Statement of Research for Nandita Dukkipati

nanditad@stanford.edu

The Internet has been a remarkable success, yet in many ways it is woefully inadequate in meeting the needs of applications and users. Its performance is unpredictable, it is frequently unavailable when we need it, and it does not easily support new technologies such as wireless or evolve with changing needs of users and applications. There are applications we would not think of running on it, such as a remote health monitoring and diagnostic service which requires location awareness, ubiquitous and seamless connectivity, and high availability while being able to support a large number of devices. The focus of my research is to make networks more adaptable, evolvable, self-managed, and cost-efficient, as well as improve their performance so as to make these applications feasible. While my primary goal is to innovate and build practical solutions, I also enjoy the approach of using theoretical tools wherever applicable.

My current work is on designing a novel congestion control algorithm for the Internet that is appreciably better than the widely used TCP. I describe this work below followed by my ideas for future research. A lot of my work has immensely benefited through collaboration with my colleagues.

Current and Past Research

My research focus has been in the area of resource allocation in networks. For my Ph.D. dissertation, I designed a new congestion control mechanism - Rate Control Protocol (RCP) - for fast download times (aka user response times, or flow-completion times), that is also significantly more efficient, stable, adaptive, and robust compared to the widely used TCP.

Congestion control solves the problem of sharing network resources in a distributed, cooperative, and fair fashion. It is unique to Networking and integral for the Internet to function well. Lack of mechanisms to share bandwidth effectively under over-loaded conditions in the early Internet led to several network meltdowns, known as congestion collapses. It spurred a large body of work on how to prevent a congestion collapse, nearly all of which universally focusing on metrics such as throughput, bottleneck utilization, fairness, packet drops/delay, and network stability. While these metrics are interesting – particularly for the network operator – they are not very interesting to a network user. When users download a web page, transfer a file, send/read email, or involve the network in almost any interaction, they want their transaction to complete in the shortest time; and therefore, they want the shortest possible flow completion time (FCT). Today, most transactions are of this type and it seems likely that a significant amount of traffic will be of this type in the future. That is the problem I wanted to explore: *how do we design congestion control algorithms and mechanisms that minimize flow-completion times?* Unfortunately, it is intractable to minimize flow-completion times. So instead congestion control algorithms are focussed on an easier to achieve goal of efficiently using bottleneck links. But, I believe that instead of being deterred by the complexity of the problem, we should find algorithms that come close to minimizing FCTs, even if they are heuristic.

I discovered that with typical Internet flow sizes, existing (TCP Sack) and newly proposed (XCP) congestion control algorithms make flows last much longer than necessary - often by one or two orders of magnitude. RCP makes flows finish close to the minimum possible, leading to a perceptible improvement for web users, distributed computing, and distributed file-systems. It achieves this by explicitly emulating Processor Sharing at each router. In RCP, a router assigns a single rate, R(t), to all flows that pass through it. I designed RCP to be simple and practical - it is an adaptive algorithm that updates the rate assigned to the flows, to approximate processor sharing in the presence of feedback delay, without any knowledge of the number of ongoing flows. The flow rate, R(t), is picked by the routers based on very little information (the current queue occupancy and the aggregate input traffic rate), requires no per-flow state or queue, and has very few per-packet computations.

Congestion control must ensure the network's health and stability even under abnormal and extreme traffic patterns. Using techniques in control-theory, I showed that RCP is provably stable under severe network overloads, irrespective of round-trip times, link-capacities, and the number of flows. Not only is it stable, it also recovers to normal behavior within a small number of round-trip times even when confronted with sudden and unanticipated network changes.

Determining the correct size for router buffers has piqued a lot of interest recently because of its profound implications on router architecture and power consumption, flow performance, and network stability. It has been demonstrated experimentally and shown theoretically that TCP congestion control works well even under very small buffers. I demonstrated that RCP does not need huge buffers to deliver its promise of short FCTs. Along with a colleague, I showed that even with buffers as small as 5% of the bandwidth-delay product, flow-completion times are at most 20% longer than with very large buffers for a wide range of network and traffic characteristics.

One of the promises of RCP is the simplicity and practicality of its implementation in routers and end-hosts. Along with colleagues at Stanford, I demonstrated that RCP's implementation is practical in both the end-hosts as well as the routers.

Some newer applications in the Internet, such as High Definition video streaming, need extremely low packet-loss to operate well. I extended RCP to RCP-AC (Rate Control Protocol with Acceleration Control) which can be tuned to achieve low packet-loss and extremely well-behaved queues even under sudden and large traffic changes at one end of the spectrum and short flow-completion times at the other end. RCP-AC is designed with optimization in mind, and is adaptable to changing traffic patterns and expectations of applications using it.

I spent a significant amount of time in my Ph.D. to explore the impact RCP can have on Internet congestion control. Feedback from practitioners in networking industries pointed out that RCP congestion control for short download times is not only a good idea for the Internet, but is also useful in niche networks with atypical network and traffic characteristics such as satellite and data-center networks.

In the past, I had also worked on mechanisms to deliver hard QoS guarantees. A key issue in delivering Quality-of-Service guarantees is resource allocation at the network nodes to satisfy the end-to-end delay requirements. I devised and implemented algorithms to determine the optimal capacity allocation satisfying end-to-end delay requirements in a network of Generalized Processor Sharing (GPS) schedulers. I demonstrated that these algorithms outperform previously proposed algorithms.

