



Storage Networking, Part 1: SANs and Fibre Channel

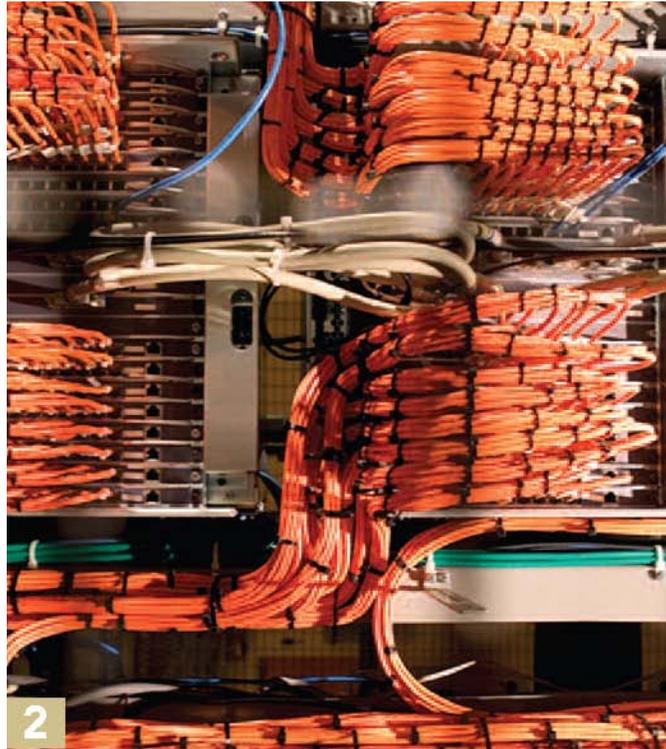


101001010110110100101011010101101100101
010010011001100101010001110010101
010010111001001001010100100
111011001010100101010110101
1010010101101101001010110101011
010010011001100101010001110010101001000010

an **internet.com** Storage eBook

contents

[Storage Networking, Part 1: SANs and Fibre Channel	1
---	---	---



This content was adapted from Internet.com's Enterprise Networking Planet Web site and was written by Charlie Schluting.

2 Understanding SANs and Storage

4 Understanding Fibre Channel

6 Understanding the Fibre Channel Protocol

11 Understanding Fibre

Storage
Networkin
g, Part 1:
SANs and
Fibre
Channel,
an
Internet.co
m Storage
eBook. ©
2008,
Jupitermedi
a Corp.

Understanding SANs and

An FCSAN, or Fibre Channel SAN, is a SAN comprised of the Fibre Channel protocol. Think of Fibre Channel (FC) as an Ethernet replacement. In fact, Fibre Channel can transport other protocols, like IP, but it's mostly used for transporting SCSI traffic. Don't worry about the FC protocol itself for now; we'll cover that later.

A fairly new type of SAN is the IPSAN: an IP network that's been designated as a storage network. Instead of using FC, an IPSAN uses Ethernet with IP and TCP to transport iSCSI data. There's nothing to stop you from shipping iSCSI data over your existing network, but an IPSAN typically means that you're using plumbing dedicated for the storage packets. Operating system support for the iSCSI

Storage

By Charlie Schluting

A storage network is any network that's designed to transport blocklevel storage protocols. Hosts (servers), disk arrays, tape libraries, and just about anything else can connect to a SAN. Generally, one would use a SAN switch to connect all devices, and then configure the switch to allow friendly devices to pair up. The entire concept is about flexibility: in a SAN environment you can move storage between hosts, virtualize your storage at the SAN level, and obtain a higher level of redundancy than was ever possible with directattached storage.

protocol has been less than stellar, but the state of iSCSI is slowly improving.

Another term you'll frequently see thrown around is NAS. Network Attached Storage doesn't really have anything to do with SANs — it's just file servers. A NAS

device runs something like Linux, and serves files using NFS or CIFS over your existing IP network. Nothing fancy to see here; move along.

