Signaling Transport Options in GMPLS Networks: In-band or Out-of-band

Malathi Veeraraghavan and Tao Li
Department of Electrical and Computer Engineering, University of Virginia
351 McCormick Rd., P. O. Box 400743, Charlottesville, VA 22904, USA
Email: {mv5g, taoli}@virginia.edu

***Abstract*—Signaling protocols for GMPLS networks have been standardized and implemented in switch controllers. Most switch vendors allow for signaling messages to be carried over in-band signaling channels as well as through out-of-band networks. In this paper, we compare these two signaling transport options. In carrying out this analysis, we allow for both software-implemented signaling protocol processors, as is common in most off-the-shelf switches today, as well as hardware-accelerated signaling protocol engines. Our motivating application is file transfers, which have high call arrival rates and low call holding times. The resulting high signaling message load makes the question of which signaling transport solution to use important. Network delays are lower in the in-band option given that there are no IP routers on these paths, but the per-channel transmitter rates are lower for in-band channels. To study this tradeoff, we set up queueing models and obtained delay estimates. Our analysis shows that with hardware-accelerated signaling engines, in-band signaling is the better option to keep end-to-end call setup delays low. With software signaling protocol processors, the most significant component is the message processing delay, which is likely to include queueing delays, making the question of in-band or out-of-band transport less important.**

***Index Terms*—GMPLS networks, signaling protocols, in-band, out-of-band, hardware acceleration, signaling transport**

I. INTRODUCTION

The Generalized Multi-Protocol Label Switching (GMPLS) architecture [1] for Synchronous Optical NETwork (SONET)/Synchronous Digital Hierarchy (SDH) and Wavelength Division Multiplexing (WDM) networks, and associated signaling protocols, such as Resource reSerVation Protocol-Traffic Engineering (RSVP-TE) [2], have been defined to enable the deployment of dynamically controlled high-speed circuit-switched networks. The RSVP-TE specification [2] allows for signaling messages to be carried directly in IP packets or UDP datagrams, but it does not constrain the transport options needed to carry these IP packets in any way.

Equipment vendors for SONET/SDH crossconnects have implemented the RSVP-TE signaling protocol [3], and support two options for the transport of signaling messages: (i) in-band signaling channels, which are realized as bandwidth set aside within user-plane interfaces between two switches, e.g., the Data Communication Channel (DCC) within each SONET signal, and (ii) out-of-band signaling channels, e.g., an Ethernet interface from the control (call) processor of the switch to an IP network. This work is intended to provide a quantitative analysis on the question of which option to use.

We are motivated to answer this question as the next step in our continuing goal to reduce call setup delay. Call setup delay is the unavoidable overhead associated with circuit-switched/virtual-circuit networks. Given the positive features of these networks to offer quality-of-service guarantees for admitted calls, we have undertaken several projects and published papers on reducing this delay.

Call setup delay consists of (i) signaling message processing delays (also referred to as "call processing" delays), and (ii) signaling message transport delays. Toward reducing the first component, signaling message processing delays, we implemented a hardware-accelerated RSVP-TE signaling board, as part of an NSF-sponsored project [4]. With this implementation, we demonstrated processing delays of $3\mu s$ for an RSVP-TE *Path* message, in contrast to 92ms measured in an off-the-shelf switch [5]. Having achieved this significant reduction in the first component, we now turn our attention to the second component of call setup delay, i.e., signaling message transport delays. The goal of this paper is compare in-band and out-of-band signaling transport solutions to find the best solution to lower this component.

In our quantitative analysis of these two solutions, our assumptions for parameters, such as call arrival rates, call holding times, etc., are based on our experience with implementing an experimental wide-area network testbed called Circuit-switched High-speed End-to-End ArcHitecture (CHEETAH) [3]. The CHEETAH network uses off-the-shelf SONET switches, with Ethernet interfaces to connect end hosts, and built-in RSVP-TE signaling software. By mapping Gigabit Ethernet ports to equivalent-rate virtually concatenated SONET circuits, the CHEETAH network offers a dynamic 1Gbps-rate circuit service with call-by-call bandwidth sharing. **File-transfer** applications constitute the primary usage of this network. File transfers are ideally suited for high-speed circuits because a file transfer can take advantage of "any" data rate (the higher the better), and furthermore, a file transfer requires the movement of stored data, which means there is no inherent burstiness in this application, making it ideal for a full utilization of a circuit. We have noted in previous publications, posted at [3], that the elastic nature of bandwidth sharing in TCP/IP packet-switched networks, which is often cited as an advantage for file transfers, becomes a disadvantage for large

