EQA

CHAPTER 21

Network Planning and Design

The business user of data communications most often applies the technical material in this book to the planning and design of a data communications system, or to the operation and management of such a system. In this chapter, we deal with planning and design of data communication systems. We look first in at the larger issues of how the organizational strategy, culture, and policies affect the planning and designing of data communication systems. Next, we look at systematic methods for planning and design. Section 21.3 is an overview of design algorithms and tools. Appendix 21.A gives some of the more straightforward of the quantitative design techniques. Finally, Appendix 21.B is a case study of online book sales.

Planning and designing of data communication networks is immensely complex. We narrow the scope considerably. First, we limit ourselves to planning and designing medium size networks. These are most frequently owned by organizations for their own use; that is, private networks. This excludes the very large networks, especially those public networks implemented by communication service vendors such as the telephone companies, and the large Internet service providers. On the other end, we do not consider networks that are so small that they can be purchased "out of the box" and for which the planning, design, and implementation can all be carried out by a very few people, perhaps only one. We focus mainly on the network planning and design problems of user organizations with significant coordination issues; this usually means wide area networks. However, even those who work for common carriers and other communication service providers will find much of the material useful and certainly insight into the user (customer) perspective on these issues is valuable. With this reduction in scope, we are still left with much to consider. We give an overview of the most important aspects.

21.1 THE PROJECT ENVIRONMENT: THE BIG PICTURE

Before a data communications project even gets to the formal feasibility studies that are part of the development methodology that we propose in this section, it is useful to make a top-down, qualitative evaluation of a proposed data communications system. Such an evaluation need not take much time or resources and may result in stopping unwise ventures early. This evaluation should start from a clear understanding of the structure, policies, and culture of the organization or organizations that will be using the system. The business role of the proposed application must also be clearly understood. For example, one should be sure that the project is not implemented just because some advanced or new technology seems interesting. On the other hand, one must be careful that focusing too narrowly on the business need does not unnecessarily limit or misdirect the technical approach. Since data communications projects take place in an environment of rapid technological advancement, it is important to closely examine technological risk. Finally, external factors such as government policy and regulation, the competitive situation, and available technological services and products must be considered. We now consider these in order.

Organizational Strategy and Culture

Ideally, any data communications project should be planned in the context of an organizational information strategy and policy. Formal and informal policies regarding outsourcing, turnkey procurement, buying of services, and in-house development are important. Sometimes policies affect the use of public versus private networks. The amount of human and technical resources in the data communication functions of the organization also strongly affects these choices. Developing a sensitive awareness of the organizational culture going into a project will help avoid later grief. For example, it is very important to know where your organization is on the centralized/decentralized management continuum. Usually, but not always, management of an organization's network will be centralized or decentralized according to whether the general management structure is centralized or decentralized.

Unfortunately, electronic communication is so ubiquitous in modern business that it is hard to develop an overall strategic vision that is comprehensive and at the same detailed enough to be useful. But a modest effort can yield a strategy to guide the development.

At this point you need to understand who are you connecting with the system, what the users are going to communicate, and what resources your organization has—financial, human, and time—to implement the project.

Business Role of Applications in the Organization

When deciding on a data communication project, there can be two types of mistakes: attempting a project that is not justified, and not implementing a project that is necessary and/or valuable. You can often avoid these mistakes by asking yourself, What happens if the project fails, and then, what happens if the project succeeds? If the success of the project would not make a substantial positive difference in your organization's activities, then the project may need rethinking. Perhaps a more aggressive approach is needed to make the project offer clear advantages. On the other hand, if there are significant and unfortunate consequences of not doing the project, or if major opportunities will be lost, then not only should the project go ahead, but a conservative path should be taken in its development to make success more likely. In any case, it is important to recognize whether the application is seen as a requirement of doing business or as an opportunity for the organization. These initial evaluations do not substitute for, and should be followed by more formal return on investment, or cost benefit analyses. But, it should not take numerical evaluations of several significant figures in financial models or the successful application of extreme and risky technological approaches.

Technology Push/Demand Pull

The impetus to implement technologically oriented projects—which most data communications projects are—is often characterized as pushed by technology, or pulled by demand. In the first case, the availability of new technology with major new capability leads to an evaluation of whether the technology can be used profitably within the organization. That is, a consideration of the technology precedes the determination of the business application. Demand-pull represents the situation where the planners start with a business need and look for the appropriate technology to satisfy it. A good example of both is e-commerce. Few traditional organizations that were early users of the technology felt a requirement to do business electronically. Rather, they saw the availability of the technology that might reduce costs and expand markets. This is an example of technology push. Later, as electronic businesses became significant, electronic commerce became a competitive requirement. For an example, see the Case: "Selling books, ... online" in Appendix 21B.

Technological Risk; The "Bleeding Edge"

The aggressiveness in which new technology is used in projects can strongly affect the chances of project success. If you are too aggressive in using new technologies before they are well proven, the technologies may not be available when advertised, or they may not work as advertised. This can delay the project, prevent it from meeting its specifications, or, ultimately, make the project fail. On the other hand, too timid a use of technology can make the project obsolete the day it is completed.

External Factors

The many external factors affecting your project should not be neglected. These include government(s) regulation, activities of your competitors, and the current and projected availability of technology.

21.2 PLANNING

It is important to have a formal planning procedure for any nontrivial project. There are many project-planning methodologies; however, most are similar. Many organizations have their own, "blessed" versions, but the mapping from the methodology we suggest here to other methodologies should be reasonably straightforward. It is sometimes argued that most projects involve modifications of existing systems, and therefore formal system planning is too time consuming and offers meager benefits. This argument is often false in the premise and/or the conclusion. The exponential growth of Web-based communications, particularly e-commerce, using the Internet calls for new networks or radical redesign of existing networks, not an evolutionary change from previous networks. But even if the proposed project is a seemingly straightforward enhancement to existing systems, a sequence of incremental changes without a well-thought-out strategy guiding the development results in baroque networks that are opaque to the user and difficult to manage.

