CHAPTER 4

Highly Available Networks

Introduction

Without a backbone, most systems, both organic and mechanic, cannot exist. The backbone in the world of computers and software is the network. When designing highly available systems, whether they are load-balanced, failover, SANs, or simply redundant systems, a weak or flimsy backbone, an unstable or poorly designed network, presents the greatest risk to availability. If the hosts cannot communicate with each other, they are useless.

This chapter focuses on building a solid network and backbone before any high-performance, highly available architecture is put in place with respect to the actual systems themselves. We discuss backbone design for availability and scalability, network interface cards, hubs, switches, and routers, and then finish the chapter with a SAN topology primer. We also introduce the storage redundancy solutions like disk mirroring and replication.

Backbone Design for High Availability

When architecting a resilient network to support high availability or high-performance solutions, it is important to understand that your work usually begins with the backbone. This is, essentially, the core networking services, which begins with the provisioning of a root, a primary, or main switch, or router, that supports your primary network. Another term for this part of the network is the *core*, and some engineers call it the *head* or the *top*.

If your network is small or located in a single office supporting less than, say, 24 devices, your backbone really lives out of a single switch or *hub*—which is the device that lets all devices (hosts) communicate with each other.

You install additional routers and switches as soon as you have good reason to partition servers from clients on a single floor, or you need to connect two floors or buildings to each other. It doesn't take much equipment or much money these days to install a fast and highly sophisticated backbone, comprising of a core router, a switch for servers, a switch for client devices and printers, and a switch or more to connect floors, buildings, or workspaces to each other.

Regardless of whether you are installing one switch or if your backbone is comprised of numerous switches, the key is to architect and design a system of switches to form a *hierarchy*. A hierarchy lets you scale out and in easily and protects you against catastrophic failures.

In a hierarchy, the topmost router ports are devoted to interconnections between buildings (dedicated network services and segments) and router-to-switch or switch-to-switch connections. Of course, if you have a very small network, then the hierarchy is relatively shallow and narrow, for instance, one or two switches deep and wide. If you have a large network and are provisioning for one or more high-availability solutions, then your network contains many switches across the network and at several levels. This is demonstrated in Figure 4.1.

When architecting high-availability solutions, you should install redundant LAN *network interface cards* (NICs) on the servers with one NIC connecting to one port of the first redundant switch and another to the second redundant switch. You should never connect your servers directly to the root or core switch. They should be connected at the second- or third-level switches. By creating a hierarchy, you are guarding against communications failure by connecting your servers to the second or third tier's switch. This protects you from failures that may occur at the root of the hierarchy. And, if the network engineers ever need to rearchitect the root or upgrade equipment, the lower or base part of the hierarchy can still function.

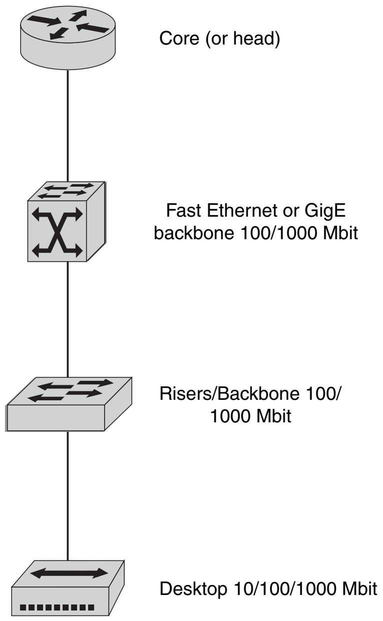
Core (or head)
Fast Ethernet or GigE backbone 100/1000 Mbit
Risers/Backbone 100/ 1000 Mbit
Desktop 10/100/1000 Mbit

Figure 4.1 A scalable, hierarchical, highly available network.

Typically, your high-end, switch-cum-routers that connect backbones only have a few ports for switch-to-switch connections. The high-end Cisco 6500s series, the 7000 series, and so on are good examples of core equipment in large networks. Your 24- and 48-port switches, such as the series 2000, 3000, and 4000 switches, connect up to these core switches. If for any reason the backbone between the root or top-level switch goes, it will not take the extended backbone further down the hierarchy.

This design also lets you scale much easier; the ability to scale is one of the key components of a good network architecture.

NOTE: A discussion of what constitutes a good switch or router is beyond the scope of this book. Network equipment is being improved every day; so, it is best to consult your Cisco, Brocade, or Dell representatives, to mention a few, for this information.

Bandwidth Field Notes

Now that we have discussed the enterprise network, let's turn to what we never seem to get enough of on a network—bandwidth. Bandwidth is the second key issue when it comes to provisioning network for high availability. The advent of cost-effective high bandwidth technology has allowed us to start entertaining what was, for years, too expensive and too complex to entertain—clusters and load-balanced services that span geographical divides. It is now possible to split a cluster over data centers in a city, across a state, and even across a continent.

We briefly discuss what it takes to build geographic clusters and the options for distributing services over a large geographical area in Chapter 11, "Load Balancing," but most of us rarely are called upon to implement such systems. Instead, we are all architecting high-bandwidth networks within our local data centers and the backbones that service individual locations in the enterprise.

As we consolidate core network services, such as email, print, and file in corporate data centers, bandwidth becomes a critical focus of the solution, as much as the availability of the service. When consolidating, you offer more clients service on a smaller number of servers; so, you need to architect solutions that cannot fail. At the same time, having a highly available print server that cannot push print jobs over your network is pointless. So, before you go spending a lot of time and effort on your clusters to serve a bunch of remote sites, spend some time making sure you have the bandwidth faucet well greased.

Ethernet

First, a primer on the new gigabit technologies takes root. It's been more than 30 years since Ethernet became the de facto networking protocol, and it is very rare these days to find Token Ring, *Fiber Distributed Data Interface* (FDDI), and *Asynchronous Transfer Mode* (ATM), especially in server and client networks. Since Xerox introduced Ethernet in the 70s, we have seen it mature from standard Ethernet at 10 megabits per second (Mbps) to 100Mbps.