---

This work is sponsored by an NSF EIN grant, 0335190, an NSF ITR grant, 0312376, and a DOE grant DE-FG02-04ER25640.

files across high bandwidth-delay-product (BDP) networks, making circuits a better option for the latter. When dedicated high-speed circuits are used, file transfer times can become quite short. For example, it takes only 800ms to transfer a 100MB file on a 1Gbps circuit. Therefore, this application can be regarded as rather extreme in its generation of calls, and corresponding signaling message load. The in-band/out-of-band question becomes important in this application because of its need to support high signaling message load and low call setup delays. We will consider this question of in-band vs. out-of-band transport under assumptions of both hardware-accelerated signaling implementations [4] and software signaling implementations [5].

After reviewing related work in Section II, we describe the in-band and out-of-band transport options in Section III. We describe our queueing models for the two options in Section IV, and provide numerical results of our comparison in Section V. Our conclusions are presented in Section VI.

II. RELATED WORK

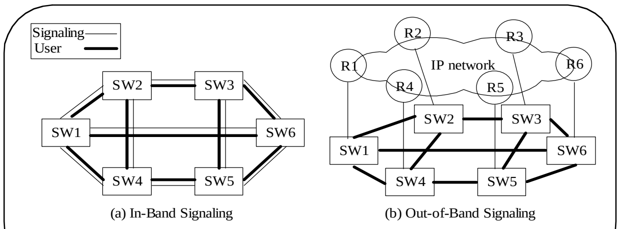
This paper is a revised version of an IEEE workshop paper [6] on the same topic. Specifically, we have improved the model based on our practical experience with the CHEETAH network, and obtained new numerical results.

Papers on this topic are relatively few because this question of choosing a transport option is not critical for the types of applications envisioned for dynamically shared GMPLS networks. Specifically, the industry cites rapid provisioning and fast restoration as two applications for RSVP-TE-enabled GMPLS switches. In the “rapid provisioning” application, RSVP-TE is used for only the provisioning step, i.e., switch fabric configuration. It is a mechanism to improve the turn-around time for leased circuits. Service providers expect relatively low call arrival rates since these circuits are expected to be held for long durations, on the order of months to years. Therefore, provisioning delay is not a major concern, which implies that either in-band or out-of-band transport can be used.

For fast restoration, we do not see how RSVP-TE based bandwidth allocation and circuit provisioning can be used if off-the-shelf switches take on the order of 100ms to process a single *Path* message. For reliability reasons, switch vendors and service providers expect to use in-band signaling for this application, but without hardware-accelerated RSVP-TE implementations, we do not foresee RSVP-TE based signaling being used for fast restoration.

III. IN-BAND AND OUT-OF-BAND SIGNALING OPTIONS

Fig. 1 illustrates the two options for signaling transport: *in-band signaling* and *out-of-band signaling*. In the in-band signaling option, the signaling traffic shares the same channel as the data traffic. For example, the DCC channel in the SONET signal can be used to transport signaling messages, as shown in Fig. 1a. In each OC-1 signal, there is a Section DCC channel with a bandwidth of 192kbps, and a Line DCC with a bandwidth of 576kbps. These data rates are much smaller than those of out-of-band channels, which are typically 10Mbps or 100Mbps Ethernet. Thus, per-message transmission delay will be higher if a message is sent on a single DCC channel; however the aggregate signaling bandwidth is high because there is one DCC per OC1. For example, in an OC192, this results in a total of $192 \times 576 kbps = 110 Mbps$. Thus fairly large signaling message load can be handled by this aggregate set of DCC channels.



(a) In-Band Signaling (b) Out-of-Band Signaling

Fig. 1 In-band and out-of-band signaling options.

