Section VI
Networking Tools and Techniques

THE FIELD OF NETWORK DESIGN HAS EVOLVED CONSIDERABLY since the use of communications to link remote users to mainframes began during the early 1960s. Today, the control unit-based, mainframe-centric network has evolved into workstations on LANs accessing mainframes in a client/server environment via bridges, routers, and gateways. Instead of only having to consider text-based applications, network managers and LAN administrators must now consider the effect of different types of multimedia applications on current and planned networks. To facilitate the network design process, a number of tools and techniques have evolved over the past few years that facilitate the design process and are the subject of this section.

In this section are three chapters focused on providing detailed information concerning the use of different networking tools and techniques which, when used, can considerably facilitate the network design process. In Chapter 43, author G. Thomas des Jardin's **A Simulation Method for the Design of Multimedia Networks** introduces three basic phases of the simulation method used for multimedia network design. In the preparation phase, the author describes how to define goals in measurable terms. Next, in the baseline phase, information concerning data capture and the validation of such data are presented. The third phase of the design process is referred to by the author as the delta phase. During this phase, changes are applied to the baselined network, after which the results are analyzed and summarized through information presented in the three phases. This chapter provides a methodology one can use to simulate a wide variety of networks.

The second chapter in this section focuses attention on techniques one can use to determine a number of quantitative metrics involved in client/server computing via bridges and routers. In Gilbert Held's chapter, titled **Determining Remote Bridge and Router Delays**, the author uses queuing theory to illustrate various performance metrics associated with interconnection geographically separated LANs via remote bridges or routers. Although the author illustrates how to project queuing delays, of far more importance is the presentation of how one can use queuing theory to answer a classic network design problem. That problem is the appropriate selection of a WAN operating rate to interconnect two geographically separated LANs.

In concluding this section, author Gilbert Held adds a second chapter. In Chapter 45, titled **Network Baselining as a Planning Tool**, Held reviews a core set of popular communications tools one can consider using to facilitate the network design process. Examples of planning tools covered in this chapter include Triticom, Inc.'s SimpleView, an easy-to-use SNMP management platform; NetManage's Newt, a program that provides statistics on network activity associated with individual users; Ethervision, a program from Triticom that provides statistical information on individual Ethernet network users as well as information concerning the overall activity of the network; and Foundation Manager, a product of Network General Corporation, which is now part of Network Associates, and which provides information on both local and remote networks via SNMP. Through the use of the products covered in this chapter or similar products, one can develop a baseline of network activity that is extremely valuable for planning network upgrades or for considering the replacement of a current network by a new network.

Chapter 43

A Simulation Method for the Design of Multimedia Networks

G. Thomas des Jardins

VENDORS OF NETWORK DESIGN TOOLS usually provide ample documentation and examples concerning the use of the tool. What is often lacking, however, is a methodology for applying simulation techniques in general.

Multimedia networks and the tools that represent them are complex, and simple examples are often insufficient to impart an understanding of the process or methodology that a network designer needs to use to achieve accurate simulation results. This chapter describes one such methodology, which is applicable to the use of tools for the design and analysis of multimedia networks that carry voice, video, and data. The chapter provides the structure and approach that are missing from the user manuals.

Furthermore, the information is intended to be helpful regardless of the simulation tool chosen. The chapter also discusses approaches to solving particular modeling problems that arise in multimedia networks.

THE MODELING PROCESS

Networks that are complicated enough to have multimedia segments are frequently too complex and too large for simple rule-of-thumb calculations. Simulation is the best means of collecting performance data on networks that are in the planning and design stages.

Complexity requires a great deal of organization in data collection, model validation, and analysis of results. A defined process increases confidence in the results by increasing organization and thereby managing complexity.

0-8493-0859-3/00/$0.00+$.50
© 2000 by CRC Press LLC

This chapter specifically draws to the reader's attention the following aspects of simulation:

- important data that will be required
- means for obtaining this data
- suggestions for modeling the data
- possible interpretations of results
- how the modeling procedure might be segmented into tractable units

Simulation results are highly dependent on the quality of data being input. The large amount of detailed data being handled increases the margin for error. By paying attention to the process, the network manager can maintain the data in an organized fashion, track the validation of the model, and increase the confidence in the results.

There is no such thing as being too organized. The corollary is that if there is any confusion about the results, the procedure should be interrupted and investigated until there is no longer any confusion. Bad data and bugs are the worst enemies of accurate simulations.

The process described in this chapter contains the following three basic phases:

- Phase I: Preparation
- Phase II: Baseline
- Phase III: Delta

The tasks for each phase are outlined briefly here, then discussed in detail:

- Phase I tasks include:
  — *Goals.* These must be stated in measurable and clear terms.
  — *Data collection.* The topology and traffic of the existing network is captured.
- Phase II tasks include:
  — *Capture.* The collected data is captured in the model.
  — *Validation.* The captured model is validated.
- Phase III tasks include:
  — *Delta.* Changes are applied to the baselined network.
  — *Analysis.* The results are analyzed and summarized.

PHASE I: PREPARATION

This phase includes the definition of goals and the collection of topology and traffic data for the baseline network. The first task is identifying the goals.

Identifying Goals

A simulation should have clearly defined goals. For the hypothetical case discussed in this chapter, there are two principal goals: The first is to develop a validated baseline model of the network in its current configuration; the second is to model the introduction of an asynchronous transfer mode (ATM) backbone.

A brief summary of the modeling strategy entails the following actions:

1. Decide if modeling is appropriate.
2. Determine simulation goals.
3. Describe the network in one or two slides.
4. Combine each goal and its network description into a series of scenarios, each with a simple, testable model description and a clearly defined goal.
5. For each model description, define the data to be collected, the results expected, and how the model will be validated.
6. Combine these individual documents into a simulation notebook.
7. After the individual models have been validated, repeat the process by combining the models into more complex models and validating each in a stepwise, iterative fashion.

Sample Goals of a Modeling Project. Frequently, network managers embark on projects without ensuring that they have defined achievable goals, although they may have an intuitive idea of what problem they seek to resolve and what steps might be useful to take. The difficulty is in translating these qualitative statements to quantitative goals based on known metrics. Without these goals, even beginning a modeling project is a waste of time.

Problem. Users are experiencing delays of 1.5 seconds running application X (problem could also be expressed in terms of low throughput).

Goal. The goals of a modeling project can often be stated simply. For example, the goal in solving the previous problem may be to "run application X, reducing delay to 0.5 seconds."

Experiments to consider to solve the problem include:

- *Segmentation.* Can further segmentation of the existing Ethernet LAN improve the performance to the desired level?
- *Backbone increase.* Can upgrading the backbone improve the performance to the desired level?
- *Segment upgrade.* Will upgrading the network improve the performance to the desired level?

Projecting Costs. Next, a spreadsheet for anticipated costs should be created. For example:

Experiments Considered	***Cost of Experimental Upgrade***
Segmentation	Bridges = $
	Routers = $
	Switching hubs = $
Backbone increase	New backbone hardware = $
	Plant (e.g., cabling) upgrade = $
Segment upgrade	Adapter cards = $

This step is usually followed by some qualitative analysis as to the amount of room for future growth, improvements in supportability, and so on.

Preparing a spreadsheet is relatively easy; however, it is not that easy to decide what precise changes are required to give needed quality of service. In addition, it is difficult to set parameters for wide area network interfaces because the answers to request for proposals for WAN services may use different sets of metrics.

Figures of Merit. The key is to define what measurements are important; these are often called figures of merit.

For example, is end-to-end delay the figure of merit that must be improved to provide users with the required quality of service? Are hardware costs more important than recurring costs? Is on-line transaction processing running across some segments? Voice? Video? For each set of users, there must be some number that can be extracted from the analysis that can quantitatively define the quality of service (QoS) being provided for each group.

To begin, a spreadsheet should be created for each group of supported users, showing the applications they are using currently or will be using in the future. For each of these applications, the metrics that describe the desired QoS should be defined (the network manager should already, to a large extent, be familiar with these).