Most corporate backbones are still running at 100Mbps (Fast Ethernet) because of the cost and past slow vendor support for gigabit technology. However, gigabit is now much cheaper and easier to implement than ever before, with many standard NICS and switching supported and huge interest from vendors. Gigabit Ethernet and beyond is here.

Looking back at Fast Ethernet, you can see exactly where gigabit technology is going to go, and very soon it will completely replace Fast Ethernet as the simple, cost-effective backbone option. We like to think of Gigabit being bandwidth for the backbone, but Gigabit NIC prices are dropping so fast it will not be long before gigabit extends to the desktop and becomes the de facto standard for corporate networking. As of mid-2004, a Gigabit NIC can be purchased for less than a hundred US dollars.

How do we compare Gigabit Ethernet (or GigE) to its predecessors? Fast Ethernet is the foundation for Gigabit Ethernet. It sits atop of the Ethernet protocol but ramps up the transmission speed almost ten times to 1000Mbps, or 1 gigabit per second (1Gbps). The protocol is not so new either; it was standardized in mid-1998, but it took until late 2003 to drop enough in price to warrant fast adoption.

How does it work? To ramp up from 100Mbps, Fast Ethernet engineers made a number of physical interface changes to the interfaces of the day. Only from the data link layer upward does Gigabit Ethernet differ from Fast and standard Ethernet. Engineers focused on merging two Ethernet technologies: IEEE 802.3 Ethernet and ANSI X3T11 *Fibre Channel* (FC).

For networks that grow to more than a single floor in a building, or grow beyond 48 devices, GigE presents the ideal backbone. In high-availability environments, it has become essential to have GigE on the backbone. For example, print-server clusters under Windows Server 2003 have become highly efficient and can now host thousands of queues. Print spooler files are growing larger than ever. Thus, it goes without saying that while you might have seen sufficient throughput in the earlier years with Fast Ethernet coming out of Microsoft Cluster Server, today's needs and requirements are very different, especially when it comes to pushing print and spool files over the WAN.

Without spending too much time on the fine points of GigE, today's new copper interface cards present low-cost entries into the gigabit market. In architecting your backbone, you can place high-end gigabit switches that interconnect with fiber at the root of the architecture—connecting buildings, floors, and sites to each other. Lower down the backbone, your 24- and 48-port switches can be the cheaper copper GigE ports for direct connection to your servers.

Servers today from all vendors now come standard with 10/100/1000 GigE interfaces on the main board with copper interfaces, which still use RJ45 plugs and the standard CAT5 cable. This enables you to connect standard Ethernet cable to the interfaces, connecting the servers to the switches for server-to-server and server-to-client communications. This is demonstrated in Figure 4.2.

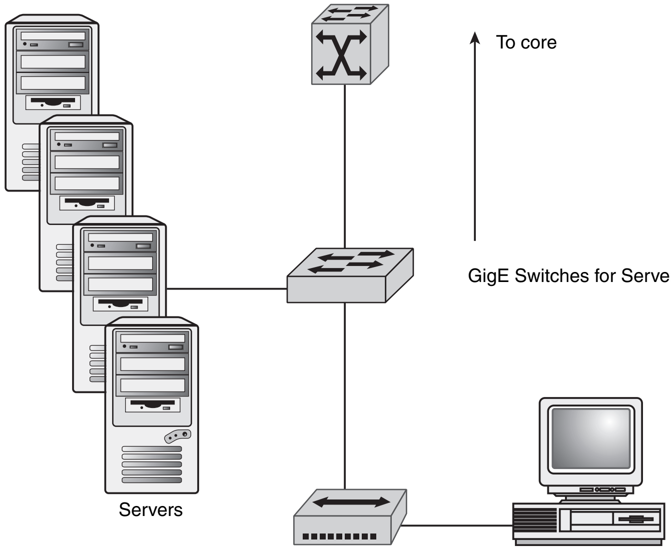

Figure 4.2 GigE on the corporate network.

Figure 4.2 shows your desktop-to-server-to-backbone architecture scaling from 10/100 at the desktop to 100Mbps up the riser to 1000 Mbps in the data center. Where do we go from here? Well, it's worthwhile to watch the advent of 10Gbit Ethernet, which is fast coming into focus on the horizon, sort of where 1Gbit was back in 1998. What this tells you is that it's not worth your time to consider architecting any longer for Fast Ethernet.

Can you have too much bandwidth? Yes, you can. The current technology in or at the server is slightly behind the curve of the advancement on the network. Internal components, computer busses, network interfaces, processors, memory, and the software cannot actually receive data thrown at it at the increasing speeds on the network.

Thus, thinking about doing anything faster than GigE at the present time may not make much sense. In fact, some systems now have technology built into it that becomes aware of data coming in too quickly by communicating with the switches and interfaces to delay transmission if the receiving host is having a tough time getting it all in.

No doubt, we will one day be plugging away at 10GB plus, but for now, sensibly architecting at 1Gbit is sufficient for the most demanding of high-availability requirements. Figure 4.1 also illustrates separating the network into three bandwidth levels: 10Mbit or 100Mbit at the desktops, 100Mbit at the risers, and 1000Mbit between the servers and connecting to the risers.

What to Look for in Network Interface Cards

Paramount in network architecture, and of prime importance when architecting a high-availability network, is making decisions about the NICs you are going to buy. Today, most servers come with at least two NICs on the main system board; so, you don't have many options but to go with what they give you. Of course, you can also choose to add additional PCI-based NICs installed at the factory (don't waste time buying them separately). Also, the high-density systems and blade servers don't have room to install NICs. Good news is all vendors put a lot of thought into the NIC and go with top-class products from the likes of Qualcom or Intel.

When it comes to larger hosts that require more than two NICs, for both LAN communications and NICs for cluster interconnects, it is up to you to decide what you need. Many engineers like to disable the onboard NICs for fear their demise requires you to replace the entire system board instead of just pulling the NICs. However, the reputation NICs have for failing has long since played out and, in any event, often the cause of NIC failure stems from handling NICs in unclean and statically charged environments.