There is one important takeaway from the NAS world, however, and that's the difference between blocklevel storage protocols and filelevel protocols. A blocklevel protocol is SCSI or ATA, whereas file protocols can be anything from NFS or CIFS to HTTP. Block

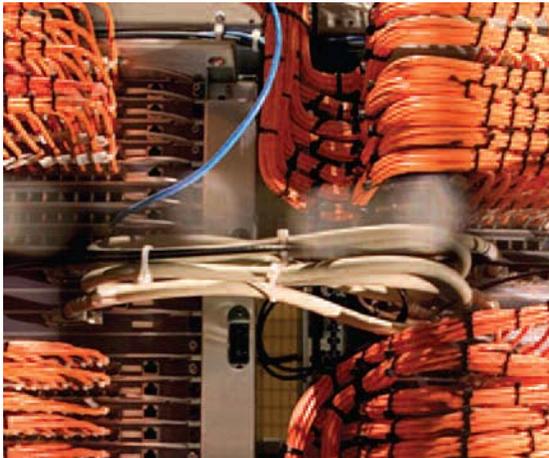
protocols ship an entire disk block at once, and it gets written to disk as a whole block. Filelevel protocols could ship one byte at a time, and depend on the lowerlevel block protocol to assemble the bytes into disk blocks.

BlockLevel Protocols

A protocol always defines a method by which two devices communicate. Block storage protocols are no

A protocol always defines a method by which two devices communicate. Block storage protocols are no different: they define how storage interacts with storage controllers.

”



“

different: they define how storage interacts with stor

ifferent category. The primary SCSI3 command set age controllers. There are two main block protocols

includes the standard commands that every SCSI3 used today: SCSI and ATA.

evice speaks, but the devicespecific

commands can d

e anything. This opened up a whole new world for ATA operates in a bus topology, and allows for two

SI, and it has been used to support many strange devices on each bus. Your IDE disk drive and CD ROM

i

S

and wonderful new devices. are, you guessed it, using the ATA protocol. There are many different ATA standards, but we'll cover just the

SCSI controllers normally contain a storage processor, important ones here. ATA2 was also known as EIDE, or

a

and the commands are processed onboard so that the enhanced IDE. It was the first of the ATA protocol we

to do know today. ATA4 introduced

host operating system doesn't become burdened

Packet

o, as with ATA. Such a SCSI ATAPI, or the ATA

s controller is called a Host Bus Interface, which allows for CD

A
d
a
p
t
e
r

·
l
n
t
h
e

S
A
N

w
O
r

t
o
k
n
o
w
a
b
o
u
t

ROM devices to speak SCSIlike

is always called an HBA.

The main thing “

FC card

with another device, SCSI device (the initiator) will initiate the communication with another device, which is known as the target. The only "traffic" sent over the

operates in a on the same bus as a regular device. producer/consumer manner. The main thing to know about

SCSI is that it

ATA

initiate the communication with ATA bus is plain electrical signal another device, which is known as the target. The host operating system

One SCSI device (the initiator) SCSI is that it operates in a producer/consumer manner. One the neat thing about ATA is that will initiate the

as the target. The roles can be is actually responsible

communication ducer/consumer manner. One the controllers are integrated.

” for imple

reversed. Most people call this a mirroring the ATA protocol, in

command/response protocol, software. This means that ATA devices will never, ever

because the initiator sends a command to a target, and be as fast as SCSI, because the CPU has to do so much

waits a response, but not always. In asynchronous work to just talk to these devices. As far as SANs are

Serial Attached SCSI, does away with this

y automatically switching back and forth. SCSI, on the other hand, is very confusing. SCSI1 and SCSI2 devices were connected via a parallel interface

CSI is tremendously more complex, but that's the gist to a bus that could support 8 or 16 devices, depending

fit on the bus width. Don't worry about

network is going to ship data. The SCSI protocol

Understanding Fibre Channel

FC wouldn't be much use without something on top of it, namely SCSI. FC is the lowlevel

mode, the host (initiator) can simply blast the target concerned, ATA isn't that important. There are some

with data until it's done. The SCSI bus, parallel in ATAbased devices that allow you to connect cheap

ature, can only support a single communication at a disks, but they translate operations into SCSI before

ime, so subsequent sessions must wait their turn. SAS, sending them out to the SAN.

limitation

the details unless you're used enough to have some older SCSI

need to understand SCSI to know how our storage gear is lying around.

plays a

enormous role in storage networking, so you may see SCSI3 separated the device-specific commands into a even want to look at it more in depth. ■

AS we dive deeper into SAN technology, it's Fibre Channel's turn to be examined. Fibre Channel, or FC, is the underpinning of all SAN technologies these days, as it won the protocol war roughly 25 years ago.

transport that ships data, but hosts are normally communicating via SCSI as far as they're concerned. The hubs, switches, and HBAs in a

m
w
n
t
W

SAN all speak FC, while the applications that use SAN storage continue to use familiar protocols, like SCSI.

