BROCADE

WHITE PAPER

Brocade Open Systems Tape Pipelining, FastWrite, and Storage-Optimized TCP

The pipelining and optimization technologies on the Brocade Extension products improve throughput and mitigate the negative effects of latency when transporting storage traffic over distance. This paper details the operation and advantages of these technologies in extended RDR and tape applications.

Table of Contents

Introduction

The Brocade® 7840 Extension Switch, Brocade 7800 Extension Switch, and Brocade FX8-24 Extension Blade for the Brocade DCX® 8510 and DCX Backbones are purpose-built products for Fibre Channel over Internet Protocol (FCIP).

FCIP (RFC 3821) is a method of transporting FC frames between two geographically distant locations using IP. The two most common applications that require FCIP are RDR (Remote Data Replication) and tape. RDR is the replication of data between two storage arrays. There are a number of challenges when transporting storage (SCSI [Small Computer Systems Interface]). In some cases, storage is sensitive to latency, and throughput is a big concern. Brocade Open Systems Tape Pipelining (OSTP) and Brocade FastWrite are mechanisms that improve throughput and mitigate the negative effects of delay.

Brocade OSTP applies to writing to tape over a WAN connection. Brocade FastWrite applies to Remote Data Replication (RDR) between two storage subsystems. Tape is serial in nature, meaning that data is steadily streamed byte by byte, one block after another, onto tape, from the perspective of the host writing the file. Disk data, on the other hand, tends to be bursty and random in nature. Disk data can be written anywhere on the disk at any time and is not predictable. Because of these differences, tape and disk are handled differently in terms of extension acceleration techniques.

Transmission Control Protocol (TCP) is vitally important to storage extension using FCIP. Brocade has developed an optimized TCP stack, called Storage Optimized TCP (SO-TCP), specifically for use with storage. Brocade SO-TCP takes into consideration assumptions about storage RDR and tape that standard TCP stacks that are typical across the Internet do not—and cannot—take into account.

These technologies are described in detail in this paper.

Brocade Open Systems Tape Pipelining (OSTP)

Brocade OSTP is an acceleration technology for streaming data in such a way as to maintain optimal utilization of the IP WAN. OSTP is a Brocade innovation. The nature of tape traffic without an acceleration mechanism results in periods of idle link time and becomes more inefficient as link propagation delay increases. The inefficiency problem is further exacerbated when files to be written or read are relatively small, forcing a large number of small exchanges. Below are descriptions of write and read Brocade OSTP operations, as well as diagrams outlining the transactions between an initiator and a target.

Brocade OSTP Writes

The host initiator is zoned with and logged into (via PLOGI, or Port Login) a target device. When the host sends a write command, the Brocade 7840, 7800, or FX8-24 intercepts that command and behaves as if it is the target, becoming a virtual target that responds with an immediate transfer ready. The Brocade 7840/7800/FX8-24 buffers incoming data and starts sending it immediately over the WAN. The data is sent as fast as possible, limited only by the bandwidth of the link, committed rate, or ARL (Adaptive Rate Limiting). The write command is allowed to continue on to the remote target. Immediately following that command is the write data, which was enabled by the virtual target's reply of a transfer ready. After the remote target receives the command, it responds with its own transfer ready. The remote Brocade 7840/7800/FX8-24 intercepts that transfer ready and acts as a virtual initiator, then it starts forwarding the arriving data that is coming in over the WAN.

The initiator is on a high-speed FC network (typically 4, 8, or 16 gigabit connections) and may complete sending the data to the local Brocade 7840/7800/FX8-24 before the data has finished being streamed over the WAN. Nonetheless, the local Brocade 7840/7800/FX8-24 returns an "OK" FC Protocol (FCP) response. While the buffers are still transmitting the data over the link, the initiator sends the next write command, and the process repeats on the local side until the end of the entire job, when it is time to write the Filemark. This process maintains a balance of data in the remote router's buffer, permitting a constant stream of data to arrive at the tape device, and preventing "shoe shining."

On the target side, the transfer ready indicates the allowable quantity of data that can be received for that sequence. The quantity may be less than what the initiator intended for the exchange. The transfer ready on the initiator side, replied by the virtual target, indicates the entire quantity of data advertised in the initiator's write command. It is not broken into smaller multiple sequences for the sake of speed and optimization. The virtual target's transfer ready for the entire amount of data does not have to match the transfer ready with which the remote tape device responds, which may be for smaller chunks of the data (specifically, the amount that it is capable of accepting at that time). The virtual initiator parses out the arriving data to the target per the transfer readies it receives from the tape device. This may result in additional write commands and transfer readies on the tape side, as compared to the initiator side, which is inconsequential within the high-speed low-latency data center, but which limits performance if across the WAN. Buffering on the remote side helps facilitate this process.

The command to write the Filemark is not intercepted by the Brocade 7840/7800/FX8-24, and it passes unfettered from end to end. When the Filemark is complete, the target responds with a status. A status of "OK" indicates to the initiator that the job is complete.

As shown in Figure 1, tape communications without Brocade OSTP involve a large number of trips across the WAN connection. Compare this to the far fewer number of trips across the WAN when using Brocade OSTP, as shown in Figure 2.

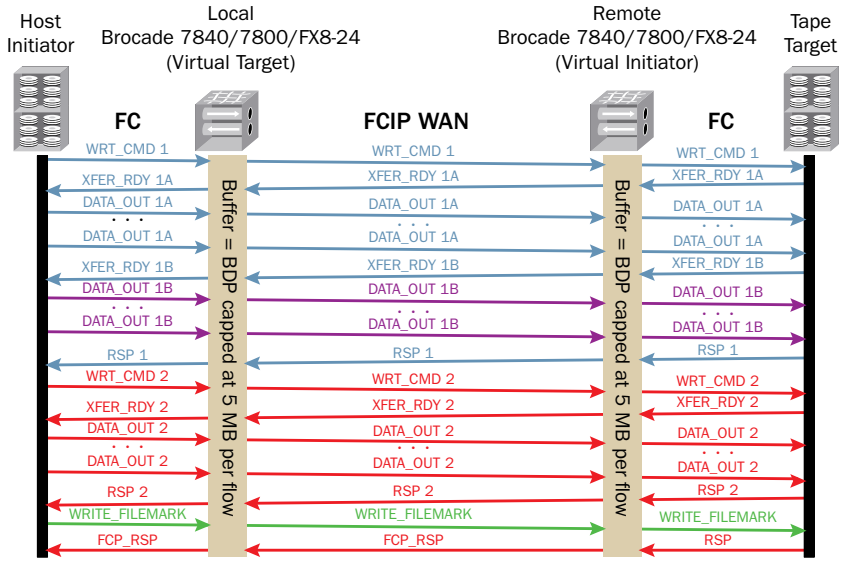

Figure 1: Tape communications without Brocade OSTP.

The diagram below illustrates how Brocade OSTP accelerates a tape write operation.

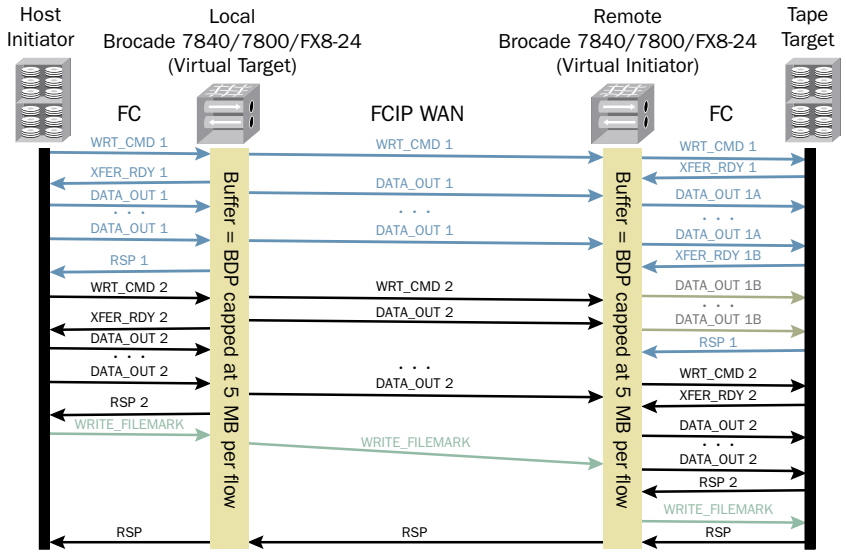

Figure 2: Tape write communications using Brocade OSTP.

Brocade OSTP is not like disk-to-disk RDR. Tape is an asynchronous application, and buffering has definite benefits. The local Brocade 7840/7800/FX8-24 observes an incoming write command (FCP_WRT_CMD) and immediately proxies a reply (FCP_XFR_RDY) to start sending the data. The data is locally buffered and a proxy response (FCP_RSP) returned upon completion, and the initiator immediately issues the next write command. These steps repeat until the allocated buffers are full and Buffer-to-Buffer

credits (BB credits) are withheld, creating flow control and preventing buffer overruns. It is via this process of local acknowledgement and buffering that tape data is accelerated significantly over high-latency connections. As soon as data enters the local buffer, it is streamed as fast as possible to the remote side. The egress out to the WAN is typically slower, due to limited WAN bandwidth as compared to the ingress from the fabric, which alone is a good reason to buffer the data with or without optimization. In addition, it is desirable to maintain a constant stream of data to the tape device, and buffering facilitates this as well.