When considering NIC, first consider the environment into which the servers are going. If you have the luxury of architecting a new backbone along with the high-availability network, you are able to decide on GigE from servers to the core and intermediate switches and install gigabit NICs from the outset.

You also need to decide on fiber versus CAT 5 (copper). CAT 5 NICs for the GigE network have come a long way, and the honor of pushing high bandwidth limits are no longer fiber's alone. Fiber is, however, more expensive in a number of areas. First, the cards themselves are expensive, although not that much more nowadays than copper GigE NICs; so, they won't break the budget of a HA data center. Second, fiber is delicate and requires a lot more care than CAT 5; pinch a CAT 5 cable and most of the time you can save the cable. A tiny pinch on fiber is likely going to “break glass” and render the cable useless. A fiber cable either works or it is broken.

NOTE: If a fiber device —NIC or adapter—stops working, do not—I repeat, do not—stare into the fiber optics to see if you see any light pulses. If the device does not have automatic shut off (*optical fiber control* [OFC]) you will damage your cornea permanently. FC emits laser beams. You will not burn a hole through your eye and out the back of your head, but you may have permanent eye damage that will only become noticeable later in life.

Fiber cables are much more expensive to replace than CAT 5. Thus, with cable, you have a much higher overhead when it comes to maintenance. Finally, provisioning every server with LAN-side fiber can be very expensive because the server must run fiber switches, and fiber switches result in you sharpening your budget pencils. Remember that it is not so much about NIC cost but about the topology. Rule of thumb: Fiber topology is more expensive and harder to maintain than CAT 5.

If you are going to use fiber for the backbone, put fiber at the switch-to-switch level. As indicated in Figure 4.3, collapse down to a copper GigE switch using a single fiber port on the CAT 5 switch, and then connect all the servers to the CAT 5 switch. Finally, building the SAN network, which we discuss later in this chapter, may require fiber provisioning; so, budget for fiber-based *Host Bus Adapters* (HBA NICs) and fiber switches for the SAN solution.

Also, remember that however you architect your network, provision for future bandwidth needs and scale-out. That copper switch you thought would do the trick for the backbone may have to be junked for a new fiber-based switch down the road. The network architects rule: Come up with a figure you think the network engineers (and purchasing) can live with, then add 20 percent for growth and another 20 percent if you need room for underestimation.

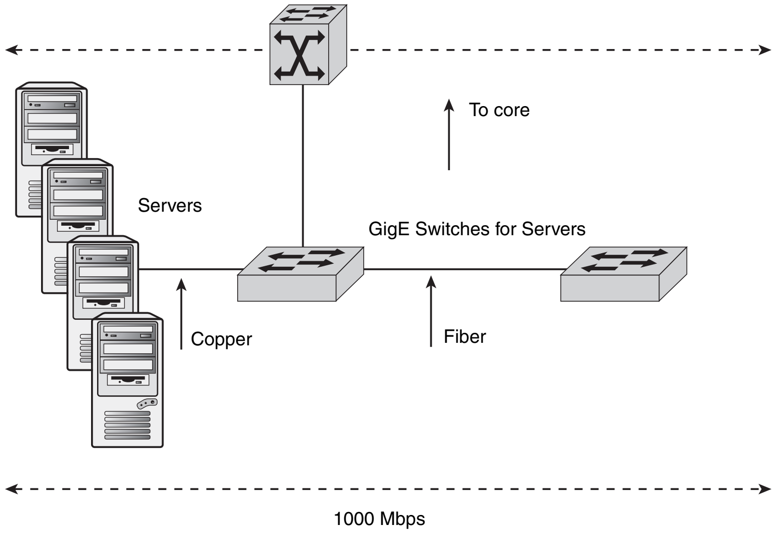

Figure 4.3 Fiber versus copper on the corporate network.

Hubs, Switches, and Routers

You may wonder how hubs are used in high-availability architecture. It is true that hubs are not intelligent like switches. Servers and devices talk to each other in hubs because the hub is a simple device in which all ports pass and receive information. You can think of the hub as a small communal swimming pool that everyone shares. While ten people in the pool may be okay, going above 30 or 40 starts to detract from the communal experience. Everyone starts bumping into each other and no one can carry on a private conversation. In essence, hubs can become very noisy, and noise on the network detracts from high-availability requirements.

There was a time when the cost of switches was such that hubs were attractive in small data centers or small offices. But nowadays you can buy a small switch the size of a small book and it costs the same and even less than a hub. Hubs are useful for small workgroups where network drops are hard to install. Everywhere else on the corporate network, switches should do the work.

Why do we even need to keep manufacturing hubs? In high availability architecture, hubs can play a very important role for hosting your cluster nodes. This is discussed in the next section.

Switches are another matter and came about as a result of many limitations encountered with hubs. Switches are able to forward packets directly to an individual *media access control* (MAC) address, direct to a port. The more expensive switches are also highly programmable, enabling you to partition the switch into routable virtual networks, otherwise known as virtual LANs or VLANS. They also offer security, monitoring, and so on. Let's now look closer at how switches, hubs, and routers are used in high-availability or load-balancing solutions.

Layer 2 Switches

The layer 2 switch, as discussed a few moments ago, is able to filter packets according to the MAC address that is interested in the packet. However, the layer 2 switch marries a port on its matrix with the MAC address that is able to talk to the server that's connected to it.

Servers communicate as usual, but the layer 2 switch forwards the packet directly to the machine for which the packet is intended. The switching matrix thus routes the packet directly from one port to another. The packet may be coming in from a port that leads to the backbone and down to the switch hosting the client machines, or it may be coming directly from another server connected to the switch.

Layer 2 technology can cause problems for both NBL and failover clusters because of the way clients connect to the virtual IP and MAC addresses used by these HA solutions. To understand the problem and how to provision for it, we must first discuss how the layer 2 switch works; however, we do not delve deeply into this subject.

On your network, when servers talk to each other and clients talk to the server by connecting to the layer 2 switch, the packets are not broadcast in a packet storm like they are in a hub. Instead, packets are switched directly to the machine for which the packet is intended, just like a telephone switch. Now the *address resolution table* (ARP) comes into play. If the switch does not have information about a particular MAC address in its ARP table, it broadcasts the address to all ports in the switch until a reply is obtained that claims to be the destination.