Next, the available tools should be examined in light of the following questions:

- Can the modeling tool provide these measurements?
- Can these measurements be extracted from the network (for validation)?
- Do these measurements provide insight into users' satisfaction?

If the answer to any question is no, the tools, metrics, or decision to model should be reevaluated. If the modeling tool does not provide the desired reports, perhaps the manufacturer will modify the tool. If the network monitoring tools are inadequate, it may be necessary to add instrumentation to the network. The network manager may need to identify additional network monitoring hardware for acquisition or lease — or perhaps simulation is not the answer to the problem. Once these criteria are satisfied, the baseline should be defined by collecting the topology and traffic of the network.

Performance Metrics

In order to manage a network, its performance must be measurable and network goals must be specified in measurable metrics. Some of the more important metrics include:

- queue buildup
- end-to-end delay
- throughput
- jitter
- goodput

Models must be instrumented according to the data collection requirements, by placing probes or turning on data collection routines in certain modules. The way that data collection is performed is unique to each simulation tool, but most tools will allow the collection of much of this information. The following paragraphs describe these metrics.

Buffer/Queue Size. Queue buildup is an important indicator of potential congestion points. Queue size is one of the largest factors in delays on ATM LANs. It is also the largest contributing factor to jitter.

Large queues always add to end-to-end delay and can indicate possible packet loss, which could adversely affect throughput by causing retransmission. When a network device such as an ATM switch has more traffic than it can send out on a given link, it is faced with two choices: either discard the traffic or save it in a buffer.

Assuming an infinite supply of buffers, frequently buffering is the optimal choice because it improves throughput. Buffering, however, creates queue buildup, and cells of a given quality of service must wait in turn to exit, in order to maintain first in, first out ordering. This means that the cell that arrives at the back of the line must wait for all of the cells in front of it to exit the switch before it can exit, increasing its latency.

Latency or End-to-End Delay. Latency is the amount of delay introduced by a particular device or link; end-to-end delay is the sum of all latencies experienced from source to destination. End-to-end delay is important because many applications require a specific quality of service.

While buffering cells, rather than dropping them, can certainly improve throughput, this waiting increases latency and contributes to jitter. In ATM networks, latency is one of the QoS metrics that a committed bit rate (CBR) or variable bit rate (VBR) traffic stream may negotiate, and therefore may seek to minimize. In other types of networks it may simply be a design goal. In other words, some applications would rather have the data right away or not at all.

Jitter. Jitter is also an ATM QoS metric for which some CBR and VBR applications desire to negotiate a low value. If a cell arrives at a switch when it has a large number of cells in its buffer, and another cell arrives when the switch has a small number of cells in its buffer, the difference between the buffer sizes in these cases is that particular switch's contribution to a cell's total end-to-end jitter.

Jitter is minimized by maintaining constant buffer queue sizes. In general, ATM available bit rate (ABR) traffic expects to have a very low cell loss probability and higher latencies and jitter. Some switches address this problem by having multiple queues.

Utilization, Throughput, and Goodput. Utilization is the amount of time a link is idle, versus the amount of time it is in use. This calculation does not necessarily show how much data is reaching the end system.

Throughput defines how much data is delivered to the end system. Throughput needs to be determined to ensure that components and links are sized correctly.

The calculation of raw bandwidth is trivial, but does not include any bandwidth lost because of retransmission or other protocol and hardware interactions. Thus, throughput is important to measure because it allows the manager to estimate reserve capacity; it also reveals those interactions in the network that may lower overall throughput.

Goodput includes the actual upper-layer contributions to performance, such as whether TCP received the packets in a usable amount of time. In some instances, goodput is a preferred metric if it can be obtained. Some simulation tools, such as CACI's COMNET III, can measure goodput, throughput, and utilization.

Data Collection

Once metrics are defined, the actual characteristics of the network as it exists currently must be collected. This is the first step in baselining the network.

First, a topology data collection sheet is created. A sample is shown in Exhibit 43.1. Using the topology data collection spreadsheet, each hardware

Exhibit 43-1. Example of a topology data collection sheet.

Network Type	Node Description	ID		
Data	10Base-T hub	DH1		
Link Name	**Link Type**	**Speed**	**From**	**To**
WAN	Coaxial	DS-3	DH1	AT&T
Backbone	Fiber	SONET	DH1	DR1

device in the network and the links that connect them are documented. The number and rate of each link are identified.

This is also a good time to note all network costs. The cost of a link consists of the local component as well as the long-distance charges; some simulation tools are integrated with tariff data bases to some extent.

Most simulation tool manufacturers have or are planning interfaces to network management tools, which will simplify information collection on the data portions of the network. Tools have different levels of integration with network management. Some tools can collect topology information; others can also collect traffic information. There are three categories of traffic: voice, video, and data.

Traffic collection is the more difficult activity, especially for data and voice. Once again, a very careful, systematic approach yields the best results.

Some useful numbers for calculating propagation delay times in various media are:

Medium	***Propagation Delay***
Coaxial cable	4 μs/km
Fiber	5 μs/km

Voice Network Information. To collect the topology and traffic information for the voice portion of the network, here are some recommendations as to how the information might be represented in the model.

Topology. All segments of the voice network have to be described, including trunks, PBXs, and additional analog lines used by fax and data equipment for each location that is serviced by the network. Links to remote users also should be documented by listing the closest points of presence (POP) of any services that remote users will be calling. Quality of service and bandwidth required for each link should be noted.

Traffic. Billing information for any voice lines should be helpful in providing usage patterns for voice links. Most PBXs collect this information.

Any simple network management protocol (SNMP)-managed devices in the voice network may deliver usage information to the management tool.

This information may be expressed in several forms, but for the simulation a probability distribution function is needed to drive the traffic sources in the models. Because voice traffic exhibits a high degree of randomness, it is frequently viewed as a Poisson distribution. Depending on the modeling tool, there should be several distributions for voice traffic:

- the distribution of addresses (i.e., who is calling whom)
- the distribution of the length of the message (i.e., call holding time)
- the distribution of the number of calls (i.e., call attempts)
- the desired quality of service

Modeling Recommendations. Typically, voice traffic is represented as a Poisson distribution. This method specifies the number of events that will occur over time. The exponential distribution is what is actually used to produce a Poisson distribution of traffic arrival because the exponential distribution determines the amount of time to the next event (i.e., the call to be generated).

Another method to simulate voice traffic is to use an interrupted Markov process to generate spurts of digitized speech packets. If the traffic is to be carried over a CBR circuit, it is entirely valid to use a constant source at the defined rate. Typical data rates for voice lines are as follows (because faxes are also essentially voice traffic, they are included in the table):

Traffic on Voice Circuit	***Rate***
Voice	64K bps
Group 3 fax	14.4K bps
Group 4 fax	64K bps

In ATM networks, voice traffic is usually represented as CBR traffic. Because it is CBR, it is tempting to subtract CBR traffic from the model in an attempt to speed it up; however, it is not recommended unless the manager is extremely familiar with the way the switch handles buffer allocation. Even though the CBR traffic is always there, some jitter may occur in the competition for buffer space, which will affect queue buildup.

Video Network Information. To collect the topology and traffic information for the video portion of the network, here are some recommendations as to how it might be represented in the model.

Topology. All video segments should be described, in the same fashion as the voice network, noting areas of overlap.

Traffic. Again, billing information may be of great assistance when determining the usage pattern for video links. Many video codecs (e.g., PictureTel's) use two switched 56K-bps lines. It can be safely assumed that current usage for such a link is at least 2x56K-bps multiplied by the holding time of the call in this case. The manufacturer of the video codec equipment should be able to provide a more accurate idea of the traffic the device generates. The holding times of the video sessions may be derived from billing or from equipment checkout logs.

For each video source there should be the following information:

- maximum bit rate generated by the codec
- holding time for each session
- number of sessions for the sample period
- address distribution of sessions (i.e., who is calling whom)

The character of these sessions may vary a great deal; if so, developing several scenarios for the model may be appropriate. One scenario might be supporting a worst-case video session during the worst-case data session and worst-case voice; another scenario could show a more normal video session.

