Weaving Measurements Into Internet Protocols and Services

Research Statement

Jasleen Kaur
Department of Computer Science
University of North Carolina at Chapel Hill

1 Research Theme

The Internet has witnessed an unimaginable growth in its infrastructure as well as usage over the past two decades—a network that barely had a thousand hosts now connects hundreds of millions of hosts and billions of users. They key enabler of this growth is the distributed and simple design of the Internet—there is no central authority that controls or maintains state about the Internet. While this stateless design has helped with infrastructure growth, it comes at the cost of increased complexity for Internet applications—when an application runs over the Internet, it has no apriori knowledge of the state of the network it is getting into. This lack of knowledge about the diverse and unpredictable Internet can result in a fairly inefficient usage of both application as well as network resources. In fact, the short life of the Internet is filled with several examples of this—a famous example is that of the "congestion collapse" witnessed in the late 1980s in which unaware applications would continue to pump data into an overloaded network at rates it could not handle.

One time-tested way of dealing with an unpredictable system is to "measure" its state and adapt to it. In the context of Internet applications, this boils down to designing protocols and services that can be used to probe for the current state of a given network path. Due to lack of support from the simple network core, this fundamentally involves estimating internal-network properties from externally-observable characteristics. Several challenges—related to accuracy, timeliness, overhead, and scalability—complicate this task. Over the past 6 years, my research has focused on addressing these challenges in order to design monitoring infrastructures, measurement techniques, protocols, and distributed services that help applications scale with the growth of the Internet. Below, I briefly summarize the research activities I've pursued at UNC.

Most of these activities have been generously supported by an NSF CAREER award and start-up funds from the University of North Carolina.

2 Research Contributions

Scalable Monitor Placement for Network Tomography

Issues Network tomography attempts to scale the task of monitoring links in a large network by using a small set of monitor nodes to probe for several remote target links. This is done by sending two closely-spaced probes along the IP routes from the monitor to the two end-points of a target link, only one of which traverses the link—link delay and loss rates are then estimated from the differences in the two probe results. The cost-saving benefits of this paradigm can be fully realized only if the number of monitors needed for completely monitoring a target network is fairly small. Consequently, recent tomographic research has focused on designing monitor placement strategies that typically identify the set of links that can be monitored by a candidate network node and then place monitors at the smallest set of nodes that can collectively monitor a complete target network.

Unfortunately, past monitor placement strategies have considered mostly statically-routed networks and are not applicable to most of current networks, in which routes can change every few minutes. Furthermore, the routing policies within individual networks may be considered proprietary information that may not be available to monitoring efforts that span multiple networks. This complicates the search for a static and small set of monitors.

Contributions In our research, we incorporate dynamically-routed networks by asking the question: *which set of links can be monitored by a candidate node, independent of the state of IP routes?* We address this question by developing the concept of *deterministically monitorable edge set* (DMES) for each network node as the set of links that can be monitored by the node under *all* possible routing configurations. The framework is fairly generic and allows us to evaluate existing monitor placement strategies as well as incorporate different node types and routing policy constraints.

Through graph-theoretic analysis, we then show that the set of all links in a network can be partitioned into two mutually-exclusive categories: the first category is of links that fall in the DMES of *all* network nodes—these links can be monitored by *any* remote node, independent of the IP routing state; the second is the set of links that belong to the DMES of *only* the two nodes directly attached to the link. We use this framework to first compute the DMES of each node in a given network and then find the smallest set of nodes that can collectively monitor all network links in the presence of dynamic routing.

We use this framework to derive efficient beacon placements for several real-world ISP topologies estimated by the Rocketfuel project and show that all links in these topologies can be monitored by merely 5-20% of nodes—this represents a reduction of more than 50% compared to existing beacon placement frameworks for dynamically-routed networks. We also show that our algorithms yield solutions that are fairly close to optimal in the number of monitor nodes needed.

Representative Publications

1. R. Kumar and J. Kaur, "Practical Beacon Placement for Link Monitoring Using Network Tomography", *in the IEEE Journal on Selected Areas in Communication*, special issue on *Sampling the Internet: Techniques and Applications*, volume 24, number 12, Dec 2006.
2. R. Kumar and J. Kaur, "Efficient Beacon Placement for Network Tomography", *in Proceedings of the ACM SIGCOMM Internet Measurement Conference*, Sicily, Italy, October 2004.