Brocade OSTP Reads

There is also a method for accelerating tape reads across links with appreciable latency. Traditional tape reads are simpler to transact, because they do not have to solicit a transfer ready. Nonetheless, without Brocade OSTP, they also suffer from the effects of link propagation delay. The read initiator sends a read command to the remote side, and the remote side sends the data as soon as it is retrieved from tape. At first impression, it seems there are no problems, because the data is sent with only a single round trip. Where the real problem occurs is between the reads, and the problem is worsened when the reads are small and numerous. When one read command ends, the next read command is not immediately initiated. This is a problem, because the next read usually consist of the next blocks on the tape, so the tape stops when it could have continued. While the response from the remote side does immediately follow the data out, the next read command is not sent until that response is obtained—and then the read command itself has to travel across the WAN. Effectively, this is an added Round-Trip Time (RTT) to initiating the next read; all the while, the WAN connection is idle.

To eliminate this unnecessary round trip, the remote-side Brocade 7840/7800/FX8-24 has to issue a spoofed read command immediately after seeing the command "OK" response being sent. This happens locally, starting the next data out within microseconds (µsec), which keeps the tape moving. The remote Brocade 7840/7800/FX8-24 can predict the next block of data to read, because tape is sequential. The 7840/7800/FX8-24 merely issues a read command for the next block. It is appropriate to keep reading blocks until there is an indication to not continue, such as: End of File, End of Tape, no more read commands from the initiator (anything pre-read in the buffers is discarded), or an error condition. The blocks are read and pipelined over to the initiator's side of the WAN. As the initiator issues the real read commands for the data, the data already exists in the local buffers of the Brocade 7840/7800/FX8-24 or is in the process of being sent across the WAN. The local Brocade 7840/7800/FX8-24 starts releasing that data to the initiator based on the specifics of the read commands it receives.

Figure 3 illustrates tape read communications without Brocade OSTP. What you should notice in the diagram is that before the next FCP_READ_CMD is issued, a response (FCP_RSP) from the prior read command had first to be received. Waiting for those responses causes delay, which accumulates.

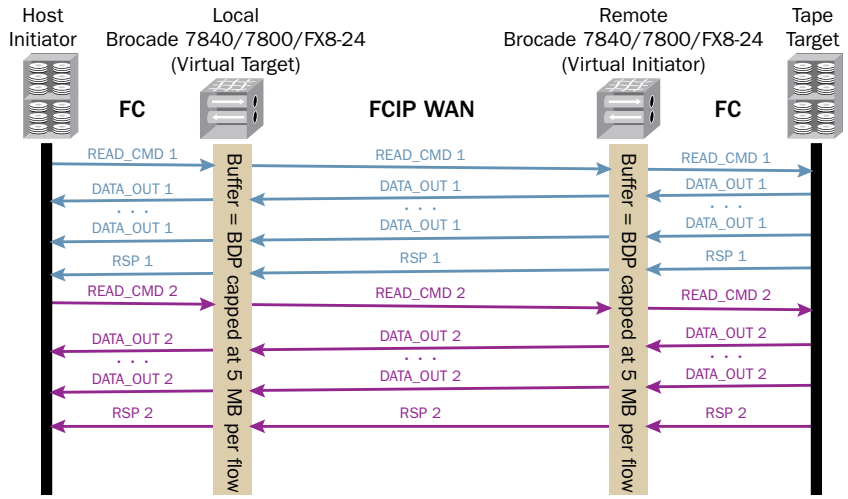

Figure 3: Tape read communications without Brocade OSTP.

In Figure 4, tape communications using Brocade OSTP is shown. The critical difference is that when the remote-side Brocade 7840/7800/FX8-24 (relative to the initiator) intercepts the response to the previous read command, it immediately sends the next read command. This process starts the next blocks of data moving in a more efficient manner and is key to fully utilizing WAN bandwidth.

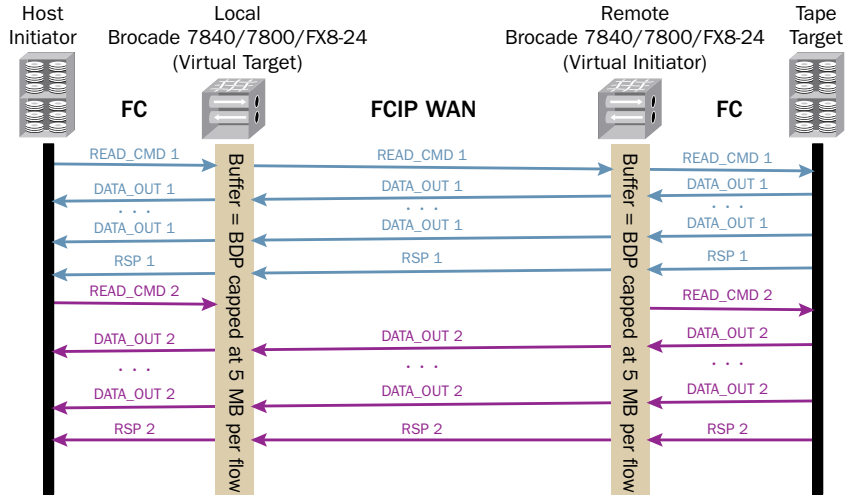

Figure 4: Tape read communications with Brocade OSTP.

Brocade OSTP General Operational Specifications

- Brocade OSTP can be used with Brocade Extension Trunking. Extension Trunking, in essence, provides a single logical IP tunnel comprised of multiple circuits. Circuits are individual IP connections within the trunk, each with its own unique source and destination IP address. A single-circuit tunnel is referred to as a tunnel. A tunnel with multiple circuits is referred to as a trunk, simply because multiple circuits are being trunked together. A tunnel or trunk carrying FCIP packets is a single Inter-Switch Link (ISL). A trunk is logically a single tunnel connecting a single VE_Port on each side. There is no return path ambiguity for OSTP when the two data centers are connected together using a single VE_Port. Extension Trunking is applicable to circuits that use GbE, 10 GbE, or 40 GbE interfaces. You can trunk circuits across multiple Ethernet interfaces.
- IPsec (IP Security) is supported with Brocade OSTP. It has no bandwidth limitations and has negligible added latency (approximately 5 µsec).
- The Brocade 7840 switch, 7800 switch, and FX8-24 blade have different capacities to handle OSTP flows. A flow is defined as the active nexus of Initiator/Target/LUN (ITL) for either a read or a write in either direction. There are two Data Paths for FCIP (DPF, also referred to as an FCIP engine) for the Brocade 7840. A total of sixteen 1/10 GbE and two 40 GbE ports are shared by the two Brocade 7840 DPFs. There is a single DPF for the Brocade 7800, which operates the 6 GbE interfaces. There are two DPFs for the Brocade FX8-24. Two 10 GbE ports are shared by the two Brocade FX8-24 DPFs. In addition, ten 1 GbE ports are dedicated to one of the Brocade FX8-24 DPFs. It is because of these separate and equal DPF engines that the Brocade FX8-24 has the capacity to run each 10 GbE interface at full line rate. Each DPF on the Brocade FX8-24 blade has its own flow scalability. Flows apply to both Brocade OSTP and Brocade FastWrite.

For example, if a tape virtualization device has 10 hosts writing to one drive each, that consumes 10 flows. If a host is writing to 5 LUNs on a storage array that has 1000 LUNs configured, it consumes 5 flows. If writing to those LUNs completes, the 5 flows are released, because they are no longer active. TUR (Test Unit Ready), Mode Sense, Log Sense, and similar control traffic that is frequently sent to tape devices is not deemed "active," is not apportioned as a flow, and does not use resources.

Each flow receives 2 MB of buffer space as a minimum, out of a total of 1.2 GB of buffer space available and a maximum of 5 MB for tape. Beyond 5 MB it becomes difficult to manage the tape's early warning that the end of tape is near, because the Brocade 7840/7800/FX8-24 has buffered more data than can be written to tape. This causes data loss, therefore it is prevented.

The actual amount of buffer space is determined by the BDP (Bandwidth Delay Product = BW × RTT).

For example, devices communicating over GbE × 20 ms get 2.5 MB of buffer, which is ample for maintaining continuous pipelining across the WAN.

Another example, devices communicating over GbE × 200 ms = 5 MB, because it is capped at 5 MB to prevent data loss due to end-of-tape situations.

The maximum number of concurrent flows that can be accommodated per DPF at full buffer size is:

Flows per DPF	Brocade 7840 with compression	Brocade 7840 without compression	Brocade 7800 with compression	Brocade 7800 without compression	Brocade FX8-24 with compression	Brocade FX8-24 without compression
Brocade FastWrite	60,000	60,000	10,000	10,000	10,000	10,000
Brocade OSTP	4,500	1,500	225	138	600 (1200 per blade)	150 (300 per blade)

The number of flows described in the table above is accommodated with no restriction on buffer size; however, if additional flows are needed, they are accommodated up to 10,000 flows per DPF by reducing buffer space on some flows.