All the methodologies consist of a number of stages to be performed in the project development process. Whatever the methodology, it is essential that at the end of each stage management make an explicit and written decision whether to abort the project, proceed to the next stage, or go back to previous stage and resolve specifically defined issues. One typical methodology is outlined in Table 21.1 and discussed in what follows.

Initial Definition of Scope and Main objectives

At the start of a project, you will often be given an informal characterization of the task at hand—sometimes very informal. A crisp, unambiguous, written characterization is necessary at this point. This description should summarize the results of the kind of strategic, high-level analysis described at the beginning of the previous section. Some of the issues to be addressed are, Who is communicating with whom? Is the project designed to support communications within the company, communications with

Table 21.1 Steps of a Development Methodology

1. Initial Definition of Scope and Main Objectives
2. Feasibility Study
3. Requirements Analysis
4. Functional or Black Box Specification
5. Options Analysis
6. System Architecture
7. Detailed Design/RFP
8. Implementation
9. Training and Cutover
10. Evaluation
11. Upgrading/Replacement

vendors and customers (business-to-business), communications with customers (retail), or a combination of these? What is to be communicated? What business functions will the proposed network support? What, in general terms, is the business rationale for the project? What is the time frame for the proposed project? Who is on the net; who is off; what classes of services are to be provided?

Feasibility Study

The feasibility study for a project is very important because it is usually the last opportunity to make major changes in the project before substantial resources are expended. At this point quantitative cost/benefit analyses are required to make sure that the project has a high expectation of success. Part of the feasibility study is to make sure that the budget and time allowance is sufficient for the objectives specified in the initial definition step. The feasibility study will be based on assumptions that must be made explicit, in writing. If, during the project, one or more of these assumptions becomes invalid, an immediate assessment of the project should be made to see if adjustments are needed to maintain feasibility. Another appraisal needed at this point is of technological risk. Choosing exactly which generation of technology to use is fundamental. Unfortunately, appropriate technology is a moving target. For most projects, available technology will improve significantly in the period of implementation. One popular indicator of the exponential growth of computer technology is Moore's law, which, in one of its manifestations, tells us that the performance of computer chips as measured by the number of transistors doubles every 18 months. In any case, a project, especially a slowly developing one, will find technology growing under its feet.

Requirements Analysis

The objective here is to refine and make quantitatively explicit the objectives of Step 1. This starts with specifying the explicit services to be provided (see Chapter 2), such as voice, data, Web services, e-commerce, and various types of multimedia. To the extent possible, future services must be provided for as well.

For each service, one must quantify current traffic and project this traffic into the future. Particularly difficult is traffic modeling for new or projected services for which there is no current traffic to use as a baseline. The least likely traffic for such a network is what you projected. Either the network fails and you get less traffic, perhaps none, or the network/application succeeds, in which case you must take rapid steps to prevent being overwhelmed. Quality of service (see Chapter 8) is also an important issue in modern requirements analysis. Differing services require differing performance guarantees. For example, video and voice require stringent delay guarantees, while data connections permit no data loss or corruption. Thus traffic volumes must not only be characterized by their sources and destinations but by their quality of service requirements as well.

The dynamic nature of traffic also offers complications. Traffic rates have trends and cyclic variations that must be considered. The load on most data communication systems grows with time. In addition, traffic levels fluctuate by the time of day, day of the week, and season of the year.

Collecting traffic data and reducing it to a form that can be used in design is extremely time consuming and error prone. The information is often incomplete and almost always come from multiple and conflicting sources. Requirements must be systematically represented. Each requirement can be represented as a list of data senders, a list of data receivers (these two lists often consist of one entry each, but, for example, multicasting applications have longer ones). For each of these requirements the type of communication service—voice, data, various types of multimedia—must be specified. For each service the traffic volume is required. Usually a dynamic specification of the volume is necessary reflecting the daily, weekly, monthly, and yearly traffic patterns and long-term trends. Quality-of-service requirements need to be specified as well. These include delay constraints (both in magnitude and variation), probability of packet loss constraints, and guaranteed capacity, availability, and reliability (e.g., diverse routing). Again, while we describe the process of collecting requirements as being independent of the design, in fact the process is iterative. For example, the use of local area networks facilitates some kinds of multicasting. When these technologies are included in the design, unforeseen requirements often materialize.

Fortunately, modern network management systems and standards offer support for requirements analysis. For example, the Management Information Base (MIB) of the Simple Network Management Protocol (SNMP) offers much useful baseline information for the objects in existing networks—hosts, bridges, routers, and hubs, as well as transmission facilities. RMON, a remote monitoring standard, allows network wide collection of network-monitoring data, particularly from Ethernet LAN segments. RMON (RFCs 2021 and 1757) makes it possible to collect automatic histories of traffic statistics such as utilization and congestion.

Finally, some global requirements must be addressed. These include privacy/security issues and network management functions.

Functional or Black Box Specification

The goal is an input/output characterization of the system from the user's perspective. How does the system look from the outside? What do users see? What can they do? A careful consideration of human factors is essential. The output of this stage is, in a sense, a contract with the user community defining what the communication system will do for them. For the credibility of the project, it is essential to have objective (and preferably quantitative) targets for service—performance, reliability, response, and so on—so that service to the users can be monitored. To the extent possible, the system should include automatic monitoring of these service objectives measures.

Options Analysis

At this point, with a good grasp of the objectives and requirements of the project, one can turn to the identification and evaluation of available implementation options. One way to do this is to use the information so far gathered and prepare a Request for Information (RFI) to send to vendors to gain a general notion of the equipment, facilities, and services they can provide that are relevant to the objectives and requirements. In any case, you need to systematically collect data on the devices, transmission facilities, software, and services that may be useful. In each case you need to know the features, costs, financing options (lease, buy, etc.), availability, reliability of the vendor, and vendor customer support.

