,ch01.24172 Page 1 Friday, November 9, 2001 12:23 PM

CHAPTER 1

Networking Objectives

The American architect Louis Henry Sullivan described his design philosophy with the simple statement "form follows function." By this credo he meant that a structure's physical layout and design should reflect as precisely as possible how this structure will be used. Every door and window is where it is for a reason.

He was talking about building skyscrapers, but this philosophy is perhaps even more useful for network design. Where building designs often include purely esthetic features to make them more beautiful to look at, every element of a good network design should serve some well-defined purpose. There are no gargoyles or frescos in a well-designed network.

The location and configuration of every piece of equipment and every protocol must be carefully optimized to create a network that fulfills the ultimate purposes for which it was designed. Any sense of esthetics in network design comes from its simplicity and reliability. The network is most beautiful when it is invisible to the end user.

So the task of designing a network begins with a thorough study of the required functions. And the form will follow from these business requirements.

Business Requirements

This is the single most important question to answer when starting a network design: why do you want to build a network? It sounds a little silly, but frequently people seem confused about this point. Often they start building a network for some completely valid and useful reason and then get bogged down in technical details that have little or nothing to do with the real objectives. It is important to always keep these real objectives in mind throughout the process of designing, implementing, and operating a network.

Too often people build networks based on technological, rather than business, considerations. Even if the resulting network fulfills business requirements, it will usually be much more expensive to implement than is necessary.

If you are building a network for somebody else, then they must have some reason why they want this done. Make sure you understand what the real reasons are. Too often user specifications are made in terms of technology. Technology has very little to do with business requirements. They may say that they need a Frame Relay WAN, or that they need switched 100Mbps Ethernet to every desk. You wanted them to tell you why they needed these things. They told you they needed a solution, but they didn't tell you what problem you were solving.

It's true that they may have the best solution, but even that is hard to know without understanding the problem. I will call these underlying reasons for building the network "business requirements." But I want to use a very loose definition for the word "business." There are many reasons for building a network, and only some of them have anything to do with business in the narrow sense of the word. Networks can be built for academic reasons, or research, or for government. There are networks in arts organizations and charities. Some networks have been built to allow a group of friends to play computer games. And there are networks that were built just because the builders wanted to try out some cool new technology, but this can probably be included in the education category.

What's important is that there is always a good reason to justify spending the money. And once the money is spent, it's important to make sure that the result actually satisfies those requirements. Networks cost money to build, and large networks cost large amounts of money.

Money

So the first step in any network design is always to sit down and list the requirements. If one of the requirements is to save money by allowing people to do some task faster and more efficiently, then it is critical to understand how much money is saved.

Money is one of the most important design constraints on any network. Money forms the upper limit to what can be accomplished, balancing against the "as fast as possible" requirement pushing up from below. How much money do they expect the network to save them? How much money do they expect it will make for them? If you spend more money building this network than it's going to save (or make) for the organization, then it has failed to meet this critical business objective. Perhaps neither of these questions is directly relevant. But in that case, somebody is still paying the bill, so how much money are they willing to spend?

Geography

Geography is the second major requirement to understand. Where are the users? Where are the services they want to access? How are the users organized geographically? By geography I mean physical location on whatever scale is relevant. This

book's primary focus is on Local Area Network (LAN) design, so I will generally assume that most of the users are in the same building or in connected building complexes. But if there are remote users, then this must be identified at the start as well. This could quite easily spawn a second project to build a Wide Area Network (WAN), a remote-access solution, or perhaps a Metropolitan Area Network (MAN). However, these sorts of designs are beyond the scope of this book.

One of the keys to understanding the local area geography is establishing how the users are grouped. Do people in the same area all work with the same resources? Do they need access to the same servers? Are the users of some resources scattered throughout the building? The answers to these questions will help to define the Virtual LAN (VLAN) architecture. If everybody in each area is part of a self-contained work group, then the network could be built with only enough bandwidth between groups to support whatever small amounts of interaction they have. But, at the opposite extreme, there are organizations in which all communication is to a centralized group of resources with little or no communication within a user area. Of course, in most real organizations, there is most likely a mixture of these extremes with some common resources, some local resources, and some group-to-group traffic.

Installed Base

The next major business requirement to determine is the installed base. What technology exists today? Why does it need to be changed? How much of the existing infrastructure must remain?