The idea behind FC was to create a high-throughput, lowlatency, reliable, and scalable protocol. Ethernet wouldn't quite cut it for highly available storage needs. FC can currently operate at speeds up to 10Gb/s (10GFC) for uplinks, and 4Gb for standard host connections.

Topologies

In reality, two different protocols, or topologies, make up the FC protocol. FC supports all topologies, but the behavior of the protocol changes depending on the topology. The following three types of topologies are supported:

PTP (point to point): normally used for DAS configurations.

FCAL (FC Arbitrated Loop): Fabric Loop ports, or FL ports on a switch, and NL_Ports (node loop) on an HBA, support loop operations.

FCSW (FC Switched): the mode when operating on a switched SAN.

FCAL operation has its share of

problems, but sometimes a device doesn't support FCSW operations, and there's no choice. A hub has no choice but to operate in FCAL mode, and therefore attached hosts must do so as well. When a device joins an FCAL, or when there's any type of error or reset, the loop must reinitialize. All communication is temporarily halted during this process, so it can cause problems for some applications. FCAL is limited to 127 nodes due to the addressing mechanism, in theory, but in reality closer to 20. FCAL is mostly relegated to

“

The idea behind FC was to create a highthroughput, low-latency, reliable, and scalable protocol. Ethernet wouldn't quite cut it for highly available storage needs.

”



niche uses now, including but not limited to internal disk array communications and internal storage for high-end servers.

FC switches can be connected any way you please, since the FC protocol avoids the possibility of a loop by nature. Ethernet isn't so lucky. The addressing scheme used does impose a limit of 239 switches though. FC switches use FSPF, a linkstate protocol like OSPF in the IP world, to ensure loop-free and efficient connectivity.

Ports

As previously mentioned, there are different port types in a SAN, and it can get confusing. Let's try to clear up some of that terminology:

N_Port: Node Port; the node connection point; end points for FC traffic

F_Port: Fabric Port; a switch-connected port, that is a "middle point" connection for two N_Ports

NL_Port: Node Loop Port; connects to others via their NL_Ports, or to a switched fabric via a single FL_Port; or NL_port to F_Port to F_Port to N_Port (through a switch)

FL_Port: Fabric Loop Port; a shared point of entry into a fabric for AL devices; example: NL_Port to FL_Port to F_Port to N_Port

E_Port: Expansion Port; used to connect multiple

FC Layers

FC has its own layers, so in fact, calling it "like Ethernet" isn't quite accurate, even if it helps for understanding. They are:

FC0: The interface to the physical media; cables, etc.

FC1: Transmission protocol or datalink layer, encodes and decodes signals

FC2: Network Layer; the core of FC

ALTERNATIVE THINKING ABOUT VIRTUAL STORAGE:

A powerful business innovation in data storage is now within your reach. The new HP StorageWorks 4400 Enterprise Virtual Array is here. It virtualizes up to 96TB of storage—

Up to 96TB virtual storage capacity.

FC networks are generally designed in one of two ways: either one big star, or one big star with edge switches hanging off it. These are commonly known as "core-only" and "core-edge" configurations. Normally a SAN will contain two of these networks, and each host's HBA or storage device's controller will attach to each. Keeping these networks separate isn't as necessary as it is with FCAL topologies, but even with FCSW setups it still provides complete isolation and assurance that a problem in one fabric won't impact the other. An FSPF recalculation, for example, could cause a brief interruption in service.

switches together via ISL (interswitch links)

G_Port: Generic Port; can switch between F_Port and E_Port operation depending on how it's connected

TE_Port: Trunked Expansion Port; link aggregation of multiple E_Ports for higher throughput

You'll generally only see F_Ports and FL_Ports when looking at a single SAN switch, and knowing the difference helps. FL means that you're talking FCAL, and there's a device attached that is either a hub, something that can't do anything but FCAL, or something strange. Ports will automatically configure themselves as an FL_Port if the attached device is loop-only, otherwise it will be an F_Port. It's also worth noting that some brands of FC switches don't allow you to have an E_Port unless you pay a higher licensing fee. It's something to think about if you ever plan to connect multiple switches together.

FC3: Common services, like hunt groups

FC4: Everything! Protocol mapping for SCSI, iSCSI, FCP, IP, and others

The bulk of FC is really in FC2. FCPH refers to FC0 through FC2, which are strangely dubbed "the physical layers."

