File transfers across optical circuit-switched networks

M. Veeraraghavan, H. Lee and X. Zheng
Polytechnic University, New York
mv@poly.edu, hlee@purros.poly.edu, zhxfifa@photon.poly.edu

Abstract:

Recent technological advances allow for the dynamic setup and release of end-to-end circuits consisting of Ethernet segments at the ends mapped on to Ethernet-over-SONET long-distance optical circuits. We call these Dynamically Reconfigurable Ethernet/Ethernet-over-SONET (DREEoS) circuits. For file transfers across end hosts that can be connected by a DREEoS circuit, we propose that, in most cases, the sending end host should first attempt to set up a DREEoS circuit, and if rejected, fall back to the TCP/IP path. If the DREEoS circuit setup is successful, the end host will enjoy a much shorter file transfer delay than with the TCP/IP path. For example, a 1GB file transfer on a TCP/IP path with a round-trip time of 50ms, link rate of 1Gbps, and a loss probability of 0.0001 takes 395.7sec, while on a DREEoS circuit with the same link rate, the transfer time is 8.08sec. The availability of the fallback TCP/IP path allows DREEoS service to be introduced gradually into optical networks. At low loads, the network can be operated at high call blocking probabilities to achieve high utilization. As loads increase, the network can be engineered to retain high utilization while simultaneously offering low call blocking probabilities. An important component of this proposal is hardware acceleration of signaling protocol implementations. This results in low call setup delays allowing for the DREEoS option to be attempted even for small file sizes. We compare mean delay incurred with a DREEoS circuit attempt against the mean delay incurred with directly choosing the TCP/IP path for different values of call blocking probability in the circuit-switched network, probability of packet loss in the IP network, round-trip times, link rates, etc.

1. Background and problem statement

There is a growing interest in improving current protocols or developing new ones to increase the effective throughput of file transfers on the Internet [1-5]. Of particular interest is the effective throughput of large file transfers, e.g., petabyte (10^{15}) and exabyte (10^{18}) sized file transfers created in Particle Physics, Earth Observation, Bioinformatics, and Radio Astronomy studies [5]. In the general Internet setting, this problem should be solved for file transfers between end hosts connected by any heterogeneous end-to-end path. Solutions for such settings are being developed by others [3][4]. We consider a slight variant of this problem. Consider the case when the two communicating end hosts are located on the same network. Then the question is should these two hosts continue using TCP/IP for their communication, or should they use protocols tailor-made for intra-network communication? The original design tenet of TCP/IP protocols was to use the same protocols irrespective of the end-to-end path to simplify protocol implementations [6]. However, this sacrifices performance. For example, consider the case of two communicating end hosts connected by a direct high-speed link. Given TCP does not examine the nature of the end-to-end path before data transfer, it will run the usual congestion control algorithms such as Slow Start. Even without packet losses, the Slow Start congestion algorithm will lead to poor performance especially in high-speed long-distance environments. For example, if round-trip propagation delay on a 1Gbps link is 50ms, then a 100MB file transfer experiences an effective throughput of 0.54Gbps because of Slow Start. As link speeds increase to 10Gbps, effective throughput will be small even in lower propagation-delay environments.

The specific intra-network setting we consider is a wide-area optical circuit-switched network. The problem statement of this work is to develop protocols for fast file transfers across an optical circuit-switched network. This is not a pure academic exercise. Instead, a few recent technological advances have made implementation of this concept quite feasible. These advances include deployment of (i) optical fiber to enterprises, (ii) Multi-Service Provisioning Platforms (MSPPs) in enterprises, and (iii) Ethernet over SONET (EoS) capabilities in MSPPs. Before we describe these advances, we clarify our problem definition with an illustration.

Fig. 1 shows the Internet as a global network of IP routers that interconnects different types of networks grouped into sets. Set 1 is a set of high-speed optical circuit-switched networks consisting of SONET/SDH/WDM[1] crossconnects, Add/Drop Multiplexers (ADMs), etc. Some of these networks could be all-optical, with optical links and all-