It would be extremely unusual to find a completely new organization that is very large, has no existing technology today, and needs it tomorrow. Even if you did find one, chances are that the problem of implementing this new technology has been broken down among various groups. So the new network design will need to fit in with whatever the other groups need for their servers and applications.

Installed base can cause several different types of constraints. There are geographical constraints, such as the location and accessibility of the computer rooms and LAN rooms. There may be existing legacy network technology that has to be supported. Or it may be too difficult, inconvenient, or expensive to replace the existing cable plant or other existing services.

Constraints from an existing installed base of equipment can be among the most difficult and frustrating parts of a network design, so it is critical to establish them as thoroughly and as early as possible.

Bandwidth

Now that you understand what you're connecting and to where, you need to figure out how much traffic to expect. This will give the bandwidth requirements. Unfortunately, this often winds up being pure guesswork. But if you can establish that there

are 50 users in the accounting department who each use an average of 10kbps in their connections to the mainframe throughout the day, plus one big file transfer at 5:00 P.M., then you have some very useful information. If you know further that this file transfer is 5 gigabytes and it has to be completed by 5:30, then you have another excellent constraint.

The idea is to get as much information as possible about all of the major traffic patterns and how much volume they involve. What are the expected average rates at the peak periods of the day (which is usually the start and end of the day for most 9–5 type operations)? Are there standard file transfers? If so, how big are they, and how quickly must they complete? Try to get this sort of information for each geographical area because it will tell you not only how to size the trunks, but also how to interconnect the areas most effectively.

In the end it is a good idea to allow for a large amount of growth. Only once have I seen a network where the customer insisted that it would get smaller over time. And even that one got larger before it got smaller. Always assume growth. If possible, try to obtain business-related growth projections. There may be plans to expand a particular department and eliminate another. Knowing this ahead of time will allow the designer to make important money-saving decisions.

Security

Last among the top-level business requirements is security. What are the security requirements? This is even important in networks that are not connected to anything else, like the Internet or other shared networks. For example, in many organizations the servers in the Payroll Department are considered sensitive, and access is restricted. In investment banks, there may be regulations that require the trading groups to be separate from corporate financing groups. The regulatory organizations tend to get annoyed when people make money on stock markets using secret insider information.

The relationship between security and geography requirements may make it necessary to implement special encryption or firewall measures, so these have to be understood before a single piece of equipment is ordered.

Philosophical and Policy Requirements

Besides the business requirements, there could be philosophical requirements. There may be a corporate philosophy that dictates that all servers must be in a central computer room. Not all organizations require this, but many do. It makes server maintenance and backups much easier if this is the case. But it also dictates that the network must be able to carry all of the traffic to and from remote user areas.

There may be a corporate philosophy that, to facilitate moves, adds, and changes, any PC can be picked up and moved anywhere else and not require reconfiguration.

Some organizations insist that all user files be stored on a file server so that they can be backed up. Make sure that you have a complete list of all such philosophical requirements, as well as the business requirements, before starting.

OSI Protocol Stack Model

No book on networking would be complete without discussing the Open System Interconnection (OSI) model. This book is more interested in the lower layers of the protocol stack. One of the central goals of network design is to build reliable networks for applications to use. So a good design starts at the bottom of the stack, letting the upper layers ride peacefully on a stable architecture. Software people take a completely different view of the network. They tend to be most concerned about the upper layers, from Layer 7 down to about Layer 4 or 5. Network designers are most concerned with Layers 1 through 4 or 5. Software people don't care much about cabling, as long as it doesn't lose their data. Network designers don't care much about the data segment of a packet, as long as the packet meets the standard specifications.

This fact alone explains much of my bias in focusing on the lower parts of the stack. There are excellent books on network programming that talk in detail about the upper layers of the stack. That is largely beyond the scope of this book, however.

The Seven Layers

The OSI model is a useful way of thinking about networking. It's important not to confuse it with reality, of course. The most commonly used networking protocols, such as TCP/IP, don't completely match the model. But it is still a useful model. Table 1-1 shows this simple model in its usual form.

