Best paper award SPIE/IEEE Optical Networking and Computer Communications Conference (OptiComm 2003), Dallas TX, October 2003.

CHEETAH: Circuit-switched High-speed End-to-End Transport ArcHitecture

M. Veeraraghavan*[a], X. Zheng[a], H. Lee[b], M. Gardner[c], W. Feng[c]
[a]University of Virginia, 351 McCormick Rd, Charlottesville, VA 22904;
[b]Polytechnic University, 5 Metrotech Center, Brooklyn, NY 11201;
[c]Los Alamos National Laboratory, P.O. Box 1663, M.S. D451, Los Alamos, NM 87545

ABSTRACT

Leveraging the dominance of Ethernet in LANs and SONET/SDH in MANs and WANs, we propose a service called CHEETAH (Circuit-switched High-speed End-to-End Transport ArcHitecture). The service concept is to provide end hosts with high-speed, end-to-end circuit connectivity on a call-by-call shared basis, where a "circuit" consists of Ethernet segments at the ends that are mapped into Ethernet-over-SONET long-distance circuits. This paper focuses on the file-transfer application for such circuits. For this application, the CHEETAH service is proposed as an add-on to the primary Internet access service already in place for enterprise hosts. This allows an end host that is sending a file to first attempt setting up an end-to-end Ethernet/EoS circuit, and if rejected, fall back to the TCP/IP path. If the circuit setup is successful, the end host will enjoy a much shorter file-transfer delay than on the TCP/IP path. To determine the conditions under which an end host with access to the CHEETAH service should attempt circuit setup, we analyze mean file-transfer delays as a function of call blocking probability in the circuit-switched network, probability of packet loss in the IP network, round-trip times, link rates, and so on.

Keywords: Signaling Protocols, Ethernet, SONET, Ethernet-over-SONET, Optical Networks, Circuit Switching

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[1-8]. Of particular interest is the effective throughput of transfers of large files, e.g., terabyte and petabyte (10^{15}) sized files created in particle physics, earth observation, bioinformatics, radio astronomy, and other scientific studies, for which current TCP has been shown to be inadequate[9]. **One** set of solutions calls for enhancing TCP to improve end-to-end throughput, thus limiting upgrades to the end hosts. Such improvements can be made via congestion control[2-4] and/or flow control[5-7]. A **second** set of solutions requires upgrades to routers within the Internet. For example, Mathis[8] proposes the use of larger Maximum Transmission Unit (MTU) to improve end-to-end throughput. Given that the Internet is a global network of IP routers that interconnects different types of networks, as illustrated in Fig. 1, to improve file-transfer performance between any two hosts connected via the Internet, research work must focus on enhancing TCP and/or IP, as is currently being done in these first two sets of solutions[2-8].

However, noting that file-transfer performance can be improved on intra-network paths by using protocols that are better tuned for that network, we propose a **third** set of solutions in which intra-network paths are used whenever possible. For example, in Fig. 1 we show that by connecting second NICs of Ethernet hosts via a wide-area Ethernet-over-SONET circuit, we can create a direct, high-speed, end-to-end circuit and achieve low file-transfer delays. Clearly this solution has limited applicability when compared to the first two sets of solutions. However, if a given network has a large coverage area, this solution will be useful in many transfers. Current-day SONET/SDH/WDM circuit-switched networks of different service providers are largely isolated, but as standards evolve, these can be interconnected directly to achieve larger coverage areas. A few research optical networks such as the Canarie network[10] extends coast-to-coast across Canada.

The problem statement of this work is as follows: Define a set of mechanisms to realize fast file transfers on intra-network paths through shared, high-speed, optical circuit-switched networks. This is not a pure academic exercise. A few recent technological advances have made implementation of this concept quite feasible. These advances include (i) deployment of optical fiber to enterprises, (ii) deployment of Multi-Service Provisioning Platforms (MSPPs) in enterprises, and (iii) the inclusion of Ethernet over SONET (EoS) capabilities in MSPPs.

MSPPs are currently used to integrate T1s from PBXs carrying voice traffic and T1s/T3s from WAN-access IP routers carrying data traffic on to the same SONET link; hence the term "multi-service." In addition, Ethernet frames can be car-

*mv@cs.virginia.edu; phone 1-434-982-2208; fax 1-434-982-2214; http://www.ece.virginia.edu/~mv