1. SONET/SDH/WDM: Synchronous Optical NETwork/Synchronous Digital Hierarchy/Wavelength Division Multiplexing

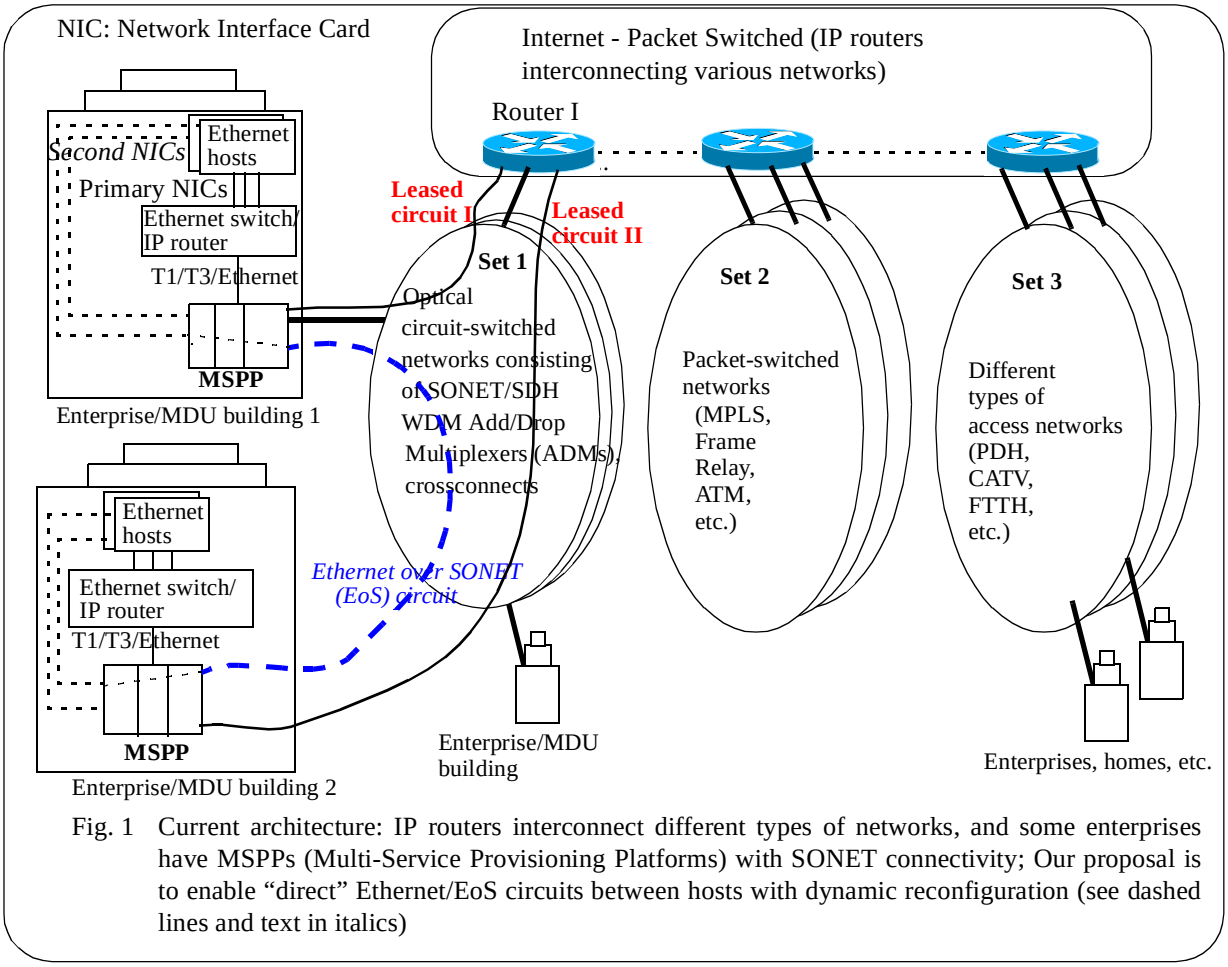


Fig. 1 Current architecture: IP routers interconnect different types of networks, and some enterprises have MSPPs (Multi-Service Provisioning Platforms) with SONET connectivity; Our proposal is to enable "direct" Ethernet/EoS circuits between hosts with dynamic reconfiguration (see dashed lines and text in italics)

optical switches. Set 2 consists of packet-switched networks, such as MultiProtocol Label Switched (MPLS) networks, frame-relay networks, Asynchronous Transfer Mode (ATM) networks, etc. Set 3 includes access networks, such as Plesiochronous Digital Hierarchy (PDH) (DS0, T1 and T3) networks, Cable TV (CATV) networks, Fiber-To-The-Home (FTTH) networks, etc. Most enterprises lease T1/T3 links for access to their ISP routers, while home users typically use dial-up or cable modems for Internet access. IP routers play the important role of interconnecting this heterogeneous set of networks, and TCP serves as the transport protocol for reliable file transfers between end hosts on different networks. Improving file transfer performance in this general context is hence a challenging problem.

As stated earlier, we focus on a more specific problem, i.e., one of improving file transfer performance for end hosts located on the same optical network. Consider a file transfer from an end host located in enterprise/MDU[2] building 1 to an end host located in building 2 of Fig. 1. Buildings 1 and 2 are representative of the many metropolitan area buildings that now have optical fiber reach. Typically, the fiber terminates on a low-cost SONET system located within the enterprise building called a Multi-Service Provisioning Platform (MSPP). An MSPP integrates multiple T1s, T3s, OC3s, etc. leased from the building into one SONET signal (e.g., leased circuits I and II in Fig. 1). T1s from PBXs carrying voice traffic and T1s/T3s from WAN-access IP routers carrying data traffic are multiplexed on to the same SONET link. In addition, various Ethernet-over-SONET (EoS) technologies have been developed for carrying Ethernet frames in a SONET frame [7]. Examples of MSPPs include Fujitsu's FLM 150 [8], Cisco's 15454 [9] and Ciena's K2 [10]. The architecture of an MSPP is shown in Fig. 2. Nodes within an enterprise are connected to interface cards, such as Ethernet (10Mbps/100Mbps), T1, T3, Gb/s Ethernet, etc. The WAN access link card has a higher rate SONET interface, such as an OC12 (622Mbps), OC48 (2.5Gbps), OC192 (10Gbps) or higher.

2. MDU: Multiple Dwelling Unit

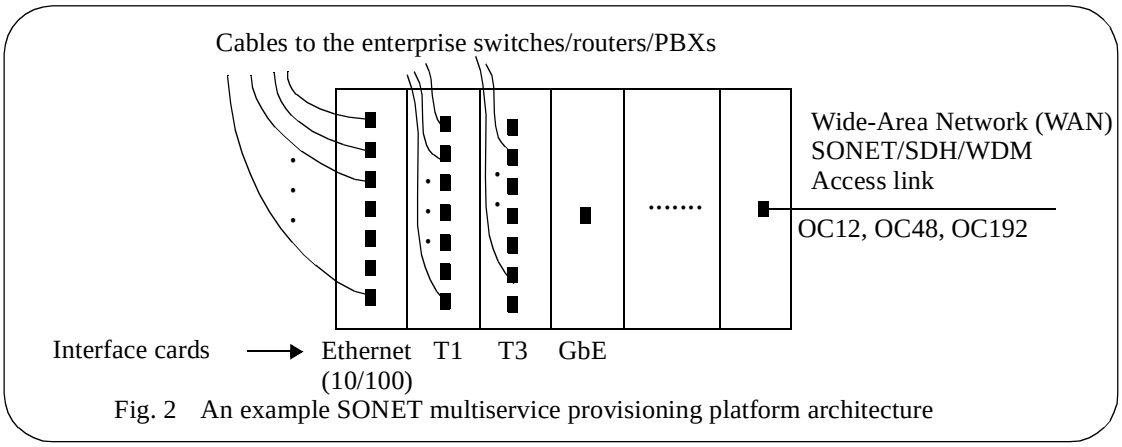


Fig. 2 An example SONET multiservice provisioning platform architecture

With the three technological advances, deployment of optical fiber, deployment of MSPPs, and EoS, we can potentially connect two end hosts with "direct" high-speed Ethernet/EoS circuits, as illustrated with dashed lines in Fig. 1. Therefore, the question arises as to whether performance can be improved by using tailor-made protocols that take advantage of the characteristics of this single-network end-to-end path. In this work, we explore various aspects of how to support file transfer applications on such end-to-end Ethernet/EoS circuits.

It is a general belief that circuit-switched networks are not well-suited for file transfers because once a circuit is set up it does not take advantage of bandwidth that may become available subsequently [4]. In other words, since files can be transferred at any rate, a networking solution that offers flexibility, allowing a flow to adjust its bandwidth dynamically, is more powerful than one that nails down the bandwidth for the duration of the transfer. However the maximum rate at which a file can be transferred is limited by the end host rates (sending and receiving ends). End host interfaces are typically of lower rates than inter-switch links. Furthermore, due to CPU limitations, the actual sending and receiving rates may be even lower than the end host interface rates. In our approach, the optical circuit-switched network will either set up a circuit at the maximum rate at which the sender and receiver can exchange data or it will block the call. Thus, the circuit-switched network will offer the same maximum rate as a packet-switched network.

The SONET/SDH/WDM circuit-switched networks shown in Fig. 1 are not limited to metropolitan areas. Wide-area optical networks interconnect SONET/SDH MANs. This means end hosts in different metropolitan areas can be connected by an Ethernet/EoS circuit. For example, the Canarie network [2] extends East-to-West across Canada. The network allows for wide-area SONET/SDH circuits or WDM lightpaths to be established end-to-end.