Table 1-1. The OSI model

Layer	Name	Uses	Examples
7	Application	User and application data	The reason for having a network in the first place
6	Presentation	Data formatting, encryption, character encoding	ASCII versus EBCDIC, software encryption of a data stream
5	Session	Negotiates and maintains connections	Name and address correlation, software flow control
4	Transport	End-to-end packet sequencing and reliability	UDP, TCP, SPX
3	Network	Routing, flow control, translation between different media types	IP, IPX
2	Data Link (MAC)	Basic framing of packets, error detection, transmission control	Ethernet packets, including collision mechanisms
1	Physical	Electrical and optical media, signaling and properties	Cabling, the electrical or optical pulses sent through the cabling

Layer 1

The Physical Layer is at the bottom. This includes the parts of the network that you can see, such as cables, patch panels, jacks, and optical fibers. Specifications for the Physical Layer have to do with the differences between categories of cables, the wavelength properties of optical fibers, the length restrictions, and electrical specifications. This is extremely important stuff, but most network designers only think about it briefly when they do the cable plant.

Other physical-layer issues, such as laser intensity, wavelength characteristics, attenuation, and so on, are important to engineers who design the equipment and cables. But for the network design they appear only in decisions to match the specifications of different pieces of hardware and cabling.

Layer 2

The Data Link Layer is where things start to get a bit more abstract, so some examples might help. This layer is where the difference between Ethernet, Fast Ethernet, and Token Ring exists. It includes all of the specifications about how to build a packet. It describes how the different nodes on this network avoid contention using collisions or token passing or perhaps some other algorithm. For broadcast media (as opposed to point-to-point media where you know that if you send out a packet, it can only be received by one other device), it defines how to actually specify for which device or devices the packet is destined.

Before going on, let me point out the ways that these first two layers are both connected and separable. For example, you have a certain physical layer, such as Category 5 twisted pair cabling. Then, when you decide to run Ethernet over this physical medium, you are constrained to use a particular type of signaling that works with this medium. It is called 10BaseT. There are other types of Ethernet signaling, such as 10Base2. In this case, though, you would have to use coaxial cable designed to have 50 Ω (ohm) characteristic impedance. But, over this twisted pair cabling, you could just as easily run Token Ring. Or, if you are working with Token Ring, you could choose instead to use Type 3 shielded cabling.

The point is that Ethernet means a particular way of forming packets and a particular way of avoiding contention (collisions). It can run over many different types of physical media. Going up the protocol stack, the same is true at each layer. You can run TCP/IP over Ethernet, or over Token Ring, ATM, or FDDI, or over point-to-point circuits of various descriptions. At each layer there is a set of specifications on how to get to the layer below. You can think of this specification as being the line between the layers of the stack. So the line between the Physical Layer and the Data Link Layer includes 10BaseT, 100BaseFx, and so forth.

Strictly speaking, these distinctions are described in sublayers of the standard OSI model. The IEEE provides detailed specifications of these protocols.

Layer 3

The Network Layer includes the IP part of TCP/IP. This is where the IP address lives. The Network Layer specifies how to get from one data-link region to another. This is called routing. See the next section on "Routing Versus Bridging" for a more detailed description of what routing means.

There are several other Network Layer protocols besides IP. One of the most popular for LANs is called IPX, which forms the basis of the Novell Netware NOS (Network Operating System). However, IPX can also be used by other systems including Microsoft Windows and Linux.

As an aside on the subject of the OSI model, it is quite common to use both IP and IPX simultaneously on the same network, over the same physical-layer equipment. But what's particularly interesting is that they don't have to use the same Data Link Layer protocol for their framing. Usually IP packets are framed using the Ethernet II data link layer. Meanwhile, IPX usually uses IEEE 802.2 with 802.3 Ethernet framing. There are several subtle differences between Ethernet II and 802.2, and it would certainly not be possible to run an IP network using both simultaneously on the same segment. But it is quite common to configure all of the devices on the network to expect their IP frames in one format and IPX in a different format.

Layer 4

At Layer 4, things become still more abstract. The IP protocol has two main transport-layer extensions, called TCP and UDP. TCP, or Transmission Control Protocol, is a connection-oriented protocol. This means that it forms end-to-end sessions between two devices. It then takes care of maintaining this session, keeping packets in order and resending them if they get lost in the network. For this reason, TCP is not useful for one-to-many or many-to-many communication. But it is perfect for building applications that require a user to log in and maintain a connection of any kind. A TCP session has to begin with a session negotiation that sets up a number of communications parameters such as packet size. At the end, it has to be torn down again.

