This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE CCNC 2010 proceedings

Performance Evaluation of Quick-Start in Low Latency Networks

Ismail Butun*, Sumit Birla†, Xuan Hung Le*, Sungyoung Lee‡, Ravi Sankar*
*Department of Electrical Engineering, †Department of Computer Science and Engineering
University of South Florida, 4202 E. Fowler Avenue, Tampa, FL, 33620
‡Department of Computer Engineering, Kyung Hee University, Korea, 446-701
ibutun@mail.usf.edu, birla@mail.usf.edu, xhle@eng.usf.edu
sylee@oslab.khu.ac.kr, sankar@eng.usf.edu

***Abstract*—The Quick-Start mechanism has been presented as a way for transport control protocols to efficiently use available bandwidth in under-utilized networks. Although studies of performance of Quick-Start have been carried out, its performance in low latency networks has not been well investigated. In this paper, we show that the benefits of Quick-Start in low latency networks are minimal. We also show that the improvement in network utilization is much lower than in networks with higher latencies. Our future work in this field will undoubtedly lead to new algorithms and improvements to existing ones.**

I. Introduction

Current implementations of TCP (Transfer Control Protocol) use Slow-Start as a way of reasonably increasing the transmission window and reducing it when a congestion is detected. However, if a file transfer is completed before the maximum window size is realized, it results in wasted bandwidth. Starting off with a small window when there is adequate bandwidth to transfer much higher number of segments results in under-utilization of the network. By starting off with a larger window, the transmission can be finished in fewer round trips. To address this issue, Quick-Start [1] offers a variant which has the potential to increase utilization in the initial phases of a TCP connection. It is different from Slow-Start in that it first asks a permission from all routers along the way for a particular bit rate. If a Quick-Start request is approved by all nodes on a given TCP connection path, the sender can start transmitting at the approved rate. The router in this case has to reserve certain allocation for the connection and keep track of its recent allocations. The existence of this bandwidth is not guaranteed. Unexpected bursts of traffic from other sources may cause packets to drop, in which case the connection would revert to Slow-Start. Also, depending on the router load and the percentage of bandwidth being utilized at each router, a request may be denied in which case also the connection would revert to Slow-Start.

A few studies of performance of Quick-Start have been carried out [2][3]. [2] evaluates several different algorithms that routers could use to process a Quick-Start request. [3] studies Quick-Start with a new implementation in the Linux protocol stack. However, the performance of Quick-Start in low latency networks such as Local Area Network (LAN) have not been thoroughly evaluated.

[1]This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2009-(C1090-0902-0002)).

Dr. Sungyoung Lee is the corresponding author.

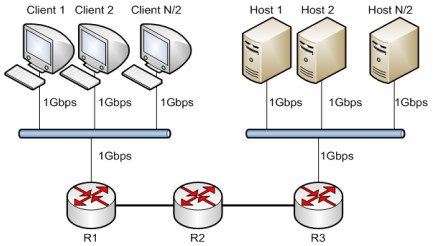

Fig. 1. Network Configuration

In this paper, we simulate Quick-Start in low latency networks under different network conditions using ns-2 simulator [4]. The simulations are built on top of the setup described in [2] and are modified to account for lower latencies and the incorporation of FTP (File Transfer Protocol).

The rest of this paper is organized as follows: Section II describes the setup of simulations used in this evaluation. Section III presents the results of experiments. We discuss the performance evaluation of Quick-Start in Section IV. Finally, Section V concludes the paper and outlines our future work.

II. Simulation Setup

We used ns-2 [4] simulator to evaluate the performance of Quick-Start in low latency networks. The network is configured as illustrated in Fig 1. It consists of three routers, *R*1, *R*2, and *R*3, arranged in a chain with hosts on both ends. Hosts can act either as web servers or web clients. Each host is connected to a router via 1 Gbps link. The routers are also connected to each other via 384 kbps, 100 Mbps and 1 Gbps links with propagation delays ranging from 1 ms to 1,000 ms.

The routers used above have a maximum queue length of 150 packets and employ drop-tail queueing. For experiments involving use of Quick-Start, all web servers start off with using this algorithm. All TCP connections use the *sack*1 TCP variant included in *ns*-2 with an initial congestion window