Fig. 1b illustrates the out-of-band signaling option. The key difference is that the path taken by the signaling channel is necessarily different from the path taken by the user-plane interfaces that it supports. For example, the signaling channel could pass through packet switches, such as IP routers, while the user-plane interfaces are direct (logical or physical) between two switches. In other words, the signaling channel is separate from and independent of the data channel. A classical example of out-of-band signaling is the Signaling System 7 (SS7) network, which is used to carry signaling messages between DS0-based telephone circuit switches. SS7 is a connectionless packet-switched technology.

Unlike in telephone networks, when it was economically feasible to create a dedicated connectionless packet-switched network just for signaling traffic, in today’s environment, with the ubiquity of the Internet, it is more likely that GMPLS-enabled SONET/SDH/WDM circuit switches will leverage the Internet for signaling message transport. A service provider could simply connect the control processors of their GMPLS-enabled circuit switches to the Internet and expect it to route signaling messages as needed. A potential drawback is latency. We will create models for such out-of-band signaling channels routed through the Internet, and analyze these models to predict performance.

IV. DELAY MODELS FOR THE TWO OPTIONS

In this section, we set up and analyze models for the two signaling transport options, *in-band signaling* and *out-of-band signaling*. Our goal is to compute the total delay incurred in processing and sending a signaling message from one switch to the next successfully. This consists of the following components: (i) queueing delay plus service time at the signaling protocol processor, (ii) queueing delay plus transmission time (or emission time) on the signaling channel, (iii) delay to successfully send the message on the signaling channel, which includes propagation delay, the time needed for retransmissions in case of message loss, and queueing delays at IP routers for the out-of-band option. We list our assumptions and notation, and then describe our queueing model in the following subsections.

A. Assumptions

The call arrival process for FTP applications is shown to be Poisson [7]. Each call, in RSVP-TE, involves *Path* messages, *Resv* messages, and *PathTear* or *ResvTear* messages. To simplify analysis, we approximate the arrival process of signaling messages to also be Poisson, even though the messages involved in a call are correlated.

The service times at signaling protocol processors and the transmission times at signaling channels are both assumed to be deterministic. The approximate size of the 4 common signaling messages is 125 bytes, which validates our assumption about message transmission times. We assume that each message fits in one IP packet, since the maximum transmission-unit size of most existing networks is larger than the size of a typical RSVP-TE message.

B. Notation

Our notation is shown in Table 1. In the rest of the paper, we use superscripts OOB and IB to represent out-of-band and in-band, respectively, for some parameters, when this distinction is required.

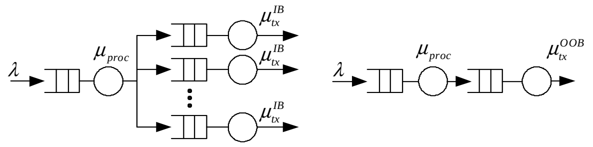
Table 1: Notation

Symbol	**Meaning**
λ	Aggregate signaling message arrival rate
μ_{proc}	Service rate of the signaling protocol processor
μ_{tx}	Service rate of the signaling channel transmitter
n	Number of GMPLS-enabled neighbors to a switch (also the number of in-band signaling channels)
T_{proc}	A random variable denoting the response time at the signaling protocol processor (waiting time plus service time)
T_{tx}	A random variable denoting the response time at the signaling channel transmitter (waiting time plus service time)
T_n	One-way network delay in sending a signaling message from one switch to the next
T_o	Initial time-out value of the retransmission timer at the sender
p	Probability of packet loss

C. Queueing model

There are two servers related to signaling at a GMPLS-enabled circuit switch: (i) signaling protocol processor, and (ii) signaling channel transmitter. We assume that the switch has a single signaling protocol processor, irrespective of whether the transport option is in-band or out-of-band. We first describe our queueing model for the signaling protocol processor and the signaling channel transmitter without considerations of message loss. We then improve this model by adding in the possibility of message loss, and corresponding retransmissions.

1) Model without retransmissions



(a) In-Band Signaling (b) Out-of-Band Signaling

Fig. 2 Queueing models of the signaling protocol processor and the signaling channel transmitters at a GMPLS-enabled circuit switch.

Fig. 2 illustrates our queueing models of these two servers for both in-band and out-of-band solutions. In the in-band solution, there are n signaling channels given our assumption of n GMPLS-enabled neighbors to the switch (see Table 1 and Fig. 1). In the out-of-band solution, we assume that there is only one out-of-band signaling channel.

