

# A Dynamic Circuits Based Wide-Area SAN Solution

H. Wang<sup>a</sup>, M. Veeraraghavan<sup>b</sup>, R. Karri<sup>a\*</sup>

<sup>a</sup>Polytechnic University, 5 Metrotech Center, Brooklyn, NY 11201;

<sup>b</sup>University of Virginia, 351 McCormick Rd, Charlottesville, VA 22904

## ABSTRACT

Fibre Channel (FC) based Storage Area Networks (SANs) have gained great success in local data centers (accounting for more than 90% of all SAN installations world wide). However, recent developments in e-commerce applications make it necessary to extend SANs across wide area. There are two approaches currently available, one is through IP networks called Storage over IP (SoIP), the other is through leased circuits (SONET/SDH circuits or WDM lightpaths). The former is low-cost and low-performance while the latter is high-performance and high-cost. We propose an approach between these two, which is based on dynamic SONET circuits. We introduce a novel circuit service called CHEETAH, and discuss how to apply the CHEETAH concept to extend Class 2 and Class 3 FC services across wide area.

**Keywords:** Dynamic circuits, GMPLS, Signaling protocols, Fibre Channel, SAN, SONET

## 1. Background and problem statement

The Storage Networking Industry Association (SNIA) defines a Storage Area Network (SAN) as “a network whose primary purpose is the transfer of data between computer systems and storage elements and among storage elements [1].” Fibre Channel (FC) is the mainstream technology used in SANs, accounting for more than 90% of all SAN installations worldwide [2]. While FC SANs are successful in local data centers, recent developments in e-commerce applications make it necessary to interconnect geographically distributed data centers.

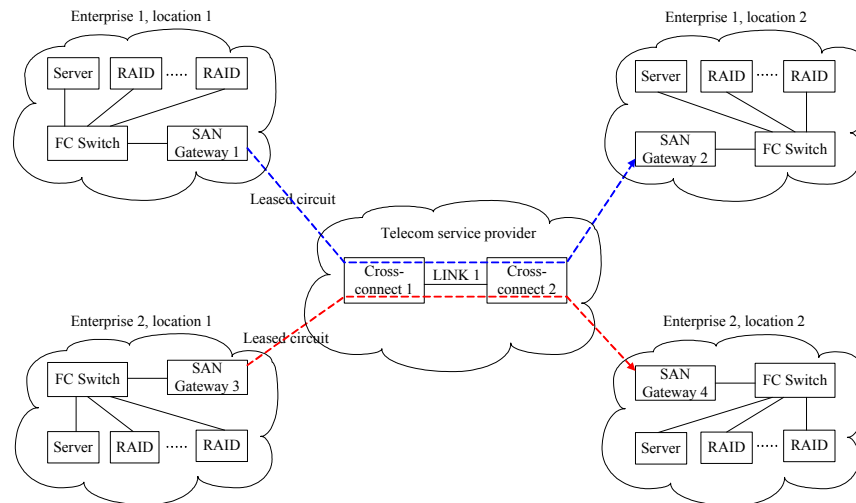


Figure 1: Storage over leased SONET circuits

A common approach for an enterprise to extend its SAN across a metro- or wide-area is to use a leased SONET<sup>\*\*</sup> circuit. Fig. 1 illustrates this configuration in which a SONET circuit is leased between two SANs of enterprise 1, and another between two SANs of enterprise 2. A SAN gateway is used to terminate the SONET circuit and serves as an interface between the SAN environment and SONET metro-/wide-area networks. An example of a SAN gateway is Akara's OUSP2000. While this product is specifically designed as a SAN gateway, an enterprise could also use a general purpose

\*haobo\_w@photon.poly.edu; phone (718) 260-4011; fax (718) 260-3906.

\*\*In this paper we use SONET networks as example, but the same concept can be applied to SDH and WDM networks.

MultiService Provisioning Platform (MSPP), such as Nortel's OPTera Metro 5100, which has interface cards for SAN protocols such as Fibre Channel.

One of the major drawbacks of the above leased SONET circuit approach is that it is expensive. For example, leasing an OC-48c circuit across a metro area has an annual cost of \$240,000 [3]. A WDM lightpath costs even more. Resource sharing on a leased circuit is limited because the only users of this circuit are the servers and storage systems at the two locations of the enterprise leasing the circuit. Often this leads to poor utilization.