System Architecture

The main task is to select from the options identified in the options analysis the networking approaches to be taken to support the requirements identified in step 3 of Table 21.1 and the functionality defined in step 4. What roles do LANs, MANs, and WANs play? Is wireless technology called for? What kind of distributed computing applications are involved and how should they be supported by communications networking (see Chapters 6 and 7)? If there are multiple networks, how do they interconnect? Parts 3 and 4 of this book provide guidance for making these design decisions. In addition, the acquisition strategy should also be identified: what elements to build, what to buy, and what to outsource. Standards play a very important role in designing communication systems. They often determine if you have the safety of alternative vendors. So you must decide which standards to require in your design (see Appendix B).

In today's environment of rapid technological change and uncertain requirements, a primary objective is to maintain flexibility: lease, don't buy; use accepted standards; don't get locked into one vendor's products or services. Pick technologies and architectures that scale; that is, that can be gracefully modified to support increasing demands without requiring radical redesign.

Detailed Design/RFP

At this stage we prepare the documents against which purchases, implementation, contracts, and other financial commitments will be made. We must specify in almost stupefying detail how the communications system is to be implemented. Consultants and vendors may help, but the owner is ultimately responsible. The users of the system must be identified. The locations of the equipment must be specified. The applications that will be supported must be detailed. The capacity and performance of the systems must be quantified. Security and reliability requirements must be set forth. The costs of equipment, transmission, and services (including support and maintenance) must be spelled out.

Deployment and cutover (transition from old to new system), together with payment schedules, must be set down. The cutover plan must make provisions for a fallback if the new system does not perform as well as expected so that essential operations are maintained. If possible, the new and old systems should operate in parallel until the new system is proved in operation. Acceptance testing should be implemented as a formal procedure to determine that the development is complete. Arrangements for user training must be made. For systems involving technical risk or other uncertainties, a pilot project might be called for.

Support for privacy and security must be specified. Network management tools to support the operation of the network must be specified in detail.

Implementation

This is the actual implementation of the network. The primary activity of the planner/designer is to establish a systematic review procedure to audit adherence to the detailed design document. In case of serious divergences, it may be necessary to cycle back to earlier steps in the development process and make adjustments. The planner/designer usually plays an important role in the acceptance testing as well, which ends this step.

Training and Cutover

A detailed schedule should have been prepared for user training that is to be completed before the cutover. If a pilot is part of the development plan, it is often useful to test the training plans as well. A critical decision here is when to allow the fallback facilities to be eliminated.

Evaluation

After the system has been in operation for some time, it is important to have a scheduled and formal evaluation of the system in light of operational experience. Some of the factors that should be considered are, Did the system achieve its operational objectives? Do the users find the system responsive and dependable? What was/is the financial performance? Did the project come in within budget? Are the operational expenses within budget? Were the financial benefits of the project realized? How does the actual load on the system compare to the projected loads?

Upgrading/Modifications/Replacement

In virtually all cases, the evaluation step will identify many surprises, frequently unpleasant. These will often need to be addressed by modifications to the system. Moreover, it is never too early to start planning for the upgrading or replacing the system. A major error is to look at network planning and design as an event rather than a process. Modifications, upgrades, and replacement will take place continuously. There will not be a point at which victory can be pronounced and the project declared complete.

21.3 DESIGN TECHNIQUES

The Model

The design process starts with a model of the system, often mathematical. The model involves variables (and two kinds of relations among them), constraints, and the design objective. The designer attempts to choose values for the variables so that the constraints are satisfied and the objective optimized. We generally assume that an architecture is given and that it is only the sizes, numbers, and locations of its elements as well as their interconnections that remain to be determined.

The model of the entire communications system is made up of models of traffic and demand, models of communication facilities, and models of terminal and switching devices.

There may be many variables, but they can be divided into a few categories. There are variables that measure (a) cost and return, (b) performance and reliability, and (c) traffic.

In most design models, the costs are divided into initial costs and recurring costs.

There are many variables characterizing performance. Delay, blocking, percent packet loss, throughput capacity, mean time between failures, and availability are examples; there are many others. These variables define quality of service.

Characterizing traffic is often the most time consuming and expensive part of the design process. The first difficulty is that, at best, you only know what traffic there was in the past and not what traffic there will be over the future lifetime of the proposed system. Especially in today's environment of rapid technological change, you often are designing a system for applications that did not previously exist or, if they did exist, were handled previously in such a radically different way from the proposed approach that past data are of little use. Internet systems to support Web traffic, multimedia, and/or electronic commerce are common examples. The next difficulty is that traffic requirements must be specified for each sender of information to each receiver or group of receivers. This gives rise to a combinatorial explosion in required data. For example, if we have 100 users, there are 9900 potential to-from pairs of users; with 1000 users, there are 999,000 possible pairs. Obviously, for major systems the users must be consolidated into groups. But doing this in an appropriate manner is not trivial. The third difficulty is dealing with the dynamics of traffic. Traffic levels vary in random ways in the short term; often have daily, weekly, monthly, and yearly patterns. The appropriate way to deal with traffic dynamics depends on the applications of the communication system. For example, many retailers make the overwhelming part of their sales in the Christmas season, and many of their communications systems must support the intense traffic during this time, which could be much greater than the average load or the load at other times of years.

The selection of relations as constraints or the objective is somewhat arbitrary. Often one is interested in the trade-offs between these relations. For example, in one context you might be interested in minimizing average delay of messages, constrained by the requirement of a given capacity. In other contexts you might wish to maximize the capacity given an upper bound on the average delay as a constraint.