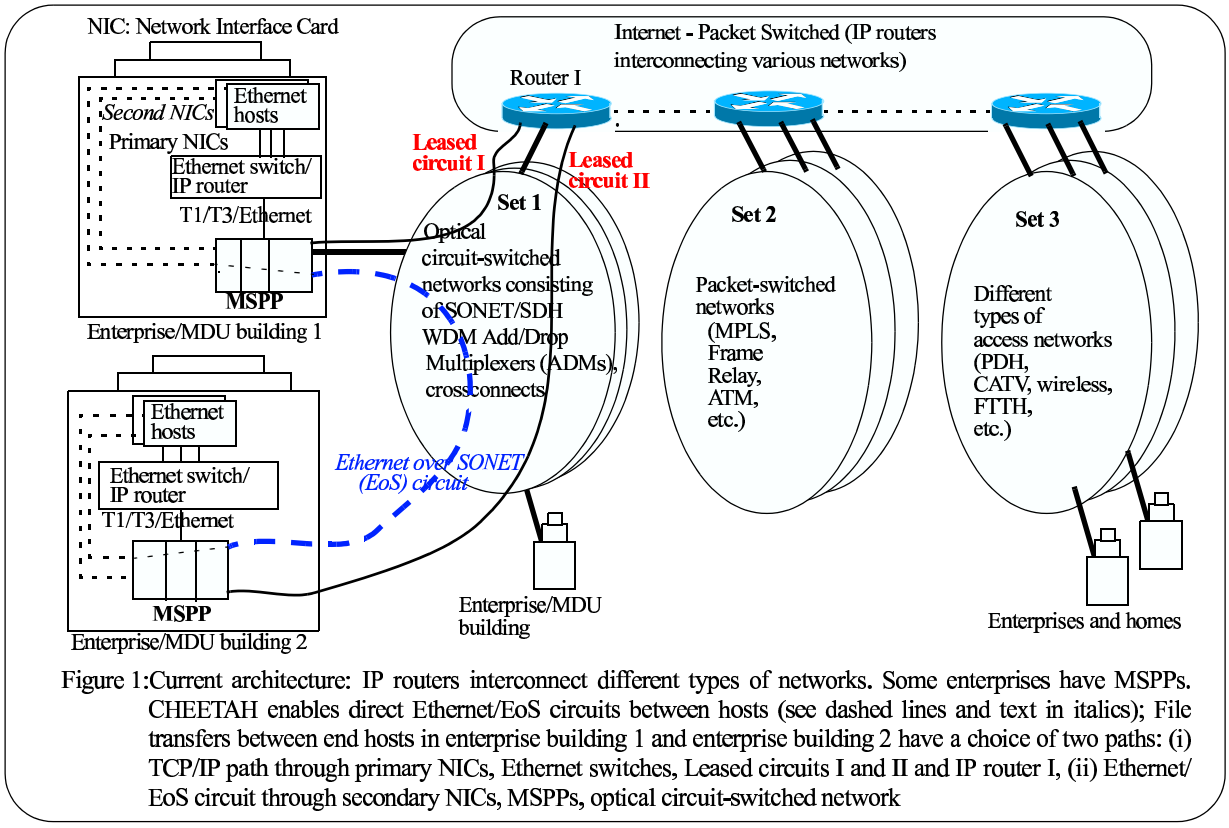


Figure 1:Current architecture: IP routers interconnect different types of networks. Some enterprises have MSPPs. CHEETAH enables direct Ethernet/EoS circuits between hosts (see dashed lines and text in italics); File transfers between end hosts in enterprise building 1 and enterprise building 2 have a choice of two paths: (i) TCP/IP path through primary NICs, Ethernet switches, Leased circuits I and II and IP router I, (ii) Ethernet/ EoS circuit through secondary NICs, MSPPs, optical circuit-switched network

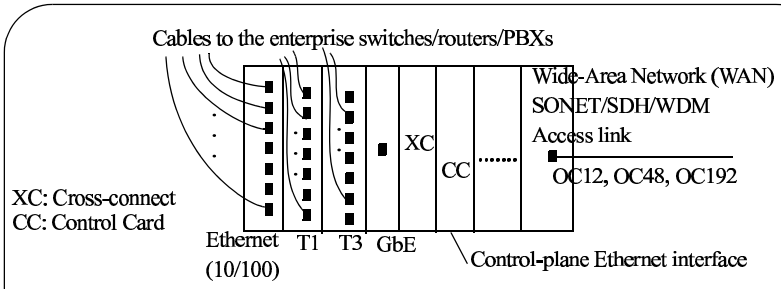


Figure 2:An example multiservice provisioning platform architecture

ried within SONET frames using Ethernet-over-SONET (EoS) techniques, such as Generic Framing Procedure (GFP)[11], which defines a method to encapsulate Ethernet frames within SONET frames, and Virtual Concatenation[12] to allow for arbitrary-bandwidth SONET signals to be created. An example MSPP architecture is shown in Fig. 2[13]. Nodes within an enterprise are connected to interface cards, such as Ethernet (10Mbps/100Mbps), T1, T3, Gb/s Ethernet, etc. The XC (Cross-connect) card is used to cross-connect signals from incoming ports to outgoing ports. The control card typically has a processor and implements software to carry out functions such as setting up and releasing cross-connections through the MSPP. Some MSPP control cards implement the latest signaling protocol standards, such as the Optical Internetworking Forum (OIF) User Network Interface (UNI)[14] specification based on the IETF Generalized MultiProtocol Label Switching (GMPLS) specification[15]. Communication with the control card is through its own control-plane Ethernet interface as shown in Fig. 2. The WAN access link card has a high-rate SONET, SDH and/or WDM interface. Typically, Ethernet, T1, T3 signals from the interface cards connected to nodes within the enterprise are cross-connected through the XC card to equivalent-level signals on the WAN SONET link. We described our proposed CHEETAH service, which is based on using these MSPPs.

2. PROPOSED CHEETAH SERVICE AND ITS APPLICATION TO FILE TRANSFERS

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 signaling protocols. **Section 2.1** describes the equipment needed to support the CHEETAH service.

Since the CHEETAH service 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 an end-to-end Ethernet/EoS circuit. In **Section 2.2**, we describe a support service for this purpose called "Optical Connectivity Service (OCS)."

Next, we consider the question of how to use the CHEETAH service for file-transfer applications. File-transfer sessions require the exchange of many back-and-forth messages in addition to the actual file transfer. We propose using a TCP connection via the primary Internet path for such short exchanges, and limiting the use of end-to-end Ethernet/EoS circuits for the actual file transfers. To achieve high utilization of the circuit-switched network, we propose (i) setting up the end-to-end high-speed Ethernet/EoS circuit just prior to the actual transfer and releasing it immediately after the file transfer, (ii) operating the circuit-switched network in call blocking mode, (iii) using circuits only for certain transfers, and (iv) using a unidirectional EoS circuit from the server to the client (since this is the primary direction of data flow).

The implication of holding circuits only for the duration of file transfers is that call holding times can be quite small. 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. Therefore, we recommend a hardware-accelerated implementation of signaling protocols at MSPPs, Add/Drop Multiplexers (ADMs), crossconnects and other optical circuit switches. **Section 2.3** describes our current work on hardware-accelerated signaling implementations.

Circuit-switched networks can be operated in call blocking mode or call queueing mode. Given that our proposal for CHEETAH is as an add-on service to basic Internet connectivity, we can afford to run the circuit-switched network in the simpler call blocking mode. In this mode, if an end host's call setup request for an Ethernet/EoS circuit is blocked, it can fall back to the TCP/IP path (see Fig. 1 for an example of the two paths available to end hosts equipped for CHEETAH service). We recognize that it is not appropriate to attempt a circuit setup for all transfers. For example if a file is small (order of a few KB), the total delay incurred in attempting a circuit setup and then transferring the file could be larger than the delay incurred in directly using the TCP/IP path. Thus, a "routing decision" needs to be made at end hosts with access to CHEETAH. **Section 2.4** analyzes metrics that impact this routing decision.

Finally, we consider the question of transport protocols for end-to-end Ethernet/EoS circuits. We found a transport protocol called Scheduled Transfer (ST), an ANSI standard[16], which is ideally suited for end-to-end Ethernet/EoS circuits. **Section 2.5** describes our data-transport approach.