We provide the reader relevant background information on ongoing standards work that enable our solution in Section 2. We present our solution for fast file transfers across optical circuit-switched networks in Section 3. Section 4 presents our conclusions.

2. Enabling protocol standards

Our solution for fast file transfers across optical circuit-switched networks builds on two sets of standards. We briefly describe these enabling protocol standards below.

Two emerging standards on EoS are Generic Framing Procedure (GFP) [7], and Virtual Concatenation [11]. GFP is a method for transporting Ethernet frames within SONET signals, and Virtual Concatenation allows for arbitrary-bandwidth SONET signals to be created to improve resource utilization. Without virtual concatenation, a 10Mbps Ethernet signal needs to be carried within 1 OC1 (a waste of 40Mbps); with virtual concatenation, it can be carried within 7 VT1.5 signals (a waste of 500Kbps). For papers on these topics, see [12]. Data sequence is maintained across a virtually concatenated signal even if its components are routed on different physical paths.

Another emerging standard that enables our solution is Generalized MultiProtocol Label Switching (GMPLS) [13-15]. The GMPLS working group of the IETF is defining standards for signaling protocols for SONET/SDH/WDM optical networks. Signaling protocols provide a distributed solution for dynamic circuit/lightpath setup and release. GMPLS signaling protocols include Resource reSerVation Protocol with Traffic Engineering (RSVP-TE) and Constraint Routing based Label Distribution Protocol (CR-LDP). The primary applications envisioned by the IETF community for GMPLS signaling are fast restoration following failures, and rapid provisioning of circuits between IP

routers. As will be described in more detail in Section 3, we propose using these GMPLS signaling protocols for the dynamic setup/release of end-to-end Ethernet/EoS circuits for fast file transfers and other applications. The Optical Internetworking Forum (OIF) is issuing specifications for the control plane of optical networks, based on RSVP-TE and other IETF signaling protocols [16].

3. Proposed solution: Dynamically Reconfigurable Ethernet/Ethernet-over-SONET

Our solution calls for equipping end hosts with second (high-speed) Ethernet NICs, and connecting these NICs directly to MSPPs, as illustrated in Fig. 1. MSPPs are then interconnected across wide-area networks using EoS circuits. The circuits are established and released dynamically using GMPLS signaling protocols. Since resource sharing in circuit-switched networks is on a call-by-call basis, we refer to our solution as "Dynamically Reconfigurable Ethernet/Ethernet Over SONET (DREEoS)." This solution exploits the dominance of Ethernet in LANs and SONET in MANs/WANs. *Section 3.1* describes the equipment needed to support the DREEoS solution.

Since DREEoS can only be used for communication between end hosts located on an optical circuit-switched network, a host requires some support to first determine whether its correspondent end host (the end host with which it is communicating) is reachable via a DREEoS circuit. In *Section 3.2*, we describe a support service for this purpose called "Optical Connectivity Service (OCS)."

Next, we define the protocols needed to support file transfers on DREEoS. Three types of protocols are required: signaling protocols, routing protocols and transport protocols. For signaling protocols, we propose using GMPLS signaling protocols. Therefore, no new protocol specification is needed. Instead, we focus our attention on the implementation aspect. We recommend a hardware accelerated implementation of signaling protocols at MSPPs, ADMs, crossconnects and other optical circuit switches. Our reason for this recommendation is as follows. To achieve high utilization of the optical circuit-switched network, circuits should be unidirectional, and should be held open only for the duration of single file transfers. File transfer applications typically have "think times" between file requests, and if a circuit is held open during these "think" times, network resources will be wasted. If calls are held only for the duration of single file transfers, call holding times will be quite small especially as link rates increase. For example, a 1MB transfer on a 100Mbps link incurs a transmission delay of only 80ms. This means call setup delays should be kept low and call handling capacities of switches should be high. The latter is important for scalability reasons. Hardware acceleration of signaling protocol implementations is an effective means to achieve these goals. *Section 3.3* describes our current work on hardware accelerated signaling implementations.

Routing protocols used within the optical circuit-switched networks are not discussed in this paper. We have not yet studied the question of whether the enhancements being proposed to OSPF for SONET/SDH/WDM networks are sufficient for the file transfer application. However, we recognize a new routing problem, one that occurs at end hosts. DREEoS service is proposed an add-on to the primary Internet access service that end hosts located on an optical circuit-switched network already enjoy. Thus an end host with access to DREEoS has to choose between using the TCP/IP path (through its primary NIC) or a DREEoS circuit (through its second NIC). This is a routing decision. An example is shown in Fig. 1. An end host in building 1 can use TCP/IP on a path through its primary NIC, Ethernet switches/routers and MSPP within building 1, Leased circuit I to router I, Leased circuit II to the MSPP in building 2, MSPP and Ethernet switches/routers within building 2, and the primary NIC in the end host in building 2. The second path is a DREEoS circuit through the second NIC, MSPP in building 1, optical network circuit switches, MSPP in building 2 and the second NIC at the end host in building 2.

We propose that the optical circuit-switched network be operated in a call blocking mode. The routing decision at the end host is whether or not to attempt setting up a DREEoS circuit. If its request is blocked, it will use the TCP/IP path. Otherwise, it will enjoy the low file transfer delay possible on a DREEoS circuit. *Section 3.4* discusses the parameters that could be considered in an end host routing decision algorithm.

Finally, we consider transport protocols to use on DREEoS circuits. The two main functions required are error control and flow control. Congestion control is not required during data transfer because of the allocation of dedicated resources to a DREEoS circuit. We selected an ANSI standard, called Scheduled Transfer (ST) protocol, for use on DREEoS. This transport protocol is ideally suited for end-to-end circuits carrying Ethernet frames. We use a combination of ST on DREEoS and the dual TCP/IP path for data transfer. The latter is used for retransmissions and control signals such as negative acknowledgments. *Section 3.5* describes our user data transport approach.

3.1 Equipment

1. Hosts that want access to DREEoS service should be equipped with second Ethernet NICs, which are connected "directly" to the MSPP Ethernet cards as shown in Fig. 1.
2. Some of the MSPPs and SONET/SDH/WDM switches (crossconnects, ADMs) should be enhanced with signaling protocol engines to handle dynamic call setup and release. Not all MSPPs/switches need signaling engines. Circuits can be provisioned between nodes that do not have signaling capability. Adding signaling engines to MSPPs allows for concentration on access links from enterprises.
3. Upgrade application software in end hosts to interface with the DREEoS service.

3.2 Optical Connectivity Service (OCS)

A support service called "Optical Connectivity Service (OCS)" is proposed to provide end hosts a mechanism to determine whether or not their correspondent end hosts have access to DREEoS service. OCS can be implemented much like DNS with enterprises and service provider networks maintaining servers with information on end hosts that have access to DREEoS service. These servers would answer queries from end hosts in much the same manner as DNS servers answer queries for IP addresses. With caching, the delay incurred in this step can be reduced.