- If the Brocade 7840/7800/FX8-24 runs out of resources, which is a very unlikely event and a difficult scenario to create, read and write operations continue to function—without acceleration—until resources become available.
- Only class 3 traffic is accelerated between tape devices. FICON® traffic is not accelerated using Brocade OSTP, and it is not positively or negatively affected by OSTP. Both FICON and open systems tape can coexist; such architectures are supported by Brocade and Brocade OEMs (IBM and Oracle). FICON tape is accelerated using Brocade FICON Emulation.
- Brocade OSTP in environments with more than one tunnel or trunk (which implies that more than one VE_Port is being used to communicate to the same remote data center) requires Traffic Isolation Zones (TIZs) or Virtual Fabric (VF) Logical Switches (LSs) to maintain a single and consistent return path. A tunnel with more than one circuit is called a trunk. Do not confuse multiple circuits belonging to a single trunk that uses a single VE_Port with multiple individual tunnels that each have their own VE_Port. Traffic isolation is not required for a single trunk containing multiple circuits within it, because logically it is a single point-to-point connection between two opposing VE_Ports. A trunk has one Fabric Shortest Path First (FSPF) cost associated with it. The individual circuits do not have FSPF costs.
- Multiple non-equal cost paths between host and tape devices forces traffic to use the lowest cost path per routing rules associated with FSPF and Brocade Fabric OS® (FOS).

Example Architectures

All diagrams are showing only the "A" fabric side. There is a mirror image of these for the "B" side. Best practice is to always have an "A" and "B" side.

Trunk

One of the most popular architectures is to simply connect a local and remote tape fabric across distance. In this case, four separate links are combined into a single logical Inter-Switch Link (ISL) called a trunk. Protocol optimization such as Brocade OSTP, Brocade FastWrite, and FICON Emulation are supported across multiple equal cost links only when using Extension Trunking. Refer to Figure 5.

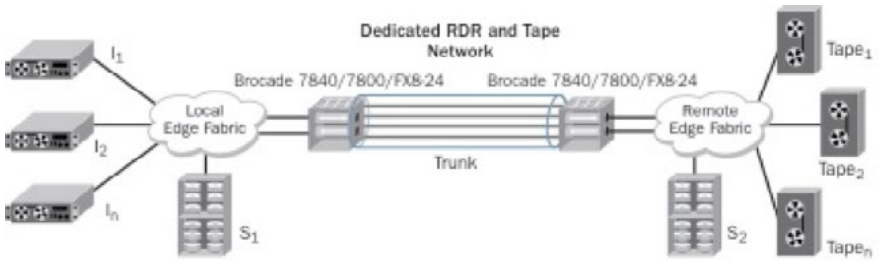

Figure 5: The Trunk—a single logical link.

Traffic Isolation

Some mainframe managers prefer to keep FICON traffic separated from the open systems traffic for a variety of reasons. The Brocade 7840/7800/FX8-24 can accommodate an architecture in which these two types of traffic can be segregated using TIZ and support protocol acceleration. It should be noted that combining FICON acceleration, Brocade FastWrite, and Brocade OSTP over the same trunk is fully supported by Brocade and the OEMs. In the example below, without TIZ, protocol acceleration cannot be used due to return path ambiguity, which breaks the optimization state machine if the wrong return path is inadvertently taken.

As shown in Figure 6, there are two isolated paths: one for mainframe tape and one for RDR and OSTP. This is a high-bandwidth solution using 10 GbE interfaces on Brocade FX8-24 blades and Brocade MLXe Series 10 GbE switches/routers. The Brocade MLXe switches/routers are particularly suited for this application, because they are feature-rich with lower latency and higher capacity compared to other Ethernet switches on the market.

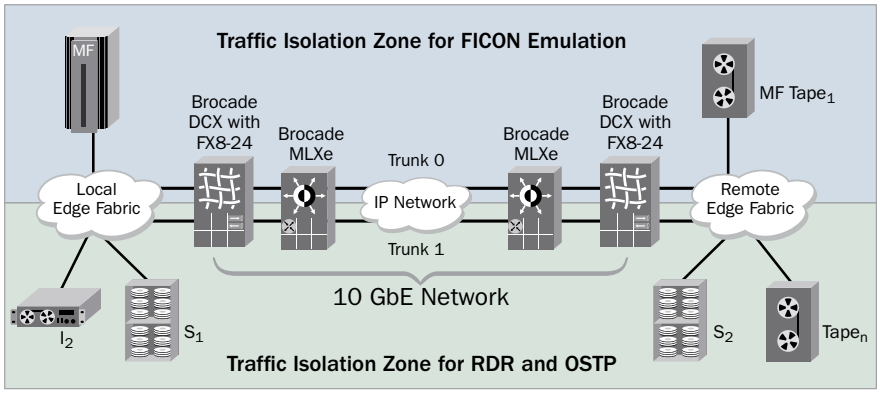

Figure 6: Two tunnel architecture with protocol acceleration and TIZ.

Site to Multisite

The site to multisite example shows a Brocade FX8-24 blade in the main data center with a 10 GbE connection to the core LAN switch. There are multiple connections within that local 10 GbE connection, which are routed to their respective remote sites. At the remote sites, devices may connect directly to the Brocade 7840/7800/FX8-24, or the 7840/7800/FX8-24 may be attached to an edge fabric. Disk RDR applications, that may or may not be using Brocade FastWrite, can coexist with tape flows over the same tunnels and interfaces that are processing OSTP. Additional separate tunnels are not required when performing extension for RDR and tape.

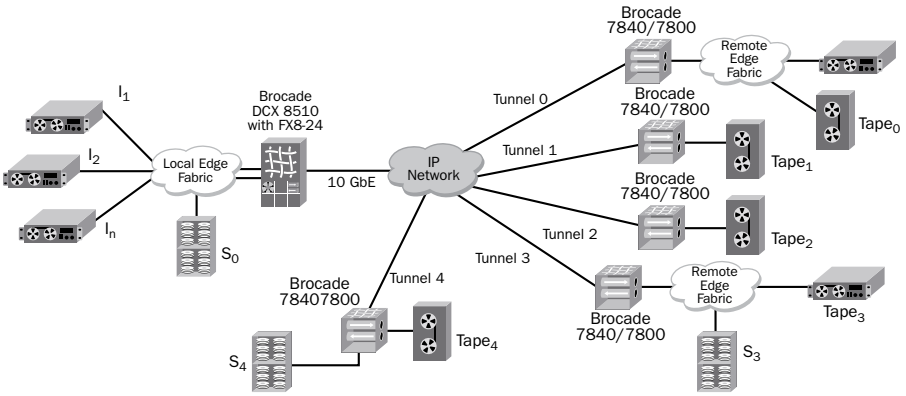

Figure 7: High Availability architecture for tape and RDR.

It is also important to maintain High Availability (HA) when replicating data or backing up to tape. Consider a data loss event. Data that has not been replicated or backed up poses a serious liability. For enterprises that cannot tolerate data loss, an HA infrastructure is built. Figure 8 provides an illustration of a common HA deployment.

Storage, tape media servers and tape virtualization are connected to redundant and physically isolated "A" and "B" fabrics. There is an "air gap" between the fabrics that stretches from storage to hosts. The fabrics are not connected together on the FC side anywhere along the path. This is important, because as a best practice FC uses fabric services that must remain completely separated. This includes any common connectivity through FC Routing (FCR) and Virtual Fabrics (VF). It is not the same on the LAN/IP side, because there are no fabric services, so connecting into a common (although redundant and resilient) LAN is an acceptable practice. The IP network here may have one large shared WAN connection or two parallel connections. The two autonomous connections may be from different service providers, garnering higher availability, although at a higher price. The need is evaluated on a case by case basis, depending on each company's aversion to risk and the value of the data being protected.

The HA concept here is that if one of the parallel paths A or B faults, storage and tape backups continue. The degree of bandwidth remaining during a fault depends on the architecture of the LAN/IP cloud. The use of Brocade Adaptive Rate Limiting (ARL) in the Brocade 7840/7800/FX8-24 products, as well as ample connectivity to the edge fabric and between the Brocade Extension product Ethernet interfaces to the IP/LAN network, maintains the maximum available bandwidth for the situation.

As for Brocade OSTP in an HA environment, the architecture is the same as the Extension Trunking scenario described above. The FCIP connections between the Brocade FX8-24 blade into the Brocade MLXe switch contain multiple circuits that are trunked together to form a single logical ISL between edge fabric A on each side. The same is true for edge fabric B on each side.

The Brocade DCX 8510 with a Brocade FX8-24 blade installed should be part of the core in a core-edge fabric design. Applications like RDR should connect directly to the core, as there are no compelling reasons to connect the dedicated ports to the edge. The storage ports that perform RDR are often dedicated to that purpose. For example, storage arrays that run EMC SRDF and HDS Universal Replicator have dedicated ports for those applications, in which no other host communications take place. Those

dedicated ports should be directly connected to the Brocade 7840/7800/FX8-24 for Extension. This is a best practice and not a requirement. If the ports are not dedicated, as is the case with some of the smaller arrays (EMC CLARiiON and HP EVA) and a virtualization appliance such as IBM SVC, which have to share ports due to a limited number of ports, connectivity through the core is still best practice, as storage is typically connected directly to the core and communicated out to the hosts via the edge. Tape is typically connected to channel extenders via the edge fabrics.