The presence of the **fallback TCP/IP path** is a key ingredient of the CHEETAH service. It enables operation of the circuit-switched network at a high utilization by allowing the circuit-switched network to be run at a high call-blocking rate (at least initially when the number of subscribers is small). It also allows for an "all-or-nothing" full-bandwidth allocation approach in which the circuit-switched network either allocates the maximum bandwidth requested by an end host (limited by the host's NIC speed or processing limits) or rejects the call. This would not have been possible in a pure circuit-switched network because of fairness considerations. The implication is that the resulting performance will be equivalent to the best scenario possible had the second NIC fed into a packet-switched network instead of a circuit-switched network.

2.1 Equipment

1. Due to the "add-on" characteristic of the CHEETAH service, hosts that want access to this service should be equipped with second Ethernet NICs that 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. 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. Application software in end hosts should be upgraded to interface with the CHEETAH service.

2.2 Optical Connectivity Service (OCS)

A support service called the "Optical Connectivity Service (OCS)" is proposed to provide end hosts a mechanism to determine whether or not their correspondent end hosts have access to the CHEETAH service. OCS can be implemented much like the Domain Name Service (DNS) with enterprises and service provider networks maintaining servers with

information on end hosts that have access to the CHEETAH service. These servers would answer queries from end hosts in much the same manner as DNS servers answer queries for IP addresses and other information. With caching, the delay incurred in this step can be reduced.

2.3 Hardware acceleration of signaling protocol implementations

Accelerating signaling protocol processing engines is a challenging task. Our work-to-date on this task has been to implement our own signaling protocol in hardware[17]. 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 in Field Programmable Gate Arrays (FPGAs), and relegated the complex and infrequently used operations (e.g., processing of optional parameters and error handling) to software. 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-confirm*. From the timing simulations, done using the ModelSim® simulator, call setup message processing consumes between 77-101 clock cycles. Assuming a 25 MHz clock, this translates into 3.08-4 μs. Compare this with the millisecond-based software implementations of signaling protocols[18].

2.4 Routing decision

In this section, we study the impact of various parameters on the routing decision of whether or not to attempt setting up an Ethernet/EoS circuit.

2.4.1 File-transfer delay models

Let $E[T_{cheetah}]$ be the mean delay incurred if an Ethernet/EoS circuit setup is attempted prior to the file transfer.

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

where P_b is the call blocking probability on the optical circuit-switched network, $E[T_{setup}]$ is the mean call-setup delay of a successful circuit setup, $T_{transfer}$ is the time to transfer the file on the Ethernet/EoS circuit, $E[T_{fail}]$ is the mean delay incurred in a failed call setup attempt, and $E[T_{tcp}]$ is the mean delay incurred in sending the file on the TCP/IP path. If the call is not blocked, mean delay experienced is $(E[T_{setup}]+T_{transfer})$, but if it is blocked, then after incurring a cost $E[T_{fail}]$, the end host has to use the TCP/IP path and hence will incur the $E[T_{tcp}]$ delay. Comparing $E[T_{tcp}]$, the delay incurred if a circuit setup is not attempted, with $E[T_{cheetah}]$, the delay incurred if a circuit setup is attempted, and approximating $E[T_{fail}]$ to be equal to $E[T_{setup}]$, results in:

$$\begin{aligned} &\text{if } \left(\frac{E[T_{setup}]}{1-P_b} \geq (E[T_{tcp}]-T_{transfer})\right) && \text{resort directly to the TCP/IP path} \\ &\text{if } \left(\frac{E[T_{setup}]}{1-P_b} < (E[T_{tcp}]-T_{transfer})\right) && \text{attempt circuit setup} \end{aligned} \quad (2)$$

Next, we obtain expressions for $E[T_{tcp}]$, $E[T_{setup}]$ and $T_{transfer}$. $E[T_{tcp}]$ is obtained using the models of [19-20], which captures 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 $E[T_{delayack}]$.

$$E[T_{tcp}] = E[T_{ss}]+E[T_{loss}]+E[T_{ca}]+E[T_{delayack}] \quad (3)$$

The first three terms on the right hand side of (3) are derived as closed-form expressions in [19], as functions of three key parameters: packet loss rate P_{loss}, round-trip time RTT, and bottleneck link rate r. We set the final term $E[T_{delayack}]$ to 0 because we assume a starting initial window size of 2^{21}· and the ACK-every-other-segment strategy. We do not include TCP connection-setup time assuming that the connection is already open (because a TCP connection needs to be opened first for sending information such as the file name/location). We assume that all file transfers start in the slow start phase because the congestion window resets to a restart window size (2 segments) whenever the session is idle for more than one retransmission timeout[21]·.

$E[T_{setup}]$ includes mean signaling message transmission delays, mean call processing delays (to process signaling protocol messages), and a round-trip propagation delay:

$$E[T_{setup}] = \frac{m_{sig}}{r_s}\times\left(1+\frac{\rho_{sig}}{2(1-\rho_{sig})}\right)\times(k+1)+T_{sp}\times\left(1+\frac{\rho_{sp}}{2(1-\rho_{sp})}\right)\times k+T_{prop} \quad (4)$$

where 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* at a load ρ_{sp}.

$T_{transfer}$ is the actual file-transfer delay:

$$T_{transfer} = \frac{f}{r_c} + \frac{T_{prop}}{2} \tag{5}$$

where f is the size of the file being transferred and r_c is the data rate of the circuit. We have not included retransmission delays here because on Ethernet/EoS circuits, retransmissions are only required when random bit errors affect a block of data, and these types of errors also impact delays on the TCP/IP path. Including this delay would in fact favor using the Ethernet/EoS circuit. This is because bit errors on the TCP/IP path would be misinterpreted as packet losses caused by congestion leading to a reduction in the sending rate.

