Effect of Head of the Line Blocking on Session Initiation Protocol Session Establishment Delays

Gonzalo Camarillo, Henning Schulzrinne, Salvatore Loreto, and Jani Hautakorpi

Abstract: **We have studied the effect of head of the line blocking (HOLB) on session initiation protocol (SIP) session establishment delays. Our results are based on experiments performed in a test bed and on the public Internet. We used the stream control transmission protocol (SCTP) as a transport for SIP because SCTP can be configured to suffer or to avoid HOLB. Our experiments show that the effect of HOLB on session establishment delays generally starts to be significant starting at fairly low packet loss rates. However, there are scenarios where network conditions are good enough to make the effect of HOLB insignificant.**

Index Terms: **Head of the line blocking (HOLB), post-selection delay, session establishment delay, session initiation protocol (SIP), stream control transmission protocol (SCTP), voice over IP (VoIP).**

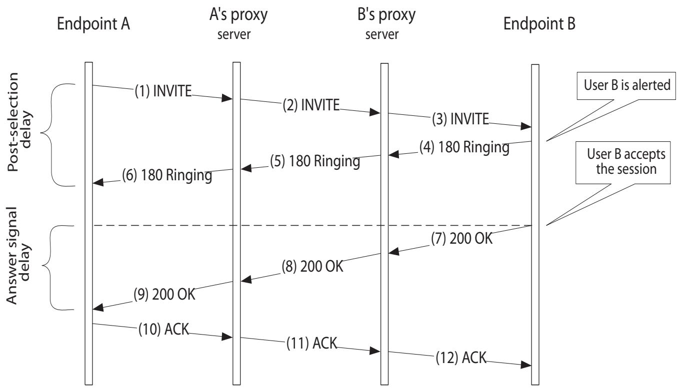


Fig. 1. SIP session establishment.

I. INTRODUCTION

The session initiation protocol (SIP) [1] is a text-based rendezvous protocol that provides user mobility and session establishment. SIP's rendezvous functionality is based on SIP-level routers, which are referred to as proxy servers. Proxy servers route SIP messages between SIP endpoints. A typical session between two SIP endpoints from different domains involves, at least, a proxy server in the originating domain and a proxy server in the terminating domain. Furthermore, the same two proxy servers generally handle all SIP sessions between those two domains. As a consequence, these proxy servers exchange SIP messages that belong to several sessions, each of which is established between different SIP endpoints.

When TCP is used between such proxy servers, the loss of a SIP message that belongs to a session between two SIP endpoints may delay the delivery of other SIP messages that belong to different sessions between different SIP endpoints. This problem is known as head of the line blocking (HOLB).

The stream control transmission protocol (SCTP) [2] is a transport protocol that provides a message delivery service. The fact that SCTP can be configured to suffer or to avoid HOLB (e.g., by using the ordered or the unordered SCTP delivery services respectively) at any given time makes it a suitable protocol to study the effects of HOLB on the transport of SIP messages.

This paper performs a general study of HOLB in SIP using experiments in a real test bed and on the public Internet. Note that, even though we used SIP in our experiments, our results can be applied to any application-layer protocol susceptible to suffer HOLB and able to run on top of SCTP.

The remainder of this paper is organized as follows. Section II introduces the delay metrics associated with session establishment in SIP and discusses how HOLB affects them. Section III describes SCTP and how it avoids HOLB. Section IV summarizes the result of previous studies on the effect of HOLB. Section V describes our test bed and introduces the experiments we performed. Section VI discusses the results of the experiments performed on our test bed under emulated random packet losses, competing long-lived TCP connections, and competing web traffic. Section VII discusses the results of the experiments performed on the public Internet. Section VIII contains the conclusions of this paper.

II. SIP

Fig. 1 shows a SIP session establishment between two endpoints through two proxy servers. Endpoint A generates an INVITE request (1) and sends it to its outbound proxy server. The outbound proxy server relays the request (2) to another proxy server, which is located at endpoint B's domain. This proxy server finally relays the request (3) to endpoint B, which was the intended final destination of the request. Subsequent messages belonging to this session typically traverse the same two proxy servers between the endpoints.

SIP responses, like hypertext transfer protocol (HTTP) responses, are identified by a three digit status code and by a reason phrase. The 180 (Ringing) response (4) generated by endpoint B indicates that user B is being alerted. The 200 (OK) response (7) generated by endpoint B indicates that user B has accepted the session. SIP session establishment involves the sending of an ACK request (10) on receiving a 200 (OK) response.

Manuscript received May 05, 2007; approved for publication by Suresh Subramaniam, Division III Editor, October 10, 2008.

G. Camarillo, S. Loreto, and J. Hautakorpi are with the Ericsson Research, Hirsalantie 11, 02420 Jorvas, Finland, email: {Gonzalo.Camarillo, Salvatore.Loreto, Jani.Hautakorpi}@ericsson.com.

H. Schulzrinne is with the School of Computer Science, Columbia University, New York, NY 10027-7003, USA, email: schulzrinne@cs.columbia.edu.

A. Delay Metrics in SIP

The international telecommunication union (ITU) E.721 recommendation [3] defines two delay metrics that relate to session establishment: post-selection delay and answer signal delay. For a SIP session establishment, the post-selection delay is the time interval from the delivery of the INVITE request to the transport layer until the reception of the 180 (Ringing) response from the transport layer. The answer signal delay is the time from the delivery of the 200 (OK) response to the transport layer until the reception of that response from the transport layer by the other endpoint. This paper focuses on the post-selection delay. In particular, this paper studies the effect of network conditions and HOLB on the post-selection delay.

ITU E.721 defines the post-selection delay in relation to the access signaling system. That is, the post-selection delay is measured from the moment the connection establishment message is passed by the calling terminal to the access signaling system until the alerting message is received by the same terminal. We have chosen to include the transport layer in our definition of signaling system so that delays introduced by HOLB are included in the post-selection delay.

From the user's point of view, the post-selection delay is the time it takes from the moment the user instructs his or her terminal to establish a session with a given destination until the user receives an alerting indication. In an old phone, that is the time since the user finishes dialing the destination number until the phone starts playing the ringing tone. Users facing too long a post-selection delay often cancel their current session establishment and attempt a new one, increasing the amount of signaling traffic the network needs to carry for that session establishment (i.e., first session establishment attempt, cancellation, and second hopefully-successful attempt) and decreasing the user's satisfaction with the service. That is why reducing the post-selection delay and making it more predictable for users (i.e., decreasing its variability) is considered important by most operators.

B. HOLB in SIP

As stated in Section I, all the SIP sessions established between endpoints from two domains may traverse the same two proxy servers. Fig. 2 shows how two proxy servers multiplex SIP messages belonging to different SIP sessions over a single transport protocol connection between them. Each SIP message can be transported in one or several packets depending on the transport protocol used (the following description of HOLB assumes a one-to-one mapping between SIP messages and transport-layer segments for clarity).

The HOLB problem appears when the transport protocol between both proxies delivers messages to the receiving application in the same order as they were generated by the sending application. For example, if the transport protocol in Fig. 2 provided an ordered delivery service, it would not deliver the message from A2 to B2 until it had delivered the message from A1 to B1. Even if the latter message gets lost but the former message arrived correctly, the transport protocol would just store the message from A2 to B2 until the arrival (possibly after a retransmission) of the message from A1 to B1. At that point, the

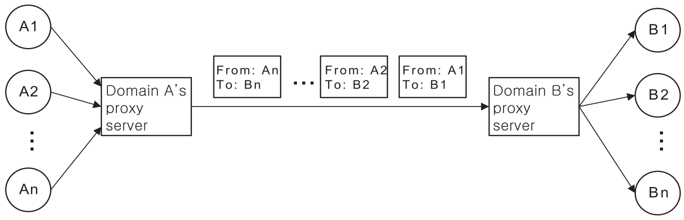


Fig. 2. SIP session multiplexing.

transport protocol would deliver both messages to the SIP proxy server application. Consequently, the loss of a message that belongs to one session (e.g., the message from A1 to B1) may unnecessarily delay the delivery of a message from a completely different session (e.g., the message from A2 to B2). TCP suffers the HOLB problem just described. SCTP, on the other hand, can be configured to avoid it by using its unordered delivery service.