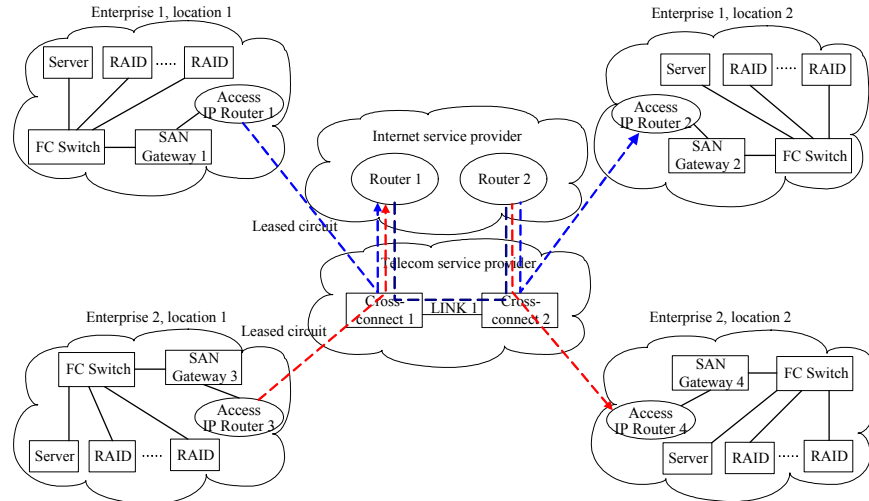


Figure 2: SoIP

A second approach is to use an IP network to interconnect geographically distributed SANs of an enterprise. The IETF IP Storage (IPS) working group is addressing issues underlying Storage over IP (SoIP) [4]. We illustrate this configuration in Fig. 2. Here, an enterprise leases a SONET circuit to an IP router of an Internet Service Provider (ISP) from each of its locations. Router-to-router circuits are leased by the ISP and are hence shared among a number of the ISP's customers. For example, bandwidth on optical link, LINK 1 of Fig. 2, is not partitioned between enterprise 1 traffic and enterprise 2 traffic as in the leased circuit configuration of Fig. 1. Instead this link is shared on a packet-by-packet basis by IP packets from enterprise 1 and enterprise 2. Because resources are shared, SoIP services can be offered at a lower cost than end-to-end leased circuits. While resource sharing can lead to lower costs, the penalty paid is the quality of service experienced by SAN users.

Resource sharing is managed in IP networks through the congestion control schemes of TCP. The SoIP proposals of [4] are all based on TCP. Unfortunately current TCP does not scale well as link bandwidths increase [5]. TCP performance is dependent on the bandwidth-delay product of an end-to-end path. Specifically TCP throughput is dependent on packet loss rate, round-trip time (RTT), and bottleneck link rate. ISPs can manage their networks to keep packet loss rates low. However, to achieve a low packet loss rate, ISPs need to limit sharing, which then adversely affects ISP revenues. Propagation delay is an important component of round-trip times. In wide-area settings, say when round-trip propagation delay is 50ms, then even with a (relatively) low bottleneck link rate of 100Mbps, we see the adverse effect of TCP congestion control algorithms. For example a 1GB file transfer will require 395.7sec on a TCP path with a bottleneck link rate of 100Mbps, round-trip propagation delay of 50ms, even when the packet loss rate is as low as  $10^{-4}$  [6] (note that this transfer would take only about 80sec without TCP rate management but at a cost to other users). The reason for this increase from 80s to 395.7s is TCP's congestion control algorithm. Not knowing the loading conditions of the network, the TCP sender starts by sending only one (or two [7]) segments and waiting for an acknowledgment (ACK). Upon receiving this ACK, it increases the congestion window (sender's buffer) and sends two (or four) segments before having to wait for an ACK. It gradually increases the congestion window and correspondingly its sending rate but in the process incurs many round-trip delays. If a packet loss occurs, the congestion window size is reduced requiring it to undergo this build-up process again. Thus packet losses impact effective throughput. The larger the RTT or the larger the packet loss rate, the worse the TCP performance.

Since wide-area scenarios (in which RTT is large) are not likely to be that common in the SAN application, we consider

the case of low propagation delays. Since TCP performance is dependent upon the bandwidth-delay product, we see that as data rates increase to 10Gbps (as planned for 10GFC [2]), even round-trip propagation delays of 0.1ms (as within a metro-area) will result in bandwidth-delay products of the same magnitude (e.g., 100Mbpsx50ms = 5 Mb; 5Mb/0.1ms = 5Gbps). For example using Mathis et al. [8]’s maximum throughput equation:

$$T = \frac{MSS}{RTT} \times \sqrt{\frac{1.5}{p}}, \tag{1}$$

and applying this with a maximum segment size ( $MSS$ ) of 1500B, round-trip time ( $RTT$ ) of 0.1ms and packet loss rate ( $p$ ) of 1%, we find that the maximum throughput is 1.5Gbps. Therefore, as link rates increase beyond this level, TCP limits the total throughput.