Table 21.2 Steps in a Development Methodology

Given	Determine	Objective
Traffic requirements, network topology, routing of traffic	Capacity of network transmission channels	Optimize trade-off between channel costs and network performance
Traffic requirements, network topology, capacity of network transmission channels	Routing of traffic in network	Minimize traffic delay
Traffic requirements, network topology	Capacity of network transmission channels, routing of network traffic	Optimize trade-off between channel costs and network performance
Traffic requirements	Network topology, routing of traffic, capacity of network transmission channels	Optimize trade-off between channel costs and network performance
Terminal locations, traffic requirements	Location of multiplexers, concentrators, and/or routers	Minimize channel costs
Terminal locations, traffic requirements, location of multiplexers, concentrators, and/or routers	Assignment of terminals to multiplexers, concentrators, and/or routers	Minimize channel costs

Computerized network design tools are often used to select the values of the variables given the relations between them, the constraints, and the objective. We may categorize these tools by the problems they solve and/or by the techniques they use to solve the problems. Table 21.2 (based on [VANS86]) summarizes some of the major categories of network design problems. Typically, network design tools provide suites of algorithms solving a variety of these problems.

Network Design Tools and Algorithms

Network design tools are systems built around suites of design algorithms. The tools support the algorithms with user-friendly graphical user interfaces. They also provide network-editing facilities so that networks can be easily modified to produce multiple "what if" scenarios. Often the tools also add some sort of version control to keep track of all these scenarios. Databases for data such as traffic, device, and tariff information are also provided. Most importantly, the tools provide integration among the various algorithms in the suite. Pointers to some current commercial design tools are given in Section 21.3.

The typical algorithms for solving the models can be characterized as exact fast algorithms, exact slow algorithms, and heuristics or approximate algorithms. In addition to these analytic techniques, discrete event simulation is also common. Exact fast algorithms such as shortest path, minimum spanning tree, and sorting algorithms are taught in beginning computer science algorithm courses [CORM01]. They can be implemented very simply and run efficiently even on very large problems. Unfortunately, they are fragile in the sense that seemingly trivial modifications to the underlying model can make the algorithms inappropriate; the algorithms are not robust with respect to model changes. There are other problems for which known algorithms are very slow, sometimes not much better than brute force enumeration. These are often not useful for practical sized problems. The traveling salesman problem (which has significant communications applications) is a well-known example of this type. For problems with no known efficient algorithms, approximate and/or heuristic methods can be used. Discrete event simulation, which is a simulation technique that is popular for modeling communication systems, is another possibility. It is the most flexible approach to modeling. However, it can be very expensive computationally, especially for large networks. The wide variation in the characteristic times of a communication network makes a unified simulation impracticable. Cycle times of computerized switches and bit times of fiber optic channels are measured in nanoseconds, bit times for wireless transmission are measured in microseconds, human response times are in seconds to minutes, and mean time between failures of communication devices ranges upward from months. This makes simulation challenging for realistically sized networks. In addition, the size of modern networks, their very high data rates, and the relatively small sizes of ATM cells on other data units makes simulation prohibitively time consuming for general use. However, the technique is very useful for modeling individual devices and complex protocols on small nets. Virtually all commercial tools that use discrete event simulation use hybrid methods that mix analytic and discrete event simulation. The algorithms for whole networks generally are analytic, while detailed behavior of switches and other devices may be simulated [VANS74]. See Appendix 21B for a more detailed discussion of some of the simpler algorithms.

21.4 SOME CAPACITY PLANNING AND NETWORK DESIGN TOOLS

NetRule, Analytical Engines, www.analyticalengines.com, a Java based tool for WAN based networks. It appears to be elegant and relatively simple to use and uses primarily analytic algorithms.

OPNET Technologies, www.opnet.com, offers a software suite designed to support network service providers. The analysis and design tools emphasize integration of analytic and simulation techniques to provide accurate results in reasonable time for large, complex networks.

The Agilent N2X suite, advanced.comms.agilent.com/n2x, provides tools for validating the performance and scalability characteristics of network equipment for voice, video, and data services.

Nagios is an Open Source host, service and network monitoring program, http://www.nagios.org.

[BRAG00] is an excellent guide to network design tools, although some product offerings have changed.

APPENDIX 21A SOME SIMPLE DESIGN ALGORITHMS

Topological Design

We first discuss how one decides the layout of a network (i.e., which locations are connected to which other locations). Suppose our organization has computer centers in New York, Chicago, Atlanta, Dallas, Los Angeles, and San Francisco (Figure 21.1).

Figure 21.1 Six Data Centers

The New York center is the central database. All the other computer centers must be able to communicate with the central database. We are planning to connect the centers by leased communication lines. For simplicity we assume the costs of the leased lines are proportional to the distances between their endpoints, which are given in Table 21.3. For our first analysis we make the simplifying assumption that the lines will have sufficient capacity for the traffic even if we relay one center's traffic through another center. We simply wish to find the least-cost network for connecting the six computer centers where the cost is just the total of the costs of the lines selected. A little thought should convince one that, under these assumptions, the cheapest network will be a tree. If the network is not a tree, you can always take away a link, reducing the cost, while still allowing all the nodes to communicate. In Figure 21.2, you can remove the link from Dallas to Chicago and all the data centers will still be connected to New York. We will take the headend to be the central database.