UDP, or User Datagram Protocol, is connectionless. It is used for applications that just send one packet at a time without requiring a response. It is also used by applications that want to maintain their own connection, rather than using TCP. This can be useful if a server needs to support a large number of clients because maintaining connections with TCP can be resource-intensive on the server. In effect, each UDP packet is a complete session. UDP is also useful for multicast type applications or for applications where the data is time sensitive, so retransmitting a packet is worse than dropping it.

TCP, being a connection-oriented protocol, is inherently reliable. It ensures that all data sent from one end to the other gets to its destination intact and in the right order. UDP, on the other hand, is inherently unreliable. This doesn't mean it's bad; it just means that the application has to make sure that it has received all of the data it needs.

The other important thing that happens at Layer 4 is the differentiation between different application streams. In both TCP and UDP (as well as in IPX/SPX at the same layer) there is a concept called a port. This is really nothing more than a number. But it is a number that represents an application. For an application to work, there has to be not only something to send information, but also something on the other end to listen. So a server will typically have a program running that listens for incoming packets on a particular port (that is, packets that have the appropriate number in the port-number part of the packet).

The network also cares about port numbers because it is an easy way to differentiate between different applications. The port number can be used to set priorities so that important applications can pass through the network more easily. Or the network can reject packets based on port number (usually for security reasons, but sometimes just to clean up artificially for ill-behaved application chatter).

Layer 5

Layer 5 is not used in every protocol. It is where instructions for pacing and load balancing of different clients will occur, as well as where sessions are established. As I mentioned previously, the TCP protocol handles session establishment at Layer 4, and the UDP protocol doesn't really have sessions at all.

To make matters more confusing, the TCP/IP telnet and FTP protocols, for example, tend to handle the session maintenance as Layer 7 application data, without a separate Session Management layer. These protocols use Layer 4 to make the connection and then handle elements such as username and password verification as application information.

Some protocols such as SNA can use a real Session Layer that operates independently from the Transport Layer. This ability to separate the layers, to run the same Session Layer protocol over a number of possible Transport Layers, or to build applications that have different options for session control, is what makes it a distinct layer.

Layer 6

The Presentation Layer, Layer 6, is also not universally used. In some cases, a data stream between two devices may be encrypted, and this is commonly handled at Layer 6. But encryption can also be done in some systems at Layer 2, which is generally more secure and where it can be combined with data compression.

One common usae of Layer 6 is in an FTP file transfer. It is possible to have the protocol interpret the data as either 7-bit or 8-bit characters. Similarly, some terminal-emulation systems use ASCII characters, while others use EBCDIC encoding for the data in the application payload of the packet. Again, this is a Layer 6 concept, but it might not be implemented as a distinct part of the application protocol. In many

cases, conversions like these are actually made by the application and then inserted directly into Layer 4 packets. That is to say, a lot of what people tend to think of as Layer 6 concepts are not really distinct protocols. Rather, they are implementation options that are applied at Layers 4 and 7.

Layer 7

And, finally, Layer 7 is called the Application Layer. This is where the contents of your email message or database query live. The Application Layer is really the point of having a network in the first place. The network needs to get information efficiently from one place to another. The Application Layer contains that information. Maybe it needs to be chopped up into several packets, maybe it needs to be translated into some sort of special encoding scheme, encrypted and forwarded through 17 different types of boxes before it reaches the destination. But ultimately the information gets there. This information belongs to Layer 7.

Where the OSI Model Breaks Down

In a sense, the model doesn't break down. It's more accurate to say that it isn't always strictly followed. And there are a lot of places where it is almost completely abandoned. Many of these examples involve concepts of tunneling.

A tunnel is a protocol within a protocol. One of the most frequent examples is a Virtual Private Network, or VPN. VPNs are often used to make secure connections through untrusted networks such as the Internet. Taking this example, suppose the users of a corporate LAN need to access some important internal application from out on the Internet. The information in the database is too sensitive to make it accessible from the Internet where anybody could get it. So the users have to make an encrypted VPN connection from their computers at home.

