



InfoStor QuickVote

In the evaluation of disk arrays, do you use industry-standard benchmarks such as the Storage Performance Council's SPC-1? (end users only, please)

[Vote Now!](#)



Extending storage over WANs

Solving the issues of latency, data integrity, and bandwidth utilization can result in an effective WAN storage solution.

By Gary Johnson

The distributed nature of IT and a heightened concern for disaster recovery and business continuity have combined to make it common today for enterprises to connect their data storage over distance. Virtually any size organization recognizes the need to have at least one copy of its data in a location outside its main data center.

For many firms, this distance requirement has meant implementing a metropolitan area network (MAN) using point-to-point fiber optics as the data transport medium. But this approach can be limited to distances of less than 100 miles, and natural disasters often have effects beyond a metro area. Extending storage over an even greater distance—a wide area network (WAN)—is often necessary.

IT managers are also being called on to connect far-flung data centers or to integrate newly added facilities, such as those resulting from acquisitions. These data centers can be thousands of miles from a company's main location.

In the first two articles in this series (see InfoStor, July 2003 and August 2003, pp. 38 and 41, respectively), we provided an introduction to extending storage over a WAN and looked at the three main issues involved: latency, data integrity, and bandwidth utilization.

In this article we describe how these issues can be solved to effectively implement a WAN storage solution. In the final article we'll discuss the key enabling technologies that will help you develop a remote storage solution that's right for your needs.

Solving the latency issue

Latency is the delay of data transfer between sender and receiver, which increases as distance increases. The general guideline for predicting latency is 1 millisecond (ms) for each 100 miles. So, for example, if the distance between the server and a tape drive is 500 miles, the transfer latency is 5ms one way. However, an I/O is not complete until the tape acknowledges receipt of the data by sending a response back to the server. Thus, the true latency of a data transfer is the round-trip time for completion of an I/O—in the above case, 10ms.

To solve latency on writes, a storage router presents data blocks to the storage controller prior to completion of the previous write. This means the storage router is operating in a "trust-me" mode with the server, so the server will issue multiple writes without waiting for the actual acknowledgment from the remote storage device. For reads, if the protocol allows, the storage router should be reading ahead of the actual commands being received from the server. This ensures that the data blocks are already in the network when the reads are issued and dramatically reduces data transfer latency.

CURRENT ISSUE

September 2003



WHITE PAPERS

[ADIC: Complimentary White paper on "Tape Backup and Restore"](#)

[Alacritech: Need TCP/IP Offload? Free software/white papers.](#)

[Access our extensive database of vendor-supplied white papers and research. Powered by Bitpipe.](#)

A term commonly used for this "trust-me" mode is "pipelining." With pipelining, the storage router is responsible for delivering the data blocks in the right order, while also ensuring data integrity. If an unrecoverable error occurs with the data transfer, the storage router notifies the server of this condition. Using the information within the error indications, most applications can then back up to the last successful block transmission and resume the data transfer.

Pipelining emulates the server's command and the storage subsystem's response sequencing, so that the storage controller appears to be "local" to the server regardless of the actual distance between the two. On both writes and reads, many applications specify how many blocks will be transferred in a given data exchange. This enables a process known as "pre-acknowledging" command completion. For example, many tape backup applications use "tape mark" as the last command in a series of command requests. To ensure actual completion of the data transfer, the last command is not pre-acknowledged. The storage router must wait until the storage controller sends command completion for the last command before acknowledging completion to the server.

In architectural terms, the storage router becomes an extension of the storage subsystem's buffer. This process is true for ESCON, SCSI, iSCSI, FICON, and Fibre Channel protocols. Since the specific commands and responses differ among these technologies, pipelining has been adapted to emulate each of these protocols. The command set used to complete a data transfer between a server and a storage subsystem is different for tape versus disk, so the pipelining technology must accommodate these differences as well.