Table 21.3 Distance Matrix

	ATL	CHI	DAL	LA	NY	SF
Atlanta	—					
Chicago	585	—				
Dallas	727	798	—			
Los Angeles	1944	1749	1251	—		
New York	748	719	1373	2462	—	
San Francisco	2145	1863	1493	344	2582	—

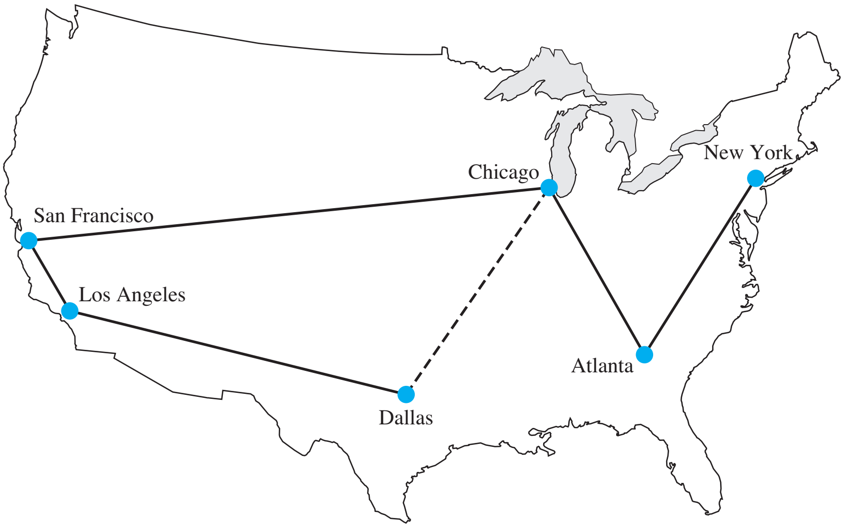

Figure 21.2 Network with Redundant Link

This type of problem is called a minimum spanning tree problem. There are several methods (or algorithms) for solving this type of problem.

We illustrate Prim's algorithm. It is quite simple. We start with one of the locations, say New York, and find the location that can be connected to it most cheaply. In our example, it is Chicago. We then find the new location that can be connected most cheaply to either New York or Chicago. For us, that is Atlanta connecting to Chicago. We then find the location that we can connect most cheaply to New York, Chicago, or Atlanta. In general, at each step we look for the shortest (least-cost) link between a location that is not yet on the tree. We add this link the tree and continue in this way until all the locations are connected. The progress of the algorithm is shown in Figure 21.3.

This algorithm always gives the correct result, and there are many efficient computer implementations of this, and similar algorithms, that can solve problems with thousands of locations in seconds ([CAHN98], [CORM01]).

Unfortunately, if you change this model even slightly, the problem becomes much more difficult. Suppose your organization had an additional facility in Louisville, Kentecky. It is not one of the data centers, so it need not necessarily connect to our network. But, perhaps surprisingly, if we do connect it, we save. Figure 21.4 shows the least-cost network that includes Louisville, and its cost (Table 21.4). We have saved 67 miles. This suggests that we could save even more by adding other inessential locations. This variation of the problem, called the Steiner tree problem, can reduce the network length. Unfortunately, solving this problem (allowing additional locations) is much more difficult than constructing a spanning tree because if you have a long list of potential locations to add, deciding which ones to include in the tree is difficult.

Another common, but difficult, generalization is to suppose that the links have capacities that limit the amount of traffic that can be carried. This small change also