2.4.2 Numerical results for transfer delays of "large" files

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 at the bottleneck link. We derive

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

Case	Input parameters			Intermediate derived results			Final results	
	Loss P_{loss}	Rate r	Round-trip prop. delay T_{prop}	Queuing delay plus service time	RTT (ms)	W_{max} (pkts)	$E[T_{tcp}]$ for a 1GB file (s)	$E[T_{tcp}]$ for a 1TB file
Case 1	0.0001	100 Mb/s	0.1ms	0.2ms	0.3	2.5	82.25	22.9 hours
Case 2			5ms		5.2	41	89.45	1 day and 1.3 hours
Case 3			50ms		50.2	418	396.5	4 days and 15.3 hours
Case 4	0.0001	1Gbps	0.1ms	0.02ms	0.12	10	8.25	2.3 hours
Case 5			5ms		5.02	418	39.6	11.1 hours
Case 6			50ms		50.02	4168	395.7	4 days and 14.9 hours
Case 7	0.001	100 Mbps	0.1ms	0.26ms	0.36	3	82.93	22.9 hours
Case 8			5ms		5.26	43.8	135.4	1 day and 0.1 hour
Case 9			50ms		50.26	418.8	1293	4 days and 15.4 hours
Case 10	0.001	1Gbps	0.1ms	0.026ms	0.13	10.8	8.64	2.3 hours
Case 11			5ms		5.03	419	129.4	11.1 hours
Case 12			50ms		50.03	4169	1287	4 days and 14.9 hours
Case 13	0.01	100 Mbps	0.1ms	0.38ms	0.48	4	92.41	22.9 hours
Case 14			5ms		5.38	44.8	471.7	1 day and 0.2 hours
Case 15			50ms		50.38	419.8	4417	4 days and 15.7 hours
Case 16	0.01	1Gbps	0.1ms	0.038ms	0.138	11.5	12.43	2.3 hours
Case 17			5ms		5.038	419.8	441.7	11.2 hours
Case 18			50ms		50.04	4169.8	4387	4 days and 14.9hours
Case 19	0.1	100 Mbps	0.1ms	0.68ms	0.78	6.5	283.56	22.9 hours
Case 20			5ms		5.68	47.33	2064.9	1 day and 0.3 hours
Case 21			50ms		50.68	422.33	18424	4 days and 16.3 hours
Case 22	0.1	1Gbps	0.1ms	0.068ms	0.168	14	61.07	2.3 hours
Case 23			5ms		5.068	422.33	1842.4	11.2 hours
Case 24			50ms		50.07	4172.3	18202	4 days and 15 hours

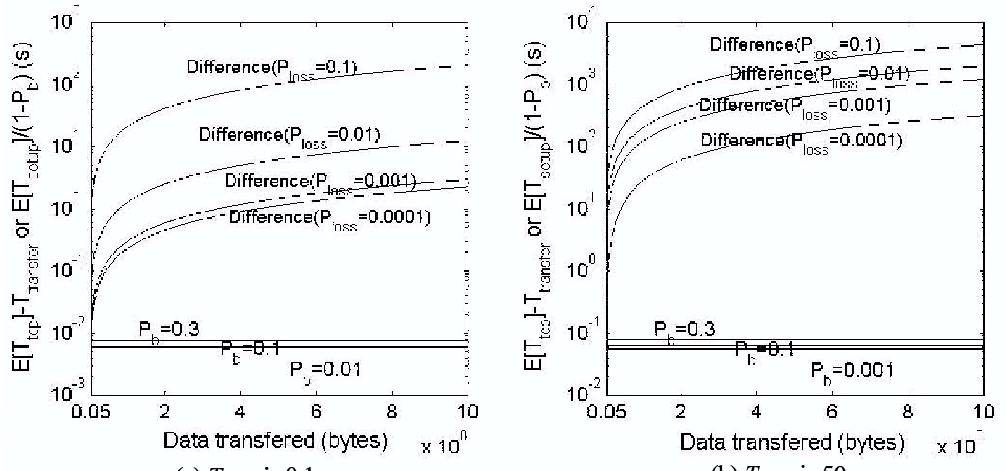
*M/D/1 queueing models are quite accurate since inter-arrival times between file transfers have been shown to be exponentially distributed[22], and signaling message lengths and call processing delays are more-or-less constant.

this estimate by determining the load at which an M/D/1/k system* will experience the assumed P_{loss} values. For all the cases, we set W_{max} to the delay-bandwidth product, i.e., $W_{max} = RTT \times r$.

Using the input parameters shown in Table 1, we compute $E[T_{tcp}]$ given by (3) for a 1GB file and 1TB file and list the values in the last two columns 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 $E[T_{tcp}]$, e.g., 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 the 1GB 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 283.56s for the 1GB file transfer when P_{loss} increases from 0.0001 to 0.1.

Compare $E[T_{tcp}]$ for a **1GB file** transfer from Table 1 with delays incurred on a successfully established end-to-end circuit. The delay on a circuit, $E[T_{setup}] + T_{transfer}$, is 80.08sec when the link rate is 100Mbps and 8.08sec when the link rate is 1Gbps. These numbers are obtained assuming $m_{sig} = 100B$, $r_s = 10Mbps$, $\rho_{sig} = \rho_{sp} = 0.8$, $T_{sp} = 4\mu s$ (see Section 2.3), $T_{prop} = 50ms$, and k, the number switches on the end-to-end path, set to 20. The major component of these values is $T_{transfer}$. $E[T_{setup}]$ is only 55.3ms. In other words, in wide-area networks or in lossy environments, the reduction in file-transfer delay using an Ethernet/EoS circuit is significant.

Compare the file-transfer delays for a **1TB file** shown in Table 1 with delays on an end-to-end high-speed Ethernet/EoS circuit. For example, with a 1Gbps Ethernet/EoS circuit, a 1TB file will take about 2.2 hours, which is comparable to the TCP/IP path numbers for the low propagation delay environment when T_{prop} is 0.1ms, but significantly less than the TCP/IP path numbers when T_{prop} is 5 or 50ms. The bulk of the 2.2 hours number is the file transfer time $T_{transfer}$; $E[T_{setup}]$ is in the order of ms as shown above. This is not a surprising result because the delay for the end-to-end circuit is possible only if the call is not blocked. Once a circuit is set up there is no reduction in delay due to competition from other users.