With improved network services, users may alter their habits and begin using the service at worst-case times. Because the network manager is usually interested in the worst-case behavior, maximum usage should be assumed for the video portion of the model.

If a model is a discrete event simulation of an ATM network, long periods will not be simulated. ATM simulations take a large amount of computer time. Therefore it is generally safe to simply presume that all of the video sessions are already established and in use for the length of the worst-case scenario. This is simpler than attempting to capture any periodic qualities they may have.

Modeling Recommendations. The data rate for a video codec can vary. Some produce constant bit rate traffic while others generate traffic with a bit rate that changes as the amount of data that can be compressed changes. Because all of the data is important, these can either be represented as a worst-case constant video source at the bit rate desired, or as "bursts of bursts."

Generally, Poisson distributions are extremely poor at representing this sort of traffic. If an ATM or frame relay network is to be the carrier for this data, one important consideration is what sort of QoS is anticipated. If this traffic will be sent over a CBR connection, then it would be appropriate to model it as a constant source. Some typical video data rates are shown in Exhibit 43.2.

Exhibit 43-2. Data rates for several types of video traffic.

Video Type	Uncompressed	Compressed
Real tIme, 30 fps (128 × 240 × 9)	8,294	2,000
Studio 30 fps (640 × 480 × 24)	221,184	4,000
MPEG	N/A	>1.86M b/s
H.261	N/A	> = 128K b/s

Data Network Information. To collect topology and traffic information for the data portion of the network, here are recommendations as to how it might be represented in the model.

Topology. Describe the network's topology, including virtual (e.g., source and destination of traffic) as well as physical aspects.

Traffic. The hardest type of traffic for which to get an accurate probability distribution is often data. To get a profile of the distribution of data, several techniques can be used. For the subnets that make up the WAN, one technique is to place a sniffer or similar device at each of the subnetwork WAN interface points and to measure the actual loading.

This method yields the most accurate and fastest-running simulation, because the actual LANs are not modeled, only the load they offer to the WAN. Many network managers only have equipment suitable for monitoring the LAN side of their interconnecting device. In this case, the capacity of the current link can be used with estimates of current user activities to derive profiles of the data traffic.

In addition, many devices have SNMP information bases that provide some useful information. Network management tools may also be useful. If frame relay or ATM is currently being used, information may be available from the provider regarding traffic patterns. For each interface (i.e., traffic source) to the WAN, the following traffic details are necessary:

- the data rate of sources
- the size of the packets
- the quality of service desired (as for voice channels)
- the distribution of the addresses

Modeling Recommendations. The best way to model data sources is to think of them as bursts with some space between them. An example would be to have fixed burst sizes with some distribution of random delay between.

However, most tools (including COMNET III, BONeS, and Opnet) can drive the simulation to some extent from sniffer-collected traces. If there is a large amount of query-and-response type data, it may be better to define

Exhibit 43-3. Representative packet sizes for use in simulations.

Network Type	Most Common Packet Size	Next Most Common Packet Size
Ethernet	64 bytes (50% of traffic)	1,500 bytes (50% of traffic)
Token Ring	64 bytes (75% of traffic)	
ATM	9,180 bytes (default for ATM)	
FDDI	4,352 bytes (default for FDDI)	

the applications' query response in the model, with the guidance of the sniffer-acquired data.

On ATM networks, data is generally sent as ABR or perhaps even as uncommitted bit rate (UBR). Ethernet packets are generally either the largest or the smallest that the Ethernet can handle; Token Ring packets are almost always small. Some packet size data for use in simulations is shown in Exhibit 43.3.

PHASE II: BASELINE MODEL POPULATION AND VALIDATION

After capturing the data required to construct the baseline, model design can begin. The goal now is to transform the collected information into a valid baseline.

Simple small steps, gradually increasing the complexity as each step has been validated, should be used. This way, data collection efforts can be validated, establishing confidence in the tool and modeling methodology.

Guidelines for Building Models

Creating Subnets. Preliminary steps should be modeling a portion of the network that the network manager understands very well — for instance, the simple case of determining the loading of the video portion of the Net. Although it may not prove to be a very interesting model, it will give confidence in the use of the tool. Later, smaller separate models for portions of each type of traffic in the network (e.g., voice, video, and data) can be built. By keeping the problems simple and only gradually adding complexity, the overall quality of the work is greatly improved.

The following paragraphs provide more detailed, step-by-step guidelines. Vendors of network design tools provide considerable support for their products that can be of additional assistance.

Step 1: Tool Use and Data Collection Validation. During this step, the goal is to learn how to use the tool and to validate the data collection techniques.

It is likely that some data requirements of the problem or model will have been overlooked. A small representative of the network should be modeled, preferably using portions that are fairly well understood.

By validating knowledge of the tool and the manager's ability to capture raw data in a model, much work will be saved. Putting off gaining this intuitive understanding of how the modeling tool represents the network's components only postpones difficulties. Later, when the model has more data in it and more processes running, it will not be possible to see what happens with the very simple cases. The idea is to conduct simple experiments until the tool is completely understood.

Some suggestions for these early experiments are video only (this is probably the simplest part of the problem), a simple WAN link, or a very simple voice link. Most networks, when lightly loaded, are of little interest so it is a good idea to experiment by reducing the bandwidth at a random bottleneck point.

Validating Subnets

Once a subnet is built, it must be validated. The process of validation requires running the model and comparing the results against data collected for the real subnet. Validation should be in three steps:

- *Topology.* The topology of the subnetwork or network should be checked. For example, are all of the links connected to the correct devices? Are they of the correct type and bandwidth? Is each traffic source connected correctly?
- *Routing.* Are the router tables set up correctly? For the path of a packet traveling through the network, does it go where it should (and on the reverse path as well)?
- *Load.* Run the simulation with only one source turned on at a time. Is the correct amount of traffic sent to each destination? Is the delay close to what the network really experiences?

Step 2: Beginning to Validate the Data Network. At this point subnets (which later will be collected into a larger model) should be created. Start with subnets that can operate independently. Rather than spending time encoding large amounts of routing or other information for larger future networks, it is better to model a small subnet correctly. In doing so, tools and models will be built up for future use in other areas.

For instance, using COMNET III, application profiles (including message response pairs) may be built, which will be very useful when building up other models. At first, entries should have depth as opposed to breadth. Once a working subnet is created, it may be copied in its entirety. In brief, the steps are to build a subnet, validate it, and repeat.

All of the subnets that may possibly be built should be built and validated before moving on. Validation can have multiple meanings. A simple validation might be a data check: Was the correct information entered for the model? Next might be a sanity check: Does it make sense? A logical check might be next: Does it behave as expected? Do all sources reach their destinations? Are source message pairs coded correctly?

Only proceed when all of this work has been done. The more time spent testing the pieces, the easier the job will be when putting them together.

Integrating and Validating Subnets. When the subnets have been validated, they must be integrated and validated in a stepwise fashion. An example might be considering replacing a collapsed Ethernet backbone with a distributed ATM backbone that also carries video and contiguous portions of the voice network. First the data is integrated, then the video, and finally the voice.

Once again, there are three tips to successfully building an accurate simulation of a network:

1. *Never proceed if there are any doubts about a result.* Stop immediately and investigate the problem until it is resolved. If the problem is not resolved, numerous bugs will creep in. This practice cannot be overemphasized.
2. *Be organized.* There is a lot of data to compile. The spreadsheet can be the biggest asset.
3. *Understand the network.* To this end, a good network analyzer is invaluable. If the expense of buying one cannot be justified, many leasing agencies and sometimes even the manufacturers rent them out.

The key to successful integration and validation is taking small steps, testing the results each time, and never proceeding if there are any doubts about a result. It is best to work up in complexity by beginning with the portions of the network that are best understood.

PHASE III: ALTERATION OF BASELINE TO ACQUIRE DATA

When a baseline with which to compare is completed and validated, alterations can be introduced. The alterations should be introduced with the same care that the baseline was constructed.