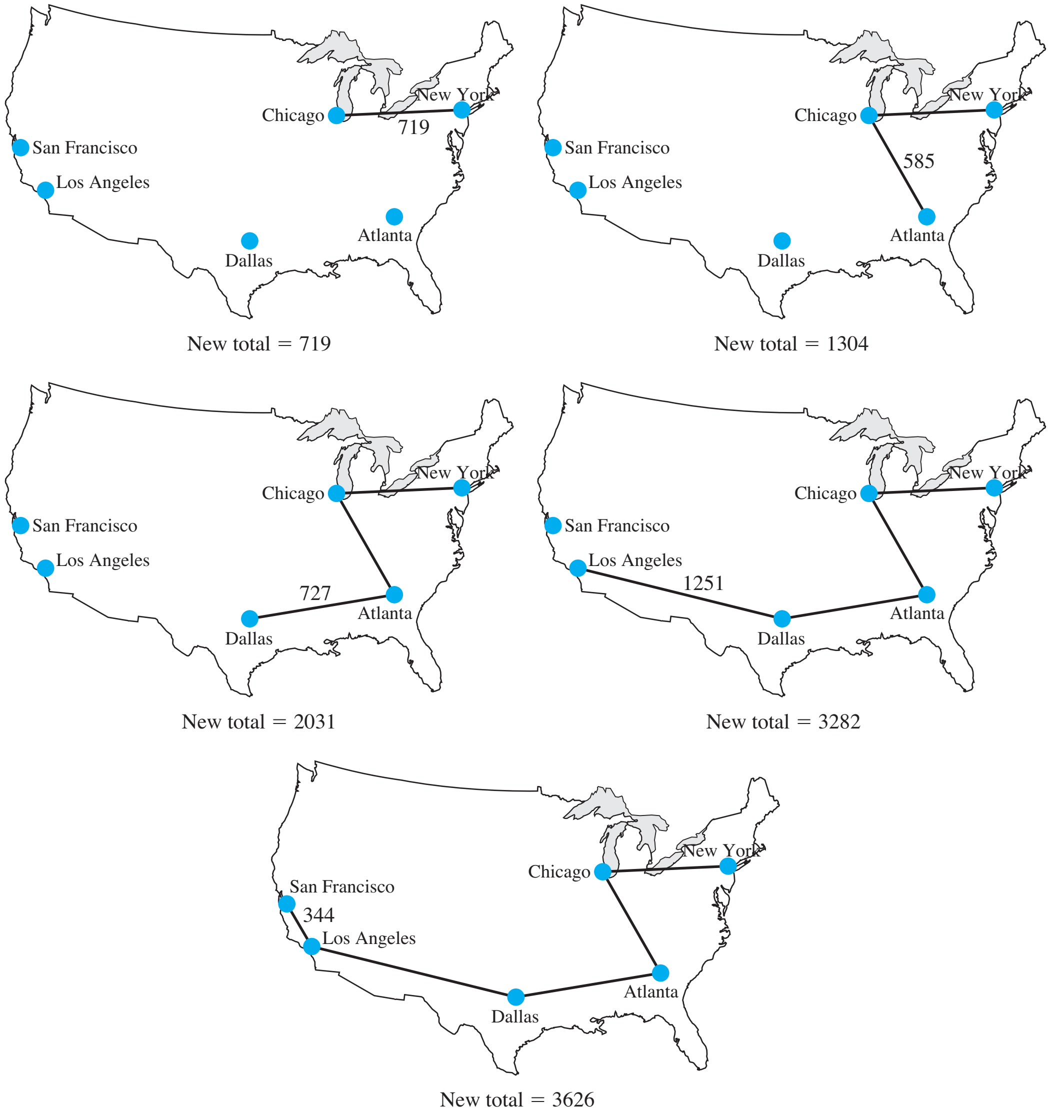

Figure 21.3 Prim's Algorithm

makes the problem much more difficult. Suppose now that that the traffic requirement between each data center and New York is 40 units of traffic, and that all the links have capacity 100. (This essentially limits each path to New York to contain three nodes in addition to New York.) Notice now that the solution of Figure 21.3 can no longer be used because the link between New York and Chicago carries all the traffic, a total of 200 units, which far exceeds the capacity, 100, of the connection. Finding the cheapest connecting network with capacity constraints can still be solved exactly because there are only a finite number of possible networks. This is an example of an exact but slow algorithm. For large networks, brute force

Figure 21.4 Least-Cost Net with Louisville

approaches such as this one are not feasible. Instead, heuristics are used that are fast approximate methods and that are shown, usually empirically, to give good if not optimal results. The Esau-Williams algorithm is a popular one for this problem [CAHN98].

By the time one has added all the essential features, including factors such as, cost, capacity, reliability, and performance, to the topological design of significant size networks, heuristics are usually the only feasible approach.

Congestion

As our second example of network design methods we look at how to deal with congestion in communications facilities such as networks or individual devices and communication lines. In this general situation, as traffic throughput increases

Table 21.4 Least-Cost Net with Louisville

	ATL	CHI	DAL	LA	LOU	NY	SF
Atlanta	—						
Chicago	585	—					
Dallas	727	798	—				
Los Angeles	1944	1749	1251	—			
Louisville	316	270	725	1839	—		
New York	748	719	1373	2462	653	—	
San Francisco	2145	1863	1493	344	1996	2582	—

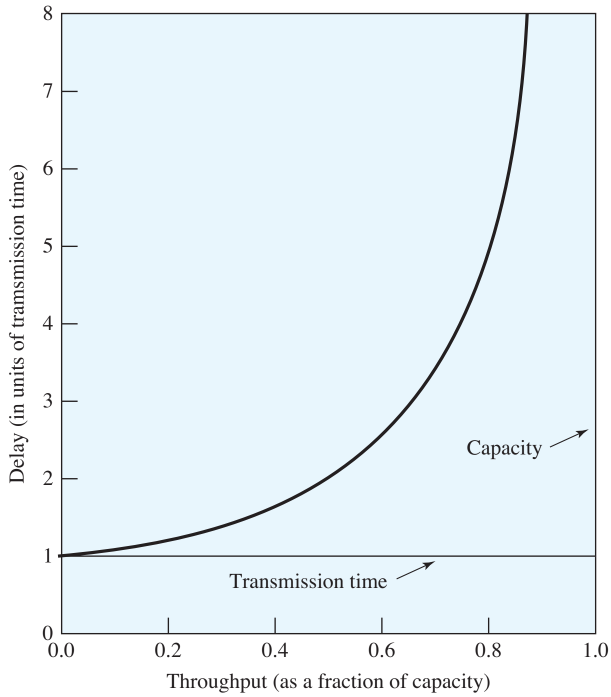

Figure 21.5 Throughput Delay

so does the average delay. Figure 21.5 is a schematic representation of the general situation. The vertical axis is the delay in getting a message across the facility, and the horizontal axis measures the traffic on the facility. The total delay consists of two parts. First, is the actual time it takes for the message to travel across the network, even in the absence of other competing traffic. This is called the message transmission time. The other part is time spent waiting in devices for access to channels, or the delay caused in going through the channels because of competing traffic. We call this the congestion time. The graph gives the average total delay of a message as a function of the traffic on the facility. The vertical intercept of the graph represents the message transmission time; that is, the delay that a message encounters even if there is no other traffic on the network. Some of the contributing factors to the transmission time are the propagation delay across the network, the transmission time for the message itself, and processing of the message in the switches and communications lines. The traffic value associated with the vertical dotted line on the right is the capacity of the facility. Again we start with simple illustrations of this general type of relation and then investigate more realistic models.

M/M/1 and M/G/1 Queues Let's start with a single communications channel. First we turn to the issue of message lengths. We will assume the lengths can vary from packet to packet. We need to characterize the distribution of these messages. We will assume the memoryless property, which is sufficient to characterize the distribution. Messages with random lengths are said to have the memoryless property that if someone tells you that the message is at least $\boldsymbol{x}$ units long, then the probability that it is $\boldsymbol{x} + \boldsymbol{y}$ units long is the same as the probability that a message is $\boldsymbol{y}$ units long given no information. That is, if you see a message coming over a channel, the fact that you know how long the message has been transmitting tells you nothing about how much longer it will transmit. This model certainly doesn't work for ATM cells—they are all the same length—or for packets—they cannot be longer than a given packet length maximum. However, this model is reasonable in many situations, such as the length of telephone calls. This model is very popular because it is easy to analyze. We start with this model and then discuss other message length models.

Next we have to make assumptions about the arrival times of messages. We make use of the same memoryless property. Here we look at the distribution of time between arrivals of messages. The memoryless property comes into play here when we assume that if for a give message sources, the probability that the time until the next arrival will be a given value is independent of how long it has been since the last arrival. We also assume that the arrivals are independent of one another and that the average rate of arrivals does not change with time. To simplify things, we also assume that the messages can take on noninteger values. For telephone calls this assumption is reasonable. For data communications, since messages are some integral number of bits, the assumption is more questionable, although for long messages it causes little error.