3.3 Hardware acceleration of signaling protocol implementations

Processing signaling protocol messages involves many data table reads/writes, parsing/constructing complex messages, maintaining state information, managing timers, etc. For example, consider call setup. Upon receiving a call setup message, a call processor needs to parse out important parameters, such as destination address, bandwidth requested, etc. and then perform several actions. First, it determines the next-hop switch through which to reach the destination typically by consulting a precomputed routing table (similar to the longest-prefix match operation in IP routers). Second, it selects an interface connected to the selected next-hop switch on which sufficient bandwidth is available. Third, it selects free time-slots and/or wavelengths on the selected interface. Finally, it programs the switch fabric by writing a switch configuration table. This table is used by the switch to route data bits received at/on a given timeslot/wavelength on an incoming interface to a given timeslot/wavelength on a corresponding outgoing interface. Other actions performed by the signaling protocol processing engines at switches include updating state information and constructing the outgoing signaling message. Similar actions are performed for circuit release.

Accelerating signaling protocol processing engines is thus a challenging task. Our work-to-date on this task has been to implement our own signaling protocol, called Optical Circuit-switched Signaling Protocol (OCSP) [17] in hardware. We designed the signaling protocol specifically for SONET networks with a goal of achieving high performance rather than flexibility. We implemented the basic and frequently used operations of OCSP in reconfigurable hardware, i.e., Field Programmable Gate Arrays (FPGAs) [18], and relegated the complex and infrequently used operations (for example, processing of optional parameters, error handling, etc.) to software. FPGAs were chosen because they offer a compromise on the performance/flexibility spectrum, offering better performance than general-purpose processors but more flexibility than ASICs for protocol upgrades. We modeled the signaling protocol in VHDL and then mapped it onto two FPGAs on the WILDFORCE™ reconfigurable board with a Xilinx® XC4036XLA FPGA with 62% resource utilization and a XC4013XLA with 8% resource utilization. The hardware implementation handles four messages, Setup, Setup-success, Release and Release-success. From the timing simulations, done using the ModelSim® simulator, while receiving and transmitting a Setup message consumes 12 clock cycles each, processing of the Setup message consumes 53 clock cycles. Processing Setup-Success, Release and Release-Confirm messages consumes about 70 clock cycles total since these messages are much shorter (2 32-bit words versus 11 32-bit words for Setup) and require simpler processing. Assuming a 25 MHz clock, this translates into 4.0ms for Setup message processing and about 2.8ms for the combined processing of Setup-Success, Release and Release-Confirm message. Thus, a complete setup and teardown of a connection consumes about 6.6 microseconds. Compare this with the millisecond-based software implementations of signaling protocols [19].

Given that GMPLS signaling protocols are more complex than OCSP, we are now undertaking a hardware-accelerated implementation of RSVP-TE in an NSF-sponsored project [20]. The complexity in GMPLS signaling protocols is the result of generalization to enable application of the same protocol to multiple types of networks. For example, RSVP was first developed for IP networks, then evolved to RSVP-TE for MPLS networks, and finally evolved to RSVP-TE for GMPLS networks. This has resulted in complex messages and parameters, with each parameter field having many options. We are handling this complexity by carefully defining a subset that is small enough for hardware implementation and yet large enough to handle most signaling requests at an optical circuit switch.

3.4 Routing decision

End hosts that have access to DREEoS circuits have a choice of paths when they communicate with each other. An end host can choose whether to attempt setting up a DREEoS circuit or not. If resources are not available, the optical circuit-switched network may reject a call setup request, in which case the end host has to fall back to using the TCP/IP path. In this section, we study the impact of various parameters on file transfer delay via the TCP/IP path and via a DREEoS circuit.

File transfer delay on the TCP/IP path is obtained using the models of [21-22], which capture the time spent in Slow Start $E[T_{ss}]$, the expected cost of a recovery following the first loss $E[T_{loss}]$, the time spent in Congestion Avoidance $E[T_{ca}]$, and the time to delay the ACK for the initial segment [Eqn. 25 of 21].

$$E[T_{tcp}] = E[T_{ss}] + E[T_{loss}] + E[T_{ca}] + E[T_{delayack}] \quad (1)$$

$E[T_{ss}]$ is a function of Round-Trip Time (RTT), W_{max}, which is a limitation posed by the sender or receiver window, w_1, the initial congestion window, the loss rate P_{loss}, the number of data segments in the file transfer, and the number of segments for which an ACK is generated (for example, if ACK-every-other-segment strategy is used, this number is 2). The $E[T_{loss}]$ term is a function of T_0 and RTT and the probability of the first loss being detected with a retransmission time-out or with a triple duplicate ACK. The reader is referred to [21] for details of these two functions. $E[T_{ca}]$ is a function of the number of data segments in the file transfer, P_{loss}, RTT, T_o, which is the average duration of a first time-out in a sequence of one or more time-outs, and W_{max}, as derived in [22].

We set the final term $E[T_{delayack}]$ to 0 because we assume a starting initial window size of 2 [23]. Also, we do not include TCP connection setup time assuming that the connection is already open. However, because of the usage of Restart Window (RW) [23], a transfer is modeled as starting in the Slow Start phase.

The input parameter values assumed for the numerical computation are shown in Table 1. We assume four values for P_{loss}, two values for the bottleneck link rate r, and three values of the round-trip propagation delay T_{prop} to create a total of 24 cases. RTT is computed from T_{prop} and a rough estimate of queueing plus service delay for the bottleneck link for a packet to get through the IP network. We derive this estimate by determining the load at which an M/M/1/k system will experience the assumed P_{loss} values. W_{max}, as stated earlier, is determined by limitations on the sender or receiver window. For all the cases, we set W_{max} to the delay-bandwidth product, i.e., $W_{max} = RTT \times r$. When the congestion window reaches W_{max}, any further increase is irrelevant because the system will reach a streaming state in which ACKs are received in time to permit further packet transmissions before the sender completes emitting its current congestion window.

Using the input parameters shown in Table 1, we compute $E[T_{tcp}]$ given by (1), and plot the results in Fig. 3. We only show the plots for the two extreme values of 0.0001 and 0.1 for clarity reasons. We also single out one data point (1GB transfer) for the discussion below and list the values in the last column of Table 1. The round-trip propagation delay T_{prop} has a significant impact on total file transfer delay. For example, for a 1GB file transfer, increasing T_{prop} from 5ms to 50ms results in a considerable increase in $E[T_{tcp}]$ from 89.45s to 396.5s. Also, at large values of the round-trip propagation delay T_{prop} (50ms), for a given P_{loss}, there is not much benefit gained from increasing the bottleneck link rate from 100Mbps to 1Gbps. Compare 396.5s for a 100Mbps link with the 395.7s number using a 1Gbps link for 1 GB file transfer. Increasing the bottleneck link rate has value when propagation delay is small. The higher the rate, the smaller the propagation delay at which this benefit can be seen. Loss probability P_{loss} also plays an important role. Even in a low propagation delay environment (T_{prop} of 0.1ms), $E[T_{tcp}]$ jumps from 82.25s to 229s for the 1GB file transfer for P_{loss} increase from 0.0001 to 0.1.