An example might be to compare the performance of both 100Base-T Ethernet and ATM backbone links. (Both should be able to handle the offered data load easily, but video traffic may be too much for the Ethernet backbone.) Loss and delay experienced by each traffic source should be measured, as well as the throughput on the backbone, to determine if these meet the required quality of service.

There are limits as to how much accuracy can be achieved by modeling. Efforts should focus on those areas that yield the highest return. Focusing on describing the offered load accurately will yield a more accurate simulation.

The goal is to have a model that yields valid information to guide decisions when doing experimentation on the network.

SUMMARY

This chapter has offered a brief description of a methodology that may be used to simulate a wide variety of networks using network design tools such as COMNET III. Because of the complexity of networks and the tools that represent them, simple examples are often insufficient to impart an understanding of the process or methodology that a network designer needs to use to achieve accurate simulation results.

Many steps described in this chapter are a necessary part of network analysis and design, yet they are often not found in the manuals of network design tools themselves. By focusing on the methodology rather than a particular tool's implementation, this chapter has attempted to provide the structure and approach that are missing from the users' manuals and to give readers insight into approaches they can use to solve some of the particular modeling problems that are expected in modeling multimedia networks.

There are many issues still untouched. Examples include parameter sensitivity analysis, confidence intervals, and length of time to run a simulation. Unfortunately, these issues are beyond the scope of this chapter. However, an excellent text on simulation that covers these issues is Dr. Raj Jam's book *The Art of Computer Systems Performance Analysis* (New York: Wiley, 1991).

Chapter 44

Determining Remote Bridge and Router Delays

Gilbert Held

THIS CHAPTER EXAMINES THE USE OF QUEUING THEORY to determine the delays associated with remote bridges and routers. In addition, it investigates the effects of modifying the operating rate of the WAN links — in particular, the effects of various communications circuit operating rates on equipment delays. There is a point beyond which increasing the operating rate of a communications circuit has an insignificant effect on equipment and network performance.

WAITING LINE ANALYSIS

Queuing theory, the formal term for waiting line analysis, can be traced to the work of A.K. Erlang, a Danish mathematician. His pioneering work spanned several areas of mathematics, including the dimensioning or sizing of trunk lines to accommodate long-distance calls between telephone company exchanges. This chapter bypasses Erlang's sizing work to concentrate on the analysis of waiting lines.

BASIC COMPONENTS

Exhibit 44-1 illustrates the basic components of a simple waiting line system. The input process can be considered the arrival of people, objects, or frames of data. The service facility performs some predefined operation on arrivals, such as collecting tolls from passengers in cars arriving at a toll booth or the conversion of a LAN data frame into a synchronous data link connection (SDLC) frame by a bridge or router for transmission over a WAN transmission facility. If the arrival rate temporarily exceeds the service rate of the service facility, a waiting line known as a queue forms. If a waiting line never exists, the server is idle or there is too much service capacity.

0-8493-0859-3/00/$0.00+$.50
© 2000 by CRC Press LLC

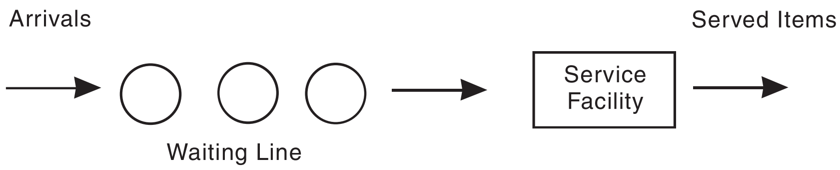

Exhibit 44-1. Basic components of a simple waiting line system.

The waiting line system illustrated in Exhibit 44-1 is more formally known as a single-channel, single-phase waiting line system — single channel because there is one waiting line, and single phase because the process performed by the service facility occurs at one location. One toll booth on a highway or a single-port bridge connected to a LAN are two examples of single-channel, single-phase waiting line systems.

Exhibit 44-2 illustrates three additional types of waiting line systems. On multichannel systems, arrivals are serviced by more than one service facility, which results in multiple paths or channels to those service facilities. On multiphase systems, arriving entities are processed by multiple service facilities.

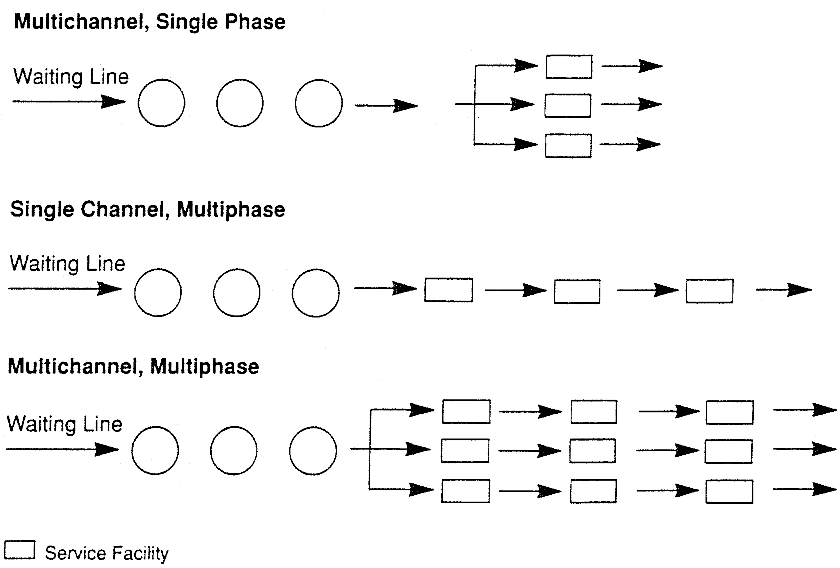

Exhibit 44-2. Other types of waiting line systems.

One example of a multiphase service facility is a toll road in which drivers of automobiles are serviced by several series of toll booths; for example, a turnpike that has toll plazas every few miles. Another example of a multiphase system is the routing of data through a series of bridges and routers. The computations associated with multiphase systems can become quite complex, and because most networks can be analyzed on a point-to-point basis as a single-phase system, this chapter restricts its examination of queuing models to single-phase systems.

ASSUMPTIONS

Queuing theory is similar to other types of theory in that it is based on a series of assumptions. Those assumptions primarily have to do with the distribution of arriving entities and the time required to service each arrival.

Both the distribution of arrivals and the time to service them are usually represented as random variables. The most common distribution used to represent arrivals is the Poisson distribution:

$$P(n) = \frac{\lambda^n e^{-\lambda}}{n!}$$

where: $P(n)$ = Probability of n arrivals

λ = Mean arrival time

e = 2.71828

$n!$ = n factorial

= $n(n-1)(n-2)\ldots 1$

One of the more interesting features of the Poisson process concerns the relationship between the arrival rate and the time between arrivals. If the number of arrivals per unit time is Poisson distributed with a mean of λ, the time between arrivals is a negative exponential probability distribution with a mean of $1/\lambda$. For example, if the mean arrival rate per 10-minute period is 3, the mean time between arrivals is 10/3, or 3.3 minutes.

NETWORK APPLICATIONS

Through the use of queuing theory or waiting time analysis, it is possible to examine the effect of different WAN circuit operating rates on the ability of remote bridges and routers to transfer data between LANs. This answers questions about the average delay associated with the use of a remote bridge or router, the effect on those delays of increasing the operating rate of the WAN circuit, and when an increase in the WAN circuit's operating rate results in an insignificant improvement in bridge or router performance.