Many schemes are being proposed to improve the performance of TCP in high bandwidth-delay product environments [5, 9-12]. The interesting aspect of some of these proposals [e.g., 9, 12] is that they effectively sense available bandwidth and adjust the TCP sending rate to decrease the chances of packet loss because packet loss leads to rate reductions. In high-speed networks, building the rate back up to the rate before the loss (time to recover) can be significant, e.g., it takes 1 hour to recover from a packet loss on a 10Gbps path across the country [11]. Furthermore, even if TCP is enhanced to work in these environments, the packet-by-packet sharing mode emphasizes fairness over differentiated services. In other words, if enterprise 1 in the configuration of Fig. 2 wants to pay more and obtain a higher share of the shared link resources, it is more difficult to realize such a service with IP than with connection-oriented techniques. Finally it is hard to provide delay guarantees across connectionless networks, a factor that could be important for the remote mirroring application of SANs.

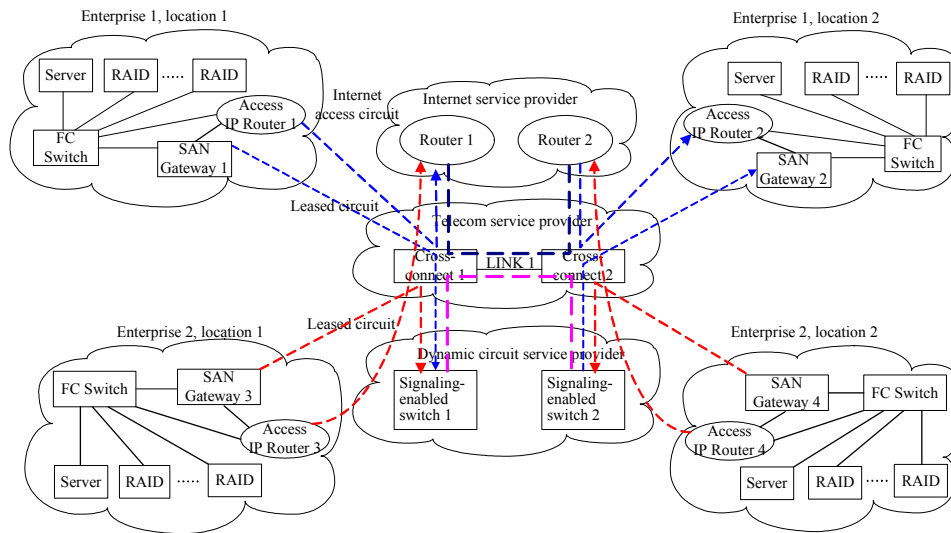


Figure 3: Storage over dynamic SONET circuits

Therefore, we propose an intermediate solution that lies between the high-quality, high-cost service of leased circuits and the low-cost, low-quality service of IP networks. This solution is to use call-by-call sharing of optical link resources. The Generalized MultiProtocol Label Switching (GMPLS) [13] working group of the IETF has defined signaling protocols for SONET circuit-switched networks to enable call-by-call resource sharing, and many vendors have implemented these protocols. We have implemented one of these signaling protocols, Resource reSerVation Protocol with Traffic Engineering (RSVP-TE) [14] in a hardware-accelerated engine for handling short-duration calls [15]. This is necessary because as optical link rates increase, the time to transfer a file from a server to a storage device decreases, e.g., 100MB file needs only 800ms on a 1Gbps circuit, which means the call setup delay overhead should also decrease.

Our solution is illustrated in Fig. 3. As in the SoIP solution, an enterprise leases bandwidth to signaling-enabled switches. The circuits leased by the “dynamic circuit service provider” between two of their signaling-enabled switches are shared on a call-by-call basis between enterprise 1 calls and enterprise 2 calls. We describe this solution in more detail in the following sections. Section 2 summarizes our prior work on a solution that we call Circuit-switched High-speed

End-to-End Transport Architecture (CHEETAH) [6]. This is a generic solution that can be used for file transfers on a combination of an end-to-end circuit and a TCP/IP path. In Section 3, we describe how we apply the CHEETAH concept to the SAN wide-area extension problem, and conclude the paper in Section 4.