Next consider two delay components incurred with DREEoS, $E[T_{setup}]$ and $T_{transfer}$. The mean call setup delay $E[T_{setup}]$ includes mean signaling message transmission delays, mean call processing delays (to process signaling protocol messages), and a round-trip propagation delay:

$$E[T_{setup}] = \frac{m_{sig}}{r_s} \times \left(1 + \frac{\rho_{sig}}{2(1-\rho_{sig})}\right) \times (k+1) + T_{sp} \times \left(1 + \frac{\rho_{sp}}{2(1-\rho_{sp})}\right) \times k + T_{prop} \quad (2)$$

m_{sig} is the cumulative size of signaling messages used in call setup, r_s is the signaling link rate, k is the number of switches on the end-to-end path, T_{sp} is the signaling message processing time incurred at each switch and T_{prop} is the round-trip propagation delay. We approximate the queueing delay for the signaling link with an M/D/1 queue at a load ρ_{sig}, and the queueing delay for the call processor also with an M/D/1 queue[3] at a load ρ_{sp}.

Table 1: Input parameters plus the time to transfer a 1 GB file

Case	Loss p_{loss}	Rate r	Round-trip propagation delay T_{prop}	RTT	W_{max} (in packets)	$E[T_{tcp}]$ for a 1GB file (s)
Case 1	0.0001	100Mb/s	0.1ms	0.3ms	2.5	82.25
Case 2			5ms	5.2ms	41	89.45
Case 3			50ms	50.2ms	418	396.5
Case 4	0.0001	1Gbps	0.1ms	0.12ms	10	8.25
Case 5			5ms	5.02ms	418	39.6
Case 6			50ms	50.02ms	4168	395.7
Case 7	0.001	100Mbps	0.1ms	0.36ms	3	82.93
Case 8			5ms	5.26ms	43.8	135.4
Case 9			50ms	50.26ms	418.8	1293
Case 10	0.001	1Gbps	0.1ms	0.13ms	10.8	8.64
Case 11			5ms	5.03ms	419	129.4
Case 12			50ms	50.03ms	4169	1287
Case 13	0.01	100Mbps	0.1ms	0.48ms	4	92.41
Case 14			5ms	5.38ms	44.8	471.7
Case 15			50ms	50.38ms	419.8	4417
Case 16	0.01	1Gbps	0.1ms	0.138ms	11.5	12.43
Case 17			5ms	5.038ms	419.8	441.7
Case 18			50ms	50.038ms	4169.8	4387
Case 19	0.1	100Mbps	0.1ms	0.63ms	5.25	229
Case 20			5ms	5.53ms	46	2010
Case 21			50ms	50.53ms	421	18370
Case 22	0.1	1Gbps	0.1ms	0.15ms	12.5	54.53
Case 23			5ms	5.05ms	420.8	1836
Case 24			50ms	50.05ms	4170.8	18195

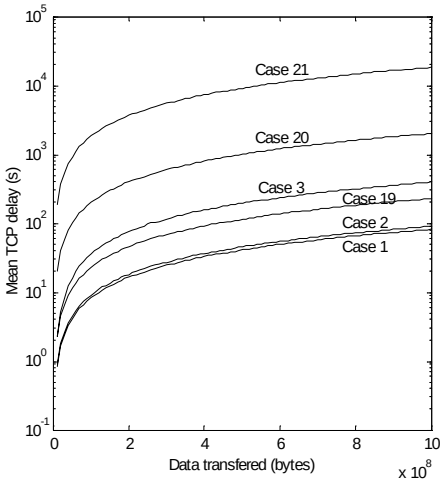
The second component $T_{transfer}$ is the actual file transfer delay:

$$T_{transfer} = \frac{f}{r_c} + \frac{T_{prop}}{2} \tag{3}$$

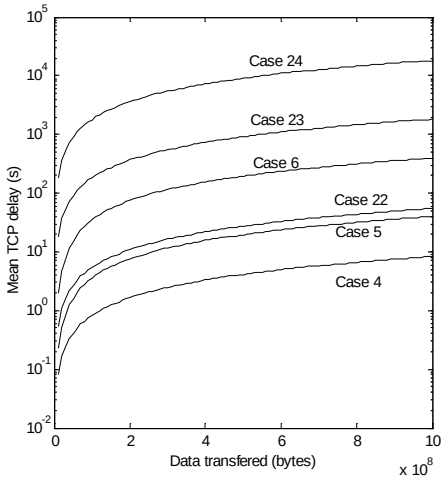
where f is the size of the file being transferred and r_c is the data rate of the circuit. Given the low error rates of optical fiber and the absence of packet switches on DREEoS circuits, the probability of requiring retransmissions is low and hence we ignore the delay component incurred from retransmissions.

Given that the optical circuit-switched network is operated in a call blocking mode, an end host has to decide whether to even attempt requesting a DREEoS circuit setup. To make this decision, compare $E[T_{tcp}]$ with the mean

3. M/D/1 queueing models are quite accurate since inter-arrival times between file transfers have been shown to be exponentially distributed [24], and signaling message lengths and call processing delays are more-or-less constant.



(a) Bottleneck link rate r is 100Mbps



(b) Bottleneck link rate r is 1Gbps

Fig. 3 Mean TCP delay $E[T_{tcp}]$

delay expected if a DREEoS circuit is attempted as follows.

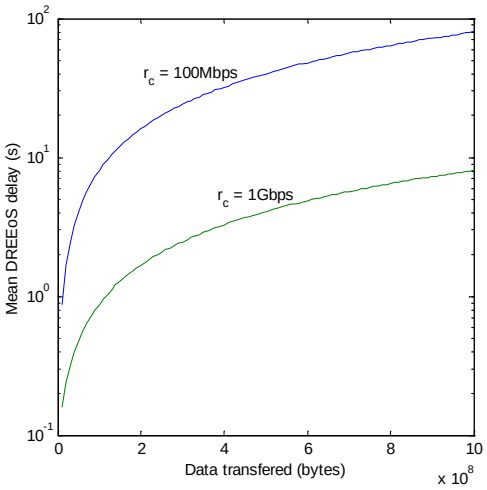
$$E[T_{dreeos}] = (1-P_b)(E[T_{setup}] + T_{transfer}) + P_b(E[T_{setup}] + E[T_{tcp}]) \quad (4)$$

where P_b is the call blocking probability. If the call is not blocked, total delay experienced is $(E[T_{setup}] + T_{transfer})$, but if it is blocked, then after incurring a cost $E[T_{setup}]$ (we assume that a failed call setup attempt incurs the same setup delay as a successful attempt), the end host has to use the TCP/IP path and hence will incur the $E[T_{tcp}]$ delay. Compare $E[T_{tcp}]$ with $E[T_{dreeos}]$. Thus,

$$\begin{aligned} &\text{if } \left(\frac{E[T_{setup}]}{1-P_b} > (E[T_{tcp}] - T_{transfer})\right) && \text{use TCP/IP path} \\ &\text{if } \left(\frac{E[T_{setup}]}{1-P_b} < (E[T_{tcp}] - T_{transfer})\right) && \text{use DREEoS circuit} \\ &\text{if } \left(\frac{E[T_{setup}]}{1-P_b} = (E[T_{tcp}] - T_{transfer})\right) && \text{use either} \end{aligned} \quad (5)$$