978-1-4244-5176-0/10/$26.00 ©2010 IEEE

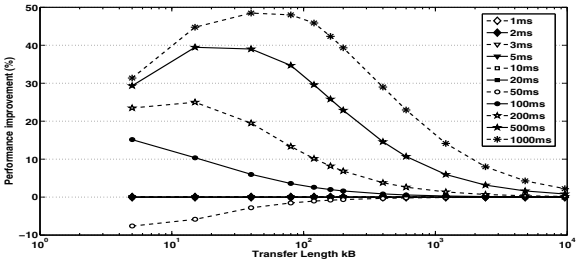

Fig. 2. Relative Improvement with Quick-Start through 384 kbps link

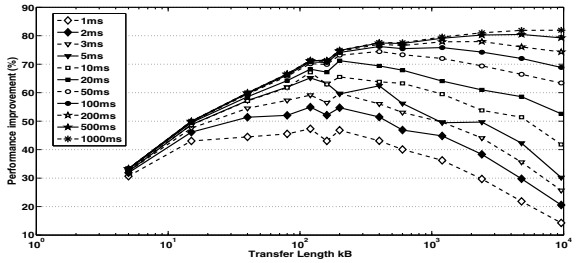

Fig. 4. Relative Improvement with Quick-Start through 1 Gbps link

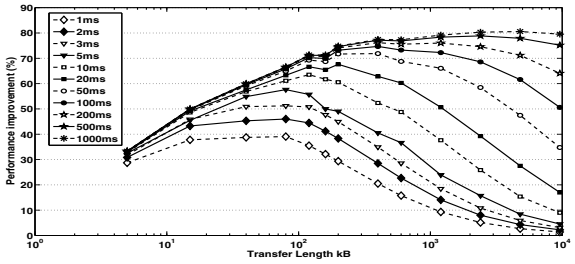

Fig. 3. Relative Improvement with Quick-Start through 100 Mbps link

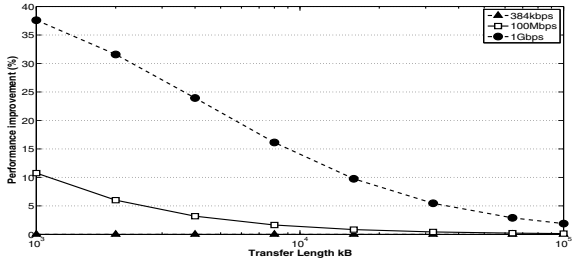

Fig. 5. Comparison of Relative Performance with Quick-Start for different link using the FTP protocol

cwnd of 3 segments, a Maximum Segment Size (MSS) of 1,460 bytes, an advertised window of 10,000 segments and the receiver acknowledging each segment.

The number of web servers (N) varies but always has a corresponding web client. Web servers are attached to *R*3 and web clients are attached to *R*1. In order to simulate background traffic, *N*/2 clients are connected to *R*1 and *N*/2 servers are connected to *R*3. For generating web traffic, the included ns-2 web traffic generator is used with an average of 30 web pages per session, inter-page parameter of 0.8 and average page size of 10 objects with average packet size of 400 packets.

The above setup is repeated with web servers and client replaced with FTP servers and client. The file transfers vary from 1 MB to 100 MB to simulate a reasonable range of file sizes.

III. Results

The results of our simulations are discussed in this section. In this work various propagation delays are used in the range of 1 ms to 1,000 ms. Among those, we consider delays from 1 ms through 50 ms as low latency links, and delays from 100 ms to 1,000 ms as high latency links. Figure 2, 3, and 4 show the simulation results on TCP protocol, while Figure 5 shows the simulation result on FTP protocol. All these figures show the relative performance improvement (%) of Quick-Start versus Slow-Start in terms of transmission time.

Figure 2 shows the plots for a link speed of 384 kbps. It can be seen that from 1 ms to 20 ms delays, there is barely any improvement in transmission times. At 50 ms delay, the percentage of performance improvement drops below zero. It means that Slow-Start outperforms Quick-Start at this delay. At 100 ms delay, there is a 15% improvement for small length of transmission packet (about 2 kB). However, this improvement becomes lower as the transmission data length increases. This is also true for all high latency cases, resulting an inversely proportional curves between transfer lengths and relative performance improvements. For 384 kbps link speed, high latency cases show improvement of up to 50%.

Figure 3 and Figure 4 show the same evaluation, but with link speeds of 100 Mbps and 1 Gbps. It is clear that Quick-Start shows improvement over Slow-Start in low latency links, but performs worse than Slow-Start in high latency links. Also, with larger transmission packet length, the improvement of low latency cases decays faster, indicating that Quick-Start is most beneficial in a certain range of packet length. It does not offer performance improvement for large data transmissions as also mentioned in [2].

For a bandwidth of 1 Gbps, performance improvement in low latency cases increases up to 74% (shown in Figure 4, an increase of average 2% compared with the 100 Mbps link). Although we increase the bandwidth of the link by ten times, the performance improvement only increases by 2%.

In Figure 5, the improvement in transmission times for large files is shown. As expected, the improvement is much lower at the larger file sizes. For 384 kpbs link speed, we observe no improvement at all. For higher transmission speeds of 100 Mbps and 1 Gbps, the improvement drops down as the transmission packet size increases. This is due to the fact that Quick-Start is most beneficial at the beginning of a TCP connection. Once the maximum window is reached, Quick-Start does not help improve utilization. Hence, for long-lived streams, we observe that the performance of Quick-Start diminishes relative to Slow-Start.