## 2. A brief review of the CHEETAH concept

The CHEETAH concept, presented in [6], is based on our thinking that file transfers are an ideal candidate for high-speed end-to-end circuits. This is because unlike in streamed audio or video sessions where there is a natural maximum bandwidth limit (e.g., even with HDTV signals, we only require ~19Mbps using compression), file transfers can use any level of bandwidth. The higher the data rate, the lower the file transfer delay. Additionally files stored at one computer disk can be streamed continuously to a receiving computer disk, which means a circuit can be used without impacting utilization unlike in bursty ON-OFF audio and video sources.

Having identified file transfers as a good candidate for high-speed circuits, we focused on the question of how to deploy a solution in which end-to-end circuits are feasible in the current Internet topology. The CHEETAH solution leverages the dominance of Ethernet in LANs and SONET in MANs/WANs, using the recently developed Ethernet-over-SONET (EoS) technology. We propose using hybrid end-to-end circuits with Ethernet segments from end hosts to enterprise Multi-Service Provisioning Platforms (MSPPs), which then encapsulate these Ethernet frames on to SONET frames. Combining these concepts with GMPLS signaling, which is increasingly being implemented in SONET switches, as demonstrated in a recent signaling interoperability experiment [16], we envision creating a dynamically controlled SONET network in which end host file transfer applications request end-to-end Ethernet/EoS circuits for individual transfers. To keep utilization high, we developed the solution as a hybrid of an end-to-end TCP/IP path and an end-to-end Ethernet/EoS circuit, leveraging the fact that end hosts are already interconnected via an IP path through the Internet. For example, we propose that end-to-end circuits be used only for the actual file transfers, requiring a prompt release of the circuit when an individual transfer is complete. All initial message exchanges (such as URLs of files) and other control exchanges are done over the TCP/IP path. Furthermore, the circuit should be unidirectional for utilization reasons because transfers typically occur only in the server-to-client direction. The transport protocol proposed for these end-to-end circuits is the ANSI standard Scheduled Transfer (ST) protocol [17]. We proposed a mode of operation for this protocol that takes advantage of the two end-to-end paths, the TCP/IP path and the end-to-end circuit.

To deploy this solution, enterprises would need to (i) add a second Ethernet Network Interface Card (NIC) in those end hosts that want to participate in the CHEETAH service, (ii) connect these second NICs to ports of the enterprise MSPP for direct mapping on to equivalent EoS circuits, and (iii) lease a second WAN access circuit, one that terminates on a signaling-enabled SONET switch as shown in Fig. 3. Circuits on the leased line are shared dynamically on a call-by-call basis for file transfers executed from the end hosts with second Ethernet NICs. Using this technique, a high-speed file transfer could be carried out, for example, on an end-to-end 1Gbps rate circuit. Such a circuit can be realized by mapping GbE signals from hosts on to a virtually concatenated OC21 signal between enterprise MSPPs, which is dynamically set up through signaling-enabled switches.

In the CHEETAH architecture, where end hosts have a choice of requesting an end-to-end Ethernet/EoS circuit for a file transfer or resorting directly to the primary TCP/IP path to its correspondent end host, we analyzed the conditions under which a circuit setup should be attempted. Allowing for a circuit request to get blocked due to a lack of resources (circuit-switched networks are typically operated in call blocking mode where a call is blocked if there are no resources), we compared  $E[T_{tcp}]$ , the mean delay expected on the TCP/IP path, with  $E[T_{cheetah}]$ , the mean delay incurred if an Ethernet/EoS circuit setup is attempted prior to the file transfer.

$$E[T_{cheetah}] = (1 - P_b)(E[T_{setup}] + T_{transfer}) + P_b(E[T_{fail}] + E[T_{tcp}]) \quad (2)$$

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, which is  $F/r_c$ , where  $F$  is the file size and  $r_c$  is the rate of the circuit, and  $E[T_{fail}]$  is the mean delay incurred in a failed call setup attempt. 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.  $E[T_{tcp}]$  is dependent upon three parameters: round-trip propagation delay,  $T_{prop}$ , bottleneck link rate,  $r$ , and packet loss rate on the end-to-end path,  $P_{loss}$ . Models presented in [18, 19] were used to compute  $E[T_{tcp}]$ .