In addition to TCP and SCTP, SIP can also run on top of UDP by using application-layer timeouts and retransmissions for reliability. However, even though UDP also avoids HOLB, its use between proxy servers like the ones in Fig. 2 is not recommended because of its lack of congestion control.

III. SCTP

SCTP [2] is a transport protocol whose flow and congestion control mechanisms are based on those of TCP. SCTP implements slow start, a congestion window, timeouts, fast retransmits, and cumulative acknowledgments. In addition to cumulative acknowledgments, which are used by receivers to acknowledge the reception of a set of data up to a point, SCTP also implements selective acknowledgments. Receivers use selective acknowledgments to acknowledge the reception of noncontiguous chunks of data. SCTP's selective acknowledgment mechanism is similar to the TCP SACK extension [4] (see [5] for a comparison between both mechanisms).

SCTP provides a message delivery service, as opposed to TCP that provides a byte-stream delivery service. SCTP delivers whole messages to the SIP application. Unlike with TCP, applications do not need to implement application-layer framing to parse SIP messages, as message framing is performed by SCTP.

An SCTP DATA chunk is the equivalent of a TCP segment. However, as noted previously, there is a direct mapping between application-layer messages and DATA chunks. A DATA chunk always carries a single application-layer message. In TCP, on the other hand, an application-layer message can span several TCP segments and a TCP segment can carry several application-layer messages (see [6] for a discussion on how to map application-layer objects to TCP segments; this mechanism could be used to implement message framing in TCP).

Other advantages of SCTP over TCP for transporting SIP messages relate to the SCTP association establishment handshake. SCTP association establishment has been designed to be more robust against denial of service (DoS) attacks. On the negative side, maintaining an SCTP association requires more state information than maintaining a TCP connection. This increase in the state information to be stored may have a negative impact in the scalability of some systems (see [7]).

In any case, the most important advantage of SCTP over TCP for transporting SIP traffic is believed to be the fact that SCTP can be configured to avoid HOLB. This paper investigates the significance of HOLB. SCTP can avoid HOLB in two ways: using multiple streams or using SCTP's unordered delivery service. Both ways of transporting SIP signaling over SCTP are described in [8].

A connection between two SCTP endpoints is referred to as an SCTP association. SCTP can establish multiple streams within an association. Messages from different streams are delivered independently from one another. A message from a stream does not block the delivery of any message from another stream. Still, messages within a stream are delivered in order. The use of multiple streams overcomes inter-stream HOLB, but not intra-stream HOLB.

SIP applications using several streams map different SIP transactions to different SCTP streams. This way, intra-stream HOLB becomes intra-SIP-transaction HOLB. This type of HOLB is only noticeable when the loss of a provisional SIP response blocks the delivery of a final response for the same transaction.

SCTP also implements an unordered delivery service. When this service is used, messages are delivered independently from one another. The unordered delivery service avoids HOLB completely. The reason why some SIP applications use multiple streams instead of the unordered delivery service is that, traditionally, the use of TLS [9] over SCTP required, at least, intra-stream ordered delivery. Since we did not use TLS in our experiments, we used the SCTP unordered delivery in order to avoid HOLB completely. We used the SCTP ordered delivery service over a single stream to produce TCP-like HOLB. We analyzed how much more delay the ordered delivery service introduces, when compared to the unordered service, in order to study the effects of HOLB in the transport of SIP messages.

IV. RELATED WORK

In [5], we performed a preliminary study of the HOLB effects in the transport of SIP using ns-2 [10] simulations. We analyzed how the average one-way delay for a set of SIP messages increased when HOLB was present. Our simulations included different network conditions; namely random packet losses, a buffer-limited router, and competing long-lived TCP connections. The simulations showed that the effect of HOLB on the average one-way delay was not significant under low random packet-loss rates and when all packet losses were caused by buffer-limited router (i.e., no random packet losses and no competing traffic). The effects of HOLB on the average one-way delay became statistically significant when we added competing TCP traffic. In the conclusions of [5], we indicated that the effect of HOLB on the 95th percentile was more relevant (to operators and regulators) than its effect on the average delay and that our future work included analysing such effect. Consequently, as discussed in Section VI, the experiments described in this paper focus on the 95th percentile of the post-selection (i.e., two-way) delay.

A study on the increase in the average one-way delay introduced by HOLB in SCTP can be found in [11]. That study was based on experiments in an emulated network environment including competing TCP traffic. The experiments in [11] show delay increases up to 18% in the average one-way delay when HOLB was present. However, given that the variability between the results of different test runs within an experiment was large (only the delay of a single message per test run was measured), the delay increases were often statistically insignificant.

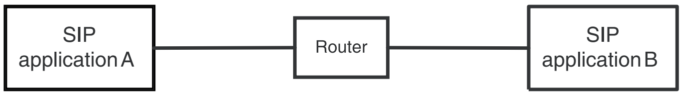


Fig. 3. The test bed.

The increase in the average two-way delay (i.e., response time in a request-response based protocol) introduced by HOLB in SCTP was studied in [12]. That study was based on experiments on an emulated network environment including random packet losses. The network environment was emulated using NIST Net [13], which is also the tool we use in the experiments described in this paper. The results in [12] show a significant delay increase in the average two-way delay caused by HOLB even for moderate packet-loss rates in the order of 2%.

The reason why previous studies on HOLB reach seemingly contradictory conclusions is that each study focused on a narrow set of scenarios. Results that are valid for the particular scenario a paper focused on do not necessarily apply to the scenario considered in another paper. In this paper, we have considered a wide set of scenarios so that our results are more general. In particular, our experiments include scenarios with different packet-loss patterns, call rates, link delays, and routers' queue sizes in emulated and Internet environments.

V. THE TEST BED

Our test bed consists of two SIP applications that exchange SIP traffic between them through a router, as shown in Fig. 3. This test bed represents the configuration in Fig. 2. "SIP Application A" in Fig. 3 corresponds to "Domain's A Proxy Server" in Fig. 2 and "SIP Application B" in Fig. 3 corresponds to "Domain's B Proxy Server" in Fig. 2.

The SIP applications are based on the kphone [14] SIP user agent. The SCTP stack used by the applications is the lksctp stack (we participated as developers in the lksctp project that developed this stack). The arrival of INVITE requests from user agents A1 through An at Domain's A Proxy Server is simulated by an INVITE request generator at SIP Application A. As a result, SIP Application A generates INVITE requests towards SIP Application B using a Poisson process.

On receiving each of these INVITE requests, SIP Application B returns a 180 (Ringing) response. These responses simulate the 180 (Ringing) responses that Domain's B Proxy Server would receive from its user agents for each of the INVITE requests. We can measure the post-selection delay for each INVITE request by measuring the time since SIP Application A passes the INVITE request to its SCTP stack until the SCTP stack passes the corresponding 180 (Ringing) response to SIP Application A. We can also modify the rate at which the Poisson process at SIP application A generates INVITE requests in order to study the effect of the call rate on the post-selection delay.

Additionally, we can also measure transport-level data such as the number of SCTP-level retransmissions needed to complete a given number of SIP-message exchanges. Moreover, we can also identify which of those retransmissions were unnecessary. That is, those DATA chunks that were retransmitted even though the original DATA chunk did not get lost and made it to the destination.

The router between the two SIP applications implements a NIST Net [13] network emulator, which is able to emulate different network conditions. NIST Net can emulate, among other things, link delays, random packet losses, and packet queues at the router. In all the experiments described in Section VI, we configured NIST Net to emulate the range of link delays we wanted to analyze. In the experiments using random packet losses, which are described in Section VI-A, NIST Net emulated random packet losses with different levels of correlation at the router. In the experiments using competing long-lived TCP connections and competing web traffic, which are described in Sections VI-B and VI-C, respectively, NIST Net emulated a router with a finite packet queue.

