Research Statement for Lili Qiu

September 2000

My research focuses on characterizing Web workloads, and developing techniques to enhance Web performance. I have analyzed the workload of busy Internet Web servers, and developed a range of techniques to improve Web performance. The techniques I have developed span several layers in the protocol stack - the application layer, transport layer, and network layer. In particular,

- At the application layer, I explore the problem of Web server replica placement, and develop several placement algorithms that use workload information, such as client latency and request rates, to make informed placement decisions.
- At the transport layer, I propose a new technique, called TCP/SPAND, to speed up Web transfers.
- At the network layer, I propose several new techniques to speed up packet classification, which has been the major bottleneck in the high-speed network.

Such an integrated approach to improving Web performance is necessary in order to get the best possible performance. Below I describe this work, as well as give an overview of my other work in the networking field, and a description of my future directions.

1 Web Workload Characterization

With the exponential growth in World Wide Web, there is a rapidly growing demand to properly design and provision Web systems. Critical to system design is a solid understanding of the nature of workload and system demands created by the Web users. Much of the work thus far in Web workload characterization has focused on proxies. Server-based studies are far fewer in number, and traditionally small departmental Web server logs were used to study the workload at the servers. It is hard to generalize the results from the study of small departmental Web servers to those large Web servers, which serve the majority of Web requests in today's Internet. In [1], we study the dynamics of the MSNBC news site, one of the busiest Web sites in the Internet today. We study both the front-end (i.e. how users access the Web content) and the back-end (i.e. how contents are created and modified at the Web site) of the MSNBC site. We find that (a) past modification history yields a rough predictor of future modification behavior, (b) modification between successive versions is small, (c) there is significant temporal stability in file popularity but not much stability in the domains from which clients access the popular content, (d) file popularity follows a Zipf-like distribution with a parameter α that is much larger than reported in previous, proxy-based studies, (e) accesses to old documents account for most first-time misses, which implies it is hard to anticipate such accesses, and eliminate these first-time misses. We also study the implications of these findings for techniques such as Web caching (including cache consistency algorithms), and prefetching, or server-based "push" of Web content.

2 Web Replication

Two primary application-layer techniques for enhancing Web performance are caching and replication. Previous work has studied many aspects of caching and replication, such as object routing, object distribution, object selection, inter-replica or inter-proxy communication, and policy management, etc. However, less attention has been given to the placement of Web proxies or Web replicas. In [5], we explore the problem of Web server replica placement in detail. We develop several placement algorithms that use workload information, such as client latency and request rates, to make informed placement decisions. We then evaluate the placement algorithms using both synthetic and real network topologies, and several access traces from large commercial and government Web servers. Based upon our results, we conclude that a greedy algorithm for solving the Web server replica placement problem can provide content distribution networks with the performance that is close to optimal. Although the greedy algorithm depends upon estimates of client distance and load predictions, we find it is relatively insensitive to errors in these estimates and therefore is a viable algorithm for use in the general Internet environment where workload and topology information will always be imperfect. To our knowledge, this is the first experimental study and evaluation on this subject.

We also examine an orthorgonal question in Web replication - how to redirect users' requests to the edge servers in CDN for better performance. We are currently developing new selection algorithms and redirection techniques, as well as evaluating their performance, and comparing them with the existing approaches.

3 Optimizing TCP Performance for Short Transfers

Web data transfers are usually very short. Such a workload is well known to interact poorly with the TCP protocol. TCP uses the slow start procedure to probe the network for bandwidth both at connection start up and upon restart after an idle period. This usually requires several roundtrips and is inefficient when the duration of a transfer is short.

In [2], we propose a new technique, called TCP/SPAND, to speed up short data transfers. In TCP/SPAND, network performance information is shared among many co-located hosts to estimate each connection's fair share of the network resources. Based on such estimation and the transfer size, the TCP sender determines the *optimal* initial congestion window size. Instead of doing slow start, it uses a pacing scheme to smoothly send out the packets in its initial congestion window. We use extensive simulations to evaluate the performance of the resulting system. Our results show that TCP/SPAND significantly reduces latency for short transfers even in presence of multiple heavily congested bottlenecks. Meanwhile, the performance benefit does not come at the expense of degrading the performance of connections using the standard TCP (i.e. TCP/SPAND is TCP friendly).