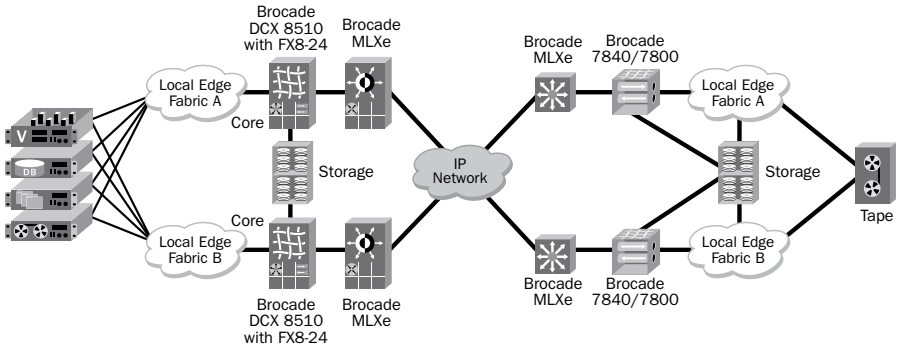

Figure 8: High Availability architecture.

Are multiple I/Os combined into one large I/O?

No, there is only one command per I/O on the host side. On the target side, the command may be broken into smaller parts but not combined into a single or fewer larger I/Os. The smaller I/Os are a function of the destination tape device and not the Brocade 7840/7800/FX8-24.

From the perspective of the target, located on the remote side relative to the initiator, the initiator appears to be local. Each I/O emanates from the Brocade 7840/7800/FX8-24 virtual initiator, as directed by each of the target's FCP_XFR_RDY responses. The buffers in the tape device may not be able to accept all the data at once and asks for smaller blocks of data to be sent. The tape device does this by indicating how much data can be accepted in each FCP_XFR_RDY response. The Brocade 7840/7800/FX8-24 acts as a proxy initiator and responds appropriately to the tape device by sending data in acceptable amounts to the drive, while FC buffer-to-buffer flow control is also used at the data link level to prevent buffer overruns.

Brocade OSTP mitigates latency by using local responses to the initiator, keeping the data flowing. Write responses from the remote target tape drive are accepted by the local Brocade 7840/7800/FX8-24. Those responses are not forwarded to the other side, where the initiator is. What is forwarded to the initiator from the target are responses to the Filemark (FCP_FM) command. When the host receives a Filemark response, it knows all the data has actually been written to tape. If this response never occurs, or an error is indicated, a recovery process must occur, or the job must be restarted.

How large can the I/O between Filemarks be?

There is no answer to this question other than, "It depends on the application." Before a Filemark is written, any size job may have occurred. Typically, this could range from kilobytes to gigabytes. If an unrecoverable error occurs between the end of one Filemark and the end of the next Filemark, all the data has to be rewritten to tape to ensure data integrity. Usually, these errors are beyond the FCP level, for example, a problem with the tape system.

Is there a local acknowledgement and data transfer with a final acknowledgement at the end of the data transfer?

Yes, there is a local acknowledgement, known as responses (FCP_RSP), after a write command (FCP_WRT_CMD) is issued from the initiator. This command/response volley continues until the apportioned buffer space in the Brocade 7840/7800/FX8-24 is full. The buffers are set to be large enough to fill the link's BDP (Bandwidth Delay Product) up to 5 MB. BDP is the amount of data in flight required to fill a link round trip. The data is streamed over the connection utilizing all the available bandwidth. When the host has finished sending the job to be archived, it issues a Filemark command (FCP_FM). Upon response back from the target that the Filemark has been successfully written, the initiator is confident that all the data has been committed to tape. The Filemark is the final acknowledgement after a large number of individual I/Os (exchanges).

There are a couple of error level tiers within Brocade OSTP. Some errors reflect current I/O problems, and others are deferred errors indicating a problem has occurred within the tape storage process, which may result in the compromise of the archive's integrity. Even though earlier an I/O may have responded successfully, a deferred I/O error is an alert that a subsequent problem has occurred, and that ultimately the job has failed.

There is a normal amount of buffering that tape systems perform to maintain a constant flow of data to the drive. This reduces problems like "shoe shining," which is a stopping and starting of the tape that requires the tape to be stopped, reversed, and re-queued before starting again. Of course, this takes time and wears on the drive and the tape itself. Adequate streaming may also eliminate or reduce the need to multiplex multiple backups onto the same tape. Multiplexing interleaves data from multiple servers onto a single tape and is often used in an effort to maintain sufficient data rates to prevent shoe shining. Multiplexing backups onto the same tape complicates management and increases data recovery time.

If an error occurs during the time an I/O is written to the buffers, it is considered a "current" error. Later, as the buffers are written to tape, if an error occurs, it generates a "deferred" error. A Filemark is used to establish that the data has been committed to tape. If the response to a Filemark comes back without error, this is an indication that the overall process was successful. In the event of a deferred error, the data has to be rewritten from the point of the previous Filemark.

TCP Error Recovery

There are a number of errors that can be recovered from at both the TCP and FCP levels. First, consider TCP, which is a connection-oriented, lossless protocol that always delivers data in the same order as it was transmitted. The unit of data transfer for TCP is called a segment. TCP sends a stream of data, and a segment is a piece of that stream, with each byte tracked by sequence numbers. TCP mechanisms can recover segments lost in transit, including those discarded by bit errors and those dropped during congestion within the IP network. Brocade SO-TCP uses SACK (Selective Acknowledgement) to optimize retransmission of missing segments.

TCP is also proficient at recovering from out-of-order segments and duplicate segments. This allows a suitable network to deliver data quickly and reliably. A suitable network does not have more than 0.1 percent transmission defects and typically has much less, even though the Brocade 7840/7800/FX8-24 is engineered to transmit with link defects as high as 1.0 percent, even at 10 GbE speeds. TCP concepts are well known and not proprietary to Brocade. Brocade SO-TCP, however, while using standard TCP mechanisms, has been designed to be a more aggressive TCP stack that gets data going quicker, is more impervious to congestion events and lost segments, and recovers faster than traditional TCP stacks. What makes this possible is that Brocade SO-TCP makes different assumptions that are congruent with enterprise WAN networks.

FCP Error Recovery

Underlying the Extension TCP/IP network are one or more FC ISLs. FC ISLs pass through the IP network. Consider error recovery on the FCP level. The examples below refer to initiator and target communications. From the perspective of FCP errors, recovery handling is local to the host or tape device; therefore, the host initiator is speaking to a virtual target on the Brocade 7840/7800/FX8-24, and a virtual initiator on the remote Brocade 7840/7800/FX8-24 is speaking to the end target. (Please note: Only the terms initiator and target are used in the explanations of the examples below. In addition, the examples do not include the TCP/IP connection between the Ethernet interfaces on the Brocade 7840/7800/FX8-24 for the reasons discussed above in the TCP Error Recovery section, and the TCP/IP connection should be deemed a reliable ISL.)

This paper discusses five error conditions:

- Write Command Lost
- Transfer Ready Lost
- Response Lost
- Data Out Lost
- Filemark Lost

Before getting into the error recovery examples, it is necessary to cover two FCP concepts: REC (Read Exchange Concise) and SRR (Sequence Retransmission Request).

REC is an FC-4 link service command. REC is used to query the other end of a link to obtain concise exchange status information. A link, in this case, is between an initiator and the local Brocade 7840/7800/FX8-24 (virtual target), or between the remote Brocade 7840/7800/FX8-24 (virtual initiator) and a target. Exchanges are specified by the nexus of: S_ID (Source ID), Ox_ID (Exchange ID), and Rx_ID (Responder ID) in the request header. The receiving port can locate the Exchange Status Block (ESB) using the S_ID and Ox_ID. If the ESB does not exist, a Link Service Reject (LS_RJT) is returned. A valid REC reply includes information valuable to recovering from various error conditions, particularly if the exchange is still open or has been completed.

There is a timer associated with REC called the REC_TOV. This timer is not configurable. Resources can be unnecessarily expended if REC commands are used too often. The REC_TOV is started each time a command or data is sent. If the transactions are not progressing as expected, and the REC_TOV expires, the initiator of the exchange can send a REC to determine the status of the exchange. Depending on that status, an action may be taken.

SRR is an FC-4 link service command. An exchange can have one or more sequences within it. A sequence that communicates a command can be retransmitted if an error condition causes that command to be lost in transit. The exchange is specified by Ox_ID and Rx_ID in the SRR header. The sequence is specified by the Routing Control (R_CTL) and the relative offset of the data. Relative offset is used when data has to be re-sent, such that the new sequence generated contains the precise portion of data required to recover.

Example of FCP command loss and recovery on the local and remote sides

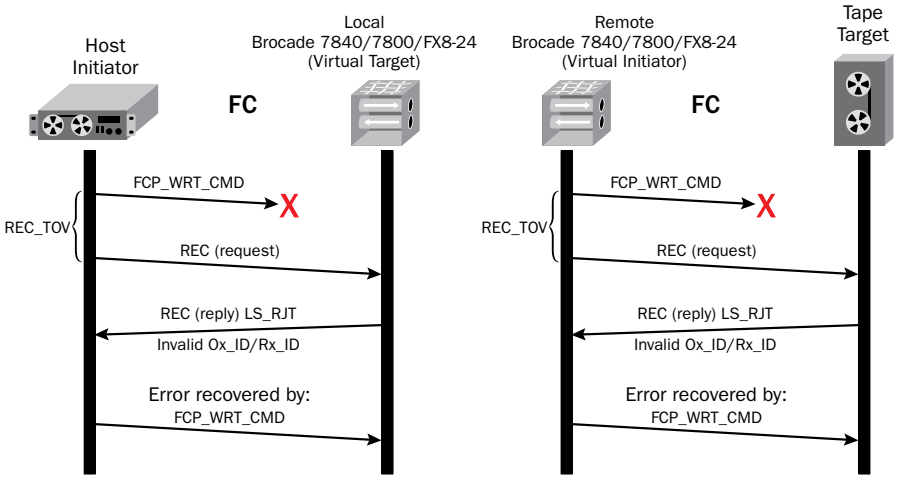

Figure 9: FCP write command lost error recovery using REC.