The links used between the SIP application and NIST Net had a bandwidth of 100 megabit per second. We did not limit the bandwidth of those links in any way because we did not want bandwidth to introduce additional delays in our experiments involving random packet losses. The computers hosting the SIP applications, the router, and the sources and sinks of the competing TCP traffic (long-lived connections and web traffic) all had an Intel Pentium IV processor and ran a Linux-based operating system (2.6 kernel).

VI. EXPERIMENTS ON THE TEST BED

We ran three different sets of experiments. In one set, we used emulated packet losses. In the other two sets, packet losses were produced by overflows at the router's queue under competing TCP traffic: Long-lived TCP connections performing file transfers that always had data to send in one set and web traffic (more bursty in nature) in the other set. We performed experiments using emulated packet losses and different types of competing traffic in order to experiment with a wide variety of different packet loss patterns.

Our test bed allows us to give different values to a set of parameters: SCTP delivery service, call rate, link delay, router's queue size, random packet loss rate, correlation between random packet losses, number of competing TCP connections, and rate at which competing web sessions are generated. Each particular set of values for these parameters defined a configuration for an experiment. We ran ten experiments for each configuration. Each experiment consisted of a one-minute exchange of SIP messages between the SIP applications through the router. For each experiment, we computed the 95th percentile of the post-selection delay for all the INVITE requests sent during the experiment. The ten experiments that were run using a given configuration gave us a distribution for the 95th percentile of the post-selection delay that consisted of ten values. We used these distributions to analyze statistically the effects of the different parameters on the 95th percentile of the post-selection delay.

We chose to analyze the 95th percentile of the post-selection delay instead of the average post-selection delay. This was because operators and regulators are usually more concerned with the number of sessions that experiment too high a delay (i.e., they do not meet their delay requirements) rather than with the average delay all their sessions experiment.

We used the following parameters values in our experiments. In order to analyze a wide range of link delays, we used 20 ms, which represents the one-way delay between two relatively close servers, and 60 ms, which represents the one-way delay between two cross-continental servers. Our goal was to analyze call rates where the throughput of the transport protocol was limited, as much as possible, by the application and not by its congestion window. This is usually the case when links are dimensioned for signaling transport. Consequently, we used 20 and 40 calls per second, which produced (approximately) 72 and 144 kilobits per second, respectively.

The active queue management (AQM) algorithm used to manage the router's queue was derivative random drop (DRD) [13]. DRD drops packets with a probability that increases linearly with the instantaneous queue length. DRD's behavior is determined by two threshold values. The drop probability starts at zero at the first threshold and reaches 95% at the second threshold. We used two different queue sizes for our experiments. The threshold values for the first queue were 10 and 20 packets; the threshold values for the second queue were 50 and 60 packets. The 10–20 packet queue represents a very short queue intended to minimize delay. Such a queue keeps competing TCP connections from saturating the link. The 50–60 packet queue represents a longer queue intended to allow applications achieve a higher throughput.

A. *Emulated Random Packet Losses*

For this set of experiments, we configured NIST Net to emulate random packet losses at the router. We could assign different values to six parameters. The packet loss rate, the packet loss correlation, the router's queue size, and the link delay determined NIST Net's behavior. The call rate (i.e., the rate at which INVITE requests were generated) determined the SIP application's behavior. Additionally, we could configure the SCTP stacks to use ordered or unordered delivery.

Table 1 shows the values we used for the experiments involving random packet losses. We wanted to experiment with packet loss rates ranging from high-quality links explicitly used for signaling (i.e., low packet loss rates) to the links that can be found on the public Internet. Given that it is not uncommon to see packet loss rates of 5% (with higher peaks) on the public Internet [15], we chose rates ranging from 0.5% to 8% for our experiments.

NIST Net produces results that are closer to reality when used with packet loss correlation values between 0.5 and 0.8. We used these two values in our experiments. In addition, we also used uncorrelated packet losses (i.e., a correlation value of zero) in order to study the effects of packet loss correlation on message delay. We analyzed the results of our experiments using a multiple linear regression model and found that the packet loss correlation did not have a statistically significant influence on the

Table 1. Parameter values (random packet losses).

Parameter	Values
Packet loss rate	0.5%, 1%, 2%, 3%, 4%, 6%, 8%
Packet loss correlation	0, 0.5, 0.8
Router's queue size	10–20, 50–60 packets
One-way link delay	20, 60 ms
Call rate	20, 40 calls/s
Delivery type	Ordered, Unordered

post-selection delay. This indicates that the NIST Net levels of packet loss correlation up to 0.8 do not affect significantly the performance of SCTP. Therefore, for simplicity, in this paper we only discuss the experiments that used a packet loss correlation of 0.8. The values in Table 1 yield 336 different configurations. Consequently, we ran 3360 one-minute-long experiments.

A.1 Effect of the Router's Queue Size

In order to study the effect of different parameter variables in different scenarios, we first divided the results of our experiments by the router's queue size used. Then, we divided the results of the experiments using a given router's queue size in four groups. Each group corresponds to the results for a given one-way link delay (20 or 60 ms) and a given call rate (20 or 40 calls per second). Fig. 4 shows the results of the experiments using emulated random packet losses and a router's queue size of 10–20 for our four scenarios. Each delay value-pair in the graph was obtained from 10 experiment runs. We calculated the 95th percentile of the post-selection delay for each experiment. The error bars in the graph show the upper and lower limits of the 95% confidence interval for the mean of the distribution consisting of the 95th percentiles of the 10 experiments.

We did not expect the router's queue size to play any significant rose in our experiments involving random packet losses. This is because, in the absence of competing traffic, the call rates used in the experiments are too low to saturate the router's queue. As expected, the four graphs obtained for a router's queue size of 50–60 were statistically equivalent at the 95% confidence level. We used Kolmogorov-Smirnov tests to compare every point in the graphs in Fig. 4 with its equivalent in the graphs of the experiments using the 50–60 queue. We did not find any statistically significant difference at the 95% confidence level between any of the pairs of points (we chose to run Kolmogorov-Smirnov tests to compare the distributions instead of t-tests because we cannot assume that the samples come from a normal distribution). Consequently, for simplicity, the analysis in the following sections focus on the results of the experiments using a 10–20 queue, which are shown in Fig. 4.

A.2 Effect of HOLB

Fig. 4, all the graphs indicate that, for any given packet loss rate, the delay introduced by HOLB is significant in every scenario. Graphically, non-overlapping confidence intervals indicate a significant difference in delay. In addition to the graphical analysis, we performed a statistical analysis to be sure that the influence of HOLB on the delay is significant, as the graphs seem to indicate. For every packet loss rate, we used