Fig. 4 shows numerical results for DREEoS. We varied propagation delay, using the same numbers as in the TCP analysis, 0.1ms, 5ms and 50ms. The impact of this variation on total transfer time is small for large files (e.g., for the range 100MB to 1GB shown in Fig. 4). We also varied the number of switches k on the end-to-end path using 4 and 20, but found little impact on the transfer times of large files. We assumed a 10Mbps signaling link rate and the circuit rates shown in Fig. 4. We assumed 0.8 for both ρ_{sp} and ρ_{sig}. Consider a file size of 1GB. The sum $E[T_{setup}] + T_{transfer}$ is 80.08sec when the link rate is 100Mbps and 8.08sec when the link rate is 1Gbps (assuming a round-trip propagation delay of 50ms and 20 switches on the end-to-end path). The major component of these values is $T_{transfer}$. $E[T_{setup}]$ is only 55.3ms. The total message length for call setup related signaling messages is assumed to be 100 bytes and the call processing delay per switch is assumed to be $4\mu s$ given our hardware-accelerated signaling implementations (see Section 3.3). Compare these numbers with the delays incurred using TCP listed in the last column of Table 1. If loss probability is low at 0.0001 on the IP path and round-trip propagation delay is 50ms, the delay incurred is 395.7sec even if the link rate is 1Gbps. The numerical value for P_b computed from (5) when $E[T_{tcp}]$ is 395.7 sec, $T_{transfer}$ is 8.025sec and $E[T_{setup}]$ is 0.0553sec is 99.98%. In other words, unless call blocking probability is higher than 99.98%, attempt the DREEoS option first when the file size is 1GB.

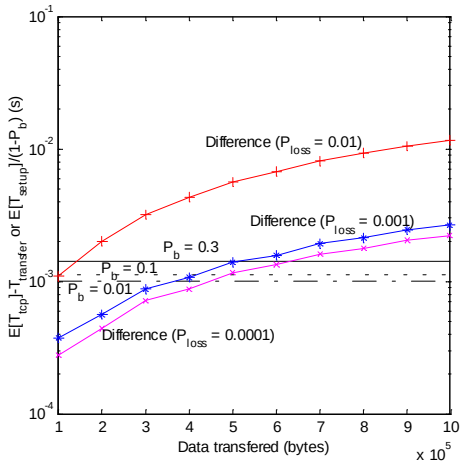
We next consider smaller file sizes. Fig. 5 compares TCP/IP path delays with DREEoS circuit delays for smaller files (100KB to 1MB). For the three horizontal lines on which P_b values are listed, the y-axis is the left-hand side of



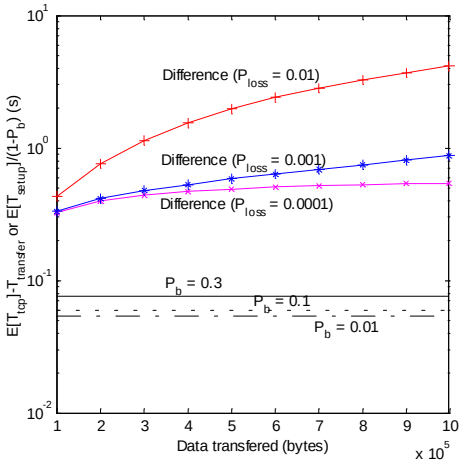
Not much dependence on propagation delay and number of switches when delay is dominated by transmission delay (for large files)

Fig. 4 Total delay for the DREEoS solution if the call gets through, i.e., it is not blocked

(5), i.e., $(E[T_{setup}])/(1-P_b)$. For the remaining three lines, which are marked "Difference" with P_{loss} values, the y-axis is the right-hand side of (5), i.e., $E[T_{tcp}]-T_{transfer}$. First consider Fig. 5(a) in which $T_{prop} = 0.1ms$. These plots show that if P_{loss} is as high as 0.01, then the cross over file size exists only for high call blocking rates, e.g., $P_b = 0.3$. When P_{loss} is 0.001 or 0.0001, then there is a crossover file size even for lower call blocking rates. For file sizes below this crossover file size, the TCP/IP option should be used immediately, but for files larger than this crossover file size, a DREEoS circuit should be attempted first. These numbers are dependent upon ρ_{sig} and



(a) T_{prop} is 0.1ms



(b) T_{prop} is 50ms

Fig. 5 Plot of equation (5) for smaller files with a link rate of 100Mbps, ρ_{sig} =ρ_{sp}=0.7

ρ_{sp}, which impact $E[T_{setup}]$. For example, if we increase these loads from 0.7 to 0.8, then the difference curve for P_{loss} of 0.01 crosses even the lower call blocking probability lines, i.e., $P_b = 0.1$ and $P_b = 0.001$. In addition, crossover file sizes for the two other difference curves are larger. Fig. 5(b) shows the delay differences for a WAN environment, e.g., T_{prop} is 50ms. For this environment, for the entire file range, a DREEoS circuit should be attempted if P_b and P_{loss} have the values shown.

It is reasonable to run the optical circuit-switched network at high call blocking probabilities especially when DREEoS is first introduced. The Erlang-B formula for call blocking probability works such that as the number of circuits is increased (corresponding to increasing loads), utilization increases. For example, consider two cases, a link

with 64 circuits with an offered load of 50 Erlangs, and a link with 117 circuits with an offered load of 100 Erlangs. In both cases, the call blocking probability is 1%. However, utilization is 77% in the first case and 84.6% in the second case. When call blocking probability is increased to 30%, then an offered load of 50 Erlangs can be handled with 38 circuits. The utilization at this operating point is 92%. Thus, when DREEoS service is first introduced the initial number of end hosts equipped with second NICs and enterprises equipped with MSPPs will be small. The network can be operated at a high utilization and high call blocking probability with many file transfers resorting to the TCP/IP path upon rejection from the optical network. But with growth in the number of DREEoS service participants (and a corresponding growth in the number of circuits), lower call blocking probabilities can be achieved without compromising utilization.