Passive Analysis of TCP Loss Detection/Recovery

Challenges A powerful approach to understanding the real-world performance of network protocols is to passively analyze network-level packet traces of Internet transfers—if traces are captured at traffic aggregation points, a fairly diverse and large set of transfers can be sampled. This approach has proved to be extremely valuable in understanding the performance of TCP, the transport protocol that carries more than 90% of Internet traffic. Existing techniques for passive analysis of TCP traces emulate a reference implementation and configuration model for TCP senders that conforms to the proposed standards for TCP. Unfortunately, current TCP implementations developed by different operating system (OS) vendors may differ (sometimes significantly) in either their interpretation or their conformance to TCP specifications/standards. Consequently, existing tools are ineffective in accurately analyzing traces of contemporary TCP transfers—this can be achieved only by conducting the analysis in an OS-sensitive manner.

Conducting an OS-specific trace analysis is, however, challenging for two main reasons. First, for any given TCP trace, it is non-trivial to identify the corresponding sender-side OS and decide which OS-specific analysis to apply to it. Second, most OSes either have proprietary code or have insufficient documentation on their TCP implementation details. Without detailed knowledge of these implementations, it is not possible to emulate these in our OS-specific analysis tools.

Approach We extract sufficient details about TCP implementations in 4 prominent OS stacks—namely, Windows XP, Linux, FreeBSD/Mac OS-X, and Solaris—by *reverse-engineering* these. For this, we develop a controlled TCP receiver that initiates TCP connections to a known server machine for each of these OSes and artificially generates a response stream that triggers different aspects of the TCP on the servers. The server response is then analyzed to infer details of the corresponding TCP stack.

We use this extracted information to develop a set of 4 OS-specific trace analysis tools. Each TCP trace is then analyzed by all 4 tools—the tool that is able to completely and unambiguously explain all events in a given trace is then used to study TCP performance in the corresponding transfer.

To the best of our knowledge, this is the *first* tool that analyzes TCP mechanisms in an OS-sensitive manner.

Impact We use our OS-specific analysis tools to analyze traces of nearly 3 million Internet transfers and study the performance of TCP loss detection/recovery mechanisms. To do this, we first formulate the fundamental trade-off between *accuracy* and *timeliness* of TCP loss detection, which impose conflicting requirements on TCP design parameters. We vary these parameters and rely on our passive analysis tools to estimate the resultant impact on accuracy and timeliness of loss detection. We then develop analytical models to quantify the impact of these properties on the overall transfer times experienced by a TCP connection. Our analysis finds that the recommended as well as in-use settings for these parameters can be fairly sub-optimal in ensuring small transfer times. We do an exhaustive search and identify a "SmartConfig" parameter configuration that helps reduce the transfer times of up to 40% of Internet transfers by more than 10%.

To the best of our knowledge, this is the *first* study that quantifies the impact of TCP-level mechanisms on application-level metrics such as transfer times in real-world transfer. Some of the results from our analysis are being used to guide the design of loss detection mechanisms in the Linux-based implementation of the recently-developed DCCP CCID-2 protocol. Gerrit Renker (University of Aberdeen) is leading this effort.

Representative Publications

1. S. Rewaskar, J. Kaur, and F.D. Smith, "A Performance Study of Loss Detection/Recovery in Real-world TCP Implementations", *in Proceedings of IEEE International Conference on Network Protocols*, Beijing, China, Oct 2007.
2. S. Rewaskar, J. Kaur, and F.D. Smith, "A Passive State-machine Based Approach for Reliable Estimation of TCP Losses", *in the ACM SIGCOMM Computer Communications Review*, volume 36, issue 3, July 2006.

Evaluation of Bandwidth Estimation Techniques

Issues It is often claimed that several Internet protocols and services can benefit from the knowledge of end-to-end available bandwidth on a given network path. While this has motivated considerable recent research on the design of bandwidth probing techniques, few of these have found their way into (improving) protocols/services. There are at least two key factors responsible for this state of affairs. First, it is not clear which among the myriad of tools available is better than the others—most existing tool evaluations are not comprehensive and are often biased by differences in implementation inefficiencies. Second, the interfaces of most tools are not in tune with application semantics—indeed, most tool designs do not consider critical application requirements on the measurement timescales and sampling intensities of interest. As a result, protocol designers are not well-informed about either which techniques work well in practice, or even how best to adapt and incorporate a technique in the protocol mechanisms. We address this state of affairs in the field of bandwidth estimation by making two main contributions summarized below.

Contributions Our first main contribution is a comprehensive experimental evaluation of a wide selection of bandwidth estimation techniques. Two key features make our evaluations fairly useful. First, we eliminate the impact of implementation inefficiencies by developing and relying on a common reference implementation framework in a simulated environment. Second, we consider performance under fairly diverse network, traffic, and probing conditions. These features prove fairly useful, as some of our findings contradict those of past work—techniques that were previously reported to under-perform, in fact, perform quite well once considered without implementation inefficiencies and under the same probing semantics as other tools.