Future Research

Over the years many applications have proliferated over the Internet ranging from short web downloads to high bandwidth high-definition video streaming. However, there is a much richer set of applications we would like to run in the future including those to monitor health remotely, manage, store, and share vast amounts of personal data online, discover local services from mobile clients and vehicles, and facilitate rich interaction among people spanning continents. While on one hand these represent scaling up from where we are to a much better Internet, more than half of the world does not even have basic web access and the technical challenges to overcome are of a different nature including making the infrastructure cost-effective, power-efficient, and tolerant to intermittent disruptions. As it stands, our current infrastructure cannot support either of these two extremes. My future research in broad terms will be on designing protocols and mechanisms to improve our networks' performance, make them more scalable, evolvable and adaptable while lowering costs, to get us from where we are to where we would like to be. I have listed below a cross-section of the problems I am inspired to work on in the near term:

Designing protocols for different network requirements

Networks with requirements very different from those of main-stream Internet are proliferating such as storage area networks and networks in data-centers, enterprises, homes, automobiles, and planes. They all have very different needs of speed, reliability, operating costs, and network conditions (such as very high or low round-trip times, link-speeds, unusual traffic patterns, and different wireless technologies). Yet the same mechanisms and protocols designed for the long-haul Internet are being retrofitted to work in these environments. As a consequence they are inefficient at best and simply do not work at worst.

As an example: Packet loss due to temporary network congestion is only a minor annoyance in the Internet and up until now there was no motivation to achieve a loss-free network. After all, not only does TCP retransmit lost packets but loss is its only indication of congestion. However, networks where even occasional congestive losses is no longer an option are proliferating. Examples include edge networks used for High Definition video-on-demand transfers where current practice is to vastly over-provision links so as to achieve a loss-free environment. This is not only an expensive solution but one that does not scale as video transfers become more wide-spread. Similarly, performance of distributed applications in data-centers takes a big hit due to packet losses and so a loss-free network is highly desirable. Clearly, TCP congestion control is not a good answer for either of these networks. I want to explore routing and congestion control mechanisms to fulfill the needs of these newer applications on networks with very different requirements.

Leveraging different infrastructures to provide efficient and low-cost connectivity for mobile users

Connectivity is fundamental. We have invested in abundant connectivity and there is more being put in place. We have multiple broadband, cellular (3G, 4G), Mesh networks, Satellite, WiMax, and Wi-Fi systems. There are many open access points available and more coming up, and while the coverage is not complete, it gives a sense of what is possible. Recent developments in software radios allow a single device to be reconfigured on the fly to access all of these networks. We have these multiple parallel infrastructures in place, but what users eventually care about is effective connectivity that also satisfies their needs. For example, it is inconceivable today for mobile users to seamlessly switch between blasting packets all the way to cell towers and using open Wi-Fi and mesh networks where available. The Internet does not support this and is completely unaware of the ephemeral mobile nodes.

I am interested in developing protocols that would leverage these different infrastructures to provide seamless connectivity that would also meet users' needs, especially for mobile users. Needs can be of different forms: lowest cost, best reliability, highest speed possible or lowest latency. Protocols should be able to adapt quickly to changing network conditions. How do we design mechanisms and protocols that would be native to this dynamic view of the infrastructure? One of the challenges would be to come up with adaptive protocols that anticipate discontinuous connectivity and packet loss and achieve efficient resource usage in such an environment.

Scaling networks to support Peer-to-Peer traffic

Peer-to-Peer (P2P) file-sharing has often been criticized because of its association with illegal distribution of copyrighted content. I am intrigued by P2P networks because of the possibilities they open in providing low cost communications without the need of a huge managed infrastructure. As more users download video and other high bandwidth content, the architecture of today's infrastructure with large data-centers located near cheap electricity with thousands of PCs, disk drives, uninterruptible power supplies, and lots of fiber can get prohibitively expensive. P2P changes the economics of the game and not surprisingly many commercial content providers (including Hollywood, BBC, Linux distributions) are increasingly providing their content via P2P networks.

P2P applications hog bandwidth and it is estimated that they account for 60 to 80% of traffic on consumer ISP networks. Service providers complain that networks are frequently clogged with P2P traffic, and yet, not all parts of the network are equally congested at the same instant. I want to explore solutions that will scale networks to support P2P traffic and treat it as a first-class application as opposed to unwanted traffic. For example, I want to investigate mechanisms that combine routing, traffic engineering and congestion control to diffuse P2P traffic in real-time to make use of paths with available bandwidth, deflect it away from congested paths, and use more of local bandwidth.

Integrating evolution into network protocols and mechanisms to adapt to unanticipated changes

The Internet's protocols and mechanisms are resilient in the sense that they continue to function even under unexpected changes in operating conditions. However, they do not necessarily function efficiently. While we are generally good at making a specific protocol work efficiently for some specific range of operating conditions, we do not quite know how to make them continue working efficiently under unanticipated changes. This is especially important in networks because unlike many traditional systems they do not have the luxury of being replaced very frequently.

The question I am interested in is: how do we integrate evolution in network controls including congestion control, Active Queue Management schemes, routing protocols, and traffic engineering, such that they are adaptable, tunable and can be optimized to be efficient under changing (and often unanticipated) conditions and constraints. Changes can be of different forms: traffic patterns, user behavior and expectations, and radically different applications. I want to explore solutions by drawing techniques from different disciplines including Machine learning, Statistics, Algorithms, Optimization Theory, and Distributed Systems.

The above problems present a flavor of the research I am inspired to work on. They all share one or more of the following themes at their core: creating solutions to lower costs; making networking systems more efficient, adaptable, and evolvable. I enjoy innovating solutions that appeal to theorists and make practical sense at the same time. In the past, I had a fulfilling experience collaborating with disciplines outside of Networking and I will actively pursue the same for my future research. In addition, I am immensely interested in collaborating with practitioners in Industry to both learn of realistic problems as well as to find a way for my research to have an impact in the real world.