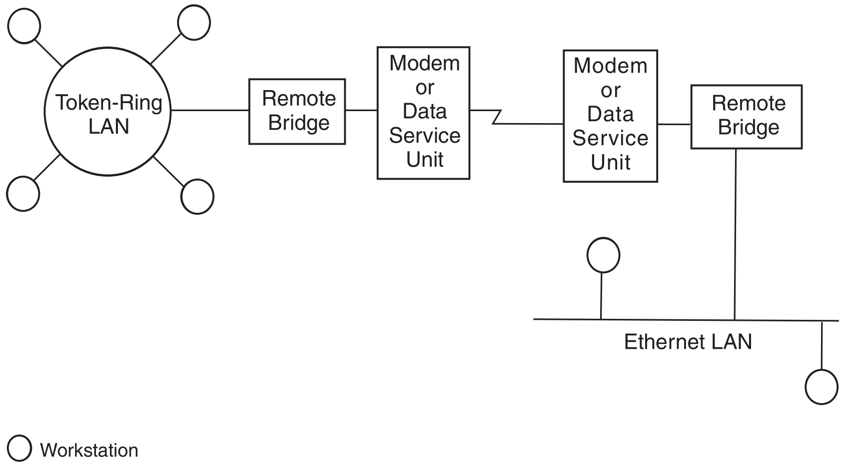

Exhibit 44-3. Internet consisting of two LANs connected through remote bridges.

The following example illustrates the application and value of queuing theory to network problems.

Two geographically distant LANs are to be connected through a pair of remote bridges as illustrated in Exhibit 44-3. Today, most remote bridges support both RS-232 and V.35 interfaces, permitting the bridge to be connected to modems, or to higher-speed data service units (DSUs) for digital transmission. This means that there are a wide variety of circuits that can be used to connect a pair of remote bridges, both analog and digital, and at different operating rates.

This example assumes that, on the basis of prior knowledge obtained from monitoring the transmission between locally interconnected LANs, the data communications manager has determined that approximately 10,000 frames per day can be expected to flow from one network to the other. The average length of a frame is 1250 bytes.

QUEUING THEORY CALCULATIONS

The 10,000-frame-per-day rate must be converted into an arrival rate. In this case, the next assumption is that each network is active only eight hours per day and both networks are in the same time zone. A transaction rate of 10,000 frames per eight-hour day is equivalent to an average arrival rate of 10,000/(8 * 60 * 60), or 0.347222 frames per second. In queuing theory, this average arrival rate (AR) is the average rate at which frames arrive at the service facility for forwarding across the WAN communications circuit.

Monitoring has determined that the average frame length is 1250 bytes. Because a LAN frame must be converted into a WAN frame or packet for transmission over a WAN transmission facility, whatever header and trailer information is required by the WAN protocol is added to the frame or packet. Thus, the actual length of the WAN frame or packet exceeds the length of the LAN frame. This example assumes that 25 bytes are added to each LAN frame, resulting in the average transmission of 1275 bytes per frame.

To compute an expected service time requires knowing the operating rate. If the WAN communications circuit illustrated in Exhibit 44-3 operates at 9600 bps, the time required to transmit one 1275-byte frame or packet becomes 1275 * 8/9600, or 1.0625 seconds. This time is more formally known as the expected service time and represents the time required to transmit a frame whose average length is 1250 bytes on the LAN and 1275 bytes when converted for transmission over the WAN transmission facility. Given that the expected service time is 1.0625 seconds, the mean service rate (MSR) can be computed easily. That rate is the rate at which frames entering the bridge destined for the other LAN are serviced and is 1/1.0625, or 0.9411765 frames per second.

Two key queuing theory variables — the arrival rate and the mean service rate — have now been computed. The service rate computation was dependent on the initial selection of a WAN circuit operating at 9600 bps.

Exhibit 44-4 illustrates the results of the initial set of computations for one portion of this Internet. In this case, 10,000 frames per eight-hour day flow in each direction, so only one-half of the Internet has to be analyzed. If this had not been true, each half could be analyzed and a circuit operating rate selected based on the highest arrival rate's effect on LAN performance.

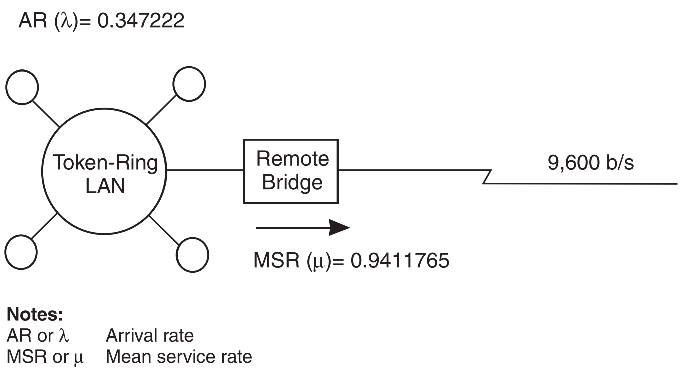

Exhibit 44-4. Initial computational results.

In Exhibit 44-4, the queuing theory designators are indicated in parentheses. Thus, in a queuing theory book, the average arrival rate of 0.347222 transactions or frames per second would be indicated by the expression $\lambda = 0.347222$.

Although the mean service rate exceeds the average arrival rate, on occasion the arrival rate results in a burst of data that exceeds the capacity of the bridge. When this occurs, queues are created as the bridge accepts frames and places those frames into buffers or temporary storage areas. Through the use of queuing theory, the expected time for frames to flow through the bridge can be examined and the circuit operating rate adjusted accordingly.

A communications network that uses remote bridges or routers corresponds to a single-channel, single-phase queuing model. The utilization of the service facility (*P*) is obtained by dividing the average arrival rate by the mean service rate. That is,

$$\begin{aligned} P &= AR/MSR \\ &= 0.347222/0.9411765 \\ &= 0.3689 \end{aligned}$$

Use of a circuit operating at 9600 bps results in an average utilization level of approximately 37 percent. In queuing theory texts, the preceding equation is replaced by $P = \lambda/\mu$, where λ is the average arrival rate and μ is the mean service rate.

The utilization level of the service facility (remote bridge) is AR/MSR, so the probability that there are no frames in the bridge, *Po*, becomes:

$$Po = 1 - AR/MSR = 1 - \lambda/\mu$$

For the remote bridge connected to a 9600-bps circuit:

$$Po = 1 - 0.37 = 0.63$$

Thus, 63 percent of the time there will be no frames in the bridge's buffers awaiting transmission to the distant network.

For a single-channel, single-phase system, the mean number of units expected to be in the system is equivalent to the average arrival rate divided by the difference between the mean service rate and the arrival rate. In queuing theory, the mean or expected number of units in a system is designated by the letter *L*. Thus,

$$L = AR/(MSR - AR) = \lambda/(\mu - \lambda)$$

Returning to the network example, the mean or expected number of frames that will be in the system, including frames residing in the bridge's buffer area or flowing down the WAN transmission facility, can be determined:

$$L = 0.347/(0.941 - 0.347) = 0.585$$

Thus, on the average, approximately 6/10 of a frame resides in the bridge's buffer and on the transmission line.

By multiplying the utilization of the service facility by the expected number of units in a system, the mean number of units in the queue or, in common English, the queue length, is obtained. The queue length is denoted by *Lq* and thus becomes:

$$\begin{aligned} Lq = PL &= \left(\frac{AR}{MSR}\right)\left(\frac{AR}{MSR - AR}\right) \\ &= \left(\frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu - \lambda}\right) \\ &= \frac{\lambda^2}{\mu(\mu - \lambda)} \end{aligned}$$

Again returning to the network example

$$\begin{aligned} Lq &= \frac{(0.347)^2}{0.941(0.941 - 0.347)} \\ &= 0.216 \end{aligned}$$

On the average, 0.216 frames are queued in the bridge for transmission when the operating rate of the WAN is 9600 bps and 10,000 frames per eight-hour day require remote bridging. There were 0.585 frames in the system, so the difference, 0.585 – 0.216, or 0.369 frames, is flowing on the transmission line at any particular time.

TIME COMPUTATIONS

In addition to computing information concerning the expected number of frames in queues and in the system, queuing theory furnishes tools to determine the mean time in the system and the mean waiting time. In queuing theory, the mean waiting time is designated as the variable *W*, whereas the mean waiting time in the queue is designated as the variabe *Wq*.

