Network Design
Unit 1
Gathering Requirements

This unit begins by outlining a formal five-phase approach to network design. This phased process covers the majority of areas that must be considered in most networking analysis and design projects. It can be used by an outside consultant or an internal Information Services (IS) group.

This network design process, as presented here, is very methodical. This level of detail helps ensure that the designer gathers all necessary information, considers all options, and keeps all key players well informed. Experienced network designers have found that this approach is essential to keep large projects on track.

Although there are many benefits to following a formal process (and we discuss them in Lesson 1), not every project requires such a detailed approach. Once you understand the reasons and methods of this process, you can modify it to fit the size and scope of most projects.

Lessons

1. The Network Design Process
2. Business Requirements
3. User Requirements
4. Application Requirements
5. Computing Platform Requirements
6. Network Requirements
7. Developing a Requirements Specification Document
Terms

24x7—This term refers to 24 hours per day, 7 days per week, and is used to describe the system uptime and support hours.

Asynchronous Transfer Mode (ATM)—ATM is a connection-oriented cell relay technology based on (53 byte) cells. An ATM network consists of ATM switches which form multiple virtual circuits to carry groups of cells from source to destination. ATM can provide high-speed transport services for audio, data, and video.

Bandwidth—Bandwidth is the total information-carrying capacity of a network or transmission channel. It is the difference between the highest and lowest frequencies that can be transmitted across a transmission line or through a network. It is measured in Hz for analog networks and bps for digital networks.

Compiler—An application that converts code written in a programming language into machine-readable instructions (an executable program) is referred to as a compiler.

Debugger—A debugger is an application that helps find programming errors.

Deterministic—Deterministic is a LAN characteristic that guarantees each node access to the shared physical medium at some level. Token-passing networks (Token Ring and FDDI) are deterministic, while networks that require nodes to compete for media access (Ethernet) are not.

Dynamic Data Exchange (DDE)—DDE is a Microsoft Windows 3 protocol that allows communication between applications using a client-server model. When a document is modified in one application, documents shared by DDE are updated as needed even if currently open in another application.

Dynamic Random Access Memory (DRAM)—DRAM is RAM that must periodically be refreshed. It is cheaper and slower and takes up more space than SRAM.

E3—See T1, T3.

Extended Industry Standard Architecture (EISA)—EISA is a 32-bit PC bus standard that is compatible with older XT and ISA bus architecture boards.

Industry Standard Architecture (ISA)—ISA is a PC expansion bus for modems, video displays, speakers, and other peripherals. It is also called an AT bus.
Fiber Distributed Data Interface (FDDI)—FDDI is a token-passing network architecture that uses two ring channels. FDDI provides 100 Mbps over optical fiber.

Global Positioning System (GPS)—GPS is a system that determines position on Earth’s surface by triangulating signals from several satellites though a receiver on Earth.

Management Information Services (MIS)—MIS is the traditional name of a department responsible for a company network or computing infrastructure.

Massively Parallel Processing (MPP)—MPP is a high-speed architecture that uses many separate CPUs (up to 200) to run the same program. Unlike SMP, each CPU of an MPP system has its own dedicated memory. This approach makes MPP applications harder to program, but prevents most memory bottlenecks. See symmetric multiprocessing.

Micro Channel Architecture (MCA)—MCA is IBM’s proprietary 32-bit bus. Designed for multiprocessing, it eliminates conflicts between peripherals. It is not compatible with many other bus standards such as ISA, EISA, and XT.

Mirroring—Disk mirroring is a method of simultaneously writing information to a hard drive while backing the same information up to a second disk. One disk controller is used to write to two hard drives.

Object Linking and Embedding (OLE)—OLE is a Microsoft protocol that connects objects from other applications with a document. An embedded object becomes part of the document. A linked file allows the object to be edited in its original application; it is then updated in the document to which it is linked.

Open Shortest Path First (OSPF)—OSPF is a popular link-state routing protocol for use within large autonomous router networks.

Optical Carrier (OC)—OC is one of the optical signal standards defined by the SONET digital signal hierarchy. The basic building block of SONET is the STS-1 51.84-Mbps signal, chosen to accommodate a DS-3 signal.

Peripheral Component Interconnect (PCI)—PCI is a local bus that provides a high-speed connection between peripherals and a CPU. It includes buffers that allow relatively slow peripherals to operate asynchronously and can be used with other buses such as ISA or EISA.
Plug-and-Play (PnP)—PnP is a standard that gives computers the ability to automatically recognize a newly installed device, without a complex process of user configuration.