FC also supports its own naming and addressing mechanism, which sheds light on the previously mentioned limitations in FCAL and FCSW topologies. ■

across numerous storage servers and platforms—simplifying storage management and speeding access. Less limitations. More freedom. Technology for better business outcomes.

Enterprise-class performance

Over 30% better capacity utilization*

Up to 75% less time needed to
configure and manage*
Easy application integration

g
e
.
V
i
s
i
t
h
p
.
c
o
m
/
g
o
/
v
i
r
t
u
a
l
s
t
o
r
a
g
e

N
o
w
,
s
t
h
e
t
i
m
e
f
o
r
v
i
r
t
u
a
l
s
t
o
r
a

Understanding the Fibre Channel Protocol

Understanding the guts of the Fibre Channel (FC) protocol itself, including the naming format and addressing scheme, allows one to better understand what's happening on a SAN. Quickly glancing at a problem and knowing what's wrong requires thorough knowledge of all the protocols involved. While it's possible to operate a SAN with only pointandclick GUIs and limited knowledge, it certainly isn't recommended. So let's learn about the FC protocol.

To reiterate: Fibre Channel is not a replacement for SCSI; SCSI generally

rides on top of Fibre Channel. Now that we have that out of the way, let's get to work.

FC generally refers to the FCPHY layers: FC0FC2, which were briefly discussed earlier. The term FCP, Fibre Channel Protocol, refers to the interface protocol for SCSI, or the FC4 mapping. We're talking about the innerworkings of FC here, not FCP.

FC data units are called Frames. FC is mostly a layer 2 protocol, even though it has its own layers. The maximum size for a FC frame is 2148 bytes, and the header FC frame itself is a bit strange, at least when compared to Ethernet with IP and TCP. FC uses one frame format for many purposes, and at many layers. The function of the frame determines the format, which is strange and wonderful, compared to our notions in the IP world.

FC frames begin with a startofframe (SOF) marker followed by the frame header, which will be described in a moment. The data, or FC content, comes next, fol

lowed by an EOF. The reason for the encapsulation is so that FC can be carried over other protocols, such as TCP if



“

desired.

The FC frame itself, the general format that is, varies in size quite a bit. In Figure 1 (next page) you can see the SOF and EOF markers we mentioned before. The strange part about FC headers is that they are word-oriented, and an FC word is 4 bytes. Up to 537 words are allowed, which gives us our 2148byte capacity.

The components of the header, with all the optional items listed, are:

- SOF (1 word): The start of a frame.

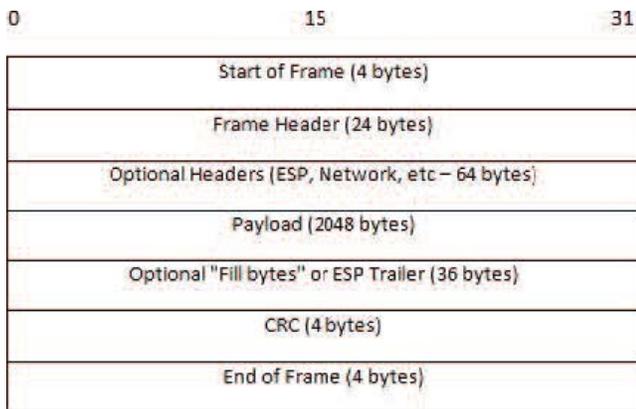
Frame Header (24 bytes): The header that specifies what protocol is being used, as well as the source and destination address. Varies depending on the protocol in question.

Optional ESP Header (8 bytes): Provides encryp

protocol itself, including the naming format and addressing scheme, allows one to better understand what's happening on a SAN.

”

Figure 1



tion; includes the SPI and ESP sequence number.

Optional Network Header (16 bytes): So that you can connect an FCSAN to nonFC networks.

Optional Association Header (32 bytes): Not used by FCP, but can be used to identify processes within a node.

Optional Device Header (up to 64 bytes): Not used by FCP, and is application specific.

Payload: The data, up to 2048 bytes.

Optional Fill Bytes (variable): Used to ensure the variablelength payload ends on a word boundary.

Optional ESP Trailer (variable): Contains check values for ESP.

CRC (4 bytes): A CRC of the header and FC data fields.

End of Frame (4 bytes): Ends the frame, and says whether or not it's the last in a sequence.