Our second major contribution to the area has been to study the impact of several probing-related temporal factors—including measurement timescales, sampling intensity and strategies, and tool run-time—on: (i) the accuracy, stability, and overhead of the bandwidth estimation process, and (ii) the application-centric redesign of tool interfaces. We show that, unlike the current practice, these factors should be given prime consideration in the tool design and interface—indeed the accuracy and overhead of existing tools varies significantly with these factors.

We have used the lessons learned from the above studies to see how bandwidth-estimation techniques can be used in the design of congestion-control as well as scalable monitoring infrastructures (each described in what follows).

Representative Publications

1. A. Shriram and J. Kaur, "Empirical Evaluation of Techniques for Measuring Available Bandwidth", *in Proceedings of the 26th IEEE INFOCOM*, Anchorage, AK, May 2007.
2. A. Shriram and J. Kaur, "Empirical Study of the Impact of Sampling Timescales and Strategies on Measurement of Available Bandwidth", *in Proceedings of the Seventh Passive and Active Measurement Conference*, Adelaide, Australia, March 2006.

RAPID: Shrinking the Congestion-control Timescale

Issues TCP congestion-control adopts a fairly slow bandwidth-search process that prevents it from efficiently utilizing end-to-end spare bandwidth in high-speed and dynamic environments. Unfortunately, while several alternate congestion-control protocols have been proposed to speed up the search process, most of these struggle to remain non-intrusive to cross-traffic while achieving high speed—consequently, even these "high-speed" protocols may still take hundreds to thousands of RTTs in searching for available bandwidth in 1-10Gbps networks. With typical error-based loss rates, such protocols can utilize no more than 60% of the spare bandwidth on 10G networks.

We argue that the basic design framework of window-based transmission and control, that has been adopted as an unquestioned legacy by most recent end-to-end protocols, is responsible for this poor scalability behavior. In fact, we show that this legacy framework is responsible for at least two other limitations of even the very best of end-to-end congestion-control designs: (i) that of poor fairness properties in heterogeneous RTT environments and (ii) that of poor friendliness to conventional TCP traffic aggregates.

Contributions In our research, we adopt a truly rate-based congestion-control paradigm as a first-order concept. Our resultant end-to-end congestion-control framework allows TCP connections to boldly search for, and adapt to, the available bandwidth within *a handful of RTTs*—this represents at least an order of magnitude of improvement over current protocols. Our key insight is to rethink the timescale at which congestion-control probes for available bandwidth—we show that by shrinking this timescale, it is possible to design a protocol that achieves a high bandwidth-search speed without significantly overloading the network. Our resultant approach relies on carefully orchestrated inter-packet gaps at the sender—that help quickly probe for several different rates using only a few packets—and estimates the available bandwidth based on gap increases at the receiver.

We use this framework to design a new protocol, referred to as RAPID, using mechanisms that promote efficiency, queue-friendliness, and fairness. Our experimental simulations with 1-10Gbps networks indicate that RAPID: (i) converges to an updated value of bandwidth within 1-4 RTTs; (ii) is fairly efficient in utilizing rapidly-varying spare bandwidth; (iii) helps maintain fairly small queues; (iv) has negligible impact on co-existing conventional TCP traffic aggregates; and (v) exhibits excellent fairness among co-existing RAPID transfers. The rate-based design allows RAPID to be truly RTT-fair.

Representative Publications

1. V. Konda and J. Kaur, "RAPID: Shrinking the Congestion Control Timescale", *to appear in the Proceedings of the 28th IEEE INFOCOM*, Rio de Janeiro, Brazil, April 2009.
2. V. Konda and J. Kaur, "Rethinking the Timescales at Which Congestion-control Operates", *in Proceedings of the 16th IEEE Workshop on Local and Metropolitan Area Networks*, Transylvania, Romania, September 2008.

Scalable Bandwidth Monitoring Infrastructures

Issues Several distributed applications—including content-distribution, overlay routing, and peer-to-peer downloads—are likely to benefit from a network monitoring service that provides the end-to-end available bandwidth between nodes. Due to the heavy-duty nature of bandwidth-estimation tools, doing an all-pairs bandwidth probing between all participating nodes is, however, not a scalable approach for such a monitoring service—frequent n^2 probing would impose significant probe-traffic overhead on the underlying network. Furthermore, the time taken to complete n^2 available bandwidth measurements can be prohibitively large for even moderately-sized networks—this limits the frequency with which available bandwidth measurements can be updated. While the problem of designing scalable monitoring services has received considerable attention in the recent past for metrics such as end-to-end latency, the issue of estimating end-to-end available bandwidth in a scalable yet accurate manner has not been solved adequately. This task is especially challenging because of the high overhead and run-time of most bandwidth-estimation tools, and the fact that available bandwidth can vary rapidly at relatively short timescales.