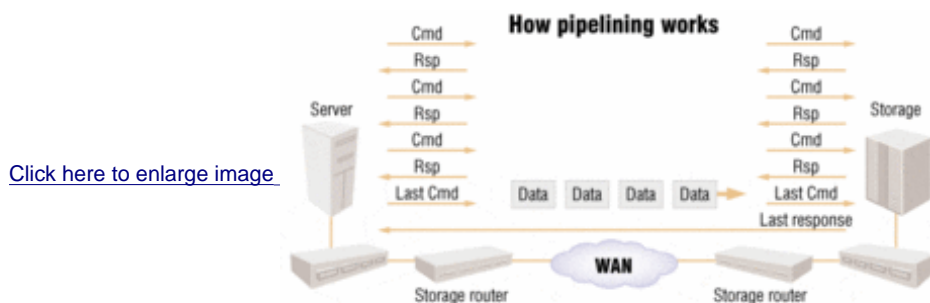


Figure 1: With pipelining, the storage router is responsible for delivering data blocks in the right order, while also ensuring data integrity.

Figure 1 is a simplified illustration of pipelining, while Figure 2 (see p. 44) shows the impact of pipelining on tape performance.

Solving the data integrity issue

As explained in last month's article, data integrity refers to keeping the contents of a data block from changing as it is sent to and from the receiver. Data corruption occurs when the value represented by the data byte is changed to some other value. When this happens, the integrity of the data is lost. The more electronic components a byte of data must pass through between the server's memory and a storage device, the higher the probability that the data byte will be corrupted. When you extend your storage over a WAN, additional components such as storage routers, routers, and multiplexers handle the data blocks. Just passing through a physical link can subject the data blocks to bit transmission errors and cause corruption.

To solve the issue of data integrity, there is a feature called cyclical redundancy check (CRC), which is a highly reliable method for detecting data corruption. CRC algorithms in common use today are simply long division problems that generate a polynomial with binary coefficients that represents the contents of a given data block. This polynomial is transmitted with the data and used to validate that the data block was not corrupted during the transfer. This polynomial (also called a CRC) can be 16 bits, 24 bits, or 32 bits long. The larger the data block, the larger the CRC to ensure proper error detection.

We
made
it First

We
made
it Better

We
made
it Faster

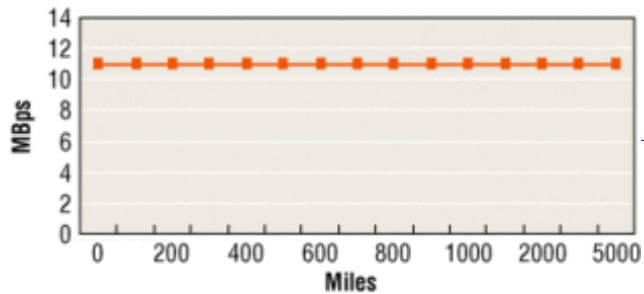
Guess
Who?

BrightStor
ARCserve
Backup v9



Computer Associates®

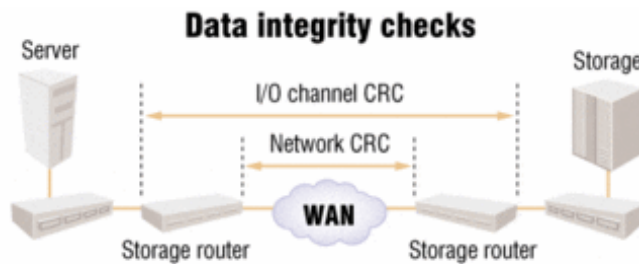
Pipelining and tape performance



[Click here to enlarge image](#)

Figure 2: Pipelining enables tape drives/libraries to maintain throughput levels as transmission distances increase.

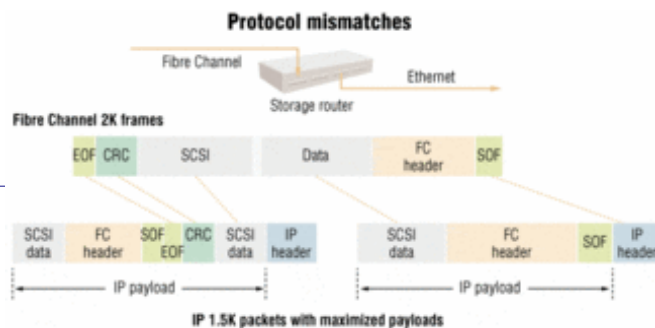
In order for CRC checking to be effective in a networked storage environment, the storage router must calculate the CRC as the data block is being received from the sending device and append the CRC to the data block for transfer across the network. The receiving storage router then calculates a CRC on its own and compares it to the CRC in the data block. If they match, data integrity is assured. If they don't match, an error is flagged, the data block is discarded, and a retransmission of the original block takes place from the sending device.