The FC frame format includes FCspecific information, including the source and destination, among others. Hopefully it is clear now why FC is so flexible, which also explains why there's so many FC protocols available to give you a headache.

The actual FC Header, depicted in Figure 2 includes the following fields:

Routing Control (1 byte): The routing portion says if this is a data frame or a linkcontrol frame (either an ACK or a Link_Response), and the information portion indicates the type of data.

Destination ID (3 bytes): The FC address of the destination.

Class Specific Control/Priority (1 byte): Essentially,

Figure 2

8-Gig Fibre Channel Arrives ... Slowly

By John P. Mello Jr.

Storage companies began rolling out their first 8 Gigabit per second Fibre Channel products during the summer of 2007, but customers won't get their hands on the devices until sometime this year, and complete systems composed of host bus adapters (HBAs), switches, and storage arrays will take a lot longer than that.

One reason for the long lag is the rigorous process imposed on new products before they reach users, said Tam Dell'Oro, founder and president of Dell'Oro Group. "The testing process typically takes six months or more," she said. "It's lengthy and thorough."

"This equipment has to be highly robust — super, super reliable," Dell'Oro explained, "and it has to be able to operate with a bunch of other stuff."

As a result, adoption of new technology like 8Gbps Fibre Channel can take years. For example, according to Dell'Oro, switches and HBAs incorporating the technology's predecessor, 4Gbps, began falling into users' hands in 2004, but it hasn't been until this year that it has begun to dominate shipments of new equipment. In 2007, 97 percent of Fibre Channel switches and 80 percent of HBAs will use 4 Gbps technology, she said.

Storage arrays, she added, are usually slower than other system components when it comes to falling in line with an evolving Fibre Channel technology. "We didn't see the first four-gig storage arrays come to market until the end of 2006," she said, "and at that time, Hewlett Packard, which is a pretty significant manufacturer of storage equipment, still did not have a four-gig product out."

Historically, new generations of Fibre Channel technology have been shipped every three to four years. "That's the cycle we're on again," observed Scott McIntyre, vice president for software marketing at Emulex, which announced several new 8Gbps products last summer, including a family of HBAs,

R_CTL(1)	D_ID (3 bytes)	
CS_CTL/PRI(1)	S_ID (3 bytes)	
Type(1)	F_CTL(3 bytes)	
SEQ_ID(1)	DF_CTL(1 byte)	SEQ_CNT(2 bytes)
OX_ID(2 bytes)		RX_ID(2 bytes)
Parameter (4 bytes)		

Quality of Service.

Source ID (3 bytes): The FC address of the originating node.

Type (1 byte): Indicates the next protocol (what's in the Payload), unless R_CTL indicates a control frame.

Frame Control (3 bytes): Various crazy FC options, such as sequencing information and what to do in case of a problem.

Sequence ID (1 byte): A sequence number, just like IP.

Data Field Control (1 byte): Indicates the presence of optional headers, and the size.

Sequence Count (2 bytes): The number of frames that have been transmitted in a sequence.

Originator Exchange ID (2 bytes): Assigned by the initiator, used to group related sequences.

Responder Exchange ID (2 bytes): Same as the OX_ID, but assigned by a target node.

Parameter (4 bytes): Mostly used as a "relative offset" in sequences, much like IP's offset.

Yes, it is confusing, and there's a lot of new terminology, compared to the IP world. We'll continue to refer back to these headers as we continue, so hopefully the fields and their purposes will become second nature after some realworld examples.

The next important concept to grasp is the way FC assigns names. Notice that the D_ID and S_ID fields in the FC Frame Header only allow for 24 bits. Each HBA is assigned a WWN, and each port on it is assigned a Port WWN, or PWWN. These WWNs are 64bits in length,

Understanding Fibre Channel Domains

which are larger than the 24 bits in FC. The ANSI T11 Address Identifier Format says that the FCID is made up of three parts, which are the Domain_ID, the Area_ID, and the Port_ID.

8Gig Fibre Channel . . . continued

custom mezzanine cards for server blades, and an embedded I/O controller. Emulex's main competitor, QLogic, has also rolled out 8gig components, and Brocade has unveiled 8gig blades for its 48000 Director.

McIntyre noted that the ramp up for 4Gbps was the fastest in the history of Fibre Channel. "That indicates that there's a strong and consistently growing demand for I/O throughput," he said.