Assuming the bottleneck link rate on the TCP/IP path,  $r$ , is the same as the rate of the circuit that can be set up through

the SONET network,  $r_c$ , we consider the question of when a circuit setup should be attempted. Our first conclusion is that from a delay perspective, while a circuit setup should be attempted if  $T_{prop}$  is large (e.g., 50ms) (almost) independent of the file size, in low propagation-delay environments, it depends upon the file size. For “large” files, i.e., files larger than a “crossover file size”, a circuit setup should be attempted. For files smaller than this size, the application running on the end host should directly choose the TCP/IP path. As an example, we provide crossover file sizes for a sample point, where  $r = r_c = 100Mbps$  and  $T_{prop} = 0.1ms$  in Table 1. As evident from this table, the loading conditions on the two

**Table 1: Crossover file sizes when  $r = r_c = 100Mbps$  and  $T_{prop} = 0.1ms$**

Measure of loading on TCP/IP path	Number of switches on the circuit $k = 4$			Number of switches on the circuit $k = 20$		
	$P_b = 0.01$	$P_b = 0.1$	$P_b = 0.3$	$P_b = 0.01$	$P_b = 0.1$	$P_b = 0.3$
$P_{loss} = 0.0001$	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

paths, is a critical factor in deciding which path to choose. We also show the impact of the number of signaling-enabled switches  $k$  on the end-to-end circuit. From the above description it is clear that the size of the file is a critical parameter in choosing a path. This is also important for another reason. If a circuit setup is attempted for very small files, then the utilization of the circuit is impacted by the setup delay. 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%. We note however that this effect is much smaller in lower propagation-delay environments.

Implementation of this routing decision is greatly simplified by recent activities in the TCP community. Several tools have been implemented to estimate round-trip time ( $RTT$ ), packet loss rate ( $P_{loss}$ ), and available bandwidth on end-to-end paths through the Internet. Examples of such tools include *iperf* and *pathload*, presented in the NLANR web site [20], and *ABwE* [21]. Call blocking probability values can be provided by the dynamic SONET circuit service provider.

An important piece of our work on CHEETAH is hardware acceleration of GMPLS signaling protocols [15]. As seen in the above file transfer application, holding times of circuits can be very small; the higher the data rate, the smaller the time needed to transfer a file. Given our utilization reasons for holding circuits open only for the duration of actual file transfers, we note that call holding times can be very small, in the order of a few ms. This means call handling rates of switches have to be quite high, e.g., a switch that can support 3000 connections, with a per-connection holding time of 5ms, requires call handling rates of 600,000 calls/sec, which can be achieved in a cost-effective manner with hardware-accelerated implementations of signaling protocols.

### 3. Application of the CHEETAH concept to SANs

There are different SAN protocols. Of these, we consider the FC SAN solution here and describe how it can be used in conjunction with our CHEETAH concept of dynamically controlled SONET circuits. Other SAN solutions can be considered using similar techniques.

The FC specification [22] supports multiple classes of services. Of these, Class 1 is a dedicated connection-oriented service, ideally suited for interworking with SONET circuits. However, not only is Class 1 FC not been implemented in local-area SAN equipment such as host adapter cards, FC switches and SAN gateways [e.g., 23, 24], it is not even supported by the FC Backbone specifications [25, 26]. These specifications describe how SANs can be extended across the wide-area using ATM, SONET and IP networks, and only describe how Class 2 and Class 3 FC services are supported. Therefore, we describe below how the user-plane mappings described in these specifications work in conjunction with dynamically controlled circuits.

We note that the FC Backbone specifications [25, 26] do specify a call control function module within SAN gateways

(referred to as FC-Backbone WAN or FC-BBW for short) to support ATM Switched Virtual Circuit (SVC) service. The call control module is responsible for generating ATM SVC setup and release signaling messages. Given that the GMPLS signaling protocols are similar in spirit to ATM signaling protocols, we note that the reference model for interworking FC with dynamic SONET circuits is already in place.

To support our solution, we require no changes to the FC host adapter cards and FC switches already deployed in local-area SANs. Clearly we require a new SAN gateway implementation that triggers dynamic SONET circuit setup and release. Additionally, we require the application software running at the servers, which initiate the file transfers, to be upgraded. We explain why this is necessary in the subsections below.

The manner in which we apply the CHEETAH concept to this wide-area SAN problem is that the SAN gateways at the two physically separate locations of an enterprise (see Fig. 3) are connected via two networks: (i) an IP network (through the Internet access service of the enterprise) and (ii) a dynamically controlled SONET circuit network. In the example shown in Fig. 3, SAN gateway 1 within the enterprise 1/location 1 network can communicate with SAN gateway 2 within the enterprise 1/location 2 network via an IP path that traverses from the SAN gateways to the access IP routers in the two locations, then through the leased circuits that interconnect access IP routers with ISP IP routers, which are interconnected by the ISP's IP network. The second path is for SAN gateway 1 to request a SONET circuit to SAN gateway 2 through the circuits leased from the SAN gateways to the signaling-enabled SONET switches of the dynamic SONET circuit service provider's network.