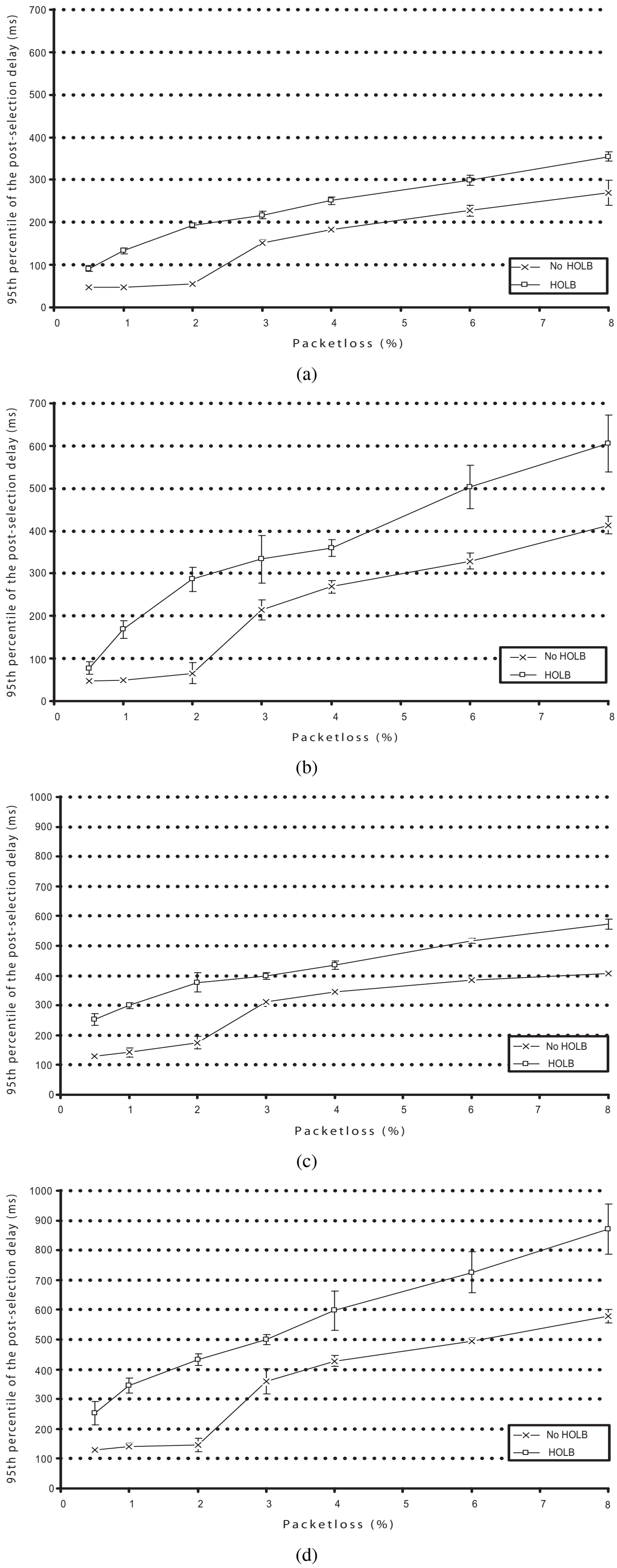


Fig. 4. Delay under emulated random packet losses: (a) 20 ms one-way delay and 40 calls/s, (b) 20 ms one-way delay and 20 calls/s, (c) 60 ms one-way delay and 40 calls/s, and (d) 60 ms one-way delay and 20 calls/s.

Kolmogorov-Smirnov tests to compare the distribution with HOLB with the distribution with no HOLB. In every case, we found, as expected, a statistically significant difference between them at the 95% confidence level.

A.3 Effect of the Packet Loss Rate

The way the packet loss rate affects the delay is different depending on whether or not HOLB is present. When there is no HOLB, SCTP associations are able to keep the 95th percentile of the post-selection delay very low under packet loss rates less than 2%. The delay increases dramatically when the packet loss rate increases from 2% to 3%. This effect is not present under HOLB.

The reason for the sudden delay increase in the graphs is that, in the absence of HOLB, the only messages that get dramatically delayed are those that get lost and need to be retransmitted. A message (i.e., an INVITE request) needs to be retransmitted if the INVITE request itself or its associated response get lost. Under a packet loss rate of 2%, the probability that a message is not lost is 98% (the probability the request is not lost) times 98% (the probability the response is not lost), which equals 96%. Under a packet loss rate of 3%, the probability that a message is not lost is 96% times 96%, which equals 92.1%.

Since we are measuring the 95th percentile of the post-selection delay, while more than 95% of the messages do not get lost, the 95th percentile corresponds to a message that did not get lost and, thus, had a very low delay. Beyond a 95% message loss, the 95th percentile corresponds to a message that got lost and had to be retransmitted. Therefore, its delay will be higher in at least one round-trip time (RTT) .

The sudden increase in the 95th percentile of the post-selection delay does not occur under HOLB. This is because the loss of a single message delays many others. Therefore, even if more than 95% of the messages do not get lost, the message corresponding to the 95th percentile is artificially delayed by HOLB.

For a network operator, the fact that, by avoiding HOLB, the 95th percentile of the post-selection delay can be kept low and nearly constant below a certain packet loss rate is important. This means that the behavior of the sessions that meet the regulator's requirements (typically, at least 95% of all the sessions) will be similar and, thus, predictable. This predictability is a property that generally improves significantly a user's perception of the quality of a service. On the other hand, under HOLB, the post-selection delays of sessions meeting the regulator's requirements will have a much higher variability. Therefore, users will find a less predictable behavior.

A.4 Effect of the Call Rate

Fig. 4 shows that for a given one-way link delay (i.e., 20 or 60 ms), increases in the call rate decreased the delay experienced by the SIP traffic. This is because, as discussed in [16], the SCTP association carrying higher call rates can develop larger congestion windows. Additionally, increases in the call rate reduce the number of timeouts by increasing the number of fast retransmits. Lower call rates are less likely to trigger the fast retransmit algorithm before a timeout occurs because the generation of duplicate ACKs takes longer. This way, higher packet loss levels caused by higher call rates are compensated by larger congestion windows and fewer timeouts. As a consequence, the SIP traffic experiences lower delays.

Table 2. Parameter values (competing with TCP).

Parameter	**Values**
Router's queue size	10–20, 50–60 packets
Number of TCP connections	1, 2, 4, 6
One-way link delay	20, 60 ms
Call rate	20, 40 calls/s
Delivery type	Ordered, Unordered

A.5 Effect of HOLB vs. the Packet Loss Rate

For any given delay value in Fig. 4, we can observe its corresponding packet loss rate with or without HOLB. For example, when the one-way link delay is 20 ms and the call rate 40 calls per second, a post-selection delay of slightly below 200 ms corresponds to a packet loss rate of 2% with HOLB and of 4% with no HOLB.

We compared the distributions used to build the graphs using Kolmogorov-Smirnov tests. A packet loss rate of 1% with HOLB always caused significantly (statistically at the 95% confidence level) higher delays than a packet loss rate of 2% without HOLB. A packet loss rate of 2% with HOLB caused a statistically equivalent delay as a packet loss rate of 4% without HOLB.

B. Competing Long-Lived TCP Connections

For this set of experiments, we also configured NIST Net to emulate a packet queue with a limited capacity at the router. Packet losses were produced by the arrival of too many incoming packets at the router. We used the link delays (20 and 60 ms), call rates (20 and 40 calls per second) and router's queue sizes (10–20, and 50–60 packets) discussed in Section VI.

In order to perform experiments with different packet loss rates, we established a number of TCP connections between the computers hosting the SIP applications. Those TCP connections competed with the SCTP connection carrying SIP traffic for the same bandwidth. We let the TCP connections settle before starting the SIP traffic. We could assign different values to five parameters: The router's queue size, the number of competing TCP connections, the link delay, the call rate, and the delivery type. Table 2 shows the values we used for the experiments involving competing long-lived TCP connections. The values in Table 2 yield 64 different configurations. Consequently, we ran 640 one-minute-long experiments.

B.1 Effect of the Router's Queue Size

Figs. 5 and 6 show the results of the experiments using the 10–20 and 50–60 packet queues respectively. The graphs show that the 95th percentile delay was significantly lower for the larger router's queue (i.e., 50–60) in most scenarios. This is because SCTP can develop a larger congestion window for larger queue sizes. Therefore, in this case, increasing the throughput of the SCTP association decreases the delay.

Note, however, that a dramatic increase in the router's queue size would increase the throughput of SCTP but would also increase the delay. This is because the competing TCP connections would saturate the router's queue. Such a long queue, when saturated, would introduce high delays to the all packets. That is why networks used for signaling traffic that competes with other types of traffic need to be carefully dimensioned.

B.2 Effect of HOLB

In all scenarios shown in Figs. 5 and 6, delays increase as the SCTP association carrying SIP traffic needs to compete for bandwidth with more TCP connections. Kolmogorov-Smirnov tests confirmed that HOLB introduces a statistically significant delay at the 95% confidence level in all scenarios except for a 50–60 queue and a single competing TCP connection. One TCP connection with a 50–60 queue does not produce enough packet losses to make HOLB relevant, since even when HOLB is present the delay is kept at minimum (i.e., close to the RTT of the empty network).

B.3 Effect of the Packet Loss Rate

As discussed in Section VI-B.2, in all scenarios delays increase as the SCTP association carrying SIP traffic needs to compete for bandwidth with more TCP connections. As it can be observed in Figs. 5 and 6, the same number of competing TCP connections cause higher delays with the 10–20 queue than with the 50–60 queue.