They first open a TCP connection from their home computers to the VPN server through the corporate firewall. This prompts them for usernames and passwords, and they log in. At this point everything seems to follow the OSI model. But then, through this TCP session, the network passes a special VPN protocol that allows users to access the internal LAN as if they were connected locally (although slower). They obtain a new IP address for this internal connection and work normally. In fact, they also can pass IPX traffic through their VPN to connect to the corporate file server. So the VPN is acting as if it were a Layer 2 protocol because it is carrying Layer 3 protocols. But in fact it's a Layer 6 protocol.

Now, suppose the users' own Internet connection is made via a DSL connection. One of the most popular ways to implement DSL in North America is to emulate an Ethernet segment, a Layer 2 protocol. But the connection over this Ethernet segment is made using PPPoE (PPP over Ethernet), a Layer 3 protocol that carries PPP, a Layer 2 protocol.

To summarize, there is a Layer 1 physical connection to the DSL provider. Over that the users run Ethernet emulations (Layer 2). On top of the Ethernet is PPPoE, another Layer 2 protocol.* Over that they run IP to communicate with the Internet at Layer 3. Then, using this IP stack, they connect to the VPN server with a special Layer 4 connection authenticated at Layer 5 and encrypted at Layer 6. Over this is new Ethernet emulation (back to Layer 2). The users can then run their normal applications (Layers 3–7) on top of this new Layer 2. And, if you wanted to be really weird, you could start over with another PPPoE session.

Things get very confusing if you try to map them too closely to the OSI model. But, as you can see from the previous example, it is still useful to think about the various protocols by function and the layers that represent those functions.

Routing Versus Bridging

Chapter 3 will discuss the design implications of the differences between routing and bridging. The discussion of the OSI model here makes it a good place to define them and talk about their technical differences.

I will use the terms "bridging" and "switching" interchangeably throughout this book. This is because early manufacturers of multiport fast bridges wanted to make it clear that their products were distinct from earlier products. The earlier products, called "bridges," were used primarily for isolation and repeating functions; the newer products tended to focus on reducing latency and increasing throughput across a network. Technically, they perform the same basic network functions. But these vendors wanted to make sure that consumers understood that their products were different from the earlier devices: so they gave them a different name.

To make matters more confusing, it has become fashionable to talk about "Layer 3 switches." These are essentially just routers. But, in general, they are special-function routers that route between like media, which allows certain speed optimizations. So, where you might use a Layer 3 switch to route between two VLANs, both built on Fast Ethernet, you would never use one to control access to a WAN. You probably would want to think very carefully before using a Layer 3 switch to regulate traffic between a Token Ring and an Ethernet.

Routing means sending packets from one Layer 3 network region to another using Layer 3 addressing information. These two Layer 3 regions could use different Layer 1 or 2 protocols. For example, one may be Ethernet and the other ATM. So part of the routing process requires taking the Layer 3 packet out of the Ethernet frame in

* PPPoE is a particularly interesting protocol when studied on the OSI stack because it looks like Layer 3 protocol to the Ethernet protocol on top of which it sits. But it presents a standard Layer 2 PPP interface to the IP protocol that lives above it on the stack.

which it was received, deciding where to send it, then creating ATM cells to carry this packet. Because ATM uses a cell size that is much smaller than the Ethernet packet size, the router has to chop up the Layer 3 packet and wrap each fragment in an ATM cell before sending it. When receiving from the ATM side, it has to wait until it receives all of the ATM cells that form one Layer 3 packet, reassemble the fragments in the correct order, and wrap it up in an Ethernet frame before sending it on. This allows easy transfer of data between LAN and WAN or between different LAN types.

Technically, bridging has some overlap into the Network Layer as well, because it specifies how the broadcast domains that are part of the Data Link Layer can interact with one another. But the special difference between routing and bridging is that in routing the Data Link properties don't need to have anything in common. It is easy to route IP from Ethernet to Token Ring without needing to consider anything but the IP addresses of both ends. But in bridging, the MAC (Media Access Control) addresses from one side of the bridge are maintained as the frame crosses over to the other side.

It is possible to bridge from Ethernet to Token Ring, for example. But the Token Ring devices must believe that they are talking to another Token Ring device. So the bridge has to generate a fake Token Ring MAC address for each Ethernet device, and a fake Ethernet MAC address for each Token Ring device taking part in the bridge.