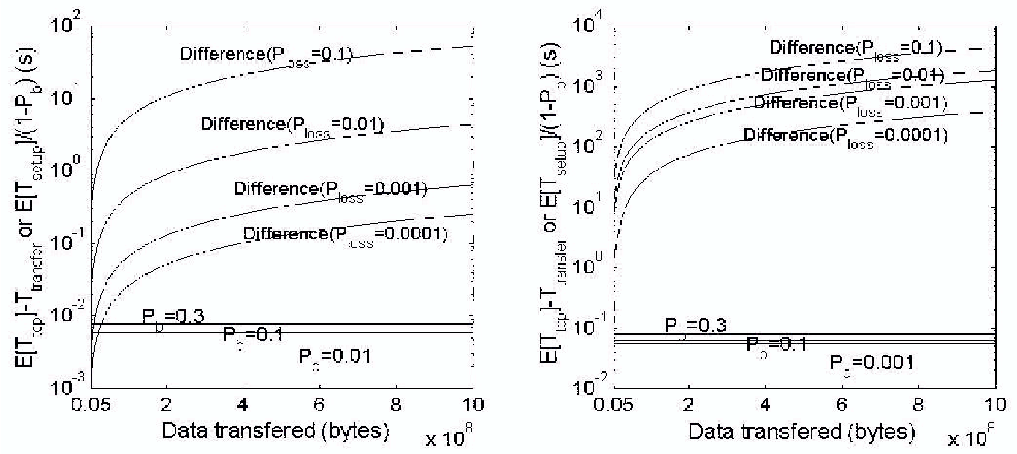


(a) T_{prop} is 0.1ms (b) T_{prop} is 50ms

Figure 3: Plot of equation (2) for large files with a link rate of 100Mbps, $\rho_{sig} = \rho_{sp} = 0.7$, $k = 20$

To take into account blocking probability, we next plot (2), the basis for the routing decision, in Fig. 3 and Fig. 4 for the 100Mbps and 1Gbps link rates, respectively. For the three horizontal lines on which P_b values are listed, the y-axis is the left-hand side of (2), 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 (2), i.e., $E[T_{tcp}] - T_{transfer}$. In Fig. 3, when the link rate is 100Mbps for the entire file range (5MB, 1GB), an Ethernet/EoS circuit should be attempted if P_b and P_{loss} have the values shown. This is

*While packet transmission (service) time is more-or-less deterministic because of MTU restrictions, the packet arrival process at a buffer feeding the bottleneck link is known not to be a Poisson process[22]. However we use this approximate model to obtain a rough estimate of queueing plus service delay.



(a) T_{prop} is 0.1ms (b) T_{prop} is 50ms

Figure 4:Plot of equation (2) for large files with a link rate of 1Gbps, $\rho_{sig} = \rho_{sp} = 0.7$, $k = 20$

because $(E[T_{setup}])/(1-P_b)$ is always less than the difference term $E[T_{tcp}] - T_{tranfser}$ (see (2)).

However, when the bottleneck link rate increases to 1Gbps (Fig. 4), while we see a similar pattern when T_{prop} is 50ms (WAN environments), in a lower-propagation delay environment (Fig. 4(a) in which T_{prop} =0.1ms), we see that there are crossover file sizes below which an end host should resort directly to the TCP/IP path and above which it should attempt an Ethernet/EoS circuit setup. These crossover file sizes are listed in Table 2.

Table 2: Crossover file sizes in the [5MB, 1GB] range when $r = 1Gbps$, $T_{prop} = 0.1ms$, $k = 20$

Measure of loading on: ckt. sw. network / TCP/IP path	$P_b = 0.01$	$P_b = 0.1$	$P_b = 0.3$
$P_{loss} = 0.0001$	22MB	24MB	30MB
$P_{loss} = 0.001$	9MB	10MB	12MB
$P_{loss} = 0.01$	<5MB	<5MB	<5MB

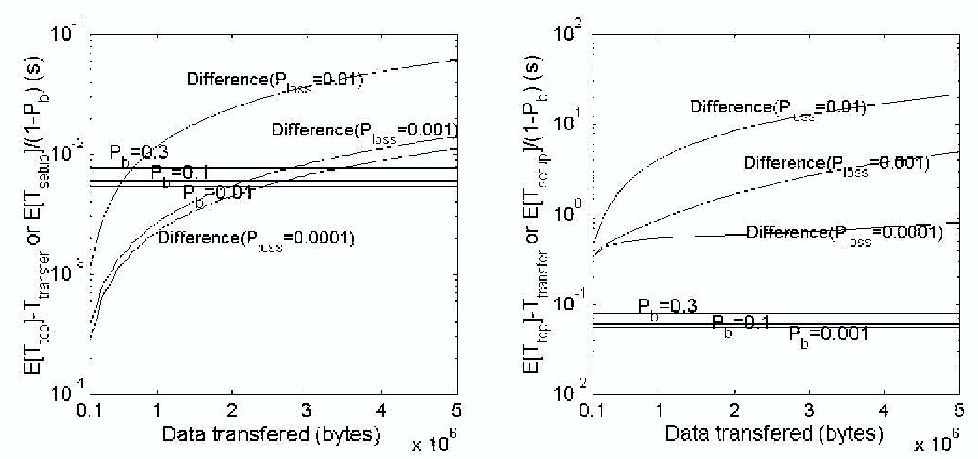
In summary, in the current-day Internet, where bottleneck link rates are in the order of Mbps for enterprise users, it is worthwhile attempting a circuit setup for files 5MB and over in most MAN and WAN environments (T_{prop} of 0.1ms, 5ms, 50ms). This holds true even as rates increase to 100Mbps. But as links become upgraded to the Gbps range, such circuit attempts should be made mainly in wide-area environments or for larger files.

2.4.3 Numerical results for transfer delays of "small" files

Even though our motivation for this work comes from high-end scientific applications with very large files, we wanted to understand whether the CHEETAH service could be used for smaller files. Fig. 5 and Fig. 6 plot (2) for smaller files (100KB to 5MB). Unlike larger files, where we studied the impact of link rate, here we study the impact of the number of switches on the end-to-end path keeping the link rate at 100Mbps. Fig. 5 plots the results for the case when the numbers of switches on the end-to-end path k is 4 and Fig. 6 plots the $k = 20$ case.

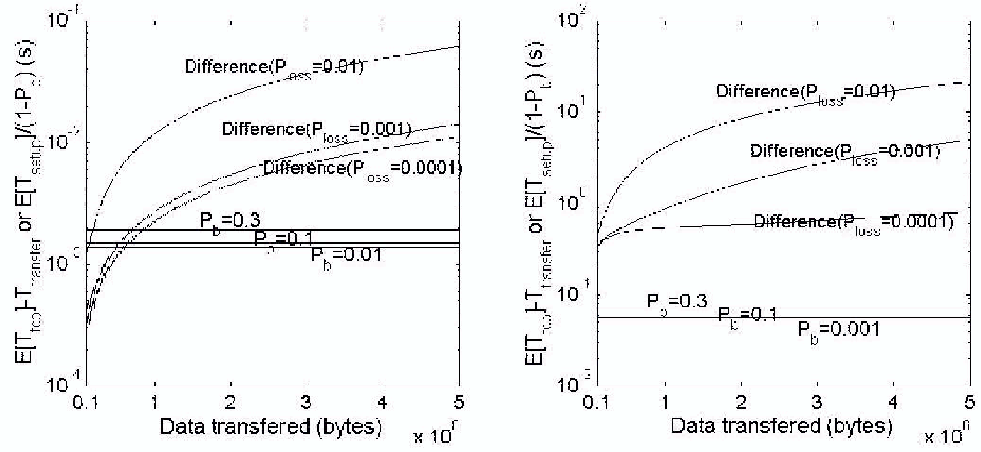
Our first observation is that in wide-area network scenarios shown in Fig. 5(b) and Fig. 6(b), for the entire file range (100KB, 5MB), an Ethernet/EoS circuit should be attempted if P_b and P_{loss} have the values shown in these plots. This is because the difference term $E[T_{tcp}] - T_{tranfser}$ is always greater than $(E[T_{setup}])/(1-P_b)$.