To design a routing decision algorithm for the end host, we use the results of the above analysis. For the range of "large" file sizes that we considered as an example in this paper, i.e., 100MB to 1GB, the benefit of attempting a DREEoS circuit is clear. For the range we considered as "smaller" file sizes, i.e., 100KB to 1MB, when RTT is "large," e.g., 50ms, then again there is an advantage to attempting a DREEoS circuit. But if RTT is "small," e.g. 0.1ms, there is a crossover file size below it is wasteful to attempt a DREEoS circuit. The routing decision algorithm implemented at an end host could use exact values of RTTs, P_b, P_{loss}, link rate, etc. to determine which path to use for each file transfer, in which case we would not need to define what constitutes a "large file" or a "large RTT," etc. However, such as a dynamic algorithm could be complex. While RTT measurements can be made during the TCP connection establishment handshake, other parameters are harder to estimate. Tomography experiments have shown that P_{loss} can be estimated by end hosts [25]. Other options are to have network management stations track these values and respond to queries from end hosts. Since the benefit of using DREEoS may not be significant for small file sizes, we need to carefully study the value of introducing this complexity. Alternatively, we could define static values for "large files," "large RTT," etc. based on nominal operating conditions of the two networks and simplify the routing decision algorithm implemented at end hosts. This needs further study.

3.5 Transport protocol used over the DREEoS circuit

After considering a number of high-speed transport protocols [26-27] and OS bypass protocols [28-30], we have selected the Scheduled Transfer (ST) protocol [31], an ANSI standard [32]. The reason for considering high-speed transport protocols is obvious, given our DREEoS circuit solution is aimed at providing high-speed transport (100Mbps, 1Gbps, 10Gbps end-to-end). The reason for considering OS bypass protocols is that a significant component of the end-to-end delay is incurred by transport protocol implementations in the end hosts. OS bypass protocols aim to reduce this component. The impact of lowering transport protocol implementation delays at end hosts was demonstrated for TCP in the Trapeze project [33] and a hardware accelerated implementation [34].

ST provides sufficient hooks to allow for an OS bypass implementation. It does this by having the sender send a control message, called a *Request-To-Send (RTS)*, specifying the *transfer* length to a receiver. The receiver, in turn, pins memory in the user space in the form of *buffers*, and replies with control messages, called *Clear-To-Send (CTS)*, one per *block*, where a block is the unit of error control and flow control. Each CTS carries the pinned memory address for a block. When the sender sends the data, the header of the data unit carries the corresponding memory address. This allows the NIC receiving the data unit to use Direct Memory Access (DMA) to write the payload in a "silent" manner into the receiver's memory. The result is that the transport protocol adds a low end host delay to the total file transfer delay.

ST offers flexibility in its flow control and error control schemes. For flow control, the RTS carries a parameter called *CTS_req*, which specifies the number of blocks that the sender would like to send back-to-back or concurrently. The receiver responds with multiple CTSs, one per block. For our application, we propose the receiver respond with as many CTSs as it has reserved space in main memory. For a large file, it is unlikely that the receiver can reserve enough space in the main memory to hold the whole file (e.g., petabyte and exabyte files), and hence such a transfer will have to be carried out as a sequence of transfers, each limited by the memory space in the receiver. Recycling buffers is not possible if there is a speed mismatch between the DREEoS circuit and the rate at which buffers are depleted by the receiving application. If the file transfer size is larger than available receiver memory space, the network runs the risk of the DREEoS circuit lying unused while waiting for the receiver to clear its buffers. Hence we propose a Maximum File Transfer Size (MFTS).

Another consideration for limiting transfer sizes is that in this circuit-switched network operation, contention for resources amongst various users is on a call-by-call basis. If one user is allowed to hold resources for a very long period of time (say with an exabyte transfer), other end hosts will be forced to use the TCP/IP path for lack of

DREEoS resources (which means they will suffer increased delays). Hence fairness is a second consideration for limiting transfer sizes to an MFTS. This is similar to Maximum Transmission Unit (MTU) used in packet-switched networks to prevent one user's packet from hogging a link. A sending end host with a very large file (petabyte/exabyte sized file) will have to break up the file into multiple file transfers and compete for DREEoS resources for each transfer. We need further study to select an appropriate MFTS.

For error control, we propose using ST's support for negative acknowledgments (NAKs) given that data blocks will be delivered in sequence on the DREEoS circuit. Since optical circuits have low bit error rates, and there are no packet switches on DREEoS paths (and hence no dropped packets due to buffer overflows), the probability of error/loss is quite low. If a block is received out of sequence or there is an error in a received data block, the receiving ST module will send a CTS for the missing/errored block requesting a retransmission (this is equivalent to a NAK). The sender then sends only the missing block (selective repeat).

RTS/CTS exchanges, NAKs (in the form of CTSs), retransmissions are sent on the TCP/IP path. The reason for sending retransmissions on the TCP/IP path is that a retransmission may be required at the very end of the file transfer in which case the circuit will have to held open in idle state until the reverse NAK is received. This will impact utilization. Since the number of retransmissions will be very few given the reliability of optical circuits and the absence of packet switches, we propose sending all retransmissions on the TCP/IP path.

4. Conclusions

We conclude that for intra-network file transfers, protocols specific to that network should be used instead of TCP/IP to achieve better performance. Specifically, we considered the case of optical circuit-switched networks. File transfers between end hosts connected by such a network can be handled by setting up a circuit dynamically, transferring the file, and then releasing the circuit. This will make file transfer durations very short especially as link rates increase. To handle calls with short holding times, we demonstrated the advantages of hardware acceleration of signaling protocol implementations. The circuit-switched network is operated in a call blocking mode and limits the maximum file transfer size (MFTS) for fairness reasons. To make long-distance connections between end hosts using optical circuits, we showed that current deployments can be leveraged to create a type of circuit that we called Dynamically Reconfigurable Ethernet/EoS (DREEoS). We envision this capability being introduced into the current Internet gradually. This means two end hosts that can use a DREEoS circuit for a file transfer would also have a TCP/IP path, a feature that affords us great flexibility in designing our protocols. Through analysis, we showed that an end host should first attempt setting up a DREEoS circuit if (i) file sizes are large (ii) file sizes are small but round-trip times (RTT) are large and (iii) file sizes and RTT are small, then for files larger than some crossover file size. If a call request is blocked, then the TCP/IP path is used for the file transfer. Exact crossover points (between "large" and "small" files or RTT) depends upon probability of packet loss on the TCP/IP path, call blocking probability on the optical circuit-switched network, and link rates.

5. Acknowledgments

We thank Wu-chun Feng and Mark Gardner of Los Alamos National Laboratory for bringing ST to our attention. We are currently defining a project in collaboration with them to implement the concepts described in this paper.