In this example, the FCP_WRT_CMD (Write Command) being issued by the host, or the same command proxied by the Brocade 7840/7800/FX8-24, is lost due to an error in transit, indicated by the red X in Figure 9.

After REC_TOV expires, the initiator sends a REC request. The proxy target tries to match the request to an exchange and determines that no ESB exists for the S_ID and Ox_ID specified. Because no ESB exists, the response back to the initiator is an LS_RJT, and the reason code given is Ox_ID/Rx_ID invalid. This tells the initiator that the write command never made it to the target, and a replacement write command is generated. Command execution continues normally from this point.

Example of FCP XFR_RDY loss and recovery on the local and remote sides

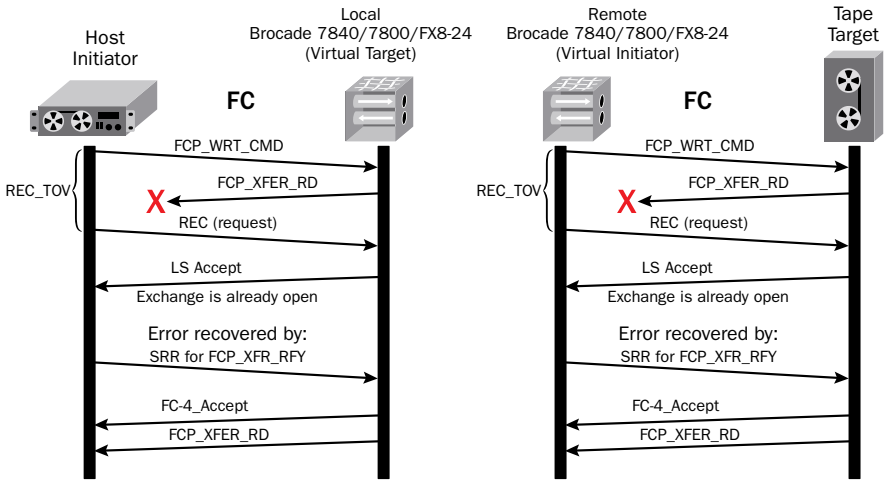

Figure 10: FCP Transfer Ready error recovery using REC.

In the example shown in Figure 10, the FCP_XFR_RDY (Transfer Ready) response to the write command is lost in transit due to an error condition. After a minimum period of time defined by REC_TOV, the initiator sends a REC to the target to obtain information on the status of the exchange in progress. An LS_ACC (Link Service Accept) is returned, indicating that the exchange is currently open and the initiator has Sequence Initiative. When the initiator has Sequence Initiative, it means that the target is waiting for the initiator. After the target sends a FCP_XFR_RDY, it is waiting for data from the initiator.

To recover from this situation, the target must re-send the FCP_XFR_RDY. To start this process, the initiator must request the sequence containing the FCP_XFR_RDY to be retransmitted. This is accomplished via the SRR FC-4 link service command.

The target responds to the SRR request with an FC-4 Accept, followed by a retransmission of the FCP_XFR_RDY. Command execution continues normally from this point.

Example of FCP_RSP loss and recovery on the local and remote sides

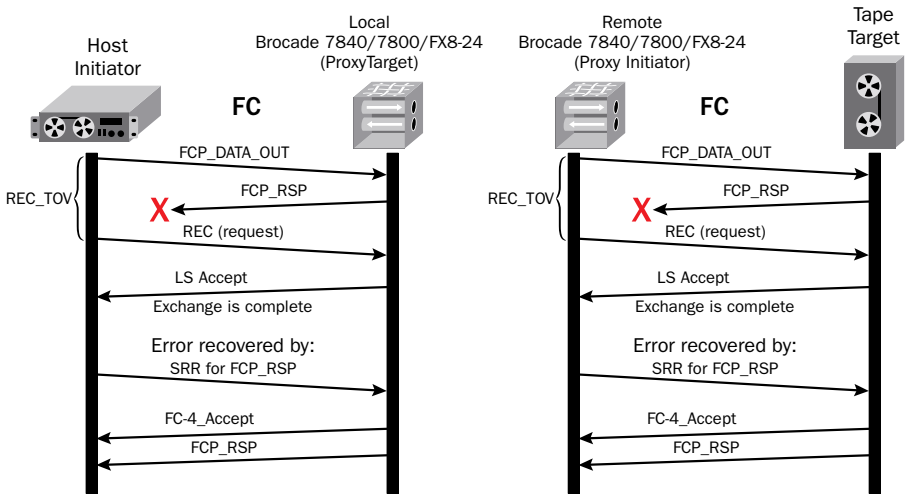

Figure 11: FCP Response error recovery using REC.

In this example, the FCP_RSP (Response) has been lost in transit, as shown in Figure 11. The initiator is expecting the response soon after the last bit of data is sent. At the time the data send is initiated, the REC_TOV is started, and after it expires an REC can be sent to the target to inquire about the exchange in progress. The target accepts the REC and returns an LS_ACC, indicating that the exchange was completed. This tells the initiator that the FCP_RSP was lost in transit, vs. some portion of data not arriving at the target.

To recover from this situation, the initiator must request from the target the retransmission of the sequence containing the FCP_RSP. The target accepts the SRR request and replies with a FC-4 Accept, followed by the FCP_RSP.

Example of FCP_DATA_OUT loss and recovery on the local and remote sides

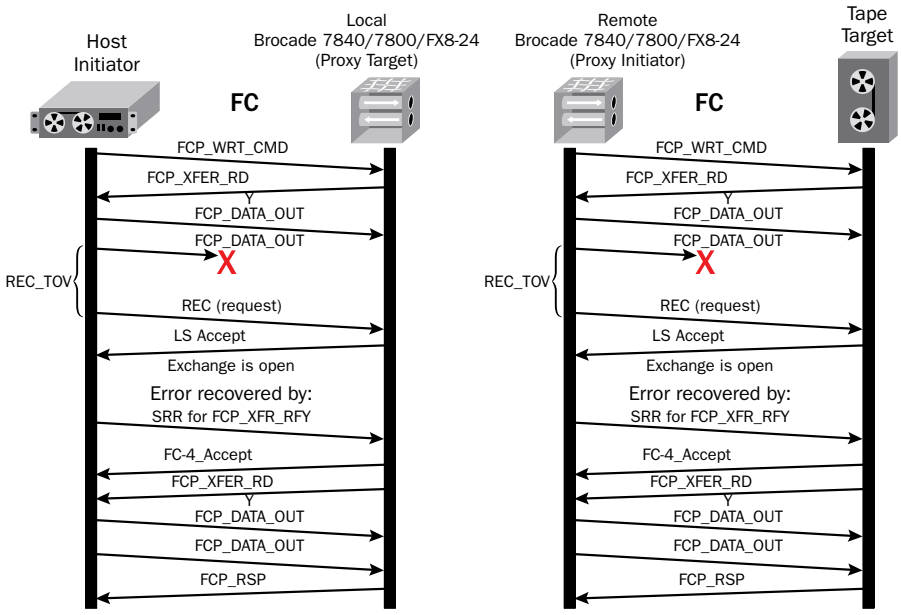

Figure 12: FCP Data Out error recovery using REC.

In the example shown in Figure 12, data has started to be sent, yet along the way there is an error of some type, and data is lost in transit. REC_TOV was started when the last data sequence was sent. After REC_TOV times out, a REC is sent to the target. The LS_ACC response from the target indicates that the exchange is still open and the initiator is holding the Sequence Initiative. The initiator is holding the Sequence Initiative because the target never received all the data, and the target is expecting more.

Recovery from this situation requires that the entire sequence of data be retransmitted. The initiator generates an SRR requesting a FCP_XFR_RDY. When the initiator receives that FCP_XFR_RDY, it starts transmitting that block of data again. Command execution continues normally from this point.

Example of FCP_FM (Filemark) loss and recovery on the local and remote sides

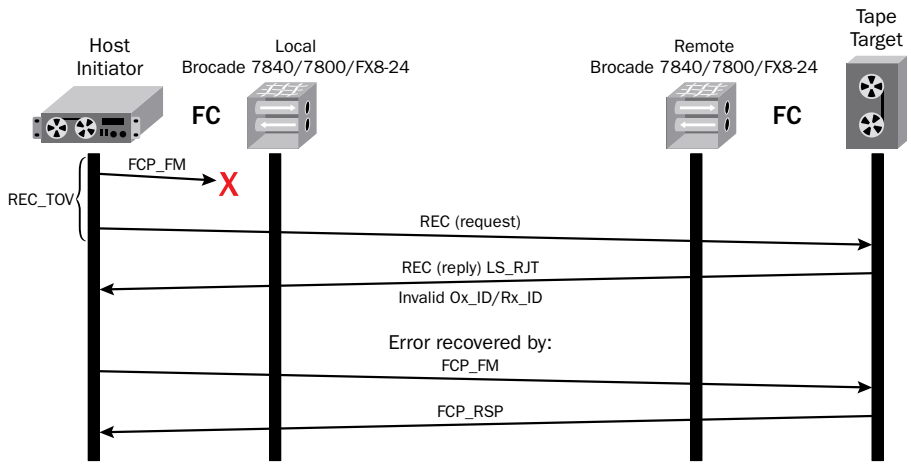

Figure 13: FCP Filemark error recovery using REC.