For lower propagation-delay environments, e.g., T_{prop} is 0.1ms, in Fig. 5(a) and Fig. 6(a), we see crossover file sizes below which an end host should resort directly to the TCP/IP path and above which it should attempt an Ethernet/EoS circuit setup. These crossover file sizes are listed in Table 3. The number of switches on the end-to-end path k has little impact on the total transfer times, but it does affect $E[T_{setup}]$ especially when T_{prop} is 0.1ms. As a result, crossover file



(a) T_{prop} is 0.1ms (b) T_{prop} is 50ms

Figure 5:Plot of equation (2) for smaller files with a link rate of 100Mbps, $\rho_{sig}=\rho_{sp}=0.7$, k=4



(a) T_{prop} is 0.1ms (b) T_{prop} is 50ms

Figure 6:Plot of equation (2) for smaller files with a link rate of 100Mbps, $\rho_{sig}=\rho_{sp}=0.7$, k=20

Table 3: Crossover file sizes in the (100Kb, 5MB) range when $r = 100Mbps$ and $T_{prop} = 0.1ms$

Measure of loading on ckt. sw. network / TCP/IP path	Number of switches on the path $k = 4$			Number of switches on the path $k = 20$		
	$P_b = 0.01$	$P_b = 0.1$	$P_b = 0.3$	$P_b = 0.01$	$P_b = 0.1$	$P_b = 0.3$
$P_{loss} = 0.0001$	610KB	640KB	840KB	2.4MB	2.65MB	3.4MB
$P_{loss} = 0.001$	490KB	550KB	730KB	2MB	2.2MB	2.8MB
$P_{loss} = 0.01$	120KB	140KB	180KB	500KB	550KB	650KB

sizes in Fig. 6(a) are much larger than those in Fig. 5(a), as seen in Table 3.

The conclusions we draw from this user file-transfer delay analysis is that a circuit setup should be attempted if T_{prop} is 50ms for files 100KB or larger for 100Mbps links. In low propagation-delay environments, it depends upon the file size. For "large" files, a circuit setup should be attempted. The size at which a file is considered "large" depends upon the bottleneck link rate and the loading conditions on the two paths, the TCP/IP path and the circuit-switched network path.

2.4.4 Optical circuit-switched network utilization considerations

While file-transfer delay is an important user measure for making the routing decision of whether or not to attempt a circuit setup, service provider measures such as utilization should also be considered since utilization ultimately does impact users through prices charged. Total network utilization has two components: aggregate network utilization u_a and per-circuit utilization u_c, which are given by:

$$u_a = \frac{(1-P_b)\times\rho}{m}, \text{ where } P_b = \frac{\rho^m/m!}{\sum_{k=0}^{m}\rho^k/k!} \text{ (Erlang-B formula)}, \tag{6}$$

$$u_c = \frac{E[T_{transfer}]}{E[T_{setup}]+E[T_{transfer}]}, \text{ where } E[T_{transfer}] = \frac{E[X]}{r_c}, \tag{7}$$

ρ is the offered traffic load, m is the number of circuits, $E[X]$ is the average file size, and r_c is the circuit rate.

Restricting transfers on the circuit-switched network to files larger than some crossover file size, χ, we can compute the fractional offered load ρ' and the average file size $E[X|(X\geq\chi)]$ if we know the distribution of file sizes. Reference[23] suggests a Pareto distribution for file sizes. Using this distribution, we compute the fractional offered load ρ' as

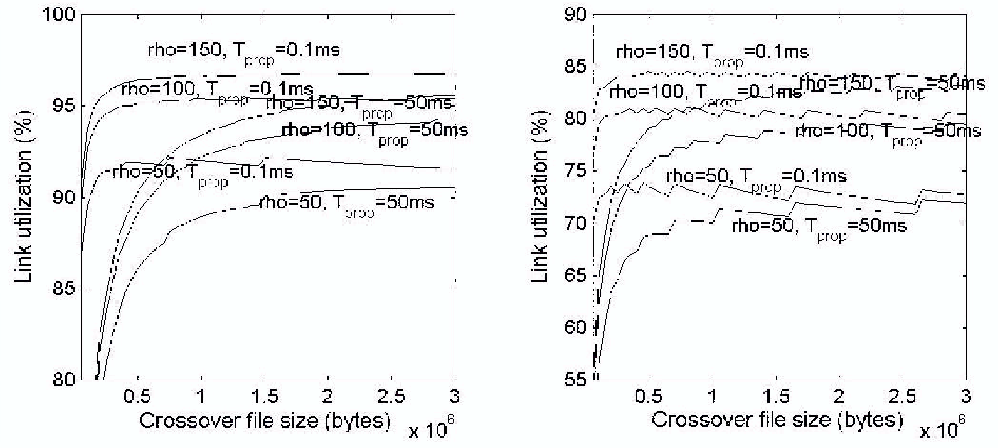
$$\rho' = \frac{\rho}{E(X)}P(X\geq\chi)E[X|(X\geq\chi)] = \frac{\rho(\alpha-1)}{\alpha k}\left(\frac{k}{\chi}\right)^{\alpha}\frac{\alpha\chi}{\alpha-1} = \rho\left(\frac{k}{\chi}\right)^{\alpha-1} \tag{8}$$

where α, the shape parameter, is 1.06 and k, the scale parameter, is 1000 bytes as computed in[23], and ρ is the total offered load. We note that the offered load decreases as χ increases, which means aggregate utilization u_a decreases for a given P_b. However, as χ increases, per-circuit utilization u_c increases.