In NLB clustering arrangements, as you will later learn, all the servers in the cluster can claim the same IP and MAC address through the address virtualization process. Your switch may attempt to assign the virtual MAC address to a particular port on the switch. This causes the requests for the address to be targeted to only one node in the cluster instead of being load-balanced across all the cluster nodes, which defeats the purpose of the NLB cluster.

With fail-over clustering, a slightly different problem can occur. With all Microsoft fail-over clustering technology, the servers perform what is called a *gratuitous Address Resolution Protocol (ARP) request* when a failover occurs. The gratuitous ARP request is performed to update the network (computers, routers, and switches) with the MAC address that now owns the virtual server's IP address.

The problem is that your device may not forward the gratuitous ARP request to other devices. So, devices on the other side of the switch or router end up with the wrong information in their ARP tables and, thus, incorrect MAC address for the virtual server that has failed over. Most likely the situation corrects itself after a router or switch learns of the failure and updates its ARP table by performing a broadcast. By default, routers and switches are configured not to forward ARP traffic between subnets. This is done to prevent the occurrence of what is known as an *ARP storm*.

In a failover cluster, each computer node has a network adapter attached to the LAN, and each cluster node has its own IP address, network name (NetBIOS name), and MAC address. The virtual server also has a resolvable IP address and network name, but it uses the MAC address of the cluster node that is the current owner of the virtual server resources.

So, when failover takes place, the node receiving IP resources sends the gratuitous ARP request to update all devices with the new MAC address assigned to an existing IP address. If a switch or router does not pass the updated MAC-to-IP address mappings, network devices on other segments contain the old MAC address for the cluster node that is down and clients no longer are able to communicate with the virtual server. Essentially, the clients are waiting for the data train on the wrong platform.

For failover clusters, you should configure your switches and routers as follows: Have them forward the gratuitous ARP requests across all networks, so all devices receive the updated MAC-to-IP address mappings. This is done inside your switch or router; so you need to telnet to the router or log in to its private Web site, if it has one, to make the changes.

When configuring your systems for NLB clustering, you can configure the cluster to use *Unicast* or *Multicast* mode. Unicast is the default and, essentially, connects the IP address to a unicast MAC address. Unicast mode uses a MAC address mask, a substitute address, as its source MAC address when it sends the packets to the switch. This behavior prevents the switch from updating its ARP cache with the address and causes it to forward the packet to all ports in the switch, which is what you want.