Portable Operating System Interface for UNIX (POSIX)—POSIX is a set of standard operating system interfaces based on UNIX. This standard approach serves as a common denominator among several “flavors” of UNIX.

Preemptive Multitasking—Preemptive multitasking is a type of multitasking in which a scheduler can suspend or interrupt a running task to run another task. The scheduler allots time for tasks based on their priorities or resources they require, makes sure tasks are interrupted only at a safe state, and makes sure they do not interfere with each other.

Redundant Array of Inexpensive Disks (RAID)—RAID is a hard disk control technology that allows multiple hard disks to act as one device. RAID systems often mirror data on multiple disks of the array, providing built-in redundancy for important data.

Scalability—The ability of a network design to accommodate both current and future needs, by ensuring that the network and its applications can be easily expanded.

Scalable Processor Architecture (SPARC)—SPARC is a widely used 32- and 64-bit microprocessor architecture. Sun Microsystems developed SPARC for hardware that uses UNIX-based operating systems; however, Sun made SPARC open for licensing to other manufacturers.

Stakeholders—Stakeholders are persons or groups who have a direct interest in a decision-making process.

Static Random Access Memory (SRAM)—SRAM is RAM that retains its contents as long as power is supplied to it. It is faster and more expensive than DRAM.

Symmetric Multiprocessing (SMP)—SMP is a high-speed computer architecture that uses one copy of the operating system to control multiple CPUs (usually no more than 16). All CPUs also share the same memory and I/O data path. Because any free processor can be assigned any task, the SMP approach is good for dynamically balancing loads across multiple processors. SMP is supported on UNIX and Windows NT platforms. See massively parallel processing.
T1, T3—T1 and T3 are two services of a hierarchical system for multiplexing digitized voice signals. The first T-carrier was installed in 1962 by the Bell system. The T-carrier family of systems now includes T1, T1C, T1D, T2, T3, and T4 (and their European counterparts E1, E2, etc.). T1 and its successors were designed to multiplex voice communications. Therefore, T1 was designed such that each channel carries a digitized representation of an analog signal that has a bandwidth of 4,000 Hz. It turns out that 64 Kbps is required to digitize a 4,000-Hz voice signal. Current digitization technology has reduced that requirement to 32 Kbps or less; however, a T-carrier channel is still 64 Kbps. A T1 line offers bandwidth of 1.544 Mbps; a T3 offers 44.736 Mbps.

Video Electronics Standards Association (VESA)—VESA is an industry organization that sets standards for PC video and multimedia. It has established, among other standards, the SVGA and VLB standards.

Uninterruptible Power Supply (UPS)—UPS is an emergency backup power source that instantly takes over when the regular electrical power fails.

Version Control—An application that tracks changes made to a document (or programming source code), and can retrieve a particular version of a document when necessary, is referred to as version control.

Virus—A virus is a self-replicating malicious program that spreads by attaching itself to a file. Viruses can spread quickly through a network, with effects that range from mildly irritating to highly destructive.

X.25—X.25 is a standard protocol for packet-switch data networks. It specifies the interface between the data terminal equipment and network. It describes the standard Physical Layer, Data Link Layer, and Network Layer, and how the data is assembled into packets.
Lesson 1—The Network Design Process

This lesson introduces the five-phase network design process that we will follow throughout this course, and shows how it is similar to the phased development approach commonly used in software development and engineering.

We discuss why a formal process can prevent many of the most common problems in any technical design work, then describe how each phase of the network design process forms a logical sequence of events referred to as the systems development life cycle.

Objectives

At the end of this lesson you should be able to:

- Understand why more formality is needed in the network design process
- Describe the two main systems development life cycles, including their similarities and differences
- List and describe the phases of a common network design process
- Explain the importance of a requirements analysis

Key Point

Following a formal design process increases your chances of success.

The Case for Formality

As with any technical discipline, a process needs to be followed when designing a network that fills a particular business need. Rather than a bureaucratic burden that interferes with the “real work” of network building, a good formal design process makes the designer’s work simpler, more productive, and more satisfying.

Time pressure is a fact of life, and many technical professionals are continually tempted to skip a formal design and “get right to work.” However, even the simplest development process can help a network avoid the following problems:
• Failure to meet requirements—If you do not find out what the requirements actually are, it is impossible to create a network that meets them.

• “Creeping” requirements—Specification additions and changes can disastrously increase the amount of time, effort, and money spent on a project. All change requests must be clearly documented, communicated, and evaluated.

• Missed deadlines and budget overruns—Haphazard projects almost always take longer and cost more than well-planned ones, often because work must be redone. Also, when you “shoot from the hip,” it is easy to miss cost-saving opportunities.