The models shown in Fig. 2 are networks of queues. We cannot obtain an exact analytical solution for this network of queues because the output process of the first server is not Poisson. This is caused by our assumption of deterministic services times, as stated in subsection A. Although an exact analytical derivation is hard to obtain, it turns out that if we take into account practical considerations, this network of queues reduces to a single M/D/1 queue plus a constant delay line. Our reasoning is as follows.

If the signaling protocol implementation is in software, and message processing delays are on the order of 50ms [5], while transmission delays are on the order of 1.7ms for the lower-speed DCC channel ($1000 bits/576 kbps$), the waiting time for the second server (the transmitters in Fig. 2) is actually zero. For the out-of-band channel, transmission delays are even lower. Therefore, we assume $\mu_{proc} \le \mu_{tx}$ for software signaling. The multiple-server queueing network of Fig. 2 thus reduces to a single-queue system with the server processing messages at a rate of μ_{proc}, followed by another server operating at rate μ_{tx} with no waiting time. Denote $E[T_{sw}]$ as the mean response time for software signaling. We have

$$E[T_{sw}] = E[T_{proc}] + \frac{1}{\mu_{tx}}, \tag{1}$$

where T_{proc} and μ_{tx} are defined in Table 1.

For the hardware-accelerated signaling protocol processor, the reverse is true. As indicated earlier, per-message processing delays are on the order of μs with hardware signaling. If data-plane considerations are taken into account (see Appendix for details), call arrival rates will likely be limited compared to

μ_{proc} even for our aggressive file-transfer application. We assume $\mu_{proc} \gg \lambda$ for hardware signaling. Queueing delays in the signaling processor queue can thus be largely neglected. We approximate the output process of the first server to be a delayed version of the input Poisson process, and consider an M/D/1 model for the transmitters shown in Fig. 2. A queueing delay could be incurred at the transmitter, especially in the in-band case, when per-message service time is around 1.7ms. The queueing network model is thus approximated to a constant delay line plus a single-queue system. Denote $E[T_{hw}]$ as the mean response time for hardware signaling. We obtain

$$E[T_{hw}] = \frac{1}{\mu_{proc}} + E[T_{tx}], \tag{2}$$

where $E[T_{tx}]$ is the mean response time at a channel transmitter.

2) Model including retransmissions

In this model, we include the third component of the delay, which consists of the time needed to successfully send the message to the next node on the in-band channel or across the out-of-band network. We define a term called the one-way network delay, denoted as T_n (see Table 1). For out-of-band signaling this delay includes queueing delays at IP routers along with propagation delay. For in-band signaling, this one-way network delay includes just propagation delay.

This network delay, T_n, and the transmission delay, T_{tx}, could be incurred multiple times if there is message loss followed by a retransmission. Message loss is possible irrespective of whether the signaling channel is in-band or out-of-band. Reasons for message loss in the in-band case include *bit-* and *burst-errors* on the links, and *receive-buffer overflows* (flow control problem). Bit- and burst-link errors arise due to noise and interference on the physical media. Even though optical fiber, the physical medium of these high-speed circuit-switched networks, is fairly reliable, link errors are unavoidable. Receive-buffer overflows will occur if the signaling protocol processor at the sending switch is faster than that at the receiving side. Since different switch vendors could use different implementation techniques for the signaling protocol processor, this flow control problem may arise. In the out-of-band solution, an additional cause of message loss is buffer overflow at any of the IP routers on the path between the sending and receiving switches.

Since RSVP-TE is specified to use UDP or raw IP, and these protocols do not offer a reliable service, RFC 2961 [8] proposed an exponential back-off retransmission algorithm to provide reliability. Each message has a unique MESSAGE_ID, and a timer starts counting after the message has been transmitted. If the corresponding acknowledgment (MESSAGE_ID_ ACK) is not received within a time-out threshold, the signaling message is retransmitted, and the corresponding time-out threshold is doubled (starting from an initial retransmission time-out value, T_0).