If we are told a light bulb has been burning for several thousand hours, it is more likely to burn out in the next five minutes than a new bulb. On the other hand, electronic devices tend to fail early. If they don't fail in a short time, then they last a considerable time. So an electronic device that has been "burnt in" and works for some initial period is less likely to fail in the next five minutes as compared with a new, unused device. In between these two cases is the memoryless case. Here knowing how long the device has been working tells us nothing about the future life.

A single channel with memoryless arrival distribution and memoryless message length distribution is called an M/M/1 queue. The trade-off between average delay and average throughput looks like Figure 21.5; in fact, the relation between delay and traffic represented in Figure 21.5 is that of a M/M/1 queue. Let us try a simple application. Messages with an average length of L bits are to be sent over a channel with capacity C bits per second. We suppose the messages arrive to the channel on average, every A seconds. When a message arrives at the channel, if the channel is free the message is sent on the channel. Otherwise, the message waits until all the other messages that arrived before it have been sent; then it is sent. Then the amount of traffic that arrives, on average, to the channel is L/A. If $L/A > C$, then the traffic will back up indefinitely because the traffic is arriving faster than it can be transmitted. The difference can only go to the queue of waiting messages which gets longer and longer. We will define traffic relative to the channel capacity. That is, we will work with $f = (L/A)/C$; this is known as the utilization of the channel. Therefore, in the units of f, the queue will have capacity 1. We can also easily compute the transmission time for the message. If the channel is free when the message arrives, then it only takes, on average, $t = L/C$ seconds to transmit the message; this is the transmission time.

Table 21.5 Types of Traffic

Type of Traffic	Message Model	c
ATM cells	Fixed-length messages	0
Data packet communications	Bounded-length messages	$0 < c < 1$
Circuit switched messages (e.g., telephone calls)	Memoryless	1
Multimedia	Fractals	>> 1

One can show without too much difficulty (see, for example [KLIE75]) that the relation between the utilization, f, of the channel and the message delay, d, is

$$d = \frac{L/C}{1 - f} = \frac{t}{1 - f} \tag{21.1}$$

Note that this has the correct properties. For $f = 0$, we get as the transmission time L/C, and the delay blows up as we approach the capacity $f = 1$. Equation (21.1) can be generalized to remove the memoryless assumption for the message length. This is important because many types of traffic that we commonly encounter do not have the memoryless property. ATM cells and packets in a TCP/IP network are examples that we have already mentioned. The generalization of the M/M/1 formula (21.1) is the M/G/1 formula:

$$d = \frac{\frac{t}{2}(1 + c^2)}{1 - f} \tag{21.2}$$

where c is the coefficient of variation and measures how variable the message lengths can be; the bigger c is the more the variation of the message length (Table 21.5). For the case of fixed-length messages such as in ATM cells, there is no variation, then $c = 0$. For memoryless length distributions, $c = 1$. For packet communication with maximum length for the packets, generally we have $0 < c < 1$. So, for example, if the message lengths are have a deterministic length, the average delay will be 1/2 the delay for the case of memoryless message lengths. For some types of multimedia traffic with multiple types of traffic sharing the channel c can be quite large. Researchers have observed fractal-like behavior in traffic with many types of services ([LELA94], [PAXS94]). This type of traffic causes major problems for both analytic and simulation quantitative design techniques.

Little's Law A fundamental, and simple relation with broad applications is Little's Law. We can apply it to almost any system that is statistically in steady state, and which there is no leakage. The general setup is that we have a steady state system to which items arrive at an average rate of A items per unit time. The items stay in the system an average of W units of time. Finally, there is an average of L units in the system at any one time. Little's Law relates these three variables as $L = AW$. To illustrate the use of Little's Law we return to the M/M/1 delay formula (21.1). We now ask ourselves what is the average number of messages in the system, including both those waiting and those being transmitted? We use for W the average time in the system the delay given by (21.1). The arrival rate of messages, A, is the A used in developing (21.1). The average number in the system is then just the L in Little's Law. Thus:

$$L = AW = \frac{At}{1-f} = \frac{A\frac{L}{C}}{1-f} = \frac{f}{1-f} \tag{21.3}$$

We now look at another example that illustrates why simulation of modern networks is so difficult. Suppose we have a wide area ATM network with a DS-3 link from New York City to Los Angeles. From Table 21.3 we see that the link is at least 2462 miles long. A DS-3 link has a capacity of 44.836 Mbps. An ATM cell contains 53 bytes or 424 bits. Assuming no overhead, we could transmit (at most) $44.836 \times 106/424 = 105{,}745$ cells per second. Assuming no congestion and just propagation delay, and further assuming, optimistically, that the cells propagate at the speed of light, 186,000 miles/second, the delay is at least $2462/186{,}000 = 0.013236$ seconds. Little's Law with $A =$ cells per second, and $W = 0.013236$ seconds, tells us that there are $105{,}745 \times 0.013236 = 1{,}400$ cells "in flight"; that is, up to 1,400 cells could all be in transit at a given time between New York and Los Angeles. A brute force, discrete event simulation of this system would have to have the capacity of keeping track of at least 1,400 cells and this is just for one link. This should make it clear why so much effort and ingenuity is expended by vendors of network design tools to avoid straightforward simulation of networks.

SUMMARY

The rapid pace of technology with broad band channels, multi-media traffic, and thousand node networks is overwhelming network design techniques. Here we have given a few simple methods that can often be used in informal analyses or for rough estimates. In the discussion, we also try to illustrate some of the issues that more sophisticated design methods must face.