We address the following questions in the subsections below:

1. Under what conditions should a circuit setup be attempted and how is the setup triggered?
2. How is flow control and error control handled on these hybrid FC/SONET circuit paths?

### 3.1 Under what conditions should a circuit setup be attempted and how is the setup triggered?

Unlike in the leased-line scenario (illustrated in Fig. 1), with CHEETAH are concerned about SONET circuit utilization. Therefore, we need to equip the SAN gateway with intelligence to decide when to attempt a SONET circuit setup and when to resort directly to the TCP/IP path to the far-end SAN gateway. As shown from the results presented in Section 2, if the round-trip time between the two SAN gateways is high or the file size is large, the SAN gateway should attempt a circuit setup. Information on the former is relatively easy to obtain through TCP's RTT measurements, but unfortunately, if we use the SAN applications as they are currently implemented, the SAN gateway will have no knowledge of the file size of a given transfer. One alternative to the file size based routing decision is to use an available-bandwidth based decision. For example, the SAN gateway can determine the throughput capability of the far-end device through information in Management Information Bases (MIBs) [27] or by storing and retrieving this information from the Internet Storage Name Service (iSNS) databases [28]. The SAN gateway can estimate the achievable throughput on the TCP/IP path using (1) and plugging in measured values of round-trip time ( $RTT$ ) and packet loss rate ( $P_{Loss}$ ). As mentioned earlier, there are several tools to obtain estimates of  $RTT$  and  $P_{Loss}$ . Alternatively, other tools such as *pathrate* and *ABwE* can be used to directly obtain measures of available bandwidth. If the available bandwidth on the TCP/IP path can satisfy the throughput requirement of the end device, the TCP/IP path is used. Otherwise, the SAN gateway can set up a SONET circuit to achieve the higher rate of transfer.

If the SAN gateway uses this relative available bandwidth information to set up a connection for an FC exchange without any knowledge of the file size, it could be a waste if the exchange only has a few data frames. Interworking solutions such as TCP Switching [29], which were proposed to interwork IP networks with connection-oriented networks such as MPLS or ATM have an answer for this question that does not require knowledge of file sizes. For example, solutions propose counting the number of packets arriving on a given TCP flow within a certain duration, and if this number exceeds some threshold, then triggering a circuit setup. However, these techniques are not ideal to determine whether a circuit setup is worthwhile or not. For example if the entire file transfer is complete as this decision to set up a circuit is made, the effort of setting up the circuit is wasted. Hence, we propose upgrading application software at the servers to send the size of the file to be transferred prior to initiating an actual data transfer. Control software at the SAN gateway can receive this information and based on its knowledge of loading conditions on the two paths and bottleneck link rates, it can decide whether or not it is worthwhile setting up a circuit for a given file transfer.

Because Class 2 and Class 3 services are by nature connectionless, the server will not wait for the SONET circuit to be established before it starts sending data. For Class 2 service, the server may send out certain number of frames without

acknowledgement (the number depends on the available credit); for Class 3 service, the server just blindly sends out frames without waiting for acknowledgement. In order to avoid possible frame loss, the first few frames can be routed by the SAN gateway through TCP/IP path. After the SONET circuit is successfully established, the SAN gateway can switch the traffic to the circuit. Again, this solution only favors bulk data transfers.

The holding time of a circuit should be just long enough to transfer a single file. Whether a complete file is sent as one sequence of data frames within an exchange or multiple sequences within an exchange depends upon how the application software is implemented. A data frame, which is the basic unit of communication in FC, consists of 36 bytes of overhead and up to 2112 bytes of payload, for a total maximum frame size of 2148 bytes. A sequence is a group of related frames transmitted unidirectionally from one node (port) to another. Each sequence can have at most 65K data frames (SEQ\_CNT, the field used to identify a data frame in a sequence, has 16 bits). Given that the maximum size of a data frame is 2148 bytes, the total length of a sequence is no more than 130MB. Such size of data is not justifiable to set up a 1Gbps SONET circuit. Exchange is the highest level communication unit in FC, consists of a series of sequences. Each exchange can have at most 256 sequences (8 bits SEQ\_ID), which means an exchange can involve at most 33.2GB data transfer. It is reasonable to set up a SONET circuit for an exchange.