The graphs with no HOLB in Fig. 5 show a faster increase than under HOLB between one and two competing TCP connections (after two connections, the graphs with HOLB are steeper). The same effect can be observed in Fig. 6 between 4 and 6 competing TCP connections and a 20 ms link delay (we would need to introduce more than 6 competing TCP connections to see this effect with a 60 ms link delay). We discuss the reasons for this faster increase focusing on Fig. 5. As discussed in Section VI-A.3, a sudden increase in the 95th percentile delay occurs when more than 5% of the messages get lost. Therefore, we could expect the effective packet loss rate in the experiments in Fig. 5 to surpass 5% when the second competing TCP connection was added.

To check that this was indeed the case, we measured the number of SCTP DATA chunks (each SCTP DATA chunk carries a SIP message) the SCTP stacks retransmitted in every experiment. Not all these retransmissions were caused by the loss of a packet. SCTP's fast retransmit algorithm retransmits a given DATA chunk when the sender receives three duplicated acknowledgments from the receiver. However, from the time the receiver generates those duplicated acknowledgments until the sender receives them, the sender continues sending new DATA chunks. The higher the link delay and the call rate are, the more DATA chunks the sender generates before retransmitting the lost DATA chunk.

On receiving each of these DATA chunks, the receiver generates a new duplicated acknowledgment. If the sender, after retransmitting the lost DATA chunk, receives three new duplicated acknowledgments, it retransmits the lost DATA chunk once again (note that although some TCP stacks do not retrans-

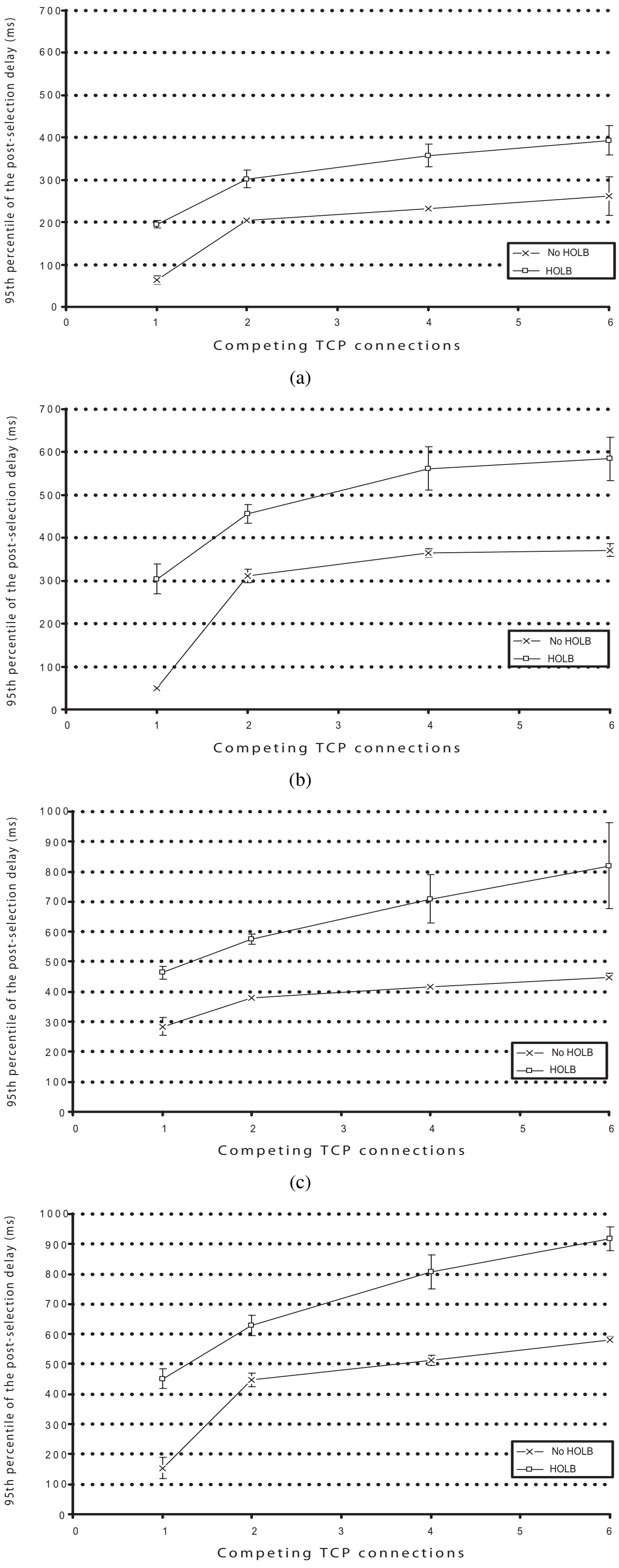
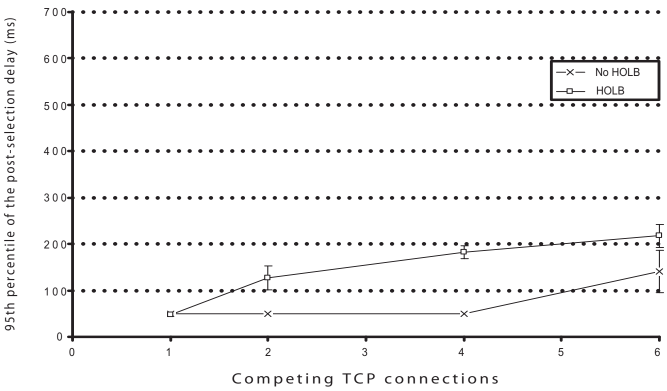
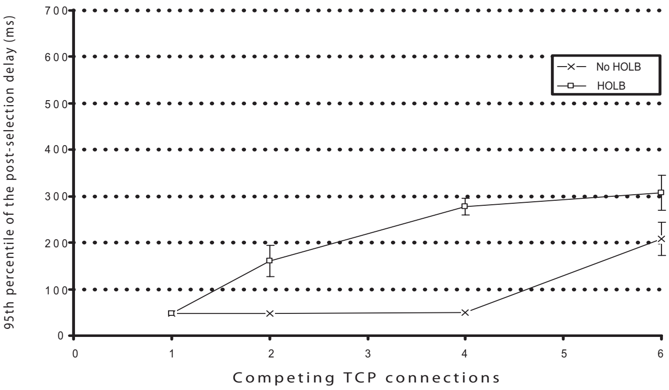


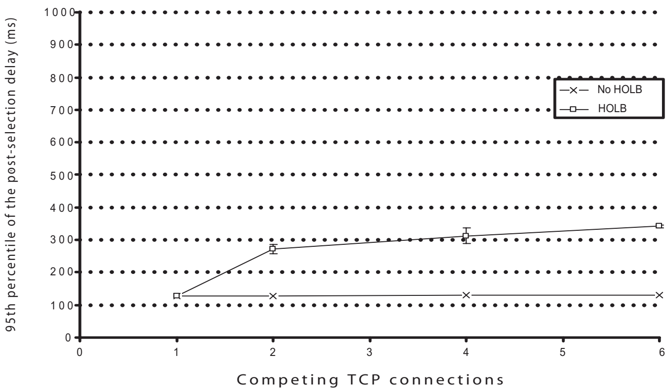
Fig. 5. Delay under competing long-lived TCP connections using a 10–20 packet queue: (a) 20 ms one-way delay and 40 calls/s, (b) 20 ms one-way delay and 20 calls/s, (c) 60 ms one-way delay and 40 calls/s, and (d) 60 ms one-way delay and 20 calls/s.