[Click here to enlarge image](#)

Figure 3: Some storage routers perform two levels of cyclical redundancy checks (CRCs) on the same data block to make error detection and recovery more efficient.

Some storage routers perform two levels of CRC checks on the same data block to make error detection and recovery more efficient. The first level is the CRC appended to the data block when it's received from the sending device. The second-level CRC is calculated by the storage router prior to the data block being placed on a WAN link.



[Click here to enlarge image](#)

Figure 4: A Fibre Channel frame is approximately 2,000 bytes, while an IP packet is only about 1,500 bytes. Without payload matching, each Fibre Channel frame would require two IP packets.

This CRC is called a "network CRC," which is added to the data block and then verified at the other end of the WAN link. If a CRC mismatch occurs here, the data block is discarded and retransmitted by the sending storage router. This is more efficient because the sending device doesn't need to be involved in network error recovery. The CRC process creates an extremely high level of data integrity assurance, as noted below.

Given the level of error detection coverage established by a 32-bit CRC, in addition to the low level of BER (biterror rate) of typical fiber and WAN media, the probability of an undetected data error nears 1 in 10⁴⁰. In other words, it would take about 317 sextillion years before an undetected error would be encountered (see Figure 3).

Solving bandwidth utilization issue

Bandwidth represents a significant portion of the total cost of ownership of any storage-over-WAN solution. For example, recent pricing for two OC-3 circuits from Minneapolis to Chicago cost about \$33,000 per month—or almost \$400,000 per year. Optimizing available bandwidth is a fundamental responsibility of the storage router.

Several technologies are available today and, when working together, make this objective possible.

The first is "pipelining," as previously described. Since it is pre-acknowledging data transfer commands, it increases the number of data blocks flowing across a link at any point in time.

The second technology is "payload matching." Payload refers to the size of the data packet supported by a given type of link protocol. Often, there is a difference between the size of the data block received from the device and the size of the link packet.

Applications use a variety of block sizes, as do different link protocols. Payload matching divides up or concatenates data blocks to fill each link packet that is sent across the network.

A good example of why this is important is when you are sending Fibre Channel frames over an IP network. A Fibre Channel frame is approximately 2,000 bytes long, while an IP packet is only about 1,500 bytes long. Without payload matching, each Fibre Channel frame would require two IP packets. This means that every other IP packet would carry only 33% of its payload capacity. In this scenario, the maximum link utilization would only be 67%, even with pipelining (see Figure 4).

Another technology having an effect on bandwidth utilization is "load leveling," which is advantageous when more than one WAN link is available for the data transfer. Load leveling evenly spreads the link packets across all available links.

This has the effect of creating a single, but larger data pipe. Some technologies fill up the first link before using any additional links. Other technologies require separation of the links by application type. Load leveling uses all of the links, no matter what the application.

Bandwidth utilization is based on the number of data blocks traversing a link (or links) in a given period of time. The larger the number of concurrent data transfers sharing the links, the better the bandwidth utilization.

Bandwidth utilization can be further improved through the use of compression technology on the storage router. One type of compression technology looks for like data patterns, strips out redundant patterns, sets a value representing how many were stripped out (compressed), and then combines this shortened data block with others to fill a link packet, putting the combined blocks on the link(s).

This increases the number of data blocks traversing the network in a given period of time, increasing actual link utilization beyond its signaling speed. For example, a 3:1 compression ratio yields actual data throughput of three times the signaling speed of the link(s).

Compression results are very data pattern-dependent. A data pattern of all "ones or zeros" will yield the highest compression ratios, while data already compressed by the server will yield the lowest compression ratios. Be wary of vendor claims of high compression ratios. These could be lab results using compression-friendly data patterns. In reality, you should expect at least a 2:1 compression ratio on uncompressed data.

In the next article, we'll discuss various enabling technologies that have been developed to address the issues associated with deploying storage over a WAN.

Gary Johnson is vice president of solutions at CNT (www.cnt.com) in Minneapolis, MN.

InfoStor September, 2003

Author(s) : Gary Johnson

Interested in a subscription to InfoStor Magazine?

[Click here](#) to subscribe!



Copyright © 2003 - PennWell Corporation. All rights reserved.