One of the drivers of that throughput hunger is the spread of virtualization technology. "What we're seeing is very strong adoption of server virtualization technologies by our enterprise customers," McIntyre said. "That means they're stacking up more and more virtual machines and more and more applications on a single server, and in many cases driving them to larger servers to accommodate many more virtual machines, and that's obviously creating a higher demand for I/O throughput on each server." ■

FC networks are broken up into hierarchies, dynamically. The Domain_ID is assigned to each switch when a fabric comes online using a Domain_ID distribution process. Normally the Domain_ID is administratively configured. The Domain_ID, along with the Area_ID, a second hierarchical level, are combined with a Port_ID (assigned by the switch) to identify each FC node in a fabric. So the WWN doesn't really mean anything as far as SAN routing goes.

Domain_IDs are distributed by a Principal Switch, which ensures that everyone has the correct information. In short, an FCID will be completely random the first time a node connects, which is generally fine, unless an administrator manually configures it. Some Domain_IDs are reserved for multicast and other purposes, but the details are a bit outside our scope here. Refer to the ANSI T11 FCSW3 specification for more details. ■

Understanding the way Fibre Channel identifies domains, and a new mechanism for virtualizing your fabric, enables you to exploit these concepts to your advantage. Building a SAN isn't difficult — you just plug things in — but to make it resilient in the face of changes, there's the rub. Let's take a look at FC domains, address assignment, and VSANs.

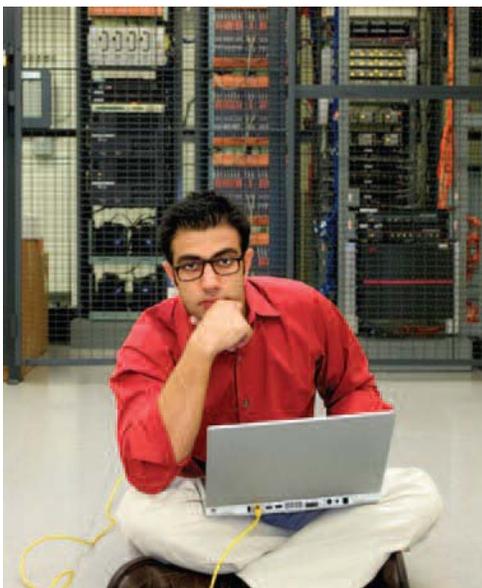
First, we must understand how a SAN fabric exists without loops. Everything you see here will look suspiciously familiar to Spanning Tree. A few terms are different, of course, but the same concept applies.

The Domain_ID is dynamically assigned to a FC switch when it comes online. The Principal Switch (PS) election begins, which is very similar to a root bridge election in Spanning Tree, followed by the Domain_ID Distribution process.

Before the switch can talk to other switches, it will first configure itself to know what's attached. Skipping over link initialization, we simply need to know that the hardware works out what port mode is present, and determines the addresses of attached N_Ports. A switch assigns the FCID to each attached node, which is derived from the Domain_ID, Area_ID and WWN of the attached node.

Briefly, this is the election process for determining the PS:

Clear Domain_ID list



Configuring the Domain_ID is important, because merging fabrics can be disruptive if conflicting Domain_IDs are present. When you have a single switch, and want to extend the fabric by connecting the two together, everything goes fine unless they're both

- On each interswitch link (EPorts), transmit the Build Fabric (BF) frame; do not send one on a port that you've received a BF on, to prevent loops

Wait for the Fabric Stability Timeout, to ensure the BF frames have been flooded throughout the entire fabric

Transmit an EFP (Exchange Fabric Parameters) frame, and send SW_ACC (Switch Accept) to each transmitter of these frames

Examine the EFP frame, looking for PS_Priority, PS_Name (the Node WWN of the switch), and the Domain_ID list

Concatenate the PS_Priority and PS_Name to select the winner; lowest number wins

Repeat until everyone attached agrees on the PS

After completion of the PS election, a switch must begin the Domain_ID Distribution process. Even if the Domain_ID is manually configured, the distribution process still occurs, because the PS needs to compile a list of Domain_IDs. The Domain_ID election process isn't really important, because most people configure the domain manually. Just know that a change of a Domain_ID results in everyone sending an EFP frame with the updated information.

switches, it will first configure itself to know what's attached. Skipping over link initialization, we simply need to know that the hardware works out what port mode is present, and determines the addresses of attached N_Ports. A switch assigns the FCID to each attached node, which is derived from the Domain_ID, Area_ID and WWN of the attached node.

tion, a switch process. Even the distribut needs to co election pro people conf

Fibre Channel has more security mechanisms built in than is largely underutilized and misunderstood, so SANs are

Domain_ID 1, as some vendors set by default. Every new switch that's brought online needs to be configured with a unique Domain_ID before connecting it to the fabric.