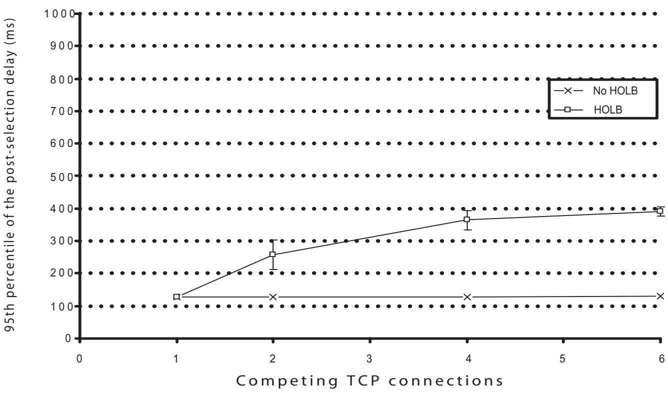
(a)



(b)



(c)



(d)

Fig. 6. Delay under competing long-lived TCP connections using a 50–60 packet queue: (a) 20 ms one-way delay and 40 calls/s, (b) 20 ms one-way delay and 20 calls/s, (c) 60 ms one-way delay and 40 calls/s, and (d) 60 ms one-way delay and 20 calls/s.

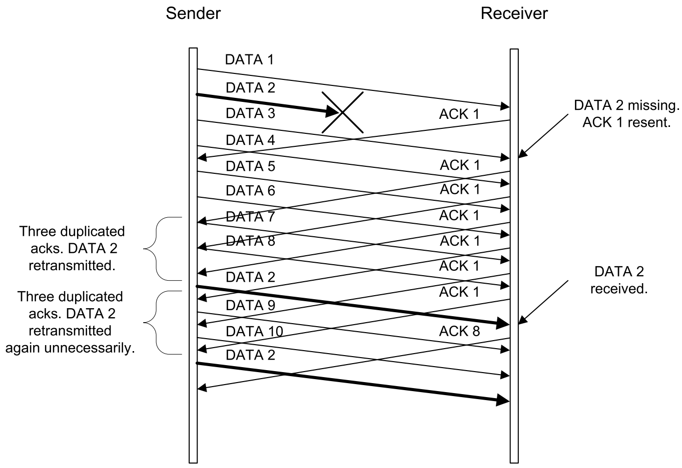


Fig. 7. Unnecessary retransmission.

mit a segment more than once using the fast retransmit algorithm, SCTP stacks do). This retransmission may be unnecessary, because the first retransmission of the DATA chunk may have already arrived to the receiver, as shown in Fig. 7.

We measured the number of duplicated DATA chunks received by the SCTP stacks (i.e., they were unnecessarily retransmitted) and subtracted this figure from the total number of retransmitted DATA chunks. That yielded the number of retransmissions caused by packet losses. As expected, a single competing TCP connection caused a packet loss rate lower than 5%. Two competing TCP connections caused a packet loss rate higher than 5%. The only exception was the experiments corresponding to a 60 ms link delay and 40 calls/s, where a single competing TCP connection managed to produce more than a 5% packet loss rate. That is why its graph shows a less sudden delay increase. As in the experiments analyzed in Section VI-A.3, the sudden increase in the 95th percentile of the post-selection delay does not occur under HOLB.

B.4 Effect of the Call Rate

Figs. 5 and 6 show that for a given one-way link delay (i.e., 20 or 60 ms), increases in the call rate decreased the delay experienced by the SIP traffic. The reasons for this behavior are discussed in Section VI-A.4.

B.5 Effect of HOLB vs. the Packet Loss Rate

Fig. 5 shows the results of the experiments using the 10–20 packet queue for our four scenarios. For any given delay value in Fig. 5, we can observe its corresponding number of competing TCP connections with or without HOLB. For example, when the one-way link delay is 20 ms and the call rate 40 calls per second, a post-selection delay of around 200 ms corresponds to one competing TCP connection with HOLB and two with no HOLB.

We compared different points in the graphs using Kolmogorov-Smirnov tests. A single competing TCP connection with HOLB caused a statistically equivalent (at the 95% confidence level) delay as two competing TCP connections without HOLB in all but one scenario (60 ms link delay and 40 calls/s), where it

Table 3. Parameter values (competing with web traffic).

Parameter	Values
Router's queue size	10–20, 50–60 packets
Web session rate	1, 2, 3
One-way link delay	20, 60 ms
Call rate	20, 40 calls/s
Delivery type	Ordered, Unordered

caused significantly higher delays. Six competing TCP connections with no HOLB caused a statistically equivalent (at the 95% confidence level) delay as between one and two competing TCP connections with HOLB in all scenarios.

As discussed in Section VI-B.2, the effect of HOLB with the 50–60 queue and a single competing TCP connection is negligible. However, when the number of competing TCP connections grows beyond one, the effect of HOLB is even higher than for the 10–20 queue, as it can be seen in Fig. 6. With a link delay of 20 ms, two competing TCP connections with HOLB caused a statistically equivalent delay as six competing TCP connections without HOLB. With a link delay of 60 ms, two competing TCP connections with HOLB caused a statistically higher delay than six competing TCP connections without HOLB.

C. Competing Web Traffic

For this set of experiments, we configured NIST Net to emulate a packet queue with a limited capacity at the router. Packet losses were produced by the arrival of too many incoming packets at the router. We used the link delays (20 and 60 ms), call rates (20 and 40 calls per second), and router's queue sizes (10–20 and 50–60 packets) discussed in Section VI-B.

In order to perform experiments with different packet loss rates, we established a number of web surfing sessions between the computers hosting the SIP applications. Those web sessions competed with the SCTP connection carrying SIP traffic for the same bandwidth. In order to generate the competing web traffic, we used the httperf tool [17]. We configured httperf to generate web surfing sessions at a given rate for each experiment. As shown in Table 3, we used values ranging from 1 to 3 sessions per second. Each web surfing session consisted of three downloads separated by 10 seconds, which represents the user's thinking time between consecutive downloads. In each download, the HTTP client downloads a web page including all its objects from the HTTP server. The web page consisted of 3 objects (and html file plus two pictures), which resulted in a total size of 85 kilobytes per download. The parameters to generate the web traffic (i.e., thinking time and web page structure) simulate the behavior of a user surfing the web. Since we needed to generate web traffic simply to emulate competing traffic of a bursty nature, we did not have the need to use a finer traffic model.

In these experiments, we could assign different values to five parameters: the link delay, the router's queue size, the web session rate, the call rate, and the delivery type. Table 3 shows the values we used for the experiments involving competing web traffic. The values in Table 3 yield 48 different configurations. Consequently, we ran 480 one-minute-long experiments.

C.1 Effect of the Router's Queue Size

When we used a 50–60 packet queue at the router, the SCTP association was able to keep the delay of the SIP traffic at minimum regardless of the value of the rest of the parameters. That is, the 95th percentile delay in all the experiments using the 50–60 packet queue was equivalent to the 95th percentile delay with an empty network (i.e., no competing traffic). Section VI-B.1 discusses why a longer queue size (within limits) helps keep the 95th percentile delay at a lower level.

C.2 Effect of HOLB

As discussed in Section VI-C.1, the effect of HOLB on the delay was negligible when we used the 50–60 packet queue. Fig. 8 shows the results of the experiments using the 10–20 packet queue for our four scenarios. The variability of the 95th percentile of the delay in the experiments under HOLB was high, especially for the 60 ms link. Despite this high variability, which results in a high variance and thus in wide confidence intervals, the effect of HOLB was statistically significant in all scenarios.

C.3 Effect of the Packet Loss Rate

Fig. 8 shows that, in all scenarios, the delay increased as the SCTP association carrying SIP traffic needs to compete for bandwidth with more web sessions. More competing web sessions caused higher packet loss rates.