### 3.2 How is flow control and error control handled on these hybrid FC/circuit paths?

First consider Class 3. The FC specification for this class [22] uses R\_RDY primitive signals for buffer-to-buffer flow-control and has no error control signals since it is an unacknowledged service. Annex A of the FC BB specification states that primitive signals are stripped at the interface boundaries. This means the R\_RDY flow control signals are stripped at the SAN gateways. This is effectively the “spoofing” concept described in [30] where the SAN gateway acts as a proxy for the remote end and either generates or consume such signals.

The FC BB and BB2 specifications [25, 26] propose the use of a Selective-Repeat (SR) or Simple Flow Control (SFC) mechanism for the FC-BBW frames carried between the two SAN gateways. These solutions are effectively “buffer-to-buffer” mechanisms for the wide-area “link” between the SAN gateways. First consider error control.

The SR error control solution works well with our unidirectional SONET circuits because it primarily requires the receiving SAN gateway (FC-BBW) to issue SR\_SREJ messages specifying the sequence numbers of those frames that need to be retransmitted. As with our CHEETAH solution for using the ST transport protocol, we propose that these SR\_SREJ messages be carried on the TCP connection between the two SAN gateways because the SONET circuit is unidirectional for utilization reasons. Thus the two FC-BBW modules communicate using both paths.

We considered the question of whether to use the TCP path or the SONET circuit for retransmissions. 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 SONET 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.

The flow control mechanism in the SR scheme is an ON-OFF solution using SR\_RR (Receive Ready) and SR\_RNR (Receiver Not Ready) supervisory signals. For utilization reasons, we do not want the SONET circuit to lie idle; this means we should avoid having the receiving SAN gateway send an SR\_RNR signal to the sending SAN gateway. We can achieve this by effectively implementing a rate-based flow-control scheme. Given that SONET supports virtual concatenation, the rate of the circuit should be determined by the rate at which the disk can receive data. This information can be obtained from MIBs [27] or iSNS databases [28]. As described in Section 3.1, we require the application software to be involved in the setting up of a SONET circuit (by passing file size information to the SAN gateway). To implement a rate-based flow control scheme we will need the application software at the server to regulate its sending rate based on the end-to-end bottleneck rate (which could be the receiving end device or a link in the network). Otherwise the buffers at the sending side SAN gateway could overflow. The bottleneck rate can be determined during the circuit setup phase, with the SAN gateway on the receiving end device side extracting information on the end device receive rate from appropriate

---

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

databases, and returning this information in a signaling message. Bandwidth negotiation may be needed if the requested throughput does not match the bottleneck link rate. Bandwidth negotiation is supported by GMPLS signaling protocols. When the circuit setup is complete, the application at the server sets its sending rate and the SAN gateways set their SR related parameters, T1, T2, K2, and K to match the agreed bandwidth, so that the circuit between SAN gateways is always filled. In other words, our goal is to avoid the production of SR\_RNR and keep the SR\_RR signals flowing on the TCP/IP path. We propose that the SAN gateways route the SR control signals through the TCP/IP path because the SONET circuit is unidirectional.

For Class 2 services, in addition to the buffer-to-buffer flow control mechanisms, there is an end-to-end credit based flow control mechanism. If the rate value is set correctly, the end-to-end credit mechanism should also run smoothly without causing gaps in the transmission on the SONET circuit.

#### 4. Conclusions

We propose to interconnect geographically distributed SANs through dynamic SONET circuits. It is a solution that lies between the low-cost low-performance SoIP solution and the high-cost high-performance leased SONET solution. When a certain criteria is satisfied (RTT or file size), a dynamic SONET circuit is attempted. If successful, FC frames are carried on this dynamically setup circuit. If the attempt fails, the SAN gateways route FC frames on the default TCP/IP path. We showed how this combination circuit/TCP solution, which we call a CHEETAH, can be applied to support Class 2 and Class 3 FC services on wide-area SONET networks.

#### Acknowledgments

We thank Xuan Zheng, University of Virginia, and Hojun Lee, Polytechnic University, for providing us the detailed models of CHEETAH, and Wu Feng and Mark Gardner, Los Alamos National Laboratory for their contributions on the CHEETAH concept.