Conflicting Domain_IDs frequently happen when using VSANs. A VSAN is the same as a VLAN, but for FC net-

works. You can configure a VSANcapable switch (usually a Cisco) to segment ports into separate fabrics. One node connected to switch port 1 may be in fabric 322, while the node right next to it lives in fabric 4; two completely separate fabrics. Each fabric may have a domain 31, for example. For the most part, excluding some fanciness implemented by a few vendors, there is no interfabric routing, so nodes in different fabrics won't be able to talk to each other. This is wonderful, but often times it's necessary to merge two fabrics together.

Merging two fabrics is normally accomplished by connecting multiple switches together. If a "core" switch already had a link to two switches, and suddenly decides to merge the fabrics by placing them in the same VSAN, those switches better have unique Domain_IDs. If not, traffic will suddenly be spotty, since the FCIDs include the Domain_ID. Furthermore, each PS in a domain runs its own name server containing information about N_Ports, and when receiving a frame, a switch will not know which way to send it if it has conflicting information.

Just like VLANs, a VSAN can be used to implement arbitrary boundaries, in ways that make administration much more tolerable, compared to manually moving wiring. The Cisco VSAN technology is gaining widespread adoption since ANSI blessed its implementation, calling it "Virtual Fabrics." The neat thing about a VSAN is that it's more capable than the Ethernet's VLANs.

The Virtual Fabric model takes virtualization to the next level. It is possible to configure a zone server, so that all

fabricattached nodes know how to reach it. FC services run on a switch, unlike the IP world where services like DHCP and DNS normally run on a host. In a VSAN environment, the switch actually runs each service multiple times, once in each fabric.

Speaking of fabric services, there are a few wellknown FC addresses associated with SAN services. The brief list is:

- 0xFF FF F5: Multicast server
- 0xFF FF F6: Clock Sync server
- 0xFF FF F7: KDC (key distribution)
- 0xFF FF F8: Alias server (for multicast, or hunt groups)

- 0xFF FF F9: QoS information
- 0xFF FF FA: Management server
- 0xFF FF FB: Time server
- 0xFF FF FC: Directory server
- 0xFF FF FD: Fabric Controller
- 0xFF FF FE: Fabric Login server

FC addresses (the FCID) aren't actually necessary for SCSI over FC operation. Unicast FC frames are sent to and from the WWN of the node, so the FC address is really only needed in two cases: during link initialization, or when sending IP over FC. When sending IP over FC, and IP address needs to be turned into the FCID. Very similar to the Ethernet world, ARP is used in FC land. Either "ARP over FC" or FARP, which are two distinct protocols, is possible, depending on what the devices support. And you wondered why FC has so many interoperability issues? ■

Understanding Fibre Channel Zones

Fibre Channel has more security mechanisms built in that

most people realize. They are largely underutilized and misunderstood, so SANs are said to be a security problem. Let's explore FC zones: the easiest and most incorrectly configured feature of FC switches.

Any decent FC switch will allow you to configure zones. Zoning is very similar to Ethernet VLANs: it allows you to fence off traffic. Zoning is more effective than VLANning because there's no chance that traffic will "leak" between the partitions.

An FC Zone is much more than a VLAN, conceptually. Zones seem more complex at first glance, but hidden within their complexity is simplicity. A

device node, or WWN, can live in multiple zones at the same time. This capability should really be abused. Creating sane and manageable zone configurations requires a certain structure —more on that in a minute.

There are two types of zones: soft and hard.

Soft Zones

Soft zoning means that the switch will place WWNs of devices in a zone, and it doesn't matter what port they're connected to. If WWN Q, for example, lives in the same soft zone as WWN Z, they will be able to talk to each other. Likewise, if Z and A are in a separate zone, they can see each other, but A cannot see Q. This is the

complexity part; a feature that isn't widespread in Ethernet switches.