In this last example, the FCP_FM (Filemark) is lost in transit, as shown in Figure 13. The example shows that the Filemark was lost between the initiator and the local router; however, this example applies to the Filemark being lost anywhere along the path from the initiator to the final target (not the virtual target).

In this case, the REC is forwarded to the end target. If the target has not received the Filemark, it has no ESB for the exchange initiated from the host containing the Filemark. If it does not have the ESB, the reply is an LS_RJT with the reason of invalid Ox_ID/Rx_ID. This indicates to the host that the original FCP_FM never arrived, and the host recovers by generating a replacement FCP_FM. The final FCP_RSP indicates that the new FCP_FM succeeded.

Brocade FastWrite

Brocade FastWrite, another Brocade innovation developed in 2001, is an algorithm that reduces the number of round trips required to complete a SCSI write operation. Brocade FastWrite can maintain throughput levels over links that have significant latency. The RDR (Remote Data Replication) application still experiences latency; however, reduced throughput due to that latency is minimized for asynchronous applications, and response time is cut in half for synchronous applications.

Typical SCSI behavior without Brocade FastWrite is shown in Figure 14. There are two steps to a standards-based SCSI write. First, the write command is sent across the WAN to the target. The first round trip is essentially asking transfer permission from the storage array. The target responds with an acceptance (FCP_XFR_RDY). The initiator waits until it receives a response from the target before starting the second step, sending the actual data (FCP_DATA_OUT). Within the confines of a data center, where the latencies are measured in microseconds (µsec), there are no issues. However, across a WAN where the latencies are measured in milliseconds (ms), there can be negative ramifications. One µsec is 1 millionth of a second, and one ms is 1 thousandth of a second.

This simplex communication over a WAN link, with any appreciable latency, aggregates the time it takes to complete the entire write procedure. In the example diagram below, there are two round trips required to complete the write; therefore, the write takes at least 2 × RTT, plus the time for the data out. Multiple write commands can be

outstanding at any one time; however, every command must wait for a response before moving on. Multiple outstanding commands can help solve the throughput and link utilization problem.

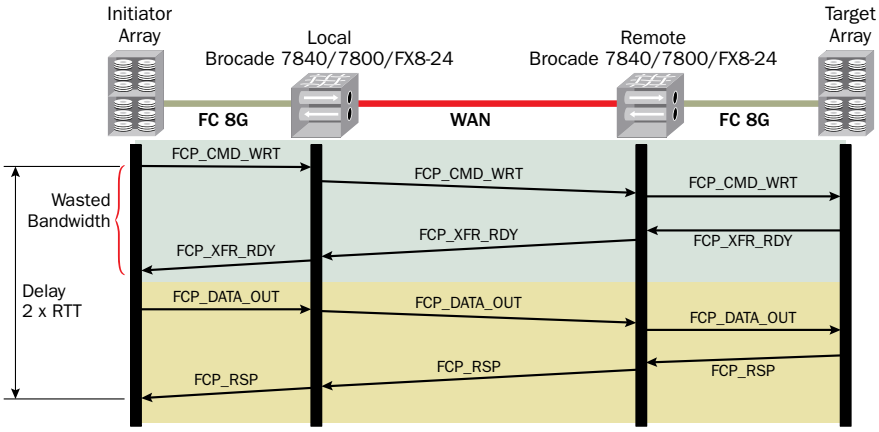

Figure 14: SCSI Write without Brocade FastWrite.

With the Brocade FastWrite algorithm, the local SAN router intercepts the originating write command and responds immediately, requesting the initiator to send the entire data set, as shown in Figure 15. This happens in terms of a couple of microseconds. The initiator starts to send the data. BBCs are tightly coupled to the TCP cwnd (cwnd is the congestion window). As segments are sent over the IP connection, R_RDYs are sent back to FC. The buffer space for TCP equals the BDP, plus an additional amount to compensate for links with up to 1 percent packet loss and jitter. The Brocade 7840/7800/FX8-24 has a continuous supply of data in its buffers to completely fill the WAN driving optimized throughput. Note: If a link has more than 1 percent packet loss, there are serious IP network problems that must be dealt with prior to a successful implementation of RDR and tape.

The Brocade 7840/7800/FX8-24 sends data across the link until the committed bandwidth is consumed. The receiving router acts on behalf of the initiator and opens a write exchange with the target over the local fabric or direct connection. This technology often allows a write to complete in a single round trip, speeding up the process tremendously and mitigating link latency by 50 percent.

There is no possibility of undetected data corruption or data loss with Brocade FastWrite, because the final response (FCP_RSP) is never spoofed, intercepted, or altered in any way. It is this final response that the receiving device sends to indicate that the entire data set has been successfully received and committed. The local router does not generate the final response in an effort to expedite the process, nor does it need to. If any single FC frame is corrupted or lost along the way, the target detects the condition and does not send the final response. If the final response is not received within a certain amount of time, the write sequence times out (REC_TOV) and is re-sent. In any case, the host initiator knows that the write was unsuccessful and recovers accordingly.

IT professionals have a choice with synchronous applications (RDR/S): links can be twice the distance, or response times can be halved. Asynchronous applications (RDR/A) benefit by enabling fewer but larger I/Os to fully utilize the connection, as well as less

idle time on the link before data out starts. This is especially important if the Bandwidth Delay Product (BDP) is big enough to exceed the storage subsystem's ability to generate enough concurrent I/Os to fill the pipe.

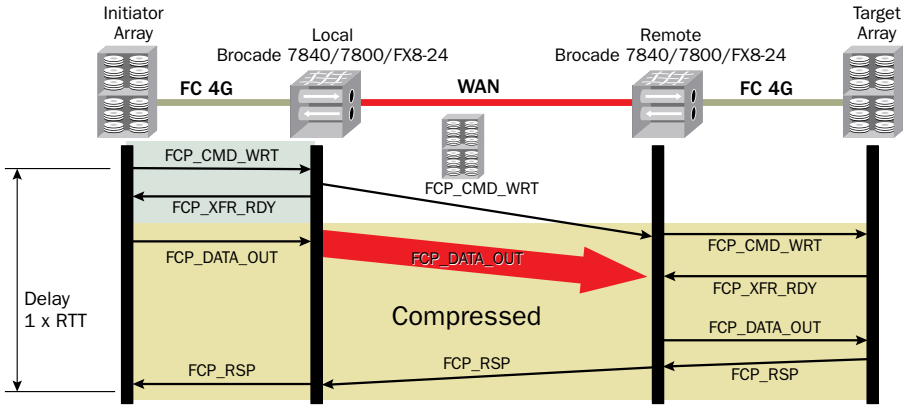

Figure 15: SCSI Write with Brocade FastWrite.

Buffer Capacity on the Brocade 7840/7800/FX8-24 for Brocade FastWrite

Buffer capacity for Brocade FastWrite is equal to the BDP, with some elasticity for jitter and already-sent R_RDY, and with the simultaneous possibility of TCP experiencing congestion events due to lost segments. Retransmitted segments are not purged until acknowledged. However, BB credits have already been issued to the storage array, resulting in imminent data arrival. Therefore, the buffer must be somewhat elastic and accommodating. Enough data is accepted by the channel extender to meet the BDP, at which point BB credits (R_RDY) are withheld, stopping the inflow from the storage array. After this point, it is a matter of repeating this process of sending a segment and R_RDY, which equates to sending data out on the wire and getting more data to refill the buffer. Data is sent as the TCP cwnd slides forward.

Generally, queuing of data for RDR is not preferred. For synchronous applications (RDR/S), the less buffering the better, because response time increases as data sits idle. For asynchronous applications (RDR/A), it is better to leave the data on the storage array and reel it off using BB credits. When TCP segments are acknowledged and the cwnd slides, more data is sent from the Brocade 7840/7800/FX8-24 to the WAN, and an R_RDY is returned to get more data from the storage array.

RDR applications do not have a mechanism like the Filemark, which is characteristic of tape systems. By leaving the data on the storage array, the RDR application running on the array maintains responsibility for the data and its integrity. From the perspective of the storage array, the amount of time the data is in transit is minimized. The response (FCP_RSP) returned for a write command (FCP_WRT_CMD) and its data (FCP_DATA_OUT) ensures to the upper layer protocol that the data has been successfully received on the remote side. Final responses are never spoofed or modified, and they come from the end target device itself, not spoofed by the Brocade 7840/7800/FX8-24. That is why buffering data beyond the completion of the current write and into the next write is not done with disk-to-disk RDR. This should not be confused with multiple outstanding write exchanges, which are independent of each other and are each waiting for their own response.