• Dissatisfied end users—Regardless of how good a network appears, it is a failure if it does not satisfy those who must use it.

• Dissatisfied management—A haphazard and unprofessional development project can hurt your credibility and create ill will among decision makers.

A formal process does not have to be burdensome, or any more complex than necessary. A design process is like a construction blueprint. Large office buildings require many complex drawings and schedules, but even a tool shed should start with a simple sketch.

Therefore, a small network project may only require a process as simple as documenting the initial requirements, implementing the solution, and documenting the resulting changes in the network. Larger and more complex jobs often require a formal, highly documented process.

The Systems Development Life Cycle

The process of creating a new system, or changing an existing system, is called a life cycle. During this cycle, a new network or feature is planned, implemented, and maintained. The process begins anew with each change. This cycle is very similar to the one long used by software engineers and system analysts.

Although no single life cycle perfectly describes all development projects, two general life cycle patterns have been identified by software engineers: the waterfall cycle and the spiral cycle. One of these life cycles describes every network development project to some extent.
Waterfall Cycle

The waterfall life cycle is defined by distinct stages. Different waterfall-based processes have different names for the stages, but they all tend to follow these five general steps, in order:

1. Analyze
2. Design
3. Build
4. Test
5. Deploy

This life cycle is called a waterfall, because work “flows down” from one stage into the next, as shown on the Waterfall Cycle Diagram. After the system is deployed, the life cycle begins again for the next update.

Waterfall Cycle

When a development process follows the waterfall model, each stage must be completed before the next stage can begin. Returning to a previous stage is often not permissible. In this case, changes that are not possible during the current development cycle are scheduled to be part of the next. When returning to an earlier stage is permissible, there are usually repercussions. The completion date is often extended as a result, and significant budget overruns are common.

The major advantage of the waterfall cycle is that all planning is done in the early stages. All system stakeholders know exactly what is expected and what stage the process is currently in. Com-
Completion dates can be determined at an early stage, and coordination is simplified.

Although the rigidity of the waterfall is appealing to many developers (who can use it as a shield against users who suggest late project changes), it can be cumbersome for any but the smallest projects. In addition, because the requirements of a project often change before the project has been completed, the rigidity of the waterfall cycle can lead to development setbacks.

**Spiral Cycle**

The spiral cycle, or whirlpool cycle, is a variation of the waterfall cycle. It is a more recent approach, meant to overcome some of the limitations of the waterfall cycle. This cycle is often used in multiple-version software development projects; however, some of its principles can be applied to network development as well.

The guiding principle behind the spiral cycle is change management. Unlike the waterfall cycle, the spiral cycle can adapt quickly to new requirements. This is accomplished by looping through all stages several times, producing a limited version of the project each time, as shown on the Spiral Cycle Diagram.
By building a subset of the eventual features in each iteration of the network design, its users get an opportunity to provide feedback on the project before it is finished. Their feedback is then incorporated into the next iteration of the spiral. With each iteration, new features are incorporated and prior problems are fixed.

Although the spiral life cycle handles changing requirements much better than the waterfall cycle, it also has significant limitations. Because there is no way to guess what new features may be requested, it is difficult to estimate the total eventual cost and release date. In addition, major features that require longer development times are difficult to implement. Most importantly, when following a spiral development life cycle, it is very easy to fall into a never-ending series of upgrades.

The Network Design Process

The waterfall and spiral cycles do not perfectly describe all network design projects, and a single project may change from one cycle to another. For example, a waterfall model may describe the process of designing and launching a new network, while a spiral model might better describe its ongoing updates and maintenance.

The network design process describes the general tasks that must be accomplished when designing a network. However, each project has its own unique needs that may require a different process with different tasks.

Process Phases

The phases of a process break a large project down into understandable, manageable pieces. If you think of a project as a long list of tasks, these phases are simply task categories. In other words, each phase includes certain jobs that must be performed to prepare the project to move to the next phase. The life cycle of a typical network design project consists of the following phases, as illustrated on the Network Design Process Diagram:

1. Requirements Gathering
2. Analysis of the Existing Network
3. Logical Design
4. Physical Design
5. Installation and Maintenance
Lesson 1—The Network Design Process

Network Design Process

This process can apply to either a waterfall or spiral systems development life cycle. In other words, the process only defines the phases of a life cycle. The decision whether to complete each phase before starting another (waterfall), or work through several iterations of the process in one life cycle (spiral), is up to an organization.

Deliverables

You can also think of a project in terms of what it is trying to produce—its “deliverable.” For example, if someone asks, “What is the project’s deliverable?” you could answer, “a network.” However, to get to the final goal of a functioning network, the development team must produce many supporting products, such as design documents, estimates, or reports. Each phase produces its own deliverables that become the input to the next phase.