Fig. 3 shows a retransmission model for lost messages. To model exponential doubling, we show the time-out value as a function f of the initial time-out value T_o. The loopback, as shown in Fig. 3, only includes the transmitter, and not the signaling protocol processor, since triggering a retransmission upon a time-out can be implemented in hardware irrespective of the MESSAGE_ID parameter processing.

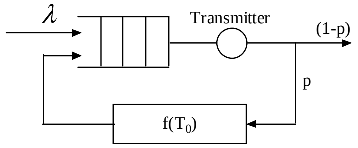


Fig. 3 Signaling channel transmitter model including retransmissions.

Rewriting (1) to include the effect of message loss and retransmission, we obtain a revised mean response time for software signaling as

$$E[T_{sw}] = E[T_{proc}] + T_n + \left(\frac{1}{\mu_{tx}}\right)[(1-p) + 2p(1-p) + 3p^2(1-p) + \ldots] + T_0[p(1-p) + 3p^2(1-p) + 7p^3(1-p) + \ldots] . \tag{3}$$

The explanation for the above equation is as follows. The first two terms, $E[T_{proc}]$ and T_n, correspond to the processing delay and the time to transmit a message successfully one-way from the sending switch to the receiving switch. The transmission delay is incurred once with probability $(1-p)$ if the first transmission is successful, twice if the first transmission is unsuccessful but the second transmission is a success, and so on. The last term in (3) corresponds to the time-out value, which is being doubled after every retransmission.

Using the mean waiting time for an M/D/1 queue to compute $E[T_{proc}]$, after some arithmetic manipulation we obtain

$$E[T_{sw}] = \left(\frac{1-\frac{\rho}{2}}{1-\rho}\right)\frac{1}{\mu_{proc}} + T_n + \frac{1}{1-p}\left(\frac{1}{\mu_{tx}}\right) + \frac{p}{1-2p}T_0, \tag{4}$$

where $\rho = \lambda/\mu_{proc}$. Assuming $T_0 = 3T_n$ (counting round-trip network delay and an extra T_n for variability), (4) can be re-written as

$$E[T_{sw}] = \left(\frac{1-\frac{\rho}{2}}{1-\rho}\right)\frac{1}{\mu_{proc}} + \frac{1}{1-p}\left(\frac{1}{\mu_{tx}}\right) + \frac{1+p}{1-2p}T_n. \tag{5}$$

For the hardware-signaling case, since a queueing delay could build up at the transmitter, we need to increase the message arrival rate because of retransmissions. The message arrival rate at the transmitter queue is given by

$$\lambda + \lambda p + \lambda p^2 + \lambda p^3 + \ldots = \frac{\lambda}{(1-p)}. \tag{6}$$

Using a similar argument as that used for (3), we obtain a revised response time for hardware signaling as

$$E[T_{hw}] = \frac{1}{\mu_{proc}} + \frac{1}{1-p}\left(\frac{1 - \frac{\rho}{2(1-p)}}{1 - \frac{\rho}{(1-p)}}\right)\frac{1}{\mu_{tx}} + \frac{1+p}{1-2p}T_n. \tag{7}$$

The M/D/1 queueing model is applied to the transmitter in this case, with the message arrival rate modified as per (6). The traffic load ρ should be adjusted for in-band channels since each in-band channel will only receive $1/n$th the message load at the signaling protocol processor (see Fig. 2).

V. NUMERICAL RESULTS

A. Input parameter values

Table 2 shows our selected values for the various input parameters. For software signaling, we vary message arrival rate to create a load ρ between 0.05 and 0.95 on the signaling processor. For the hardware-signaling/in-band transport option, the arrival rate, λ, is chosen so that the per-channel message load (which is assumed to be the same for all n in-band transmission channels) is varied between 0.05 and 0.95. For hardware signaling with out-of-band transport, the arrival rate equals the aggregate arrival rate across the n transmission channels of the in-band solution.