4 Fast Packet Classification

The exponential growth in Web traffic not only demands efficient techniques at both the transport and application layers, but also demands radical advances in the more fundamental network technology. In particular, we need high speed links, high router data throughput, and high packet forwarding rates. Readily available solutions exist for the first two: for example, fiber-optic cables can provide faster links, and switching technology can be used to move packets from the input interface of a router to the corresponding output interface at gigabit speeds. But efficient packet forwarding is still an open problem. The traditional packet forwarding, which is soly based on the destination address, has been well studied. However, users are demanding, and some router vendors are providing, a more discriminating form of router forwarding. For example, as many people envision, in the future, routing will not only be based on destination address, but also be based on source host or even source users, as well as destination URLs and specific business policies etc. This new vision of forwarding is called Layer 4 Forwarding, since routing decisions can be based on headers available at Layer 4 or higher in the OSI architecture. Layer 4 forwarding enables service differentiation in a connectionless network. For example, in the emerging diffserv architecture core routers classify packets based only on IP TOS fields; however, edge routers determine Per Hop Behaviors (PHBs) by setting IP TOS bits based on destination, source and even port fields. More importantly (at least today), many router products allow a firewall capability, such as Cisco Access Control Lists (ACLs), which allow packets to be blocked based on the same fields.

It is well known that performing classification on an arbitrary number of fields has poor worst case cost: searching over N filters in k dimensions, either one takes $(logN)^{(k-1)}$ worst-time or one pays N^k memory. In [7], we re-examine two basic mechanisms that have been dismissed in the literature as being too inefficient: backtracking search and set pruning trees. We find using real databases that the time for backtracking search is much better than the worst case bound; instead of $\Omega((logN)^{k-1})$, the search time is only roughly twice the optimal search time. Similarly, we find that set pruning trees (using a DAG optimization) have much better storage costs than the worst case bound; it has memory requirements similar to the RFC scheme of Gupta and McKeown. We also propose several new techniques to further improve the two basic mechanisms. Our major contributions are a novel compression algorithm, the ability to trade smoothly between backtracking and set pruning, and algorithms to effectively make use of hardware if hardware is available. We quantify the performance gain of each technique using real databases. We show that on real firewall databases our schemes, with the accompanying optimizations, are close to optimal in time and storage.

5 Other Work

Besides my thesis, I also worked on adaptive applications, active queue management, and TCP performance analysis. I describe them in more detail below.

5.1 Adaptive Applications

Internet telephony (the provisioning of telephony services using the Internet) has emerged as an important service, poised to replace circuit-switched telephony service in the future. However, the currently available, best effort Internet service model was never engineered to handle voice traffice, since it provides variable loss rates and delays. Efforts are underway to address these problems at the network layer using differentiated services and Integrated Services. However, we anticipate that many public and private internets will continue to provide best-effort delivery for a long time. Therefore, compensation for loss and end-to-end delay using adpative applications which adjust their encoding parameters, such as bitrate, in response to network feedback on loss, delay, and jitter, is an important problem.

Packet loss is handled primarily through a variety of different forward error correction (FEC) algorithms and local repair at the receiver. Jitter is compensated for by means of adaptive playout buffer algorithms at the receiver. Traditionally, these two mechanisms have been investigated in isolation. In [3], we investigate the interactions between adaptive playout buffer algorithms and FEC, and demonstrate the need for coupling. More specifically, we find that when FEC is introduced, the playout delay not only depends on the network delay, but also depends on the network loss rate. For example, when the network loss rate is zero, the playout delay should be set to the maximum network delay in order to play out all the packets. However, as the network loss rate increases, the playout delay has to increase in order to receive the FEC packets. Based on the insight, we propose a number of novel playout buffer algorithms which provide this coupling, and demonstrate their effectiveness through simulations based on both network models and real network traces. To our knowledge, our work is the first one that investigates the coupling effect, and proposes algorithms to take into account of this coupling.