Contributions In our research, we propose and evaluate an alternative scheme for scalable available bandwidth monitoring of a network. Our approach called SABI (Scalable Available Bandwidth Inference) is based on a two-step approach: (i) we group together nodes with similar views to the rest of the nodes (that are likely to share bottleneck links), and (ii) we select a well-provisioned head-node for each cluster of nodes. Measurements are made from the group-heads to nodes outside the cluster, while the members of a group make measurements to the group-head. The available bandwidth between any pair of nodes is then inferred by using these measured values. We develop three SABI algorithms, that incur decreasing probe overhead while maintaining reasonable inference accuracy. We evaluate our architecture on Planet-Lab using a network sensing service called S^3. Our results show that our SABI algorithms can reduce the number of measurements to O(N), while incurring an average estimation error within 15%—this significantly outperforms previous techniques that typically incur 50% estimation error.

Representative Publications

1. A. Shriram, S. Banerjee, J. Kaur, and P. Yalagandula, "Scalable End-to-end Available Bandwidth Inference", *pending submission*.

Sizing Router Buffers with Queue-friendly Congestion-control

Issues Two aspects of TCP NewReno congestion-control—the dominant protocol used in the Internet—can lead to large packet queues in network routers. First, due to its reliance on only packet losses for detecting congestion, TCP induces large queue buildups in routers—a TCP transfer will keep increasing its sending rate till buffers overflow and drop packets. Second, due to the aggressive reaction of TCP to packet losses (multiplicative decrease), a network carrying TCP transfers can maintain high link utilization only by provisioning large buffers in its routers. Thus, TCP requires large buffers *and* keeps them persistently full.

There have been two kinds of significant efforts in the research community to address these issues. The first is the design of alternate congestion-control strategies that use "early" feedback other than packet losses for congestion-detection—the hope is that the use of such strategies will cause smaller router queues than TCP. Unfortunately, when evaluated with representative TCP traffic mixes, each of the prominent protocols fails to *simultaneously* maintain small queues and high TCP throughput.

A second significant effort has been in the formulation of alternate strategies for router buffer provisioning—these argue that routers carrying large aggregates of TCP traffic can maintain high link utilization even with fairly small buffers. Unfortunately, other studies have shown that such routers can experience high packet losses and can adversely impact TCP throughput.

Contributions In our research, we address the above issues by taking a two-pronged approach: first, we explore the design of alternate congestion-control protocols that can successfully induce small queue buildups in routers without limiting TCP throughput. Our key insight is to (i) rely on a frequent router-assisted congestion

signal in order to ensure that TCP responds to queue buildups before they grow very large, and (ii) use a less aggressive (additive decrease) response to a congestion signal, in order to ensure high link utilization even with small router queues.

We then develop a buffer provisioning strategy that relies on the queue-friendliness of such protocols and estimates the buffer space needed for only accommodating *asynchrony* between an aggregated mix of TCP transfers. We show that when our resultant buffer provisioning rule is used in a network running queue-friendly congestion-control, it can help maintain high utilization as well as high TCP throughput—while requiring only small router buffers. This work provides a vital perspective on the open issue of router buffer sizing.

Representative Publications

1. R. Kumar and J. Kaur, "Towards a Queue Sensitive Transport Protocol", in Proceedings of the 27th IEEE International Performance Computing and Communications Conference, Austin, TX, December 2008.

2. R. Kumar and J. Kaur, "Additive Increase Additive Decrease: A Vital Perspective on Router Buffer Sizing", *pending submission*.

3 Research Vision

My research over the next 5-7 years will be guided by the desire to de-mystify the enormous Internet traffic beast. Specifically, I will focus on: (i) better understanding (and controlling) the interactions between factors operating at the network-, transport-, and application-layers, and (ii) studying their impact on the burstiness and queuing behavior of traffic aggregates. I will continue to rely on an empirical, modeling, and measurement-driven approach. I believe that such an investigation will lead to radically-improved protocol designs and will help address the scalability of the Internet infrastructure.

I hope to continue to benefit from the excellent resources and expertise in related fields (especially modeling expertise in the Department of Operations Research and Statistics, and experimental expertise in RENCI) that UNC offers.