RDR applications include: EMC SRDF, EMC MirrorView, EMC SANcopy, IBM GlobalMirror, IBM MetroMirror, IBM SVC, HDS TrueCopy, HDS Universal Replicator (HUR), and HP Continuous Access. HDS Universal Replicator cannot take advantage of Brocade FastWrite, because the remote DR side reads (pulls) data from the local production side. SCSI read commands use only a single round trip; therefore, there is no initial round trip to eliminate. Simply, the local side communicates with the remote side indicating that data is available to be read. IBM GlobalMirror/MetroMirror/SVC and HP CA for EVA do not utilize Brocade FastWrite, because of proprietary SCSI methods of sending data unsolicited. Essentially, they do not send a write command, and they wait for a transfer ready before sending their data.

Brocade FastWrite Operational Specifications

The operational specifications are the same as those described in the Brocade OSTP operational specifications earlier in this paper. There are no required settings or adjustments for latency or bandwidth; everything is automatic. There is nothing to configure for Brocade FastWrite or OSTP other than enabling it on the trunk or tunnel. FastWrite is not configured on a per-circuit basis.

Brocade Storage Optimized TCP (SO-TCP)

Brocade developed SO-TCP specifically for use with storage, taking into consideration assumptions about storage RDR and tape. The Brocade enhancements to the TCP stack are discussed below. Each trunk circuit is transported using TCP and there is one TCP session for each QoS within a circuit.

Maximum TCP cwnd (Congestion Window) Size

TCP was designed to operate reliably over almost any transmission medium regardless of transmission rate, delay, corruption, duplication, or reordering of segments. Production TCP implementations can adapt to a wide range of conditions, including bandwidth rates, round-trip times, and packet loss. For example, bandwidth can range from dialup at 56 kbps to 100 Gbps. Round-trip delays (RTT) can range from a few microseconds (µsec) to hundreds of seconds.

The Brocade 7840/7800/FX8-24 Gigabit Ethernet (GbE) interface TCP implementation has a maximum RTT of 200 ms before experiencing droop. Droop is a condition in which utilization drops off because the link can no longer be completely filled. The 10 GbE interface on the Brocade FX8-24 blade can operate at a distance of

100 ms RTT without droop—a very long-distance OC-192 by today's standards.

TCP Performance

Modern TCP networks are often constructed from fiber optics. A primary characteristic of fiber networks is their high speeds, which are orders of magnitude greater than in the past. TCP has had to adapt to these newer networks by addressing issues of performance and reliability.

TCP performance depends not upon the transfer rate itself, but rather upon the product of the transfer rate and the round-trip delay (RTT). This Bandwidth Delay Product (BDP) measures the amount of data it takes to "fill the pipe" and is the minimum buffer space required at both the sender and receiver to obtain maximum throughput. The TCP cwnd grows to the BDP, allowing that quantity of unacknowledged segments to be outstanding. When the window is sufficiently large, enough data is permitted to fill the pipe from sender to receiver, and back. Data continues to be transmitted non-stop until acknowledgements for the first-arrived segments are received by the data sender. At this

time, the acknowledged segments are purged from the data sender's buffers, the cwnd slides forward, and the next segments are sent, maintaining link utilization. This process repeats continually and, when in a steady state, bandwidth is fully utilized.

Large BDP can result in performance problems. A "Long Fat Network" (LFN) is a network with a large BDP. High-capacity storage extension using GbE, OC-48 (2.5 Gbps), and 10 GbE (OC-192) often constitutes an LFN. However, within metro distances the latency tends to be small—but not small enough to allow TCP to continue to operate efficiently. RTT has to be less than 1 ms to prevent the standard TCP 64 KB cwnd from hitting the ceiling at 10 GbE speeds.

Here are some common link examples that exceed the original TCP 64 KB cwnd:

- A 10 Mbps satellite link has a BDP product of 10 Mbits, corresponding to approximately 825 outstanding TCP segments of 1460 bytes each. This example is extreme on the delay part of the equation, with 1000 ms of delay.
- An example of a low-delay, high-bandwidth network is 10 GbE over DWDM with 1 ms RTT. The BDP is also 10 Mbits, with the same number of outstanding TCP segments.

There are fundamental performance problems with non-RFC1323 TCP over LFN, centered on maximum cwnd size limits. The TCP header has a 16-bit field to report the receive window size to the sender. Therefore, the largest window that can be used is 216 = 64 KB. To circumvent this problem, "Window Scale" was defined to allow windows larger than 216. This option defines an implicit scale factor, used to multiply the window size value found in each TCP header in order to obtain the true window size. The scale factor is negotiated during the TCP three-way handshake connection origination. The maximum possible value is 1 GB; however, for the sender and receiver to use such a large window, they also have to have this much buffer space. The cwnd on the Brocade 7840/7800/FX8-24 is negotiated with a value of 9 to 20 MB per TCP session.

Congestion Handling

TCP has a variety of mechanisms designed to recover from "congestion events." Congestion events are not necessarily caused by congestion but are events in which segments are lost in transit. The mechanisms used to recover from congestion events are defined in RFC 2581 (referenced below) and then optimized for storage applications by Brocade:

- Fast Retransmit/Fast Recovery

  The TCP receiver sends an immediate Duplicate ACK (DupACK) when an out of-order segment arrives. Why is a DupACK sent? This is a notice from the receiver that segments are being received, but that something is wrong. DupACK is indicated by the sequence number being the same as the last ACK, hence the name "Duplicate ACK." The purpose of this ACK is to inform the data sender that a segment was received out of order and what the expected sequence number is.

  From the sender's perspective, duplicate ACKs can be caused by a number of network problems. First, they can be caused by dropped segments. In this case, all segments after the dropped segment are out of order and trigger duplicate ACKs. Second, duplicate ACKs can be caused by the reordering of data segments within the IP network, which is not a rare event in some IP network architectures. Finally, duplicate ACKs can be caused by replication of an ACK or data segments by the network, which is the rarest case.

If a DupACK is sent, and subsequently the TCP receiver receives the missing segment that fills in all or part of a gap in the sequence space, the receiver sends an immediate ACK. This generates timely information for a sender attempting to recover from data loss so that it can quickly move forward.

The TCP sender uses the "Fast Retransmit" algorithm to detect and repair loss based on incoming DupACKs. The Fast Retransmit algorithm uses the arrival of 3 duplicate ACKs (4 identical ACKs without the arrival of any other intervening packets) as an indication that a segment has actually been lost and is not just delayed in its arrival to the destination. After receiving 3 duplicate ACKs, TCP performs a retransmission of what appears to be the missing segment, without waiting for the retransmission timer to expire.

After the Fast Retransmit algorithm sends what appears to be the missing segment, the "Fast Recovery" algorithm governs the transmission of new data until a non-duplicate ACK arrives. The reason for not performing slow start is that the receipt of the duplicate ACKs not only indicates that a segment has been lost, but also that segments continue to arrive at the remote side.

Fast Recovery is a method of managing the cwnd such that segments sent during the Fast Retransmit process are taken into consideration, and subsequent congestion events are mitigated. Standard TCP dictates that cwnd be set equal to slow start threshold (ssthresh), and the ssthresh in turn is set to ½ the current segments in flight, which is known as the Flight Size. This cuts throughput by 50 percent.

Brocade has optimized this process for storage applications by reducing the amount the cwnd is cut to a value less than 50 percent. Dedicated connections for storage traffic do not need to mitigate congestion events, like oversubscribed shared links on the Internet do. Using a value less than 50 percent is not detrimental to storage network operations.

- Slow Starts

  The Slow Start algorithm is used for the purpose of the initial sending of data after TCP establishes its connection, and after repairing loss detected by the Retransmit Timer. Initial transmission into a network with unknown conditions, such as the public Internet, requires TCP to slowly probe the network to determine the available capacity and avoid congesting the network with an inappropriately large burst of data, hence the name "Slow Start."

  The Initial Window (IW) size plays an important part in Slow Start. Standard TCP limits the IW to 2 segments. The Brocade 7840/7800/FX8-24 uses an IW that is larger than this, with the effect being that storage traffic can obtain its committed rate much quicker. This is how it works: For each segment acknowledged, the cwnd is increased by that amount. This works out to be an exponential increase. For example, 1 ACK grows the cwnd to 2, then 2 ACKs grows the cwnd to 4, and so on. This continues until the slow start threshold (ssthresh) is reached. After reaching the ssthresh, Congestion Avoidance takes place. You can see how starting at a larger value ramps up throughput faster (fewer RTT), due to the exponential nature of Slow Start.

  Dedicated RDR links do not have unknown conditions, so using an optimized form of Slow Start that ramps up data rates more quickly is prudent in these environments.

- Congestion Avoidance

  Once the cwnd reaches the ssthresh, Slow Start ends and Congestion Avoidance begins. During Congestion Avoidance, cwnd is incremented by 1 full-sized segment per Round-Trip Time (RTT). Congestion Avoidance continues until congestion is detected or until the committed rate or interface speed is obtained. Congestion Avoidance is a very slow increase in the cwnd, approximately 1 segment per RTT. When TCP was introduced years ago, 64 KB cwnd traffic resumed to full rates very quickly, because the available bandwidth was small relative to the bandwidths we see today—10 GbE vs.128 kbps—which is 78,000 times greater. With a cwnd of 20 MB, approaching full rates one segment per RTT can take a long time.