Table 2: Input parameter values

Symbol	Value
λ	Varied to create a message load ρ of 0.05 to 0.95 to the signaling protocol processor for software-signaling, and to an in-band transmitter for hardware-signaling.
μ_{proc}	200,000/sec, hardware signaling; corresponds to 5 μs 20/sec and 500/sec, software signaling, corresponds to 50ms, and an improved value of 2ms, respectively.
μ_{tx}	$\mu_{tx} = 500$ msg/sec, in-band channels $\mu_{tx} = 10000$ msg/sec, out-of-band channel
n	10
T_n^{IB}	$0.2ms$ in metro area $25ms$ in wide area for software signaling $5ms$ in wide area for hardware signaling
T_n^{OOB}	$1ms$ in metro area $40ms$ in wide area for software signaling $10ms$ in wide area for hardware signaling
p^{IBj}	10^{-6}
p^{OOB}	1%

Based on our measurements from the hardware-accelerated signaling processor [4] and the off-the-shelf switch [5], service rate μ_{proc} is assumed to be 200,000/sec for hardware signaling, and 20/sec for software signaling. We further assume a μ_{proc} of 500/sec for a potentially improved software signaling solution.

Given the 125-byte length of signaling messages, the service rate of a 576kbps in-band DCC is roughly 500 msg/sec. For a 10Mbps out-of-band transmission channel, the service rate is 10,000 msg/sec.

For the one-way network delay, T_n, we assume that given the high message processing delays incurred with software signaling, service providers will likely provision logical high-bandwidth circuits between distant switches. For example, in telephone networks, it is not uncommon for service providers to provision high-bandwidth circuits from New York (NY) to Los Angeles (LA) through multiple digital crossconnects, so that individual telephone calls only require signaling protocol processing at the NY and LA switches. Similar designs will be required in GMPLS networks with switches using software-based signaling engines. The network delay for the signaling messages across such wide-area paths will be high. We chose 40ms for the one-way network delay T_n based on *ping* measurements across the U.S., which includes IP router queueing delays. On the other hand, with hardware signaling, no such provisioned circuits are required. Instead, signaling messages could be processed at each switch of an end-to-end path. Consider the Abilene network topology of Internet2 [9]. A call setup from Washington to Seattle could require signaling message processing at the intermediate switches of New York, Chicago, Indianapolis, Kansas City and Denver. In this case, the one-way network delay, T_n, is between consecutive switches, and hence smaller (we assume 10ms for the out-of-band path). The in-band network delay values shown in Table 2 can be similarly reasoned.

B. Results

Fig. 4 shows the numerical results for the software-signaling case. Parameters that are held unchanged for all the out-of-band plots are the transmitter rate, μ_{tx}, and packet loss rate, p. Similarly, all the in-band plots have the same transmitter rate and packet loss rate. Values for these parameters are shown in Table 2. For the signaling protocol processor, we use two values of the service rate, μ_{proc}, which in combination with metro- vs. wide-area, and in-band vs. out-of-band, leads to a total of 8 plots.

Under the metro-area assumption, the difference between OOB signaling and IB signaling is small when $\mu_{proc} = 20$. This is because T_n is small relative to the message processing delay, which is 50ms in the $\mu_{proc} = 20$ plots. At high load, the queueing delay could increase this message processing delay by a factor of ten, making this significantly larger than T_n. The difference is more with OOB having the edge if the signaling message processing service time is lowered from 50ms to 2ms. This is because, at this range of values, the effect of the lower transmitter rate of the IB channels becomes important. In the wide-area network setting, delay is smaller if IB signaling is used instead of OOB signaling. This difference is because of the network delay, which could be much higher in the OOB case as messages are likely to incur queueing delays at IP routers. In summary, given today's off-the-shelf message processing delays are on the order of 50ms, IB signaling is the better option from a delay perspective, in addition to other arguments, such as reliability. When the processing delay is so high, the effect of the IB channel transmitter rate being lower is not important.

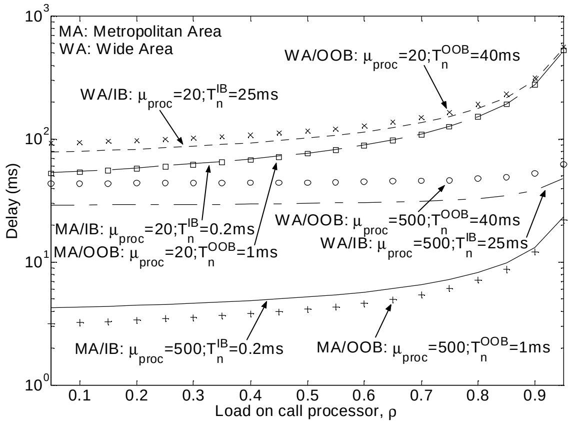


Fig. 4 Software signaling processor; plots show the effect of metro-area vs. wide-area, in-band (IB) vs. out-of-band (OOB) transport, with two values of message processing service rate, μ_{proc}.