Another lesson learned by seasoned NLB cluster architects is to connect all the NICs in the NLB "farm" to a hub instead, and then connect the hub to the layer 2 switch. The switch then records the virtual MAC address and associates it with the port connecting to the hub. The packets then arrive at the hub, which broadcasts it to all servers connected to the hub. However, if you use a hub, then you need to disable the MAC address masking behavior because now you *want* the ARP table to be updated with the MAC of the port that connects to the hub. To do this, you need to change a registry key of the Windows Server operating system. The key is found at the following location:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WLBS\
Parameters
```

You can find the following default information at this registry folder:

MaskSourceMAC
Data type = REG_DWORD
Range = 0 or 1
Default = 1

Change the value to **0** in Range. Setting the Range value to 1 masks the source MAC address in outbound packets, preventing switches from learning and forcing them to broadcast packets for unknown addresses to all ports. Setting this value to 0 disables masking of the MAC address.

You can also choose to disable port blocking (this enables unknown unicast and multicast packets to flood the specific ports). Known as Multicast mode, it makes use of a multicast MAC address for the virtual address and resolves it to a unicast IP address. This causes the switch to ignore recording which MAC address is associated with a single port. Instead, the switch now behaves like a hub and broadcasts the packets to all ports.

While the latter option is a simple quick fix, it is not a better option and detracts from the purpose of layer 2 switches. The switch does not associate the multicast MAC addresses to a port and, thus, sends frames to the MAC address on all the ports. IP Multicast pruning implementations cannot limit the port flooding and you may have to configure a virtual LAN as a solution. Multicast provides no advantage over unicast from the switches perspective. Performance decreases and will burden the layer 2 topology your networking team has spent time perfecting. Connecting the NLB hosts to a hub may be your best option.

Layer 3, Layer 4, and Beyond

The higher layer switches work in similar fashion to layer 2 switches, but the MAC address is not included in the switching equation. Instead, routing is based on IP addressing. This can be a problem for clustering because you now do not have the option of masking the MAC address as discussed earlier.

The best, and only, option in layer 3 and above switching environments is to connect your cluster nodes, NLB, and fail-over to a hub and connect the hub to a port on the layer 3 switch. If for some reason you must connect the nodes directly to a switch, then you have to provision for an intermediate layer 2 switch.

NOTE: The issues described are obviously pertinent to larger networks that have multiple switches and routers spanning the corporate network.

Routers and Routing in High Availability Architecture

Routers figure in all high-availability scenarios that cater to multiple subnets and network segments. Now that geographically dispersed clustering is becoming easier to achieve, routers actually sit between nodes of your clusters. In fact, lately it has become harder to discern the difference between switches and routers because many modern switches support router modules that are no bigger than a port on the chassis.

Some of the things you need to watch for in routed networks supporting high-availability architecture are that clients on the outer segments can connect back to a failover server or connect again upon failback. Again, the culprit that causes clients to not find the virtual servers is the ARP. The solution is to ensure routers are configured to bridge their ARP broadcasts across all the routers on the enterprise network.

Also, the use of address masking can cause problems at the router level. Routes typically do not record the ARP entry for the virtual server; so, you may need to add the entry to the router's table manually.

TIP: For better performance, configure the default gateway on your node's virtual interface only, and leave it out of the configuration dialog for the actual corporate network NICs in the machine.

Using Hubs for Failover Interconnects

Earlier we discussed the role of hubs in the high-availability architecture with respect to connectivity on the enterprise network. However, hubs can play an important role in the interconnect architecture for your failover clusters.

Later, in Chapter 8, "High-Performance File-Server Solutions," we talk about configuring failover clusters and, more specifically, configuring the interconnects. When a resource fails on a cluster node, the passive node takes over operations. It does this because it learns of the failure over a private network between the nodes in a cluster. How and why this works is discussed in Chapter 8, but for now, know that you need to set up a private network typically isolated from the main LAN or corporate network exclusively for node-to-node communications. When only two clusters are installed as part of a fail-over cluster in the same rack, the network is established using a crossover cable between the two nodes. A private IP subnet is used to enable all the nodes to talk to each other's IP address.

However, as soon as you add more nodes to the cluster, or add more clusters to the architecture, you have to install a small hub to cater to the interconnected private network. Using a hub also enables you to easily troubleshoot the interconnect traffic. You can join a client monitoring device or machine to the hub and capture the packets being sent and received on the interconnect network. The topology for hubs in the interconnect design is illustrated in Figure 4.4.

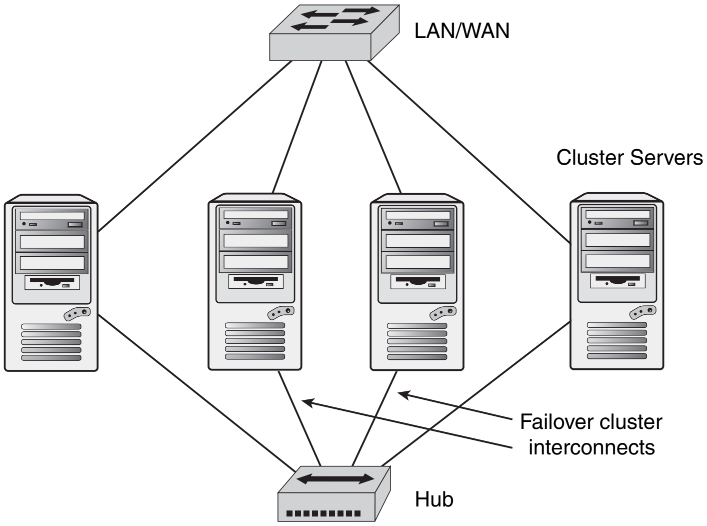

Figure 4.4 Using a hub for your interconnect network.

Almost all cluster interconnect topologies that have started as a single CAT 5 cable between two servers have ended up being removed in favor of a small hub. Remember one of the golden rules of architecting networks is to plan for the future, which is just around the next corner. So, place a cheap but reliable hub into the cluster rack from the get-go and save yourself the trouble of trying to add it later when the rack is full.