The mean time in the system, *W*, is

$$W = 1/(MSR - AR = 1/(4 - \lambda)$$

For the bridged network example, the mean time a frame can be expected to reside in the system can be computed:

$$W = 1/(0.941 - 0.347) = 1.68 \text{ seconds}$$

By itself, this tells us that an average response time of approximately 1.7 seconds can be expected for frames that must be bridged if the WAN transmission facility operates at 9600 bps. Whether this is good or bad depends on the acceptability of the 1.7-second delay.

The last queuing item is the waiting time associated with a frame being queued. That time, *Wq*, is equivalent to the waiting time system multiplied by the use of the service facility. That is,

$$\begin{aligned} Wq = PW &= \left(\frac{AR}{MSR}\right)\left(\frac{1}{MSR - AR}\right) \\ &= \frac{AR}{MSR(MSR - AR)} \end{aligned}$$

In terms of queuing theory, *Wq* is

$$Wq = \frac{\lambda}{\mu(\mu - \lambda)}$$

Returning to the bridged network example, the waiting time for a queued frame is

$$Wq = \frac{0.347}{0.941(0.941 - 0.347)} = 0.621 \text{ seconds}$$

Previously, it had been determined that a frame can be expected to reside for 1.68 seconds in the bridged system, including queue waiting time and transmission time. The computed queue waiting time is 0.621 seconds. The difference between the two, 1.68 – 0.621, or approximately 1.06 seconds, is the time required to transmit a frame over the 9600-bps WAN transmission facility.

DETERMINING AN OPTIMUM LINE OPERATING RATE

Each of the variables could be recomputed for different line operating rates, but it is simple to write a small computer program to perform this operation. Exhibit 44-5 lists the statements of a Beginner's All-Purpose Symbolic Instruction Code language program named QUEUE.BAS. The execution of this program is shown in Exhibit 44-6, which displays the values

Exhibit 44-5. QUEUE.BAS program listing.

```
REM PROGRAM QUEUE.BAS
CLS
REM AR = arrival rate
REM MSR = mean service rate
REM L = mean (expected) number of frames in system
REM Lq = mean number of frames in queue
REM W = mean time (s) in system
REM Wq = mean waiting time (s)
REM EST = expected service time
transactions = 10000          'transactions per day
avgframe = 1275               'average frame size
hrsperday = 8
AR = transactions/(8*60*60)
DATA 4800,9600,19200,56000,64000,128000,256000,384000,768000,1536000
FOR i = 1 TO 10
READ linespeed(i)
est(i) = avgframe * 8/linespeed(i)
msr(i) = 1/est(i)
utilization(i) = AR/msr(i)
prob0(i) = 1-(AR/msr(i))
L(i) = AR/(msr(i)-AR)
Lq(i) = AR^2/(msr(i)*(msr(i)-AR))
W(i) = 1/(msr(i)-AR)
Wq(i) = AR/(MSR(i)*(msr(i)-AR))
NEXT i
PRINT "Line Speed    EST    MSR    Po     p     L     Lq     W    Wq"
FOR i = 1 TO 10
PRINT USING " #######  #.#### ###.## "; linespeed(i); est(i); msr(i);
PRINT USING " #.#####   .#####"; prob0(i); utilization(i);
PRINT USING "  #.#####  #.##### #.##### #.#####"; L(i)l Lq(i); W(i); Wq(i)
NEXT i
PRINT
PRINT "where:"
PRINT
PRINT "EST = expected service time  MSR = mean service rate"
PRINT "Po = probability of zero frames in the system p = utilization"
PRINT "L = mean number of frames in system  Lq = mean number in queue"
PRINT "W = mean waiting time in system  Wq = mean waiting time in queue"
```

for eight queuing theory parameters for line speeds ranging from 4,800 bps to 1.536 Mbps. The 536-Mbps line speed is the effective operating rate of a T1 circuit because 8000 bps of the 1.544 Mbps operating rate of that circuit is used for framing and is not available for the actual transmission of data.

The utilization level *P* and the mean waiting time in the queue, *W(q),* in Exhibit 44-6, should be examined. At 4800 bps, the utilization level is approximately 74 percent, and the waiting time in the queue is almost 6 seconds. Clearly, linking the two LANs with remote bridges operating at 4800 bps provides an unacceptable waiting time due to the high utilization level of the remote bridge.

Exhibit 44-6. QUEUE.BAS execution results.

Line Speed	EST	MSR	Po	P	L
4800	2.1250	0.47	0.26215	.73785	2.81457
9600	1.0625	0.94	0.63108	.36892	0.58459
19200	0.5313	1.88	0.81554	.18446	0.22618
56000	0.1821	5.49	0.93676	.06324	0.06751
64000	0.1594	6.27	0.94466	.05534	0.05858
128000	0.0797	12.55	0.97233	.02767	0.02846
256000	0.0398	25.10	0.98617	.01383	0.01403
384000	0.0266	37.65	0.99078	.00922	0.00931
768000	0.0133	75.29	0.99539	.00461	0.00463
1536000	0.0066	150.59	0.00769	.00231	0.00231

Lq	W	Wq
2.07672	8.10596	5.98096
0.21567	1.68363	0.62113
0.04172	0.65141	0.12016
0.00427	0.19444	0.01230
0.00324	0.16871	0.00934
0.00079	0.08196	0.00227
0.00019	0.04040	0.00056
0.00009	0.02681	0.00025
0.00002	0.01334	0.00006
0.00001	0.00666	0.00002

Note:

EST = Expected service time
MSR = Mean service rate
PO = Probability of zero frames in the system
P = Utilization
L = Mean number of frames in system
Lq = Mean number in queue
W = Mean waiting time in system
Wq = Mean waiting time in queue

Press any key to continue

As the line speed is increased, each bridge can service frames at a higher processing rate. The average arrival rate is fixed, so increasing the line operating rate should lower the utilization level of the bridge as well as the time a frame resides in the queue. This expectation is verified by the results of the execution of QUEUE.BAS (Exhibit 44-6). Note that, as expected, both the utilization level and mean waiting time in the queue decrease as the line speed increases.

Queuing theory can be used to determine the operating rate of transmission lines for linking remote bridges and routers. In actuality, such use will not produce a magic number. Instead, queuing theory returns a range of values that can be used to make a logical decision. As shown in Exhibit 44-6, a line operating rate of 4800 bps is clearly unacceptable. But operating rates of 9600 bps, 19,200 bps, and 56,000 bps must be evaluated.

Exhibit 44-7 graphs the probability of a bridge containing zero frames (e.g., having empty buffers) and the utilization level of the bridge based on the 10 line operating rates. At line rates above 56 Kbps or 64 Kbps, the reduction in the utilization level of the bridge and increase in the probability of zero frames occurring in the system are insignificant. Thus, the line operating rate could safely be restricted to the maximum of 56 Kbps or 64 bps. Exhibit 44-6 shows that increasing the line rate from 19,200 bps to 64 Kbps only marginally decreases the waiting time in the queue from 0.12 seconds to 0.009 seconds. If the application is not likely to grow, a 19,200-bps analog leased line would be a good choice as the cost of that line is usually less than the cost of a digital line. If greater use of remote bridges is expected, the more expensive 56-Kbps or 64-Kbps digital l line is an appropriate choice.

