simulator and the simulation results show that TCP-LP is largely non-intrusive to TCP traffic and that both single and aggregate TCP-LP

flows are able to successfully utilize excess bandwidth. Moreover, the experiments show that multiple TCP-LP flows share excess bandwidth fairly.

B. TCP-LP: Implementation and Evaluation in the Internet

I propose implementation of TCP-LP in Linux and its evaluation in a test-bed and the Internet. In the test-bed, I propose evaluation with both artificial and HTTP background traffic in order to validate the above simulation findings. In the Internet, I propose to perform experiments with bulk data transfers using TCP and TCP-LP protocols and to compare their throughputs over long time scales. This would help the networking community to better understand and quantify the difference between the pure excess network bandwidth and the available bandwidth as experienced by TCP. Moderate differences between the two quantities would strongly support TCP-LP's candidacy for the leading bulk-data transfer protocol of the future Internet.

III. TCP and Denial of Service Attacks

I propose to analyze and evaluate TCP from the point of a DoS attacker and my goal is to detect the main fragile points of TCP. The main hypothesis is that the exponential backoff policy and the newly adopted value of minimum RTO [4] together form a predictable and deterministic mechanism that can potentially be used by malicious users for DoS attacks. In particular, an attacker can choose a strategy to provoke TCP to enter exponential backoff phase by sending at peak rates only for a short period of time (on the time scale of a connection's RTT) and to repeat this activity periodically on time scales of the minimum RTO value. In this way, the attacker can significantly intrude a background TCP flow, but with only sending at low average rates. Moreover, since *all* variants of TCP use the same exponential backoff policy with the same minimum RTO value, it is expected that the above DoS strategy could throttle an *aggregate* of TCP flows, provided that the congestion caused by the DoS flow lasts long enough to force *all* background TCP flows to enter the exponential backoff phase. Finally, considering that TCP flows within the aggregate can have heterogeneous RTT times, another hypothesis is that the above DoS strategy can throttle only a *subset* of flows within the aggregate whose RTT is shorter than the duration of the DoS flow's burst time.

I propose to examine the above hypothesis in ns-2 simulator. If shown to be valid, the

possible two implications are as follows. First, in order to prevent against such low bit-rate DoS attacks, the TCP's minimum RTO parameter should be randomized. Second, the above strategy can be used by network engineers to scalably *filter* long-lived TCP flows with short round-trip times which can monopolize network resources and significantly impede the background TCP traffic [9].

4 Related Work

While the edge-based solutions developed in this thesis are unique in their goals, there are related efforts in several areas. The first is accurate inference of network characteristics based on end-to-end measurements; the second is network protection against DoS attacks. The inference algorithms based on end-to-end measurements can roughly be divided into two main classes: static inference algorithms aim to estimate *static* (or fixed) network parameters such as bottleneck link capacities, buffer sizes or parameters of Active Queue Management (AQM) schemes; and dynamic inference algorithms aim to estimate *dynamic* (or variable) network parameters such as available bandwidth. This section first reviews static and dynamic inference algorithms and then the related algorithms from the area of network protection against DoS attacks.

Static inference algorithms relate to multi-class inference techniques developed in this thesis in the following two ways. First, the algorithms that aim to measure bottleneck or per-hop link capacities [10, 11, 12, 13, 14, 15, 16] relate to the rate-limiter parameter inference methodology. In particular, the problem of inferring bottleneck link capacity is essentially identical to the problem of inferring a *single-level* leaky bucket rate-limit parameter. However, the main difference is that the above techniques use active probing while the multi-class technique uses passive packet monitoring. Moreover, these active probing techniques are not suitable for inferring *multi-level* leaky bucket parameters since this would require them to send probes at peak rates for sustained long time periods[2], which would unacceptably degrade the performance of the background traffic. Second, the proposed techniques for inferring the most likely *multi-class* scheduling discipline relate to probing schemes that aim to infer bottleneck-bandwidth queue size and parameters of AQM schemes. Liu and Crovella [18] propose one such scheme with the

[2]The inference of multi-level leaky bucket parameters requires longer time scale measurements due to traffic constraint functions which shape the traffic differently at different time scales (see [17] for example).

goal of inferring parameters of Random Early Detection (RED) or BLUE AQM schemes. The main difference between the two inference algorithms is that multi-class inference algorithm intends to first *detect* the most likely bottleneck scheduler and then to estimate its parameters, while Liu and Crovella [18] lack techniques to detect the AQM type implemented in the bottleneck router but simply assume that it is known.