SAN Topology Primer

In Chapter 3, we touched on the subject of SAN networking and fabrics. Let's now look deeper into the subject from a SAN network architecting point of view. If you are new to SANs, then it is important to know the subject is vast, the technology is leading-edge, and often bleeding-edge. The entire subject cannot be covered adequately in one or two chapters. However, SANs are so key to high-availability solutions, we offer some insight and guidelines to get your taste buds working.

The difference between a SAN and the corporate network, from the server's point of view, is on the SAN, servers never talk to each other, they only communicate with the SAN. The only role the SAN network has is to provide the high-speed channel or data bus between the server and its hard disk drives. Sure, there is a lot of technology in between the SAN and the server, but the server only sees its LUNs and volumes as if they were local to the machine at the end of the standard SCSI cable. This is fundamentally what a SAN is—a network, instead of an 80-pin ribbon cable, that provides access to storage (see Chapter 3).

Servers interface to the SAN network with a NIC called a *host bus adapter* (HBA). While some SAN HBAs use copper media, fiber is more suitable, scalable, and functional. Fibre Channel (not fiber) SANs are still too expensive for small networks (see Chapter 3 on the subject of SCSI and ISCSI). However, if you are building a SAN that serves several clusters (and provides storage for hundreds and thousands of users), then you almost certainly should deploy Fibre Channel.

Our experience with Fibre Channel in a number of locations is that it is not a very difficult technology to deploy. However, because it is rarely deployed outside of the SAN implementation, most IT shops do not have in-house expertise on the subject. The lack of knowledge in deploying Fibre Channel for a SAN is one of the greatest contributors to the cost of your high-availability project.

Until you have expertise in-house, it is important to add an FC consultant to the project, even if just to get you up-to-speed. Alternatively, spend a few thousand dollars and get someone from the network team out to a SAN implementation course. Most of the SAN vendors, like private-label EMC or Hitachi SANS provide consultants. These come at great cost (typically no less that $10,000 for a two-day SAN project). They don't give you a lot of time, and they leave you with little transferred knowledge and a lot of PowerPoint slides.

Fibre Channel

To start you on the road to your SAN, let's talk a little about Fibre Channel. While FC is used in standard enterprise networking, it is ideal for SAN implementations for the following reasons:

- FC supports non-network protocols, such as SCSI-3 or video, and is thus ideal for data access.
- FC supports hot pluggable devices. This is critical for HA architectures.
- FC is a low-latency connection and connectionless service.

- FC offers superb *quality of service* (QoS) features. These include fractional bandwidth allocation and connections-specific bandwidth guarantee. This is useful, for example, in allocating bandwidth to high-performance backups to SAN-attached tape drive units when access to the file servers is at its lowest.
- FC offers a band rate of 1Gbps/2Gbps, with the ability to scale to 4Gbps.
- FC supports point-to-point, loop, switched, or hub topology.
- FC offers guaranteed delivery (both GigE or ATM do not offer this).
- FC offers data transfer with zero congested data loss.
- FC has a frame size that is variable to 2KB.
- FC supports both glass and copper media. Fibre Channel once only supported fiber optic cables, but, as discussed earlier, support for copper cable is now mature.

FC is actually a standard that defines a stack of protocols used for data transfer. Its upper-layer protocols include SCSI, IP, 802.2, IPI, and HIPPI. As data travels up or down the stack, it is mapped to the FC channel protocols for transfer. The layers are illustrated in Figure 4.5.

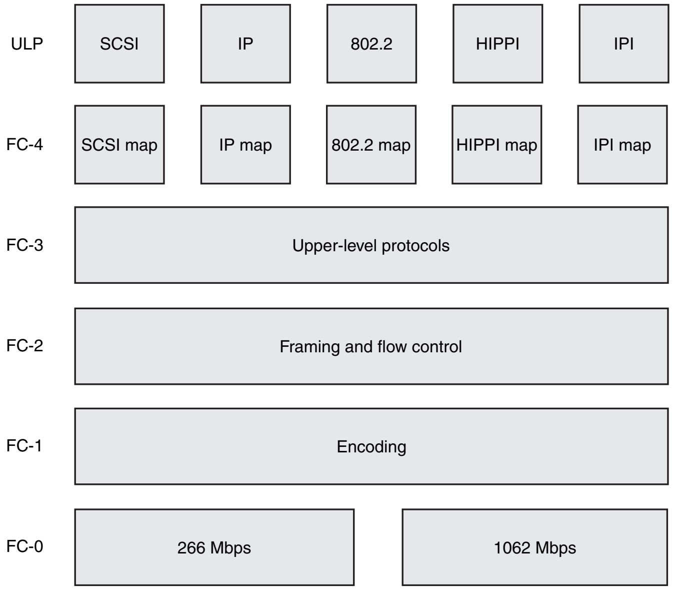

Figure 4.5 The Fibre Channel protocol stack.

When a server requests data from a disk, the SCSI-3 protocol essentially decides how the data is going to be retrieved. The data is then passed to the SCSI map of the FC protocol (see FC-4 in Figure 4.5). The data is then segmented into frames at FC-2, encoded at FC-1, and then onto FC-0 to begin its journey across the glass or copper media.

The data transfer is made possible by support for the SCSI-3 protocol at the adapter and server level. SCSI-3 defines what is known as serial SCSI. Instead of sending the data across parallel conductors in the SCSI ribbon cable, the data is sent in one transmission stream through the single line in the FC cable.

The HBA cards that you plug into your servers are the devices that take care of the aforementioned details for your servers. So, the server can talk the language they are already familiar with (SCSI) and not care how the data is sent and retrieved from the SAN and the disk array on the other end of the cable.

This magic over FC is replicated on the other side of the SAN fabric, the so-called weave of network technology that ties the SAN together (discussed in more detail in the next section). The disk arrays themselves can be thought of as servers, but their job is to handle the requests for data, fetch and store data from the arrays, and transfer them across to the servers on the other end of the fabric. SAN array processors use UNIX- or Linux-like operating systems (only accessible to the manufacturers) to do the data handling and disk management. The processor on the array controller is no different from the processor that resides in your server.

SAN Topology

When architecting a SAN network, you have the choice of three FC topologies: point-to-point, *FC arbitrated loop* (FC-AL), and fabric. The cheapest SAN topology is point-to-point and fabric is the most expensive, and most functional, of all three. We focus on fabric because it is by far the most desirable topology. But first let's discuss ports.

Ports

There are three types of ports we talk about when architecting SAN topology: N_Ports, which are ports on a disk or server system; F_Ports, which are ports on a SAN fabric switch; and L_Ports, which are ports that function in an arbitrated loop.