#### References

- [1] Storage Networking Industry Association, <http://www.snia.org>.
- [2] Fibre Channel Association, <http://www.fibrechannel.org>.
- [3] Bandwidth Market, Ltd., <http://www.bandwidthmarket.com>.
- [4] M. Rajagopal, E. Rodriguez, R. Weber, "Fibre Channel Over TCP/IP," IETF Internet Draft, <http://www.ietf.org/internet-drafts/draft-ietf-ips-fcovertcpip-12.txt>, Aug. 2002.
- [5] S. Floyd, "HighSpeed TCP for Large Congestion Windows," IETF Internet Draft, <http://www.ietf.org/internet-drafts/draft-ietf-tsvwg-highspeed-01.txt>, February, 2003.
- [6] M. Veeraraghavan, X. Zheng, H. Lee, M. Gardner, W. Feng, "CHEETAH: Circuit-switched High-speed End-to-End Transport Architecture," *Proc. of Opticomm 2003*, Oct.13-17, 2003, Dallas, TX.
- [7] M. Allman, V. Paxson, W. Stevens, "TCP Congestion Control", IETF RFC 2581, Apr. 1999.
- [8] M. Matthis, J. Semke, J. Mahdavi, T. Ott, "The Macroscopic Behavior of the TCP congestion avoidance algorithm," *Computer Communication Review*, volume 27, number3, July 1997.
- [9] 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.
- [10] T. Kelly, "Scalable TCP: Improving Performance in HighSpeed Wide Area Networks," *PFLDnet 2003*, <http://datatag.web.cern.ch/datatag/pfldnet2003/>, Feb. 3-4, 2003, Geneva, Switzerland.
- [11] B. Tierney, T. Dunnigan, M. Matthis, "Net100: Developing network-aware operating systems," <http://www.net100.org>.
- [12] L. Brakmo, L. Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, Oct. 1995, pp. 1465-1480.
- [13] E. Mannie, "Generalized Multi-Protocol Label Switching Architecture," IETF Internet Draft, <http://www.ietf.org/internet-drafts/draft-ietf-ccamp-gmpls-architecture-07.txt>, May 2003.
- [14] L. Berger, "GMPLS Signaling Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions," IETF RFC3473, January 2003.
- [15] H. Wang, M. Veeraraghavan and R. Karri, "A Hardware-Accelerated Implementation of the RSVP-TE Signaling Protocol," submitted for publication.

- [16] G. Beranano, T. Dimicelli, N. Larkin, D. Pendarakis, B. Schultz, A. Wang, "Achieving UNI and NNI Interoperability," OIF Forum, [http://www.oiforum.com/public/documents/OFC03\\_WP.pdf](http://www.oiforum.com/public/documents/OFC03_WP.pdf).
- [17] "Information Technology - Scheduled Transfer Protocol (ST)," ANSI NCITS 337-2000, Oct. 2000.
- [18] 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.
- [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] NLANR - National Laboratory for Applied Network Research, <http://www.nlanr.net>.
- [21] J. Navratil, R. L. Cottrell, "ABwE: A Practical Approach to Available Bandwidth Estimation," *PAM2003*, the Passive and Active Measurement Workshop, April 6-8, 2003.
- [22] "Fibre Channel Framing and Signaling (FC-FS)," ANSI INCITS.373:2003, July 7, 2003 (<http://www.t11.org/t11/docreg.nsf/ldl/fc-fs>).
- [23] QLogic Corporation, <http://www.qlogic.com>.
- [24] Brocade Communications Systems, Inc., <http://www.brocade.com>.
- [25] "Fibre Channel Backbone (FC-BB)," ANSI NCITS.342:200x, March 5, 2001 (<http://www.t11.org/t11/docreg.nsf/ldl/fc-bb>).
- [26] "Fibre Channel Backbone (FC-BB-2)," T11 Project 1466-D, Feb. 4, 2003, (<http://www.t11.org/t11/docreg.nsf/ldl/fc-bb-2>).
- [27] K. McCloghrie, "Fibre Channel Management MIB," IETF Internet Draft, <http://www.ietf.org/internet-drafts/draft-ietf-ips-fcmgmt-mib-04.txt>, Feb. 2003.
- [28] J. Tseng, K. Gibbons, F. Travostina, C. D. Laney, J. Souza, "Internet Storage Name Service (iSNS)," IETF Internet Draft, <http://www.ietf.org/internet-drafts/draft-ietf-ips-isns-20.txt>.
- [29] P. Molinero-Fernandez, N. Mckeown, "TCP Switching: Exposing Circuits to IP," *IEEE Micro*, vol. 22, issue 1, Jan./Feb. 2002, page 82-89.
- [30] Cisco Systems, "SAN Extension over SONET/SDH Networks," [http://www.cisco.com/warp/public/cc/pd/olpl/metro/on15454/prodlit/seosn\\_qp.htm](http://www.cisco.com/warp/public/cc/pd/olpl/metro/on15454/prodlit/seosn_qp.htm).