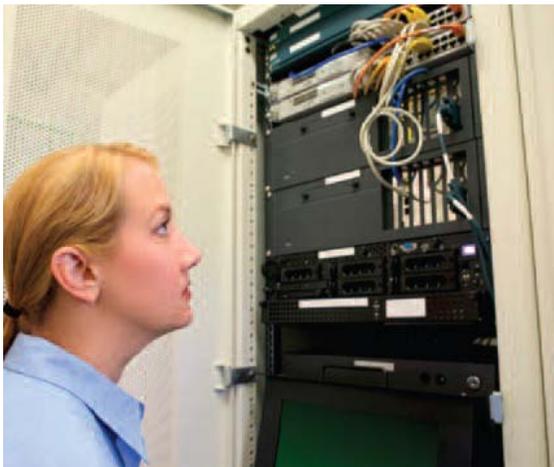
The concept of soft zones is not hard to grasp. It simply means that the enforcement relies on the WWN of the node in the fabric. The benefit to using soft zones is that you can connect to any port on a switch, and know that you'll have access to the other nodes you're supposed to see.

Is this a good thing? No. Not at all. Starting with the manageability aspect, softly zoned environments are a mess. You need to know where a node is connected, for maintenance purposes. If soft zones are used,

there can be no port description on the switch, because it will likely become out of date quickly. Next, soft zoning imposes certain security risks. Nobody, as far as everyone believes, has ever seen a hacker attempting to spoof WWNs, but it is possible. Changing a device's WWN so that it's zoned differently would be quite difficult, since the attacker would have

Fibre Channel has more security mechanisms built in that most people realize. They are largely underutilized and misunderstood, so SANs are said to be a security problem.

”



to know what WWN is allowed to access the zone he wants. You don't leave switch configurations in publicly accessible places, do you?

Hard Zones

Hard zones are more like VLANs in the Ethernet world. You place the port into a zone, and anything connecting to that port is in the zone, or zones, which are configured for that port. Sure, it is less secure in the event of a physical attack where someone is able to move fiber connections. However, do you really need to worry about that? The preferred configuration for SAN bliss is thusly: hard zoning on the switches, and WWN restrictions for LUN access on the targets. Your storage array should employ WWN masking, so that multiple initiators can be zoned such that they can both see the target.

People dream up some horrific zoning schemes. Grouping similar operating systems together may seem like a good idea, but it makes no sense in reality. Back in the day people used to scare easily at the thought of Windows servers being zoned together with storage arrays that other OSes use. Windows pops up a "do you want to initialize this new volume?" dialog when it sees new LUNs, and if the clickhappy Windows administrator decided to say yes, he just destroyed someone else's LUN. With LUN masking on the storage array this is not a concern.

Zoning Best Practices

Many schools of thought for zoning best practices exist. Most agree that soft zones are a nightmare, and they are. We're going to assume hard zoning from this point on. Remember, each node should have two HBAs, but each HBA will be in a different fabric, on different switches, for redundancy. Each switch should have the same zoning configuration.

The "single initiator zones" camp believes that you should create zones based on the initiator. This means that each zone will contain a single host, or initiator.

Internet.com eBooks bring together the best in technical information; ideas and coverage of important IT trends that help technology professionals build their knowledge and shape the future of their IT organizations. For more information and resources on storage, visit any of our category leading sites.

Multiple storage array ports can be added to the zone without violating the single initiator rule—arrays are the targets. This method makes the most sense, because you can quickly see which arrays your host can access in the configuration.

Others like to zone based on their targets. After all, each target will allow a certain number of hosts to access it, so we may as well just create a little mininetwork out of all these likeminded initiators. Some storage administrators get nervous with the thought of multiple initiators being able to see each other, but it's nice in some situations. When a server reboots, other servers in the same zone will report that "node X disappeared from fabric" in syslog. The benefit to targetbased zones is that you can quickly see which hosts have access to a specific target.

Remember, each "zone" is really just a two (or more)–way communication mapping between nodes. One port on a storage array will likely live in multiple zones (in singleinitiator style zones), each containing hosts, a.k.a. initiators.

Some people like to skip zoning altogether. For stability reasons alone, this is not recommended. A fabric reset will cause everyone to relogin at the same time, and fabric updates get sent to everyone. The potential for security issues exist as well, but in reality it's rookie mistakes that you must be most wary of.

Your zone configuration decisions are very important, so take some time to decide which style of hard zoning works best in your environment.

In Part 2, we're going to talk about configuring servers and disk arrays, and we'll look at the advantages of having a SAN. Be sure to download it from the Internet.com eBook Library at www.internet.com/ebook.

■

This content was adapted from Internet.com's Enterprise Networking Planet Web site and was written by Charlie Schluting.