5.2 Active Queue Management

Random Early Detection (RED) is the recommended active queue management scheme for rapid deployment throughout the Internet. As a result, there have been numerous research efforts in studying the performance of RED. However, previous studies have often focused on relatively homogeneous environments. The effects of RED in a heterogeneous environment are not thoroughly understood. In [6], we use extensive simulations to explore the interaction between RED and various types of heterogeneity, as well as the impact of such interaction on the user-perceived end-to-end performance. Our major observations are: (a) when competing with long-lived TCP flows, short-lived flows tend to get higher goodput with RED than with DropTail; (b) for TCP flows with different RTT's, RED in general tends to reduce the bias against long-RTT bulk transfers; (c) when the ACK path is congested, TCP gets higher goodput with RED than with DropTail; (d) when ECN-enabled traffic compete with non-ECN-enabled traffic, ECN-enabled traffic can receive up to 30% higher goodput. So overall RED improves performance at least for the types of heterogeneity we have considered. To our knowledge, this is the first study that investigates RED in the heterogeneous environment.

5.3 TCP Performance Analysis

As the most widely used reliable transport in today's Internet, TCP has been extensively studied in the past decade. However, previous research usually only considers a small or medium number of concurrent TCP connections. The TCP behavior under many competing TCP flows has not been sufficiently explored.

In [4], we use extensive simulations to investigate the individual and aggregate TCP performance for a large number of concurrent TCP flows. We develop a simple yet realistic network model to abstract an Internet connection. Based on the abstract model, we study the performance of a single TCP flow with many competing TCP flows by evaluating the best-known analytical model proposed in the literature. We also examine the aggregate TCP behavior exhibited by many concurrent TCP flows. Our major findings are: (a) When all the connections have the same propagation delay, without adding random processing time, global synchronization occurs when W_c (i.e. the total packets that the network can hold) is larger than three times the number of connections. When W_c is smaller than the number of connections, the bandwidth sharing becomes extremely unfair: some get most bandwidth, while others become totally shut-off. When the value of $\frac{W_c}{Conn}$ is in between, the bandwidth sharing falls in between. (b) Adding random processing time makes both synchronization and consistent discrimination less pronounced. So in the real Internet, these dramatic behaviors are not severe problems. (c) We have derived the general characterization of overall throughput, goodput, and loss probability. These results are very useful for network provisioning. (d) We also investigate the TCP connections with different RTT's, and quantifies the roundtrip bias analytically.

6 Future Directions

In the future, I would like to further explore content distribution of both Web pages and multimedia traffic. Given the different characteristics of Web pages and multimedia traffic, different optimization techniques will be needed. In addition, I would also like to explore the content distribution in wireless network. High loss rate, mobility, and thin clients make the content distribution in wireless environment even more challenging and interesting.

The rapid growth in link bandwidth poses new challenges to designing networking protocols and systems. In particular, it demands more efficient packet classification techniques to keep up with the line speed. In addition, it may also require different congestion control algorithms to effectively utilize the high bandwidth available. I am interested in developing techniques to address the challenges.

References

[1] Venkata N. Padmanabhan and Lili Qiu. The Content and Access Dynamics of a Busy Web Site: Findings and Implications. To appear in Proceedings of SIGCOMM 2000, Stockholm, Sweden, August 2000.

[2] Yin Zhang, Lili Qiu, and Srinivasan Keshav. Speeding Up Short Data Transfers: Theory, Architectural Support, and Simulation Results. In Proceedings of NOSSDAV 2000, Chapel Hill, North Carolina, USA, June 2000.

[3] Jonathan Rosenberg, Lili Qiu, and Henning Schulzrinne. Integrating Packet FEC into Adaptive Voice Playout Buffer Algorithms on the Internet. Proceedings of IEEE Infocom 2000, Tel-Aviv, Israel, March 2000.

[4] Lili Qiu, Yin Zhang, and Srinivasan Keshav. On Individual and Aggregate TCP Performance. In Proceedings of 7th International Conference on Network Protocols (ICNP'99), Toronto, Canada, November 1999.

[5] Lili Qiu, Venkata N. Padmanabhan, Geoffrey M. Voelker. On the Placement of Web Server Replicas. Submitted to INFOCOM'2001.

[6] Yin Zhang, and Lili Qiu. Understanding the End-to-End Performance Impact of RED in a Heterogeneous Environment. Submitted to INFOCOM'2001.

[7] Lili Qiu, George Varghese, and Subash Suri. Fast Packet Classification on Multiple Dimensions. In preparation for submission.