N_Ports only communicate with other N_Ports in point-to-point topology, or they talk to F_Ports on a SAN fabric. L_Ports alone do not exist. While they are built to function in arbitrated loops, they need to be combined with N_Ports or F_Ports to create FL_Ports. In other words, an FL_Port on a switch can connect to a node that can function in an arbitrated loop.

We also talk about E_Ports, which stands for *expansion* port. E_Ports connect switches to each other to scale the SAN fabric. Then there are G_Ports, which are *generic* ports on a switch that can function as FL_Ports or F_Ports depending on what they connect to.

Point-to-Point Topology

Point-to-point topology is comprised of two N_Ports that talk over a direct connection. This is illustrated in Figure 4.6, showing a single host server connected to an array over FC. FC-AL, pronounced "f-cal," actually came after fabric topology because, in the early days of SANs, fabric was prohibitively expensive for many small to medium implementations. FC-AL was thus offered as a cheaper alternative to fabric, but without the limitations apparent in a point to point-to-point topology.

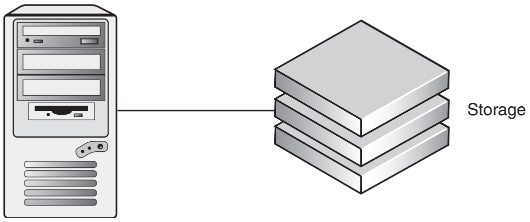

Figure 4.6 Point-to-point topology.

For obvious reasons, point-to-point topology is hardly suitable for larger high-availability solutions but is far better than SCSI arrays for clustering solutions.

FC-AL

The so-called "f-cal" loop is illustrated in Figure 4.7, where the topology is an actual loop. This topology requires the receiving and transmitting wires of the interconnecting cables be split. This was achieved with the invention of HBA adapters for FC-AL networks that actually split the fiber-optic cables. This topology is illustrated in Figure 4.7.

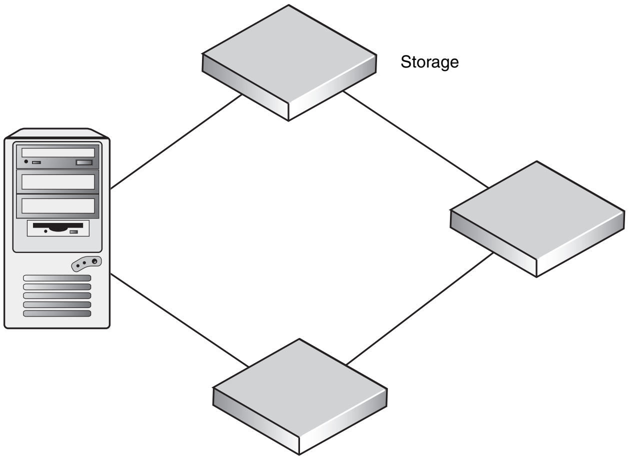

Figure 4.7 FC-AL topology.

It should be obvious from both Figures 4.6 and 4.7 that the limitations of FC-AL are distance and the number of supported devices. Another, more profound, limitation of the "loop" illustrated in Figure 4.7 is that the death of one HBA takes out the entire loop. This is unacceptable for any HA architecture. The advent of FC-AL switches allows you to overcome this limitation by forming a star topology. This is illustrated in Figure 4.8.

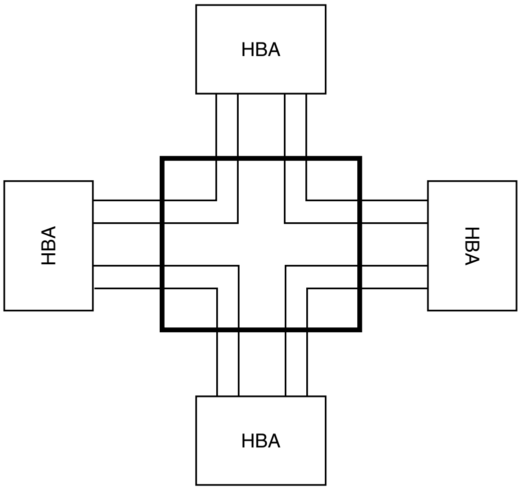

Figure 4.8 FC-AL hub.

In the star topology, you connect each NL-Port to an FC hub. The internals in the hub take care of the loop for you. Another factor of FC-AL over fabric is the arbitration factor (the "L" and the "A" in the acronym, as in *arbitrated loop*). Because FC-AL is a shared resource, nodes that wish to transmit have to arbitrate for that right.

Fabric

Fabric topology makes use of a special switch where each N_Port plugs into the F-Port on the switch. Each node is then assigned an address by the switch. To participate in the fabric, the node must log in to the fabric. Each address in a fabric is a 24-bit value; so, you can easily scale to 16 million devices on your SAN (not that the array itself would be able support that many).

The fabric topology thus allows the node that logs into the port to use the entire bandwidth of the port, which is similar to how the layer 2 and layer 3 switches work as discussed earlier. SAN fabric technology is fast making FC-AL technology obsolete, and fabric is quickly becoming the standard.

One of the advantages of fabric is the switch technology employed in the fabric is much more sophisticated and functional than a FC-AL hub. A fabric switch offers true switching and many other sophisticated features. One of the most appealing is the ability to run software that manages the switch for you. With switch software you create "zones" that enable certain servers to see only the volumes or devices you want them to see. For example, you can place a tape library in a zone and configure the switch to enable the servers to see only the tape library in that zone and not any of the other servers. The zoning, thus, allows the servers to see the tape drive and think they have exclusive use of it; that it is somehow locally attached to the server. More about this topic in the following section, "Zoning."

NOTE: As mentioned in Chapter 3, iSCSI can use standard IP switches for its shared storage topology, but fabric SAN switches are specially built for SAN topology, and they are more expensive. The three leading manufacturers of fabric switches are Brocade Communications Systems at No. 1, McData Corp at No. 2, and QLogic Corp. at No. 3.

The good news is fabric equipment is becoming cheaper by the day. The cost of the switches, from the likes of Brocade and McData, are dropping, as is the cost of the HBAs and the SAN arrays themselves.

Of course, if you want to start experimenting or have a small number of nodes to install on a high-availability SAN, FC-AL can be done much cheaper. Entry cost for FC-AL can be astonishingly low in comparison to fabric, especially for a lab setup. FC-AL hubs supporting up to nine devices can be found on the Internet for less than $200. The array controllers or processors are going to be the most expensive components, but you can still put up a small SAN on FC-AL for less than $5000. At the time of this writing, EMC's baby fabric SAN, the CSX200, is around $20,000 for a few nodes.

Now let's look at zoning in more detail.

Zoning

As mentioned in Chapter 3, zoning enables you to partition a SAN fabric in such a way that devices on the fabric are isolated from each other and only the servers zoned with certain devices get to use those devices. A very different scenario exists on legacy SCSI storage solutions, which allow all the servers on the SAN to see each other and all devices connected to the SAN. Unrestricted access has a number of implications for building in reliability, maintainability, and monitoring on your SAN.

Zoning throws off a lot of network engineers new to SAN topology; so, it's a good idea to liken it to technology with which network engineers are intimately familiar (and passionate about)—VLANs. Zoning is a lot like virtual LAN configuration. Network administrators partition a network using VLAN software in switches to group devices together to form collections of devices, protocols, ports, and addresses in the switch. A VLAN-capable switch can be easily segmented into, say, four VLANs, each VLAN routable to the other VLANs such that the switch appears to the network and the devices as four distinct networks.