C.4 Effect of the Call Rate

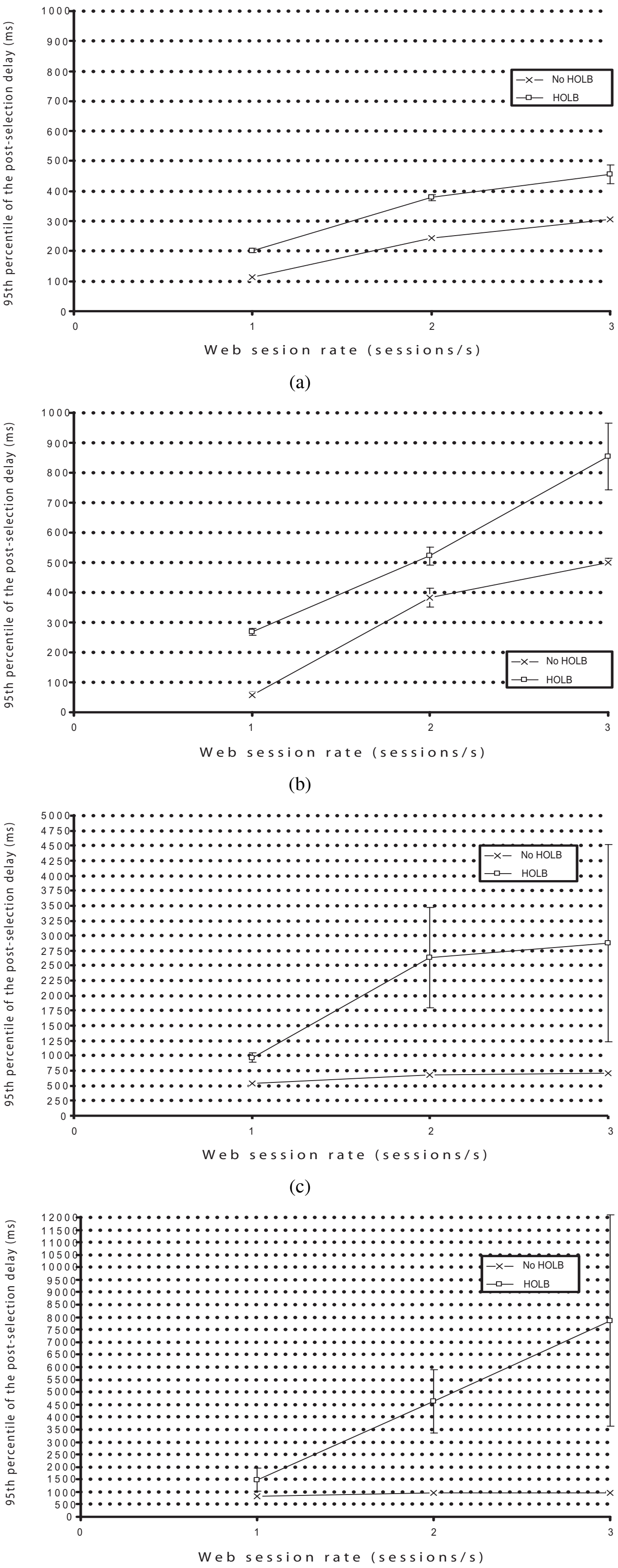
Fig. 8 shows that for a given one-way link delay (i.e., 20 or 60 ms), increases in the call rate decreased the delay experienced by the SIP traffic. The reasons for this behavior are discussed in Section VI-A.4.

VII. EXPERIMENTS ON THE PUBLIC INTERNET

In order to check the validity of our results using emulated environments (which are discussed in Section VI), we performed a set of preliminary experiments on the public Internet. Our future work includes performing more experiments on the public Internet and on distributed test beds such as Planetlab to test a wide variety of network conditions.

In this set of experiments, we replaced the router in Fig. 3 with the public Internet. We installed our SIP application on computers in Helsinki (Finland), Madrid (Spain), and New York (U.S.A.). We ran experiments between the SIP application in Helsinki, and the SIP applications in Madrid and New York. The path between the machines in Helsinki and Madrid included 14 routers and had a RTT of approximately 70 ms. The path between the machines in Helsinki and New York included 13 routers and had a RTT of approximately 120 ms. We ran two types of experiments. In the first experiment, we only ran SIP traffic between both machines. In the second experiment, we added two long-lived TCP connections transferring large files between both machines.

Table 4 shows the values we used for the experiments performed on the public Internet. The values in Table 4 yield 16



(a) (b) (c) (d)

Fig. 8. Delay under competing web traffic using a 10–20 packet queue: (a) 20 ms one-way delay and 40 calls/s, (b) 20 ms one-way delay and 20 calls/s, (c) 60 ms one-way delay and 40 calls/s, and (d) 60 ms one-way delay and 20 calls/s.

Table 4. Parameter values (public Internet).

Parameter	Values
Number of TCP connections	0, 2
Round-trip time	Madrid, New York
Call rate	20, 40 calls/s
Delivery type	Ordered, Unordered

different configurations. Consequently, we ran 160 one-minute-long experiments.

Fig. 9 shows the results of the experiments. The results of our experiments between Helsinki and Madrid show that the effect of HOLB was statistically significant when we did not generate competing TCP traffic. With two competing TCP connection, the observed delay varied considerably between experiments. These variations result in a high variance, which makes the effects of HOLB statistically insignificant. The results of our experiments between Helsinki and New York show that the effect of HOLB was statistically significant only for a call rate of 40 calls/s and no competing TCP traffic. Even though the variability in the experiments with two competing TCP connection was much lower than in the experiments between Helsinki and Madrid, it was enough to make the effects of HOLB statistically insignificant.

As it can be observed in Fig. 9, the 95th percentile of the post-selection delay under two competing TCP connections was higher than 4 seconds. In order to measure how much of that delay corresponded to the queues of the routers in the path (as opposed to delay caused by packet losses and the congestion window), we followed the method in [18] to measure queue lengths. We ran a ping application between both nodes to obtained the maximum RTT between the nodes during the experiments. The maximum RTT between Helsinki and Madrid was in the order of 1.6 seconds. The maximum RTT between Helsinki and New York was in the order of 1.7 seconds.

VIII. CONCLUSIONS

In this paper, we have studied the influence of HOLB on the performance of SCTP transporting SIP traffic. Specifically, we have studied the influence of HOLB on the 95th percentile of the post-selection delay, which is the value regulators typically use to measure the performance of network operators. Our experiments include SCTP associations under random packet losses and SCTP associations competing for bandwidth with TCP connections carrying large files and web sessions. We performed these experiments using several different link delays, call rates, and routers' queue sizes in order to make our results widely applicable.

The degree in which HOLB influences the post-selection delay depends on the scenario being studied. In general, HOLB has a statistically significant effect in the 95th percentile of the post-selection delay as soon as the network conditions are not optimal. That is, the effect of HOLB generally starts to be significant starting at fairly low packet loss rates. However, there are scenarios where the effect of HOLB is not significant. Favorable network conditions can allow an SCTP association achieve enough throughput to keep delays low even under HOLB. In