Like the invisible foundation of a building, these deliverables form a strong structure that strengthens the overall design. Therefore, all documentation that records your design assumptions, technical alternatives, customer information, and management approval should be retained for easy access and future reference.

As you work through this course, it is important to remember that not all projects require all of these phases or deliverables. Smaller projects may skip some phases, or combine them. Once you understand the reason for each phase, task, and deliverable, you can decide how much of this formal process is necessary for each of your development projects.
Phase 1: Requirements Gathering

This is the most crucial phase in the development process, because requirements provide the target your network design must hit. Although Requirements Gathering is fundamental to the network design, it can often be a challenge to collect and organize information gathered from many sources.

Gathering requirements means talking to users, managers, and other network personnel, then summarizing and interpreting the results. Often, it means resolving conflicting needs and desires among user groups. However, network personnel are sometimes distanced from the users, and might not have a clear idea of what they want or need.

Requirements Gathering is time-consuming, and it may appear to produce no immediate results. On the contrary, requirements analysis helps the designer to better understand how the new network should perform. Therefore, requirements gathering produces immediate payoffs in:

- Better view of current network
- Objective decision-making
- Ability to plan for network migration
- Ability to deliver appropriate resources to all users

Requirements for All Types of Needs

Just as different types of users have different networking needs, each aspect of the organization has its own requirements. In this course, we discuss the need to gather requirements for:

- The business or organization as a whole
- Users
- Applications
- Computing platforms
- The network itself, and the network staff

The Areas of Requirements Diagram shows these in a layered format with the associated services and requirements at each layer.
The Requirements Gathering process is a series of steps. We begin by gathering requirements from upper management or the owners of the business. Next, we work with the user community, gathering the network requirements for supporting the users, their applications, and the base of installed computing equipment. The network itself is the last consideration; we gather all other requirements before considering the network or network technologies.

Qualities of Good Requirements

The concept of “garbage in, garbage out” applies to requirements gathering. Good results depend on gathering good requirements that are both user- and business-centered, as well as detailed and specific.

It is common for network professionals to base a network design solely on a particular technology, service, or vendor (typically ones the designer has the most experience with). However, this makes as much sense as designing a house without knowing anything about the people who will live there.

A network is not an end in itself; it is a highly customized tool that helps people do their work. Thus, designers must deliberately postpone any technical decision-making, and focus instead on discovering what factors make a real difference to users. Do they have enough storage space? Do their applications perform well? Are people waiting too long for print jobs? Is the security system...
understandable and usable? Do any network problems really get in their way?

Network designers cannot be expected to understand the jobs of system users. However, users often assume that certain “essential” features will be part of a network, even though they never explicitly ask for them.

The Requirements Gathering phase is a chance to define, as precisely as possible, what users want and need. Detailed requirements make it more likely the final network will satisfy its users. Specific requirements help guard against “scope creep,” the process of gradually adding requirements until a project becomes unrecognizable. Good Requirements Gathering techniques will not only help individuals do their work, but will also improve the overall productivity of the organization, providing a competitive edge in the marketplace.

Looking to the Future

The Requirements Gathering process must consider both the current and future needs of the organization. Without proper planning for future growth, it will be difficult to expand the network later.

Deliverable: Requirements Specification Document

The network designer must formally record the requirements in a Requirements Specification document that describes exactly what the organization and users need from the network. This document should not propose solutions or designs (that will come later); instead the Requirements Specification should clearly and specifically summarize the needs and desires of the organization and users.

After the Requirements Specification document has been written, management and network designers should formally agree that it is correct. In other words, the responsible stakeholders must all sign off on the requirements. At this point, the requirements document becomes an agreement between the development team and management. Management agrees that the requirements document describes the system they want; the network developers agree to deliver that system.

After the Requirements Specification document has been formally accepted, the development process can move forward to the next phase. However, although formal requirements documents are vitally important, they are not written in stone. Things change, new factors arise, and the key players should always be willing to renegotiate the network requirements. However, a formal require-
ments process helps makes it clear to everyone that there is always a price (in time, money, or features) for any requirements changes.

Phase 2: Analysis of the Existing Network

When a network design project upgrades or enhances an existing network, it is essential to analyze the existing network architecture and performance. The Analysis phase complements the Requirements Gathering phase; requirements show you where you need to be, and analysis tells you where you currently are.