Combining the two components of utilization, we obtain total utilization u as:

$$u = \frac{(1-P_b)\times\rho'}{m}\times\frac{(E[X|X\geq\chi\rangle])/r_c}{E[T_{setup}]+(E[X|X\geq\chi\rangle])/r_c} \tag{9}$$

We plot the total utilization u in Fig. 7 for different call blocking probabilities P_b, different values of ρ and T_{prop}. As crossover file size χ is increased, the plots show utilization increasing because of the second factor, i.e., the per-circuit utilization increases. However, the drop in the offered load and the corresponding drop in the aggregate utilization slows the increase of the total utilization, making it stable at some value below 1 or even dropping it slightly. In these plots, to



(a) P_b is 0.3 (b) P_b is 0.01

Figure 7:Plot of utilization u with a link rate of 100Mbps, $\rho_{sig}=\rho_{sp}=0.7$, $k = 20$

keep P_b constant as χ is increased, we compute m for each value of χ, using the second equation of (6). The "zigzag" pattern of the plots occurs because m has to be an integer.

From our file-transfer delay analysis, we did not have a crossover file size when T_{prop} is large (e.g., 50ms), but from the utilization analysis here we see the need to place a lower bound. Without such a lower bound, per-circuit utilization can be poor. For example, for a 100KB file transfer on a 100Mbps circuit with 4 switches on the end-to-end path, we need 50.158ms setup time and 8ms total transfer time. As a result, the per-circuit utilization is only 13.7%, which is why the 50ms plots are at a lower utilization than the 0.1ms plots in Fig. 7.

Another observation is that high utilizations are possible by operating the network at a high call blocking probability (30%). For example, with ρ=50 and T_{prop}=0.1ms, with a blocking probability of 30%, we can achieve a 90% utilization at the crossover file size of 150KB, while at a low blocking probability (1%), we can only achieve a 73% utilization for the same crossover file size (150KB). Thus, when the CHEETAH 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 CHEETAH service participants (as ρ increases), lower call-blocking probabilities can be achieved while maintaining high utilization.

These plots have been generated assuming all calls are of the long-distance variety (T_{prop} is 50ms) or all calls are in small propagation-delay environments (T_{prop} is 0.1ms). In reality, different file transfers will experience different round-trip propagation delays. This means the routing decision algorithm should have different crossover file sizes for different end-to-end paths.

2.4.5 Implementation of routing decision algorithm

The routing decision algorithm implemented at an end host could use dynamically obtained values of RTTs, P_b, P_{loss}, and link rate. 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[24]. Other options are to have network management stations track these values and respond to queries from end hosts. Since the benefit of using Ethernet/EoS circuits 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 RTT and crossover file size based on nominal operating conditions of the two networks and simplify the routing decision algorithm implemented at end hosts. This needs experimental study.

Another question is whether the CHEETAH service should be implemented from IP router to router rather than end-to-end. We note the routing decision on whether or not to attempt an Ethernet/EoS circuit is difficult to make within an IP router. This is because it is hard to extract information on the file size and RTT at a router that supports many flows, and both these parameters are important in making this decision. Other attempts have been made in the past to perform flow classification within routers and then trigger cut-through connections between routers[25]. Given the difficulties with these solutions, we realize that the routing decision is best made at the end hosts where it is easier to determine these parameters, and hence propose CHEETAH as an end-to-end service.

2.5 Transport protocol used over the Ethernet/EoS circuit

In this section, we consider the question of what transport protocol to use on these end-to-end high-speed Ethernet/EoS circuits. TCP is a poor choice for dedicated end-to-end circuits because of its slow start and congestion avoidance algorithms. Also, TCP's window-based flow control and positive-ACK based error control scheme are not well suited for dedicated end-to-end circuits. Hence we considered a number of other transport protocols, some high-speed transport protocols such as[26-27] and some OS bypass protocols[28-30]. Of these, we selected the Scheduled Transfer (ST) protocol, which is an ANSI standard[16], and is ideally suited for end-to-end circuits carrying Ethernet frames.

ST provides sufficient hooks to allow for a high-speed, **OS-bypass** implementation, a feature that is necessary to achieve true high-speed end-to-end throughput. It does this by having the sender specify a receiver memory address in the data block header, which causes the receiving NIC to simply write the received payload using Direct Memory Access (DMA) into the specified memory location. This results in a low end-host transport layer delay. ST offers flexibility in its flow control and error control schemes. For **flow control**, we propose using a rate control approach in which the circuit rate is selected to taking into account the rate at which the receiving application can process received data from memory. An alternative is to have the receiver allocate a large-enough buffer space for the entire file prior to the start of the transfer. This solution however limits the maximum size of files that can be transferred, which may anyway be necessary from a network circuit-sharing perspective. This means we limit file sizes to a Maximum File Transfer Size (MFTS) per session. For **error control**, we propose using ST's support for negative acknowledgments (NAKs) given that data blocks will be

delivered in sequence on the Ethernet/EoS circuit. Missing/errored blocks resulting from bit errors will need to be retransmitted. ST supports the selective repeat approach.

At the start of Section 2, we stated that the EoS circuit set up for the file transfer would be a unidirectional circuit (for utilization reasons). However, this raises the question of how to transport reverse-path control messages, such as NAKs and any ST-related control messages. For example ST requires a control message exchange to send the address of the receiver buffer to the sender prior to the actual data transfer. If the EoS circuit is used for this exchange, utilization will suffer. Hence we propose using a dual TCP connection set up via the IP path (through the primary NIC) for such exchanges. In other words, our transport solution is a combination of TCP on the IP path in conjunction with ST on the Ethernet/EoS circuit for the data transfer.

We also considered using the TCP connection for retransmissions. However, a simple back-of-the-envelope calculation suggests that the delay consequences of such a decision could be large. For example, consider a 1 TB file transfer. With a block size of 100KB, and an effective Bit Error Rate (BER)* of 10^{-8}, possibly 80000 out of the total 10M blocks may need retransmission. Since this is equivalent to 8GB, which is a large file in itself, we recommend using the end-to-end Ethernet/EoS circuit for retransmissions. However, when the final block is sent, the server should immediately release the circuit in order to avoid having the circuit lie idle while waiting for the transmission-ending positive ACK (in a NAK-based retransmission scheme, a final positive ACK is required as assurance to the server that the file has been successfully delivered). Any retransmissions required for the final few blocks will be sent on the TCP/IP path.