The class of dynamic inference algorithms that attempt to infer the available bandwidth via probing relates to TCP-LP, which targets transmitting at the rate of available bandwidth. For example, Ribeiro et al. [19] and Alouf et al. [20] provide algorithms for estimation of parameters of competing cross-traffic under multifractal and Poisson models of cross traffic. In contrast, TCP-LP provides an adaptive estimation of available bandwidth by continually monitoring one-way delays and dynamically tracking the excess capacity. Similarly, Jain and Dovrolis [21] develop *pathload*, a delay-based rate-adaptive probing scheme for estimating available bandwidth. The key difference between *pathload* and TCP-LP is that the latter aims to *utilize* the available bandwidth, while the former only estimates it. Moreover, TCP-LP addresses the case of multiple flows *simultaneously* inferring the available bandwidth by providing each with a fair share (according to TCP fairness), an objective that is problematic to achieve with probes. Next, end-point admission control algorithms also use probes to detect if sufficient bandwidth is available for real-time flows [22]. Unfortunately, such techniques have a "thrashing" problem when many users probe simultaneously and none can be admitted. While TCP-LP targets a low rather than high priority class, its basic ideas of adaptive and transparent bandwidth estimation could be applied to end-point admission control and could alleviate the thrashing condition. In general, a probing flow should not assume that all measured available bandwidth is for itself alone, as this bandwidth will be shared among other probing flows. As TCP-LP partitions available bandwidth fairly among TCP-LP flows, this problem is eliminated.

Finally, the related algorithms from the area of protection against DoS attacks can be found in references [23, 24, 25, 26]. These algorithms first try to detect the misbehaving flows in a network router. A common assumption is that the misbehaving flows continually send packets at peak rates over long time intervals and thus cause persistent congestion. However, one of the hypotheses of this thesis is that DoS attacks can be performed by flows that send at peak rates only over short time intervals and which have low average rate. Next, the above algorithms aim to actively control the misbehaving flows at a network router once they are detected. In

contrast, this thesis proposes improvements to the existing end-point TCP protocol. These improvements present a *preventive* anti-DoS mechanism that should be used together with other existing protection mechanisms, including those from [23, 24, 25, 26].

5 Thesis Status

The progress of this thesis work is as follows.

Contribution I-A: Completed, for details and results refer to [5].

Contribution I-B: Completed, for details and results refer to [6].

Contribution II-A: Completed, for details and results refer to [8].

Contribution II-B: In progress.

Contribution III: In progress.

References

[1] J. Saltzer, D. Reed, and D. Clark, "End-to-end arguments in system design," *ACM Transactions on Computer Systems*, vol. 2, no. 4, pp. 195–206, Nov. 1984.

[2] J. Qiu and E. Knightly, "Inter-class resource sharing using statistical service envelopes," in *Proceedings of IEEE INFOCOM '99*, New York, NY, Mar. 1999.

[3] M. Allman and V. Paxson, "On estimating end-to-end network path properties," in *Proceedings of ACM SIGCOMM '99*, Vancouver, British Columbia, Sept. 1999.

[4] V. Paxson and M. Allman, "Computing TCP's retransmission timer," Nov. 2000, Internet RFC 2988.

[5] A. Kuzmanovic and E. Knightly, "Measuring service in multi-class networks," in *Proceedings of IEEE INFOCOM '01*, Anchorage, Alaska, Apr. 2001.