APPENDIX 21B SELLING BOOKS ONLINE — A CASE STUDY

In the Spring of 1994 a 30 year old, recent, Princeton graduate was investigating the Internet for D. E. Shaw's consultancy in New York City. He was astonished by data that indicated that the newly developed Web was growing at a 2300% annual rate. He quickly decided that he must seize the opportunity signaled by this phenomenon or regret it the rest of his life. But the question was, exactly what was the opportunity? Bezos, assuming that products successfully retailed by mail-order companies could also be sold on the web, made a list of 20 possible product categories that he might use the burgeoning Web technology to sell. He chose books. One reason was that there are many more books in print than any physical bookstore could possibly stock, or than a mail-order catalog could list. Moreover, the market was fragmented as contrasted with music which he also considered but initially rejected because it is controlled by a small number of large distributors. Too, the distributors had well-documented book lists already on CD ROMs ripe for online use. Less than two months later, on July 4th, Jeff Bezos left D. E. Shaw and New York, and headed west to Seattle to seize the opportunity.

Barely a year later, in July of 1995, Jeff Bezos was selling books from his newly formed bookstore, Amazon.com. He bought print ads claiming to be the "Earth's Biggest Bookstore," a not so subtle dig at the U. S.'s largest retail bookstore chain, Barnes and Noble, which called itself the "World's Largest Bookseller." But the Amazon bookstore was largely "virtual." Initially his company had about 200 books in stock. The rest of the over 2 million titles Amazon advertised were provided through distributors, or the publishers. This provided several advantages. It obviously reduced staffing, "bricks and mortar," and inventory costs. Amazon also received money from its customers up front, and it needn't (and didn't) pay the distributors and publishers for 30 days, providing the newly formed company useful "float." On the other hand, this approach didn't make for fast deliveries; so as time has passed, Amazon has accumulated huge warehouse operations throughout the United States.

Book selling has not been the same since. Nor the stock market for that matter. For example, in traditional book selling, about 80% of sales is from the best seller list, while 20% is in midlist and backlist books. These percentages are reversed in on-line book selling. Another unusual feature of Amazon's evolution is that except for a brief period of profitability in 1995, it has been losing increasing amounts of money each quarter as it invests in new product areas. While this is happening the valuation of Amazon as reflected in its stock value is becoming immense.

The Riggio brothers, Leonard and Steve, had not built Barnes and Noble into a chain of hundreds of stores by being unaware of challenges to their business. They quickly realized the significance of the Amazon and that they needed a Internet presence.

They launched their on-line business, barnesandnoble.com in May of 1997. It was a separate organization from the book chain in order to avoid having to collect state sales taxes in all the states with Barnes and Noble stores. This hindered them from integrating their on-line operation with their bricks and mortar stores. Thus, initially, they basically were only able to emulate Amazon and could not benefit from possible synergies with their bricks and mortar stores. In the Fall of 1998 they postponed a planned public offering of barnesandnoble.com shares; instead they sold half the operation to Bertelsmann, the massive German media conglomerate, which among other operations owns Random House and is the largest book publisher in the U.S. The price was $200 million to Barnes and Noble plus an additional $100 million investment in the operation. In May of 1999 a initial public offering of about $485 million was made, ending up with 40% of the shares with Barnes and Noble, 40% with Bertelsmann, and 20% public.

It is important to notice that the electronic technology may be the least important and least difficult aspect of selling on line. More essential may be the physical handling of the products. How to find them quickly in warehouses, how to make sure the cost of maintaining an inventory does not damage profitability, and how to get the products quickly, safely, cheaply, and reliably to their destinations.

Amazon confronted this issue directly. They have built up 7 distribution centers, 5 in 1998 alone, in states with few people, and no or little sales tax (they have to collect state sales tax for deliveries to customers in states where they have distribution centers). Seattle, Washington; New Castle, Pennsylvania; Reno, Nevada; Coffeyville, Kansas; Delaware, and Kentucky (2). Moreover, they clearly see their mission to sell almost any product category on-line. Books were only the first of the product categories they marketed. They started offering music CD's in June of 1998, and Videos in November of the same year. Later they added video games, jewelry, consumer electronics and home improvement tools. They have also developed auction sites. They used the high value of their stock to purchase major positions in online drug stores, on-grocery sales, pet sales, and car sales, all of which they link to their site. They have also worked with wireless vendors to enable customers to use their Palm Organizers, Sprint Wireless Phones, or other wireless devices to purchase from Amazon.

In the Summer of 1999, Barnes and Noble made a $600 million deal to purchase Ingram the largest distributor of books which has 11 distribution centers across the country. Also a significant part of the transaction was Ingram's print on demand operation, Lightening Print. However, because of challenges to the purchase by the FTC, on anti-trust grounds, the transaction was not completed. Barnesandnoble.com is now developing, in addition to its New Jersey facility, new distribution centers in Reno, Nevada, and Atlanta, Georgia.

For the future, Barnes and Noble has a different vision. They have limited themselves to books, music, video, and software. They noted that all these products are actually just information in one form or another. They look towards electronic delivery of these products to reduce the need for large physical distribution centers. For example, with electronic book inventories, no book need go out of print, and information can be easily updated. Storing bits is cheaper than storing multiple ink and paper copies too. In the near future they see books as being stored electronically, and when sold being downloaded to electronic books, "eBooks," or PCs; printed by the customer; printed on demand by the publisher or distributor in small but efficient runs; or printed individually, on demand, at bookstores. Barnes and Noble already offers the Rocket eBook which holds up to 10 books, has about a 20 hour battery life, supports audio and graphics, and offers various search and look-up features. They are also active in the Open eBook Consortium which has developed a non-proprietary standard based on HTML and XML for electronic files for eBooks. Included in the aborted Ingram purchase was Ingram's Lightening Print which has agreements with more than 100 publishers for printing out-of-print books on demand.