6. References

[1] DOE Office Of Science High Performance Network Planning Workshop, http://doecollaboratory.pnl.gov/meetings/hpnpw/workshopdescription.pdf, August 13-15, 2002.

[2] Bill St. Arnaud, "Proposed CA*net 4 Network Design and Research Program," Revision no. 8, April 2, 2002.

[3] S. Floyd, "High Speed TCP for Large Congestion Windows," August 2002, http://www.ietf.org/internet-drafts/draft-floyd-tcp-highspeed-01.txt.

[4] J. J. Bunn, J. C. Doyle, S. H. Low, H. B. Newman, S. M. Yip, "Ultrascale Network Protocols for Computing and Science in the 21st Century," September 2002, http://netlab.caltech.edu/FAST/.

[5] First International Workshop on Protocols for Fast Long-Distance Networks, PFLDnet 2003, http://datatag.web.cern.ch/datatag/pfldnet2003/, Feb. 3-4, 2003, Geneva, Switzerland.

[6] J. L. Hennessy and D. A. Patterson, "Computer Architecture - A Quantitative Approach," Morgan Kaufmann Publishers, 1990, pp. 870.

[7] ITU-T Rec. G.7041, "Generic Framing Procedure (GFP)," Oct. 2001.

[8] Fujitsu, "FLM 150 ADM: Flexible OC-3 and OC-12 Add/Drop Multiplexer," http://www.fnc.fujitsu.com/prod-

ucts/view_325.html.
[9] Cisco, “Cisco ONS 15454 Optical Transport Platform,” http://www.cisco.com/warp/public/cc/pd/olpl/metro/on15454/prodlit/ons15_ds.htm.
[10] Ciena, “CIENA MultiWave MetroDirector K2™ Next-Generation Multi-Service Access and Switching Platform,” http://www.ciena.com/products/switching/metrodirectork2/index.asp.
[11] ITU-T Rec. G.707, “Network Node Interface for the Synchronous Digital Hierarchy,” Oct. 2000.
[12] Special Issue of IEEE Communications Magazine on “Generic Framing Procedure (GFP) and Data over SONET/SDH and OTN,” May 2002.
[13] E. Mannie, “GMPLS Architecture,” *IETF Internet Draft*, draft-many-gmpls-architecture-00.txt, Mar. 2001.
[14] P. Ashwood-Smith, et al, “Generalized MPLS Signaling - CR-LDP Extensions,” *IETF Internet Draft*, draft-ietf-mpls-generalized-cr-ldp-03.txt, May 2001.
[15] P. Ashwood-Smith, et al. “Generalized MPLS - RSVP-TE Extensions,” *IETF Internet Draft*, draft-ietf-mpls-generalized-rsvp-te-04.txt, July 2001.
[16] Optical Internetworking Forum, “Supercomm 2001 OIF UNI Demonstration White Paper,” http://www.oiforum.com, 2001.
[17] H. Wang, M.Veeraraghavan and R. Karri, “A hardware implementation of a signaling protocol,” accepted to *Opticomm 2002*, July 29-Aug. 2, 2002, Boston, MA.
[18] D. Clark and B. Hutchings, “Supporting FPGA Microprocessors Through Retargetable Software Tools,” *Proc. IEEE Symposium on FPGAs for Custom Computing Machines*, Apr.1996.
[19] S. K. Long, R. R. Pillai, J. Biswas, T. C. Khong, “Call Performance Studies on the ATM Forum UNI Signalling,” http://www.krdl.org.sg/Research/Publications/Papers/pillai_uni_perf.pdf.
[20] M. Veeraraghavan and R. Karri, “Towards enabling a 2-3 orders of magnitude improvement in call handling capacities of switches,” NSF proposal 0087487, 2001.
[21] N. Cardwell, S. Savage, and T. Anderson, “Modeling TCP Latency,” *Proc. of IEEE Infocom*, Mar. 26-30, 2000, Tel-Aviv, Israel, pp. 1724-1751.
[22] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, “Modeling TCP Throughput: A Simple Model and its Empirical Validation,” Proc. of *ACM SIGCOMM 98,* Aug. 31 - Sep. 4, Vancouver Canada, pp. 303-314.
[23] M. Allman, V. Paxson, W. Stevens, “TCP Congestion Control”, *IETF RFC 2581*, Apr. 1999.
[24] V. Paxson and S. Floyd, “Wide-Area Traffic: The Failure of Poisson Modeling,” *IEEE/ACM Trans. Networking*, vol. 3, pp. 226-244, June 1995.
[25] N.G. Duffield, J. Horowitz, D. Towsley, W. Wei, and T. Friedman, “Multicast-Based Loss Inference with Missing Data,” to appear in *IEEE Journal on Selected Areas in Communications,* http://www-net.cs.umass.edu/papers/papers.html.
[26] W. Doeringer, D. Dykeman, M. Kaiserswerth, B. W. Meister, H. Rudin, R. Williamson, “A survey of lightweight transport protocols for high-speed networks”, *IEEE Trans. Comm.*, 38(11):2025-39, Nov. 1990.
[27] S. Iren, P. D. Amer and P. T. Conrad, “The Transport Layer: Tutorial and Survey,” *ACM Computing Surveys*, Vol. 31, No. 4, Dec. 99.
[28] M. Blumrich, C. Dubrucki. E. Felton, and K. Li, “Protected, User-Level DMA for the SHRIMP Network Interface,” In *Proceedings 2nd International Symposium on High Performance Architecture*, San Jose, CA, Feb. 3-7, 1996, pp. 154-165.
[29] P. Druschel and L.L. Peterson and B.S. Davic, “Experiences with a High-Speed Network Adapter: A-Software Perspective,” In *Proceedings of ACM Sigcomm’94*, Aug. 1994.
[30] S. Pakin, M. Lauria, and A. Chien, “High Performance Messaging on Workstations: Illinois Fast Messages (FM) for Myrinet,” In *Proceedings of Supercomputing’95*, San Diego, CA, 1995.
[31] I. R. Philip and Y.-L. Liong, “The Scheduled Transfer (ST) Protocol,” *3rd Intl. Workshop on Communications, Architecture and Applications for Network-Based Parallel Computing (CANC’99), Lecture Notes in Computer Science*, vol. 1602, Jan. 1999.
[32] ANSI, “Information Technology - Scheduled Transfer Protocol (ST),” T11.1/Proj. 1245-M/Rev 4.0, Oct. 2000.
[33] J. S. Chase, A. J. Gallatin, K.G. Yocum, “End-System Optimizations for High-Speed TCP,” *IEEE Communications Magazine*, vol. 39, no. 4, April 2001, pp. 68 -74.
[34] M. Benz, “An Architecture and Prototype Implementation for TCP/IP Hardware Support,” Proc. of TERENA Networking Conference 2001, http://www.terena.nl/conferences/tnc2001/proceedings/PaperBenz.pdf, May 14-17, 2001, Antalya, Turkey.