Fig. 5 shows the plots for the hardware-signaling case. For the metro area, OOB signaling slightly outperforms IB signaling when the signaling channel is moderately loaded. In the high-load region, OOB signaling is significantly better than IB signaling in terms of call setup delay. This suggests that if the load on the signaling channel is a concern in metro-area networks, service providers should either use OOB signaling, or set up multiple DCC channels between switches to avoid overloading the signaling channel if they prefer IB signaling for other reasons, such as reliability.

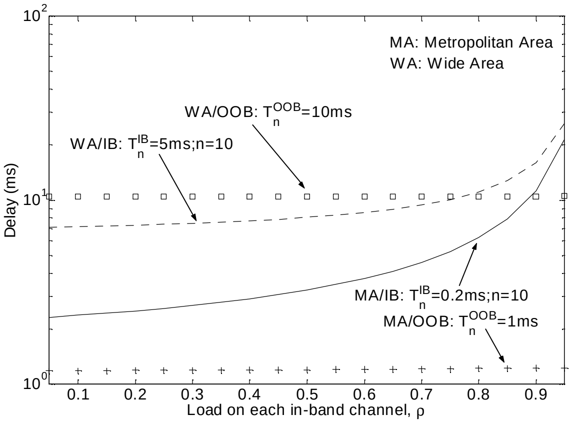


Fig. 5 Hardware signaling: plots show the effect of metro-area vs. wide-area, and in-band vs. out-of-band transport.

In the wide-area setting, we see that the plots for OOB signaling and IB signaling intersect. In the low-load region, which is to the left of this intersection point, IB signaling outperforms OOB signaling. In the high load region, the trend is reversed. We recognize that the position of this intersection point depends upon the load ρ, and the difference between T_n^{OOB} and T_n^{IB}. As described in Section III, since each OC1 has a DCC, the signaling load can be spread to keep the load per channel low. In this case, saving on the network delay by using in-band signaling will help lower end-to-end call setup delay to close to the round-trip propagation delay. Finally, for applications such as fast restoration, which requires extremely low call setup delay, high-speed (e.g., OC1 rate) in-band channels are likely to be needed, since the sum of emission delays for the low-speed DCC can be significant on a path consisting of multiple links.

VI. CONCLUSIONS

We compared in-band and out-of-band signaling transport options under assumptions of the GMPLS switches having hardware-accelerated signaling engines or software signaling protocol processors. With hardware signaling engines, if the load per in-band signaling channel can be kept low by using many of the DCC channels (one per OC1), in-band signaling is the preferred option. Since the network delay for carrying the messages through IP routers can be significantly more in the out-of-band option, in-band signaling is better in spite of the per-channel transmitter rate being lower. With software-based signaling protocol processors, queueing delays are likely to build up for the processor, making this component the most significant part of end-to-end call setup delay. In this case, the lower network delay for in-band signaling makes it a better option than the out-of-band signaling channel. The difference in transmitter rates does not matter since the message transmission service time is much smaller than the message processing service time, which means there is no queueing delay at the transmission servers for both in-band and out-of-band options.

APPENDIX

In Section IV.C, we reduced the two-server model to a single-queue model based on certain assumptions. In this appendix, we describe our reasoning behind these assumptions for the hardware and software signaling scenarios. First, consider a switch with a *hardware-signaling* module. In [10], we showed that circuits are useful for file transfers mainly in the wide-area context. In this case, if we choose a 50ms round-trip propagation delay, it will dominate the signaling message emission delays and message processing delays (given our hardware-signaling assumption). Thus, end-to-end call setup delay will be close to 50ms. Citing utilization considerations in [10], we recommended that the call holding time of a circuit should be at least 10 times the call setup delay. Applying this factor implies that call holding times should be minimally 500ms.