VLANs can also span many switches, and a matrix of VLANs in an enterprise spanning multiple switches makes life easier for network engineers. A good example is how you can move a port from one VLAN to another without ever having to physically unplug a cable from any device. A port in one VLAN is not accessible from any other VLANs.

Zones are similar to VLANs in that zones can also span multiple fabric switches and control how traffic is distributed around the switches. The difference is that zoning is a way of partnering servers with devices on the SAN, such as LUNs and tape drive systems. This is illustrated in Figure 4.9.

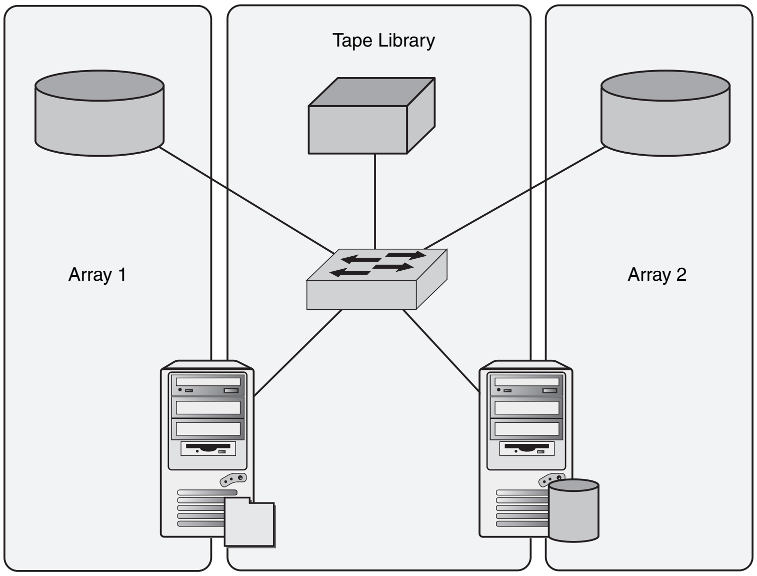

Figure 4.9 Zones associating servers with arrays.

Fortunately, zoning software has matured, but setting up the zones requires some assistance from the fabric expert, or you need to get yourself off to some intense training. Zoning is a knowledge you use time and time again, as long as you manage a SAN.

Architecting SAN Topology for High Availability

As a rule of thumb, when architecting and designing for the SAN network, the corporate network, the interconnect network, or the backbone, always determine the number of ports you need and then, as a rule, add 20 to 30 percent for expansion. Experience has shown that no matter how careful you study your network, there are some factors that always cause you to underestimate your requirements.

The larger your network, the more planning you need to do. Always start with your servers and high-availability needs. Determine the devices you are going to be installing; understand fully your backup device needs and your switching and zoning requirements. Leave port requirements for last.

One of the key reasons a SAN is so desirable in a high-availability environment is that you can design it to avoid a single point-of-failure in the path between the servers and the storage array. This is a problem in SCSI storage where hardware and network failure can trash the entire storage solution, and kill all access to the storage from the servers. Before you go architecting the SAN topology like a SCSI storage solution, consider an architecture that provides multiple paths between servers and devices.

Figure 4.10 illustrates the typical single point-of-failure SAN fabric.

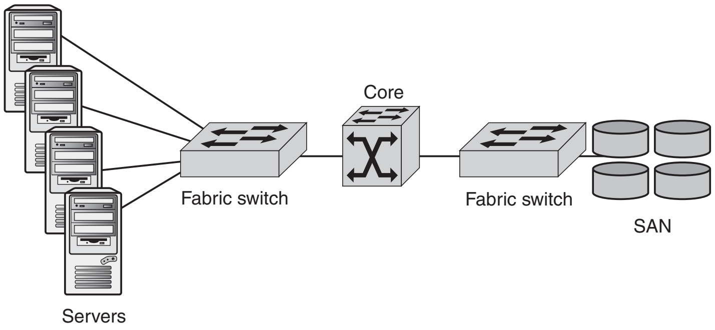

Figure 4.10 Single point-of-failure SAN fabric.

In Figure 4.10, servers only have one route to the storage array and the tape device. Thus, if you count the devices and ports between the servers and the arrays, it is not difficult to determine the number of points of failure that have a potential to bring down your cluster.

In Figure 4.11, we illustrate a multi-path topology for a highly available SAN fabric. You still have the same number of failure points, but as soon as a failure occurs in one path, you have yet another path to keep up the application.

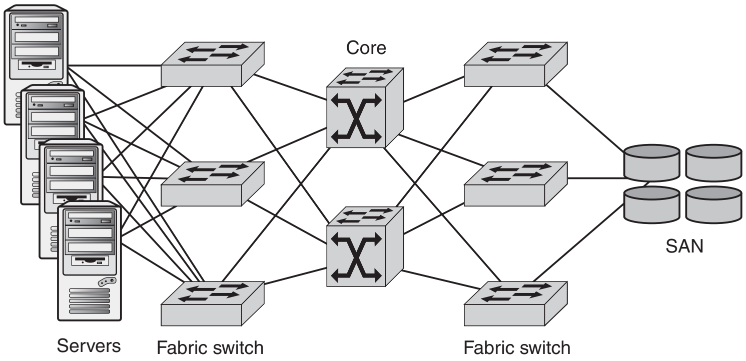

Figure 4.11 Multipath architecture for a high-availability SAN fabric.

Notice, however, that you still have only one SAN in the architecture. So, it is still possible to lose the system if the SAN itself, its processing unit, crashes. You do not lose the array if disks or switches go down because disks are configured in RAID arrays, and you typically install two or more switches in the SAN topology.

But a SAN is driven by a self-contained server, with a UNIX-like operating system, that is not crash proof. You cannot claim to have a SAN that is without a single point of failure unless you are replicating the data off the SAN to a backup mirror SAN somewhere. There are several technologies that can mirror or replicate volumes from one storage device to another to achieve a failover storage device. We will further explore the subject in Chapter 10, in architecting high availability storage for Microsoft Exchange 2003.

Time-Out

This chapter provided some food for thought on the networking subject. There is a lot to consider when it comes to crafting a sound network for a highly available solution. However, all too often network architecture is either ignored or undertaken without the consideration of the HA system needs.

When you receive a mandate to craft a system that must provide four or five nines availability, the first consideration is where to put your system. Very few businesses are fortunate enough to have a hardened server room that has fire suppressions systems, unlimited and uninterruptable power supply, generators, cooling systems, security, and more. So, the first item on the design plan is to locate the systems off-site in a data center or disaster recovery location. Depending on your design, the data center can be either primary operations or standby in the event of failure at the office.

In today's highly distributed and connected world, it makes sense to locate the production system in the hardened data center where your partners, clients, and users are primarily connected. Back at the office, you can still use local domain controllers, file and print servers, email, and database systems for local access, but you can mirror these systems with the systems at the data center in the event the systems at the office fail.

If the SLA mandates that employees should be able to continue working, even in the event the primary employee workplace is taken offline or shut down, then all employees and users should be able to telecommute to the production systems in the data center and continue working without leaving their homes.

Installing a system in a data center is going to place a lot of attention on the network architecture. You need to consider how the main and branch offices are going to connect with the data center. Are you going to connect with a dedicated private network, or use a VPN over the Internet? Are you going to route the traffic, or do you need to craft a flat address space to meet your needs? What about firewalls, routers, and network address translation? In Chapter 9, we investigate the architecture for such solutions using SQL Server applications in our design.