  Brocade has adjusted the ssthresh up to equal the BDP, enabling Congestion Avoidance to start later after traffic flow has already reached acceptable rates, or to eliminate Congestion Avoidance altogether. Congestion Avoidance is not required for dedicated enterprise connections. Dedicated links for storage applications do not have to consider contention, so a slow and prolonged congestion avoidance process becomes an unnecessary and wasteful safeguard.

- Retransmit Timer

  Sometimes Fast Retransmit cannot be triggered to recover from lost segments. Consider the case when no additional data is being sent to cause the 3 duplicate ACKs to come back. It could be the last segment of a transfer that is dropped, or perhaps only 1 or 2 duplicate ACKs occur before the transfer stops. In this case, the only way for a retransmit to occur is for the TCP Retransmit Timer to expire. Upon expiration, the lost segment is re-sent. After re-sending the segment, TCP goes into Slow Start to recover.

  The retransmit timer is measured RTT plus some value. Brocade FOS on the Brocade 7840/7800/FX8-24 has the ability to set the minimum retransmit timeout value. The Retransmit Timer is not less than this value. Setting a value too low can cause premature retransmission that might be ACKed if a little more time is permitted. This can result in a waste of network resources. A value too high prevents retransmission from occurring expeditiously. The default has proven to work best in most cases.

TCP SACK (Selective Acknowledgement): RFC 2018

Multiple packet losses from a window of data (cwnd) can have a catastrophic effect on TCP throughput. Traditional TCP uses a cumulative acknowledgement scheme in which received segments that are across the gap of missing segments are not acknowledged. Only the contiguous segments up to the last one received are acknowledged. This forces the sender, who is trying to recover, to either wait a roundtrip time before discovering that another segment is also lost or to unnecessarily retransmit segments which have been correctly received but not acknowledged, because they were not on the contiguous side of the gap. With the cumulative acknowledgement scheme described here, multiple dropped segments generally cause TCP to lose its "ACK-based clock," resulting in reduced overall throughput.

Brocade Selective Acknowledgement (SACK) is a strategy that corrects this behavior in the face of multiple dropped segments. With SACK, the receiving side can inform the sender about a finite number of holes that exist within the segments that have arrived successfully. Now, the sender needs only to fill in the holes by retransmitting the segments that have actually been lost.

Other TCP/IP Considerations

QoS on FCIP packets

There is a clear need for a relatively simple and coarse method of providing differentiated classes of service for data traffic to support various types of applications with specific business requirements. The differentiated services approach to providing Quality of Service (QoS) in networks employs a small, well-defined set of building blocks from which a variety of aggregate behaviors may be built. A small bit-pattern in each packet, referred to as the Differentiated Services Code Point (DSCP) in the IPv4 TOS octet or the IPv6 Traffic Class octet, is used to mark the packet to receive a particular forwarding treatment or Per-Hop Behavior (PHB) at each network node.

There must exist a common understanding within an enterprise about the use and interpretation of this bit-pattern before it has any applicable meaning, because the QoS of flows within an IP network becomes relative. Without some degree of consensus for inter-domain use, multivendor interoperability, and consistent reasoning about expected aggregate behaviors within a network, the concepts of QoS fail. Fortunately, there is a standardized common layout for the six-bit field of both octets, called the "DS field." RFC 2474 and RFC 2475 define the architecture and the general use of bits within the DS field.

Brocade FOS enables setting DSCP on the Brocade 7840/7800/FX8-24. Note: DSCP values are not a ranking of priorities from 0 to 63.

DSCP uses are listed below in the table:

PHB	**DSCP Value**		
Default Lowest Priority	**0**		
Class Selector (Only IP TOS within DSCP)	8,16,24,32,40 (Data Traffic) 48,56 (Network Control Traffic)		
EF (Expedited Forwarding) Highest priority	**46**		
AF (Assured Forwarding)	Low	Med	High
Class 1	AF11=10	AF12=12	AF13=14
Class 2	AF21=18	AF22=20	AF23=22
Class 3	AF31=26	AF32=28	AF33=30
Class 4	AF41=34	AF42=36	AF43=38
Private Use	**Odd-Numbered Values**		

The Brocade 7840/7800/FX8-24 does not enforce DSCP or 802.1P (L2 CoS [Class of Service]), however, they do enforce a QoS structure of High/Medium/Low. The percentage of bandwidth apportioned to each priority is user configurable. Moreover, the Brocade 7840/7800/FX8-24 implements a separate TCP stack for each priority. The effectiveness of QoS is muted when all priorities are communicated through a single TCP session. TCP is too rigid a protocol to allow the freedom within the IP network to enforce the QoS markings. QoS implementations do not function properly when using a single TCP session for all priorities. To gain priority autonomy, a separate TCP session is required.

The Brocade 7840/7800/FX8-24 uses a queuing algorithm to enforce the High/Medium/Low priorities. There is no priority advantage unless there is actual contention for the available bandwidth. When there is contention, the bandwidth is apportioned based on the levels set by the user. The defaults are 50/30/20 percent. High must be a higher percentage than medium, and medium must have a higher percentage than low. If the high priority is not using its 50 percent of the bandwidth, the unused portion is split evenly among the medium- and low-priority queues.

In a situation in which you have high-priority disk traffic that may use only 20 percent of the bandwidth and low-priority tape traffic that uses 50 percent of the bandwidth, it is not best practice to assign the disk to the low priority and the tape to the high priority to apportion the traffic bandwidth accordingly. What happens if the default settings were used is best. The tape gets to use its desired 50 percent of the bandwidth most of the time, simply because the disk was not using it. Thus there is no contention. Yet, when the disk needs to use its share of the bandwidth (and possibly more) at particular times, it is able to pull from the tape's bandwidth because it has a higher priority. This is the safest situation and is best practice for production RDR flows.

Data traffic ingress to the Brocade 7840/7800/FX8-24 experiences no benefit from a queuing algorithm, because the traffic has arrived at its final destination and is always processed at line rate.

Virtual channels, which play an important role in solving the Head of Line Blocking problem, should not be confused with priority queuing algorithms. They are two different mechanisms used for different purposes.

Interacting with MPLS

MPLS (Multi-Protocol Label Switching) is a technology that was developed to virtualize a single large layer 2 network, such that multiple autonomous layer 3 networks can be superimposed. MPLS also provides for a more deterministic, secure, and traffic engineered network for customers. Service provider layer 2 networks include a multitude of link types: TDM (Time Division Multiplexing) links, SONET (Synchronous Optical Network), DWDM (Dense Wavelength Division Multiplexing), Carrier Ethernet, and so forth. MPLS enables better utilization of infrastructure and provides customers with what appears to be their own private network. This is possible because the common component tying these link types together is an MPLS-capable router.

The Brocade 7840, 7800, and FX8-24 do not have any native MPLS capabilities, meaning that they cannot label or unlabel traffic, switch traffic based on labels, or enforce QoS or security based on labels. A service provider edge router performs those duties, with data coming to the edge router from the Brocade 7840/7800/FX8-24. DSCP and 802.1P (L2 CoS) information that is set by the Brocade 7840/7800/FX8-24 and sent into an MPLS network can be used to set the QoS attributes of the MPLS labels, if the network is configured to do so. On an MPLS network, QoS and security attributes are contained in the labels, so that at each hop the appropriate actions can be taken to comply with those attributes. Each hop can be configured to change the QoS and security settings as well.

The Brocade 7840/7800/FX8-24 is analogous to a server. It is connected to an Ethernet LAN and, in turn, communicates over a WAN. The LAN switch that a Brocade 7840/7800/FX8-24 directly connects to should be configured as if the 7840/7800/FX8-24 Ethernet interfaces are a server NIC. TCP/IP originates and terminates on the 7840/7800/FX8-24 just like client/server. Servers do not run MPLS, and neither do the

Brocade 7840, 7800 and FX8-24. MPLS operates in the network and is transparent to end devices like servers and clients, which is appropriate here as well.

Extension Trunking

Extension Trunking allows the aggregation of multiple circuits to achieve greater bandwidth links and provides lossless failover for increased resilience over IP WANs. Refer to the Brocade Tech Brief entitled "Brocade Extension Trunking" for more information on the operation and benefits of Brocade Extension Trunking.

Conclusion

Brocade OSTP, FastWrite, and SO-TCP technologies are just a few of the advanced features that have been developed for the industry-leading Brocade Extension solutions. These pipelining and optimization technologies improve throughput and mitigate the negative effects of latency when transporting storage traffic over distance. As a thought leader and innovator in extension technology, Brocade was the first to develop these and other technologies, including Extension Trunking, FC Routing, FICON Emulation, FCIP over 10 GbE and 40 GbE, Adaptive Rate Limiting, FCIP IPsec, Per-Priority TCP QoS, and much more. These critical capabilities deliver unmatched performance, efficiency, and flexibility and enable more effective data protection, data sharing, and consolidation.

In addition to the industry-leading Brocade Extension technologies, Brocade offers SAN assessment, design, and implementation services to help organizations deploy optimized networking infrastructures that meet their specific objectives.

Corporate Headquarters
San Jose, CA USA
T: +1-408-333-8000
info@brocade.com

European Headquarters
Geneva, Switzerland
T: +41-22-799-56-40
emea-info@brocade.com

Asia Pacific Headquarters
Singapore
T: +65-6538-4700
apac-info@brocade.com

BROCADE