Further, in [10], we showed that this mode of call-by-call bandwidth sharing becomes feasible (i.e., the service provider can run the link at a high utilization with reasonably low values of call blocking probability) when the number of channels into which link capacity is divided is on the order of 100. Consider a switch with two OC192 links and 180 GbE ports (based on our experience with the equipment used in the CHEETAH network [5]). If the OC192 links are the critical links from a utilization perspective, then we should allocate per-circuit bandwidth rates of 100Mbps (making the number of channels on the OC192 equal 100).

To achieve a high utilization, the offered call load, $\rho = \lambda_{call}/\mu_{call}$, should be close to 100, where λ_{call} and μ_{call} are the call arrival and departure rates, respectively. Using $\mu_{call} = 1/(500ms)$, we require the call arrival rate for the single OC192 link to be 200 calls/sec. Extending this argument to the whole switch (with 180 GbE ports and 2 OC192 ports), we obtain an aggregate call arrival rate of 2000 calls/sec for the call processor. Given that 1 call requires three messages to be processed, the total message arrival rate is 6000 msg/sec. Since this number is much smaller than the 200,000 msg/sec processing rate for the hardware-signaling engine (see Table 2), it justifies the assumption $\mu_{proc} \gg \lambda$.

With out-of-band transmitter rates of 10Mbps or 100Mbps and 1000-bit signaling messages, the service rate is at least 10,000 msg/sec, as shown in Table 2, which is also higher than the 6000 msg/sec estimate from data-plane consideration. With in-band signaling, recall that the 6000 msg/sec are sent over multiple in-band channels. If the number of in-band channels is small, say 5 (for example, one in-band channel is allocated between two neighboring switches even if the switches are interconnected by multiple data-plane links), then the per-channel message arrival rate exceeds the 500 msg/sec service rate possible for 1000-bit messages on a 576kbps DCC. On the other hand, if the number of in-band channels is much higher, then there will be no queue build-up for even the low-bandwidth DCC in-band channel.

Assuming *software signaling*, we change the end-to-end call setup delay to be on the order of 1sec, which makes the minimum call holding time increase from 500ms (in the hardware signaling option) to 10sec. The corresponding call arrival rate for the signaling protocol processor is 100 calls/sec. If we assume a 20/sec signaling protocol processor service rate (see Table 2), it becomes clear that we cannot afford to support calls with such short call holding times. Thus calls have to be on the order of minutes, if the switch signaling engine is software based. But given our file transfer application, it becomes clear that there will be queueing delays at the software signaling protocol processor, as assumed in Section IV.C.

REFERENCES

[1] E. Mannie, "Generalized Multi-Protocol Label Switching (GMPLS) architecture," IETF RFC 3945, Oct. 2004.

[2] L. Berger, "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource reSerVation Protocol-Traffic Engineering (RSVP-TE) Extensions," IETF RFC 3473, Jan. 2003.

[3] Circuit-switched High-speed End-to-End ArcHitecture (CHEETAH). [Online]. Available: http://www.ece.virginia.edu/cheetah.

[4] H. Wang, M. Veeraraghavan, R. Karri, and T. Li, "Design of a high-performance RSVP-TE signaling hardware accelerator," *IEEE JSAC*, vol. 23, no. 8, pp. 1588-1595, Aug. 2005.

[5] X. Zhu, X. Zheng, and M. Veeraraghavan, "Experiences in implementing an experimental wide-area GMPLS network," accepted for publication in *IEEE JSAC*, vol. 24, no. 4, April 2007.

[6] M. Veeraraghavan and H. Wang, "A comparison of in-band and out-of-band transport options for signaling," *IEEE Workshop On the Quest to Control Next-Generation Transport Networks: The Role of Generalized Multi-Protocol Label Switching (GMPLS), held in conjunction with IEEE Globecom 2004*, Nov. 29-Dec. 3, Dallas, TX.

[7] V. Paxson and S. Floyd, "Wide-area traffic: The failure of Poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226-244, June 1995.

[8] L. Berger, D. Gan, et al., "RSVP Refresh Overhead Reduction Extensions," IETF RFC 2961, April 2001.

[9] Internet2. [Online]. Available: http://www.internet2.edu.

[10] X. Fang and M. Veeraraghavan, "On applications for dynamically shared circuit-switched/virtual-circuit networks," submitted to *IEEE JSAC*.