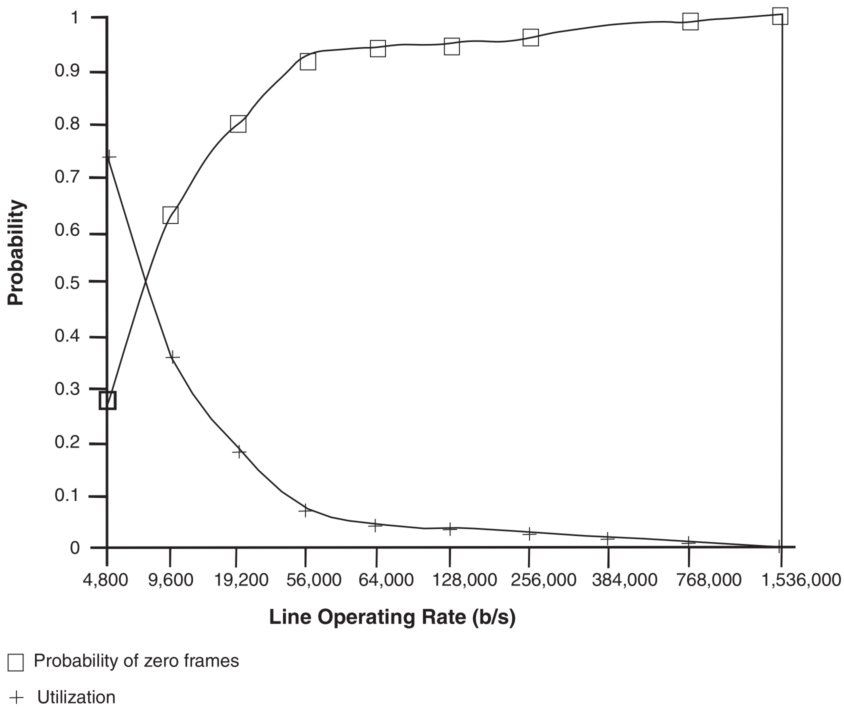

Exhibit 44-7. Bridge performance at different line operating rates.

CONCLUSION

To correctly apply queuing theory to the interconnection of LANs, the number of transactions and the average frame size of each transaction that will flow to the other network must be determined. Once this is accomplished, an allowance must be made for the increase in the average frame size when WAN protocol header and trailer information is added. The average arrival rate of frames as well as the mean service rate of the bridge or router can then be calculated, as well as the additional queuing parameters discussed in this chapter. By recalculating those parameters for different transmission line operating rates, the network manager can select an appropriate operating rate.

Note

For specific information concerning the application of A. K. Erlang's work to the sizing of parts of multiplexers and concentrators, readers are referred to *Network Design Techniques* (Gilbert Held, Wiley, New York).

Chapter 45
Network Baselining as a Planning Tool

Gilbert Held

BASELINING PROVIDES A MECHANISM FOR DETERMINING THE LEVEL OF UTILIZATION OF A NETWORK, including its computational and transmission facilities. As such, it plays a central role in a network manager's capacity planning effort because the baseline shows whether or not there is sufficient capacity available, as well as providing a foundation for future network measurements that can be compared to the baseline to indicate the direction of network utilization. Thus, the network baselining effort represents the first major step in the capacity planning effort.

In addition, baselining enables network managers and administrators to identify and respond to network capacity requirements before they become an issue, in effect providing a mechanism to head off network-related problems.

BASELINING TOOLS AND TECHNIQUES

There are a variety of network baseline tools and techniques that can be used to facilitte an organization's capacity planning effort. The actual techniques employed are commonly based on the type of tool used. This chapter focuses on a number of commercially available network baselining tools, and discusses appropriate techniques concerning their use.

SimpleView

SimpleView is an easy to use and relatively inexpensive Simple Network Management Protocol (SNMP) management platform from Triticom, Inc., of Eden Prairie, Minnesota. Through the use of SimpleView, users can retrieve statistical information maintained by Remote Monitoring (RMON) network probes. SimpleView supports a Management Information Base (MIB) walk capability, shown in the MIB Walk window, that lets a user click on an MIB group to select the group starting point, or double-click on the group to explode its elements, enabling a specific element from the group to be selected for retrieval.

0-8493-0859-3/00/$0.00+$.50
© 2000 by CRC Press LLC

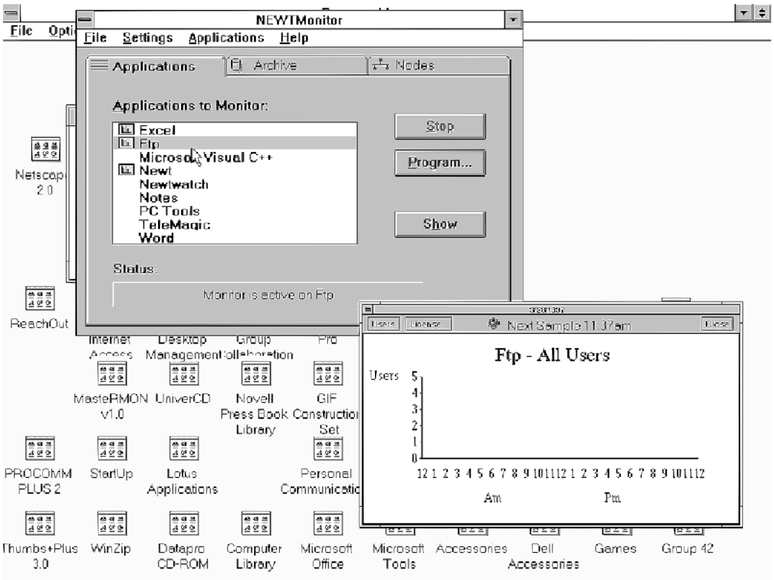

Exhibit 45-1. The NetManage NEWTMonitor program provides the capability to examine the activity on a network based upon certain types of predifined applications.

NEWT

NetManage of Cupertino, California, well known for its Chameleon suite of Internet applications, also markets a program called NEWT that can be used to monitor the use of desktop applications as well as to provide statistics on network activity associated with individual users. Exhibit 45-1 illustrates the use of NEWTMonitor on the author's computer to monitor the number of simultaneous FTP sessions occurring over a period of time. Doing so can be extremely important, especially when used in conjunction with normal RMON traffic statistics that do not look beyond the data link layer. NEWTMonitor enables the use of specific types of TCP/IP applications. In comparison, if the network probes and network management system support RMONv2, or can be upgraded to this new version of RMON, it can be used to obtain a distribution of traffic through the application layer.

Exhibit 45-2 illustrates the use of NEWTGraph to display different TCP/IP statistics by node. In the example shown in Exhibit 45-2, the author displayed Interface Errors for his node.

EtherVision

When checking the activity associated with an individual network, users can choose from a variety of network monitoring programs. One such program is EtherVision, also from Triticom, Inc., of Eden Prairie, Minnesota.

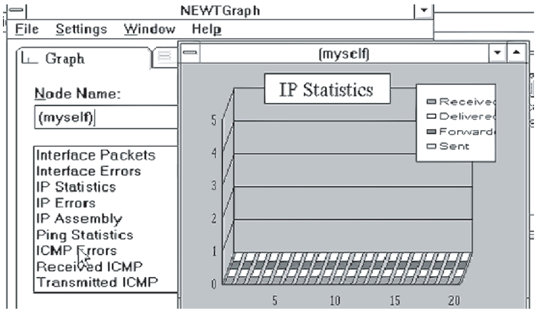

Exhibit 45-2. Through the use of the NetManage NEWTGraph program, graphs of different types of TCP/IP statistical information can be displayed.

Exhibit 45-3 illustrates the statistics summary display based on the monitoring of frames using their source address for constructing a statistical baseline. EtherVision supports monitoring by either Source or Destination address, enabling users to build two baselines. In examining Exhibit 45-3, note that the statistics summary presented indicates the frame count over the monitored period of time, current network utilization in the form of a horizontal bar graph, and a summary of “average,” “now” or current, and “peak” utilization displayed as a percentage, as well as the time peak utilization occurred. The latter can be extremely handy as it allows a user to run the program on a workstation connected to an Ethernet LAN and return at the end of the day to determine the peak percentage of network use as well as when the peak occurred.

Although not shown in Exhibit 45-3, an EtherVision user can also set the program to generate a report that will log each period of activity over a certain percentage of network activity. Then, using the logged report, a network manager or LAN administrator can easily determine the distribution of network utilization throughout the monitoring period.