3. CONCLUSIONS

We propose improving delay performance of file transfers by using intra-network paths where possible. Specifically, we propose a service called CHEETAH in which pairs of end hosts are connected on a call-by-call basis via high-speed end-to-end Ethernet/EoS circuits. This is feasible today given the deployment of fiber to enterprises, MSPPs in enterprises and EoS technologies within these MSPPs. Seeking to achieve high utilizations, we propose setting up unidirectional EoS circuits and only holding circuits for the duration of the actual file transfers. The CHEETAH service is proposed as an add-on to basic Internet access service. The latter allows for the optical circuit-switched network to be operated in call blocking mode such that if the circuit setup is blocked an end host can fall back to the TCP/IP path. If the circuit setup is successful, there is a huge advantage in total delay especially in wide-area environments. For example, a 1TB file requires on 2.2 hours on a 1Gbps end-to-end circuit but could take more than 4 days on a TCP/IP path in a WAN environment. We analyzed the conditions under which a circuit setup should be attempted. For WAN environments and large files, it is clear that a circuit setup should be attempted. We also found that for medium-sized files (MBs), it is worthwhile making this attempt in WAN environments. In lower propagation-delay environments, if bottleneck link rates are in the order of 100Mbps, for files larger than 3.5MB, it becomes worthwhile attempting a circuit setup. For higher link rates (1Gbps), or smaller files, one should consider the loading conditions on the two paths, probability of packet loss on the TCP/IP path and call blocking probability through the circuit-switched network, before deciding whether or not to attempt the circuit setup.

ACKNOWLEDGMENTS

We thank Ramesh Karri and Haobo Wang, Polytechnic University, for their effort in implementing signaling protocols in hardware, and Prof. Tim Moors, UNSW, for his input on transport protocols. This work was supported by an NSF grant, 0087487, and by NYSTAR (The New York Agency of Science, Technology and Academic Research) through the Center for Advanced Technology in Telecommunications (CATT) at Polytechnic University.

REFERENCES

1. DOE Office Of Science High Performance Network Planning Workshop, http://doecollaboratory.pnl.gov/meetings/hpnpw/workshopdescription.pdf, August 13-15, 2002.
2. S.Floyd, "HighSpeed TCP for Large Congestion Windows," *http://www.ietf.org/internet-drafts/draft-floyd-tcp-highspeed-02.txt,* February, 2003.
3. C. Jin, D. Wei, S. Low, J. Bunn, D. H. Choe, J. C. Doyle, H. Newman, S. Ravot, S. Singh, G. Buhrmaster, R.L.A. Cottrell, and F. Paganini, "FAST Kernel: Background Theory and Experimental Results," *PFLDnet 2003*, http://datatag.web.cern.ch/datatag/pfldnet2003/, Feb. 3-4, 2003, Geneva, Switzerland.
4. T. Kelly, "Scalable TCP: Improving Performance in HighSpeed Wide Area Networks," *PFLDnet 2003*, http://datat-

*BER of optical fiber is much lower, but dust and poor connectors at fiber ends often result in BERs in the 10^{-8} range.

ag.web.cern.ch/datatag/pfldnet2003/, Feb. 3-4, 2003, Geneva, Switzerland.
5. J. Semke, J. Mahdavi, and M. Mathis, "Automatic TCP Buffer Tuning," *Proc. of ACM SIGCOMM 1998*, 28(4), October 1998.
6. W. Feng, M. Gardner, M. Fisk, and E. Weigle, "Automatic Flow-Control Adaptation for Enhancing Network Performance in Computational Grids," *Journal of Grid Computing*, 2003.
7. M. Gardner, W. Feng, and M. Fisk, "Dynamic Right-Sizing in FTP (drsFTP): An Automatic Technique for Enhancing Grid Performance," *Proc. of the IEEE Symposium on High-Performance Distributed Computing*, July 2002.
8. M. Mathis, "Raising the Internet MTU," http://www.psc.edu/~mathis/MTU/.
9. W. Feng and P. Tinnakornsrisuphapá, "The Failure of TCP in High-Performance Computational Grids," *Proc. of SC2000: High-Performance Network and Computing Conference*, Dallas, TX, Nov. 2000.
10. Bill St. Arnaud, "Proposed CA*net 4 Network Design and Research Program," Revision no. 8, April 2, 2002.
11. ITU-T Rec. G.7041, "Generic Framing Procedure (GFP)," Oct. 2001.
12. ITU-T Rec. G.707, "Network Node Interface for the Synchronous Digital Hierarchy," Oct. 2000.
13. Cisco, "Cisco ONS 15454 Optical Transport Platform," http://www.cisco.com/en/US/products/hw/optical/ps2006/ps2010/index.html.
14. Optical Internetworking Forum, "User Network Interface (UNI) 1.0 Signaling Specification," Oct. 1, 2001, http://www.oiforum.com/public/documents/OIF-UNI-01.0.pdf.
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. ANSI, "Information Technology - Scheduled Transfer Protocol (ST)," T11.1/Proj. 1245-M/Rev 4.0, Oct. 2000.
17. H. Wang, M.Veeraraghavan and R. Karri, "A hardware implementation of a signaling protocol," *Proc. of Opticomm 2002,* July 29-Aug. 2, 2002, Boston, MA.
18. 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.
19. N. Cardwell, S. Savage, and T. Anderson, "Modeling TCP Latency," *Proc. of IEEE Infocom,* Mar. 26-30, 2000, Tel-Aviv, Israel, pp. 1724-1751.
20. 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.
21. M. Allman, V. Paxson, W. Stevens, "TCP Congestion Control", *IETF RFC 2581*, Apr. 1999.
22. V. Paxson and S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Trans. Networking*, vol. 3, pp. 226-244, June 1995.
23. M. E. Crovella and A. Bestavros, "Self-similarity in World Wide Web Traffic Evidence and Possible Causes," *Proc. of the SPIE International Conference on Performance and Control of Network Systems,* Nov., 1997.
24. T. Bu, N.G. Duffield, F. Lo Presti, D. Towsley, "Network Tomography on General Topologies," *Proceedings of ACM SIGMETRICS 2002*.
25. P. Newman, G. Minshall, T. Lyon, L. Huston, "Flow Labelled IP: A Connectionless Approach to ATM," *Proc. of IEEE Infocom 1996*.
26. W. Doeringer, D. Dykeman, M. Kaiserswerth, B. W. Meister, H. Rudin, R. Williamson, "A survey of light-weight 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, 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," *Proceedings of Supercomputing'95*, San Diego, CA, 1995.