With routing, though, there is only one MAC address visible, that of the router itself. Each device knows that it has to communicate with all off-segment devices through that address.

So routing scales much better than bridging when large numbers of devices need to communicate with one another. But the drawback is that the extra work of translating from one data-link layer to another means that the router has to read in every packet, decide where to send it, reformat it for the new medium, and then send it along.

With switching, however, it is possible to read in just enough of the packet to figure out where it needs to go and then start sending it out before it has all been received. This is called cut-through switching. Store-and-forward switching, in which the entire packet is read before forwarding, is also common. But the bottom line is that switching is generally faster than routing.

Layer 3 switching is sort of a hybrid. If you know that you are switching between like media, then the only things you need to change when you pass the packet along are the source and destination MAC addresses (and the checksum will also need to be corrected). This is considerably less work than the general media-independent problem of routing. So these Layer 3 switches are usually faster than a general-purpose router.

The other advantage of a Layer 3 switch over a router is that it can often be implemented as a special card in a Layer 2 switch. This means that it is able to do its work while touching only the backplane of the switch. Because the switch backplane doesn't need to go very far, and because it usually runs a proprietary high-speed protocol, it is able to run at extremely high speeds. So it is as if you were able to connect your router, not to a 100-Mbps Fast Ethernet or even to 1000Mbps Gigabit Ethernet, but to a medium many times faster than the fastest readily available LAN technologies. And this is done without having to pay a lot of extra money for the high speed access.

Chapter 3 will discuss how to use these sorts of devices effectively.

Top-Down Design Philosophy

Once the actual requirements are understood, the design work can begin, and it should always start at the top. Earlier in this chapter I described the standard seven-layer OSI protocol model. The top layer in this model is the Application Layer. That is where one has to start when designing a network. The network exists to support applications. The applications exist to fulfill business requirements.

The trick is that the network will almost certainly outlive some of these applications. The organization will implement new applications, and they will likely have new network requirements. They will form new business units, and new departments will replace old ones. A good network design is sufficiently flexible to support these sorts of changes without requiring wholesale redesign. This is why an experienced network designer will generally add certain philosophical requirements to the business requirements that have already been determined.

The network needs to be scalable, manageable, and reliable. Methods for achieving each of these topics will be examined in considerable detail throughout this book. It should be obvious why they are all important, but let me briefly touch on some of the benefits of imposing these as requirements in a network design.

Making a design scalable automatically dismisses design possibilities where switches for different workgroups are either interconnected with a mesh or cascaded one after another in a long string. Scalability will generally lead to hierarchical designs with a Core where all intergroup traffic aggregates.

Manageability implies that you want to see what is going on throughout the network easily. It will also demand simple, rational addressing schemes. Some types of technology are either unmanageable or difficult to manage. You probably wouldn't want to eliminate these outright because they may be cost effective. But you probably don't want to put them in key parts of the network.

Reliability is usually the result of combining a simple, scalable, manageable architecture with the business throughput and traffic-flow requirements. But it also implies that the network designer will study the design carefully to eliminate key single points of failure.

There are other important philosophical principles that may guide a network design. A common one is that, except for specific security exclusions, any user should be able to access any other part of the network. This will help ensure that, when new services are deployed, the network will not need to be redesigned.

Another common design philosophy says that only network devices perform network functions. In other words, never use a server as a bridge or a router. It's often possible to set up a server with multiple interface cards, but this philosophy will steer you away from doing such things. Generally speaking, a server has enough work to do already without having the resources act as some kind of gateway. It will be almost invariably slower and less reliable at these functions than a special-purpose network device.

If your network uses TCP/IP, will you use registered or unregistered IP addresses? This used to be a hotly debated subject, but these days it is becoming clear that there is very little to lose by implementing a network with unregistered addresses, as long as you have some registered addresses available for address-translation purposes.

Perhaps the most important philosophical decisions have to do with what networking standards will be employed. Will they be open standards that will allow easy interoperability among different vendors' equipment? Or will they be proprietary to one vendor, hopefully delivering better performance at a lower price? It is wise to be very careful before implementing any proprietary protocols on your network because it can make it exceedingly difficult to integrate other equipment later. It is always possible that somebody will come along with a new technology that is vastly better than anything currently on the market. If you want to implement this new technology, you may find that the existing proprietary protocols will force a complete redesign of the network.