The effectiveness of a new network design depends on whether the current computing infrastructure can support the new requirements. The existing network installation and its supporting systems may be an asset to the new development, or a liability. Therefore, after the Requirements Specification has been written, but before the design process begins, the design team must thoroughly analyze the existing network and any other resources the new network may depend on.

A thorough analysis should gather both qualitative information (such as user estimates of storage and traffic) and quantitative data (such as traffic measurements and network management statistics). A Traffic Specification Document is created during this phase of the design, and is considered a formal deliverable before proceeding to the Logical Design Phase.

Deliverable: Traffic Specification Document

The network Analysis phase should produce deliverables such as:

- Logical diagram of the current topology
- Estimated traffic volumes and patterns that describe the network capacity required for each application, each network segment, and the network as a whole
- Detailed statistics, baseline measurements, and any other direct measurements that describe the network's current level of performance
- A report on the quality of service provided by suppliers of Internet connections or wide area network (WAN) links
- A list of design constraints, such as the need to use existing cabling or devices

Phase 3: Logical Design

The Logical Design describes what the network must do, and how it must perform, to meet the requirements. A Logical Design specifies how data flows through a network, not where particular network elements are physically located (that comes in the next phase).
The designer creates a logical network structure based on the Requirements Specification and results of the network analysis. If the current hardware or software cannot meet the needs of the new network, they must be upgraded. If current systems can be reused, the new design can integrate them. If not, the team can find new systems, and test them to confirm they meet the requirements.

**Deliverable: Logical Design**

A Logical Design identifies the services, equipment, network architecture, and addressing structure necessary to create a network that satisfies its requirements. This phase should produce a Logical Design document that includes:

- Logical network diagrams
- Addressing strategy
- Security scheme
- Specification of hardware components, software, WAN links, and general services
- Specification of new hires or training for the network staff
- Initial cost estimates for hardware, software, services, personnel, and training

**Phase 4: Physical Design**

The Physical Design shows how to make the logical design work in the real world. In this phase, the network designer creates a detailed specification of the hardware, software, links, services, and cabling necessary to implement the Logical Design.

**Deliverable: Physical Design**

Physical Design outputs guide the equipment procurement and installation, thus the Physical Design document must be as specific and detailed as possible, often including:

- Physical network diagrams and to-scale wiring plans
- Detailed lists of equipment and parts
- Cost estimates for hardware, software, and installation labor
- Installation schedule that specifies the time and duration of physical or service disruptions
- Post-installation testing plan
- User training plan
Phase 5: Installation and Maintenance

Installation

A smooth installation is the reward for thorough work in the first four phases. When network developers are disciplined enough to invest real effort in the earlier phases, they find that they have already solved or prevented many common installation problems.

Of course, the main output of the Installation phase is the network itself. However, a good installation should also produce:

- Updated diagrams (logical and physical) that include all last-minute changes
- Cabling, connections, and devices that are clearly labeled
- Any notes or documents that can simplify later maintenance or troubleshooting, such as test results or new traffic measurements

Any necessary hardware or software must be purchased and tested before installation can proceed. In a broader sense, any resources the network needs before its final deployment should also be arranged. New employees, consulting services, training, and service contracts are all resources that may need to be in place.

The procurement of these resources should always occur before installation begins in earnest. If a total system cannot be procured and tested prior to installation, a complete or partial redesign may be necessary. Although painful, it is better to deal with it before the network staff has already dismantled sections of the existing network.

The objective of the whole design process is to answer questions, make decisions, and discover problems before the installation phase begins. However, nobody is perfect, and the best plans cannot always prevent unexpected problems. Therefore, it is important that the designer participate in the network's installation.

Maintenance

After the network has been installed, the network staff shifts its focus to getting input from the user community and monitoring the network itself for potential problems. As each set of additional requirements arises, the network life cycle repeats.
Activities

1. Describe at least three problems that can be avoided by using a formal design process.

   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________

2. List the five general steps or stages of the waterfall cycle.

   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________

3. List the four steps of the spiral cycle.

   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________

4. Unlike the waterfall cycle, it is relatively easy to return to a previous stage in the spiral cycle to make adjustments. True or False

5. Briefly describe one advantage and disadvantage of each of the two systems development life cycles.

   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________
6. List and describe the five phases of the network design process.

7. Describe what is meant by a project’s “deliverables,” and at what phase of the design process are they quantified?

8. In your opinion, why is it important to break down business needs into several areas of requirements?


10. Compare the Logical Design and Physical Design phases.
Extended Activities

1. Search the Internet and find out what tools are available for the following tasks:
   a. Gathering requirements
   b. Measuring availability
   c. Measuring response time

2. Review other development processes (such as manufacturing) and discuss how they relate to the network design process.