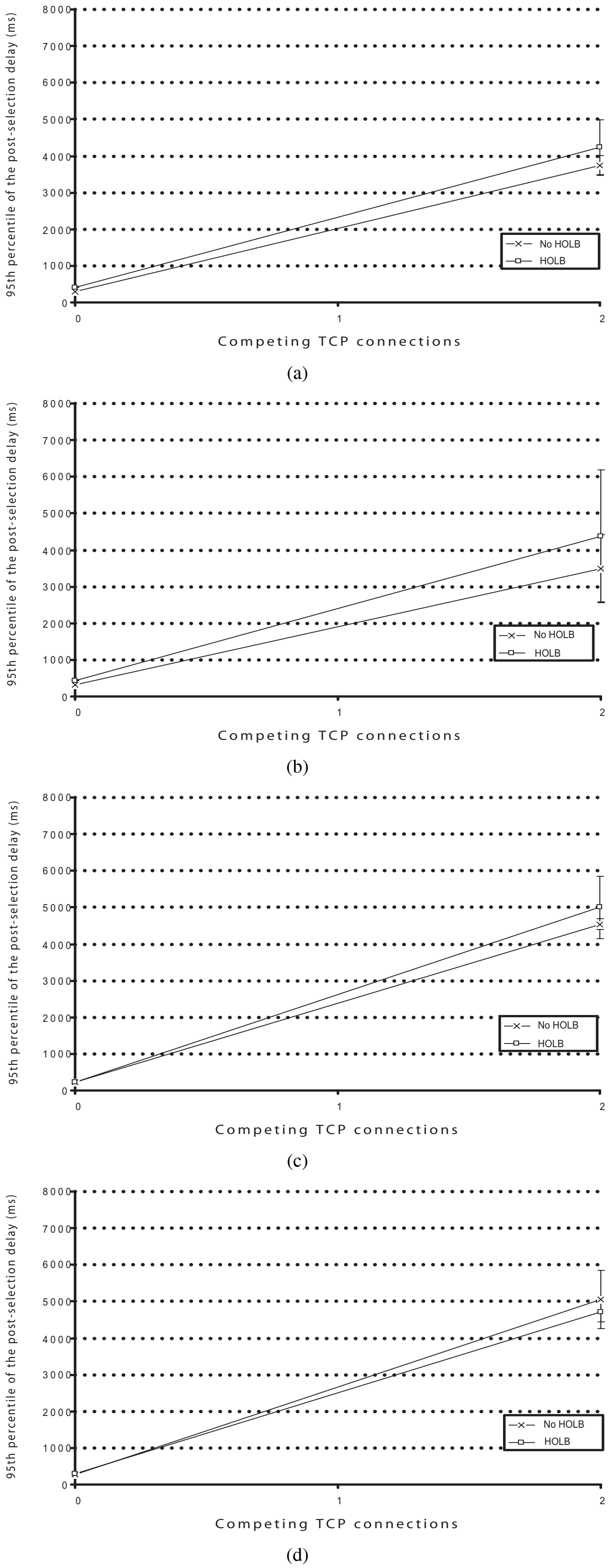


Fig. 9. Delay on the public Internet: (a) Helsinki–Madrid with 40 calls/s, (b) Helsinki–Madrid with 20 calls/s, (c) Helsinki–New York with 40 calls/s, and (d) Helsinki–New York with 20 calls/s.

some scenarios, even if avoiding HOLB systematically reduced the 95th percentile of the post-selection delay, such a reduction was not statistically significant. This was because a high variability in the results of the experiments under HOLB made confidence intervals wide enough.

Previous studies arrived to seemingly contradictory conclusions on the significance of the effect of HOLB on the delay because they focused on a narrow set of scenarios. In this paper, we have considered a wide set of scenarios so that our results have a more general applicability.

Another advantage derived from avoiding HOLB is that it reduces the variability in the post-selection delay for sessions that do not suffer any packet loss (which in normal circumstances are most of the sessions). All these sessions experiment a similar delay, which makes session establishment times more easily predictable for users.

ACKNOWLEDGEMENTS

The experiments on the public Internet were performed using the infrastructure of the Onelab project.

REFERENCES

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session initiation protocol," Internet Engineering Task Force, Tech. Rep. RFC 3261, June 2002.

[2] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream control transmission protocol," Internet Engineering Task Force, Tech. Rep. RFC 2960, Oct. 2000.

[3] ITU-T, "Network grade of service parameters and target values for circuit-switched services in the evolving ISDN," International Telecommunication Union, Tech. Rep. Recommendation E.721, May 1999.

[4] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP selective acknowledgement options," Internet Engineering Task Force, Tech. Rep. RFC 2018, Oct. 1996.

[5] G. Camarillo, H. Schulzrinne, and R. Kantola, "Evaluation of transport protocols for the session initiation protocol," *IEEE Network*, vol. 17, no. 5, 2003.

[6] X. Li, D. Salyers, and A. Striegel, "Improving packet caching scalability through the concept of an explicit end of data marker," in *Proc. 1st IEEE Workshop on HOTWEB.*, 2006.

[7] K. Ono and H. Schulzrinne, "The impact of SCTP on server scalability and performance," Dept. of Computer Science, Columbia University, Tech. Rep. CUCS-012-08, Feb. 2008.

[8] J. Rosenberg, H. Schulzrinne, and G. Camarillo, "The stream control transmission protocol (SCTP) as a transport for the session initiation protocol (SIP)," Internet Engineering Task Force, Tech. Rep. RFC 4168, Oct. 2005.

[9] T. Dierks and E. Rescorla, "The transport layer security (TLS) protocol version 1.1," Internet Engineering Task Force, Tech. Rep. RFC 4346, Apr. 2006.

[10] http://www.isi.edu/nsnam/ns/index.html.

[11] K. Grinnem, T. Andersson, and A. Brunstrom, "Performance benefits of avoiding head-of-line blocking in SCTP," in *Proc. Autonomic and Autonomous Systems and International Conference on Networking and Services*, 2005.

[12] M. Scharf and S. Kiesel, "Head-of-line blocking in TCP and SCTP: Analysis and measurements," in *Proc. IEEE Globecom*, 2006.

[13] M. Carson and D. Santay, "NIST net: A linux-based network emulation tool," *ACM CCR*, vol. 33, no. 3, 2003.

[14] http://sourceforge.net/projects/kphone.

[15] http://www.internettrafficreport.com.

[16] A. Mondal and A. Kuzmanovic, "When tcp friendliness becomes harmful," in *Proc. IEEE Infocom*, May 2007.

[17] D. Mosberger and T. Jin, "Httperf: A tool for measuring web server performance," *Performance Evaluation Review*, vol. 26, no. 53, 1998.

[18] M. Claypool, R. Kinicki, M. Li, J. Nichols, and H. Wu, "Inferring queue sizes in access networks by active measurement," Dept. of Computer Sci-

ence, Worcester Polytechnic Institute, Tech. Rep. WPI-CS-TR-04-04, Feb. 2004.

Gonzalo Camarillo received M.Sc. degrees in electrical engineering from the Stockholm's (Sweden) Royal Institute of Technology and from Universidad Politecnica de Madrid (Spain). He has co-authored, among other standards, the SIP (Session Initiation Protocol) specification (RFC 3261). He currently serves in the IAB (Internet Architecture Board). He is the IETF liaison manager to 3GPP and currently co-chairs the SIPPING and HIP working groups in the IETF (Internet Engineering Task Force). He is the author of the books "SIP Demystified" and "The 3G IP Multimedia Subsystem (IMS)". He regularly publishes scientific papers in conferences, journals, and magazines. He heads the Multimedia Research Laboratory, which is part of the NomadicLab, at Ericsson Research in Finland. His research interests include signalling, multimedia applications, transport protocols, network security, and network architectures.

Henning Schulzrinne received his Ph.D. from the University of Massachusetts in Amherst, Massachusetts. He was a Member of technical staff at AT&T Bell Laboratories, Murray Hill and an associate department head at GMD-Fokus (Berlin), before joining the Computer Science and Electrical Engineering departments at Columbia University, New York. He is currently chair of the Department of Computer Science. Protocols co-developed by him, such as RTP, RTSP and SIP, are now Internet standards, used by almost all Internet telephony and multimedia applications. His research interests include Internet multimedia systems, ubiquitous computing, mobile systems, quality of service, and performance evaluation. He is a Fellow of the IEEE.

Jani Hautakorpi works as a research scientist in the Multimedia Research Laboratory, which is part of the NomdicLab, at Ericsson Research in Finland. He holds a B.Sc. degree from Satakunta University of Applied Sciences (SAMK) and an M.Sc degree from Tampere University of Technology (TUT). Currently he is working towards a Ph.D. degree in Helsinki University of Technology (HUT). His research interests lie on decentralized interpersonal communication mechanisms, especially on aspects related to security, interworking, service replication, and incentives. More concretely, a considerable amount of his work relate to peer-to-peer session initiation protocol (P2PSIP) technology. In addition, he also contributes to standardization work, mainly in the IETF.

Salvatore Loreto received his degree in Computer Engineering from the University of Napoli Federico II, Italy, in 1999. He obtained a Ph.D. degree in Computer Networks in 2005. Since 2000, he has been working for Ericsson both in Italy and in Finland. He is currently working as a research scientist at the Multimedia Research Laboratory, which is part of the NomadicLab, at Ericsson Research in Finland. He has authored and co-authored of several papers on transport and signaling protocols. His research interests include signaling, multimedia applications, transport protocols and network security. He is also involved in Ericsson's standardization activities, mainly at the IETF.