The results above using various link speeds and propagation delays indicate that the transfer time reduction depends on the Bandwidth-Delay Product (BDP). In simulations, we observe that using Quick-Start at a link speed of 384 kbps with 1 ms through 20 ms delays yields no benefit. On the other hand, using Quick-Start at a link speed of 1 Gbps with a latency of 1,000 ms shows significant improvement (up to 85%). However, this improvement decreases drastically for small-sized and extremely large-sized packet transmissions.For mid-sized packet transmissions through normal latency links, the transmission times with Quick-Start is much lower than Slow-Start.

TABLE I
THE PERFORMANCE IMPROVEMENT % TABULATED FOR PROPAGATION DELAY VS. LINK BANDWIDTH FOR DATA TRANSFER LENGTH OF 120 KB

Bandwidth/delay	384 kbps	100 Mbps	1 Gbps
1 ms	0	35.4	47.3
3 ms	0	50.8	59.2
5 ms	0	55.7	65.2
10 ms	0	63.5	67.2
50 ms	−1.1	69.3	70.2
100 ms	2.5	70.2	70.8
200 ms	10.1	70.8	71.1
500 ms	29.6	71.2	71.3
1000 ms	45.9	71.3	71.4

An interesting observation point in Table I is the improvement percentage with bandwidth of 384 kbps and propagation delay of 50 ms. In this case, there is a deterioration in transmission time. In other words, it takes longer for 120 kB of data to be transferred when Quick-Start is employed. This is due to one of numerous causes resulting in the connection reverting back to Slow-Start.

IV. DISCUSSION

In all these observations, the simulations provide an ideal case scenario - routers have unlimited buffer sizes and are willing to allocate up to 90% of their capacity to Quick-Start requests. Also, there is a constant stream of background transfer in the reverse direction. In practice, unexpected data may arrive at any time and in any direction causing the connection to revert to slow-start. Also, if the utilization of any router on the path increases beyond its maximum allocation threshold, it will never approve any Quick-Start requests and consequently, every TCP connection on this path will take longer time to transfer data compared to traditional connection.

Quick-Start used in conjunction with Slow-Start as fallback mechanism has the potential to increase network utilization. This is especially true at the early stages of a TCP connection. However, the benefits appear to manifest mainly in the mid-size data transfer cases. For transmission packet lengths of 50 kB to 1 MB, there is an observable performance improvement. The improvement is significantly lower when the link delays are reduced to below 3 ms. These delays are seen in low-latency networks like Intranet which are generally under-utilized.

The improvement is also shown to be lower for larger transfer sizes. Current studies focus on short HTTP requests. While this may hold true for majority of data on the Internet today, we may see a shift towards larger data transfers with the popularization of streaming audio and video applications. It is also difficult to predict the nature of web traffic within five to ten years.

In real life scenarios, the improvement can be expected to be much lower than the ideal-case. If a Quick-Start request is not approved by all routers, then the TCP connection will be wasted one round-trip time and would revert to Slow-Start yielding worse utilization than a traditional connection. For protocols other than HTTP, it may be difficult for a host to determine how much bandwidth to request. Also, artificial denial-of-service attacks are possible where a malicious host requests large amounts of bandwidths and then does not use it. Hence the question whether to implement Quick-Start in routers and hosts remains an open one. The added expense of implementing new firmware for routers and updating TCP/IP stacks to take advantage of this scheme needs to be considered. The security concerns may not be a factor in controlled environments, but may pose a significant challenge for open networks.

V. CONCLUSIONS AND FUTURE WORK

In this work, we have studied the performance of Quick-Start in low latency networks and observed a number of interesting results. In low latency networks, we observed that the difference in transfer times of various sizes of data with Slow-Start is not much higher than Quick-Start. Through simulations, we have shown that when Quick-Start is employed in a LAN with low latencies, the improvement in network utilization is much lower than in networks with higher latencies. We have also shown that this mechanism is of limited benefit to protocols other than HTTP such as FTP where the transfer sizes are generally larger and does not comprise of fetching small files.

Our future work in this field will undoubtedly lead to new algorithms and improvements to existing ones. The traditional Slow-Start congestion control method has worked well, albeit not at an optimal level. But it scales to any window size and is already implemented and in use. More studies need to be done to gauge the effectiveness of Quick-Start. The questions which need to be answered include the cost of implementation, mitigation of security concerns and the benefits for data streams other than simple web page requests. Should investment be made for the installation of higher performance routers and links instead of implementing Quick-Start?

REFERENCES

[1] S. Floyd, M. Allman, A. Jain, and P. Sarolahti, *Quick-start for TCP and IP*, IETF RFC 4782 (experimental), Jan. 2007
[2] P.Sarolahti, M. Allman and S. Floyd. *Evaluating Quick-Start for TCP*, February 2005.
[3] Scharf, M., Strotbek, H.: Performance evaluation of quick-start TCP with a Linux kernel implementation. NETWORKING 2008. LNCS, vol. 4982, pp. 703714. Springer, Heidelberg (2008)
[4] The network simulator - NS-2. http://www.isi.edu/nsnam/ns/