[6] A. Kuzmanovic and E. Knightly, "Measurement based characterization and classification of QoS-enhanced systems," *to appear in IEEE Transactions on Parallel and Distributed Systems*.

[7] M. Aron, P. Druschel, and W. Zwaenepoel, "Cluster reserves: A mechanism for resource management in cluster-based network servers," in *Proceedings of ACM SIGMETRICS '00*, June 2000.

[8] A. Kuzmanovic and E. Knightly, "TCP-LP: A distributed algorithm for low priority data transfer," in *Proceedings of IEEE INFOCOM '03*, San Francisco, CA, Apr. 2003.

[9] S. Sarvotham, R. Riedi, and R. Baraniuk, "Connection-level analysis and modeling of network traffic," in *Proceedings of IEEE/ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, Nov. 2001.

[10] R. L. Carter and M. E. Crovella, "Measuring bottleneck link speed in packet-switched networks," *Performence Evaluation*, vol. 27:28, pp. 297–318, 1996.

[11] V. Jacobson, "Pathchar: A tool to infer characteristics of Internet paths," *ftp://ftp.ee.lbl.gov/pathchar/*, Apr. 1997.

[12] A. Downey, "Using pathchar to estimate Internet link characteristics," in *Proceedings of ACM SIGCOMM '99*, Vancouver, British Columbia, Sept. 1999.

[13] C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet dispersion techniques measure?," in *Proceedings of IEEE INFOCOM '01*, Anchorage, Alaska, Apr. 2001.

[14] V. Paxson, "End-to-end Internet packet dynamics," *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 277–292, June 1999.

[15] K. Lai and M. Baker, "Measuring bandwidth," in *Proceedings of IEEE INFOCOM '99*, New York, NY, Mar. 1999.

[16] K. Lai and M. Baker, "Measuring link bandwidths using a deterministic model of packet delay," in *Proceedings of ACM SIGCOMM '00*, Stockholm, Sweden, Aug. 2000.

[17] D. Wrege and J. Liebeherr, "Video trafic characterization for multimedia networks with a deterministic service," in *Proceedings of IEEE INFOCOM '96*, San Francisco, CA, Mar. 1996, pp. 537–544.

[18] J. Liu and M. Crovella, "Using loss pairs to discover network properties," in *Proceedings of IEEE/ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, Nov. 2001.

[19] V. Ribeiro, M. Coates, R. Riedi, S. Sarvotham, B. Hendricks, and R. Baraniuk, "Multifractal cross-traffic estimation," in *Proceedings of ITC '00*, Monterey, CA, Sept. 2000.

[20] S. Alouf, P. Nain, and D. Towsley, "Inferring network characteristics via moment-based estimators," in *Proceedings of IEEE INFOCOM '01*, Anchorage, Alaska, Apr. 2001.

[21] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput," in *Proceedings of ACM SIGCOMM '02*, Pittsburgh, PA, Aug. 2002.

[22] L. Breslau, E. Knightly, S. Shenker, I. Stoica, and H. Zhang, "Endpoint admission control: Architectural issues and performance," in *Proceedings of ACM SIGCOMM '00*, Stockholm, Sweden, Aug. 2000.

[23] R. Mahajan, S. Floyd, and D. Wetherall, "Controlling high-bandwidth flows at the congested router," in *Proceedings of IEEE ICNP '01*, Riverside, CA, Nov. 2001.

[24] R. Mahajan, S. Bellovin, S. Floyd, J. Ioannidisand, V. Paxson, and S. Shenker, "Controlling high bandwidth aggregates in the network," *Technical report*, Feb. 2001.

[25] D. Yau, J. Lui, and F. Liang, "Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles," in *Proceedings of IWQoS '02*, Miami, FL, May 2002.

[26] F. Ertemalp, D. Chiriton, and A. Bechtolsheim, "Using dynamic buffer limiting to protect against belligerent flows in high-speed networks," in *Proceedings of IEEE ICNP '01*, Riverside, CA, Nov. 2001.