In the upper right corner of Exhibit 45-3, note that EtherVision maintains a distribution of frames transmitted on the network based on their size or length, falling into five predefined intervals. By examining the distribution of frames based on their length, users can determine the general type of traffic flowing on a network. This is possible because interactive query-response applications are generally transported in relatively short frames. In comparison, file transfers, such as Web browser pages containing one or more images, commonly fill frames to their full length. In examining the distribution of frame sizes shown in Exhibit 45-3, note that there are relatively few full-sized Ethernet frames in comparison to the total number of frames

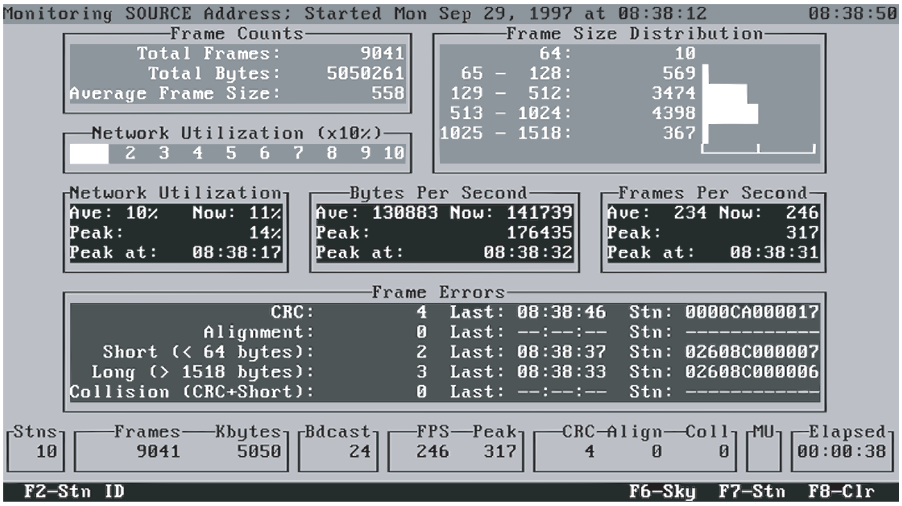

Exhibit 45-3. The Triticom EtherVision statistics summary display can be used to obtain information about network utilization and frame distribution.

encountered during the period of monitoring. This indicates a low level of file transfer and Web browser activity occurring on the monitored network.

Although EtherVision provides numeric information concerning network utilization, many users prefer to work with charts that note trends at a glance. To accommodate such users, EtherVision includes a number of built-in displays such as the one shown in Exhibit 45-4, which plots network utilization over a period of time. By examining a visual display, users can immediately note any potential capacity-related problems. In the example shown in Exhibit 45-4, the maximum level of network utilization is slightly above 46 percent. However, based on the monitored period, network traffic rose from 22 to 46 percent numerous times during the monitoring period. Because an Ethernet LAN gets congested at utilization levels above 50 percent due to its CSMA/CD access protocol, and the effect of the delay associated with the use of a random exponential back-off algorithm after a collision occurs, Exhibit 45-4 indicates a baseline of network utilization that justifies careful attention and a scheduled remonitoring effort to ensure traffic on the network does not turn into a bottleneck.

Foundation Manager

Foundation Manager, a product of Network General Corporation, is a sophisticated SNMP Network Management System (NMS) platform that

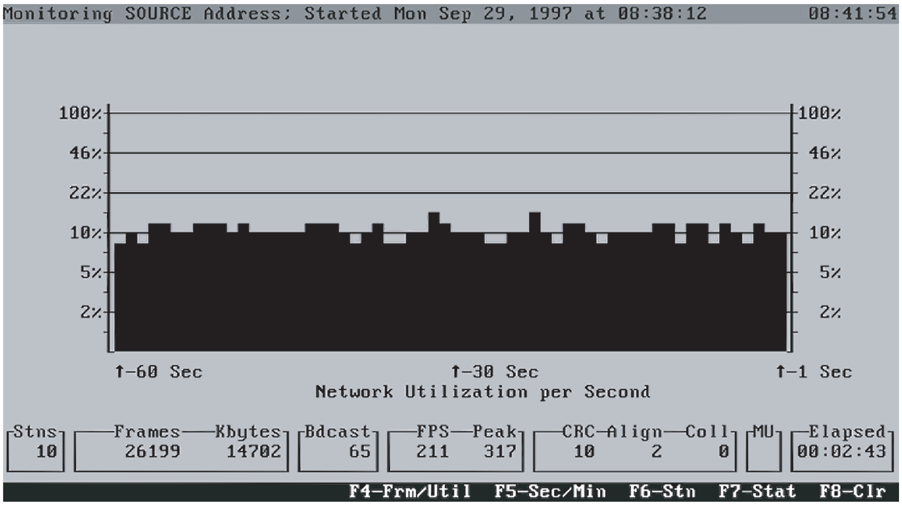

Exhibit 45-4. EtherVision supports the display of network utilization over a period of time, which facilitates observing the changing state of this important baseline metric.

operates on Intel-based computers using different versions of Microsoft's Windows operating system.

Foundation Manager was upgraded to support the emerging RMONv2 standard. When used to gather statistics from an RMON v2-compatible probe, it can provide a summary of statistics through the application layer, allowing it to replace the use of multiple products to obtain equivalent information.

Exhibit 45-5 illustrates the use of Foundation Manager to monitor a local Token Ring network. In the example, two buttons under the Local Token Ring Monitoring bar were pressed to initiate two displays of information from the Token Ring Statistics Group that an RMON probe on the locat network accumulates. The first button clicked on is the bar chart icon to the right of the icon with the upraised hand in the form of a stop sign. Clicking on the bar chart icon results in the display of the top row of eight bar charts that indicate the total number of different types of frames and level of network utilization.

For example, the second bar chart located on the left side of the top display indicates that network utilization is at three percent on a 100 percent basis. Other bar charts on the top row indicate the current number of logical link control (LLC) bytes and frames, multicast frames, broadcast frames, beaconing frames, purge events, and claim events. The second row

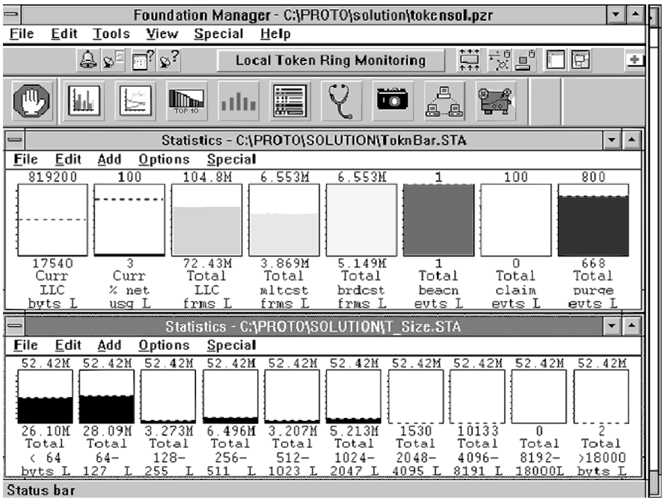

Exhibit 45-5. Using Network General's Foundation Manager to monitor the distribution of frames by length and type on a Token Ring network.

of bar charts resulted from clicking on the third icon to the right of the raised hand icon. This sequence of ten bar charts indicates the distribution of Token Ring frames in a manner similar to the method that EtherVision used to summarize Ethernet frame sizes. Foundation Manager follows the RMON standard and provides a more detailed breakdown of the distribution of Token Ring frames by their length.

Similarly, when using Foundation Manager to monitor Ethernet networks, the program retrieves RMON probe-kept frame distribution information that is more detailed than that kept by EtherVision. However, it is important to note that the retail price of EtherVision is under $500 and it can operate by itself. In comparison, the retail price of Foundation Manager is approximately $5000 and a single probe can cost approximately $1000, requiring an investment of an additional $5500 to obtain an enhanced level of frame size distributions as well as some additional features.

Two of the more interesting features of Foundation Manager are its QuickStats and discovery, and baselining capabilities. Exhibit 45-6 illustrates the use of the Foundation Manager Quick Stats feature to display a quick set of statistics for a remotely monitored network. In the example, statistics for an RMON probe connected to a network located in San Diego are displayed.

Foundation Manager is capable of displaying up to eight Quick Stats graphical reports at one time, with each report generated by clicking on an

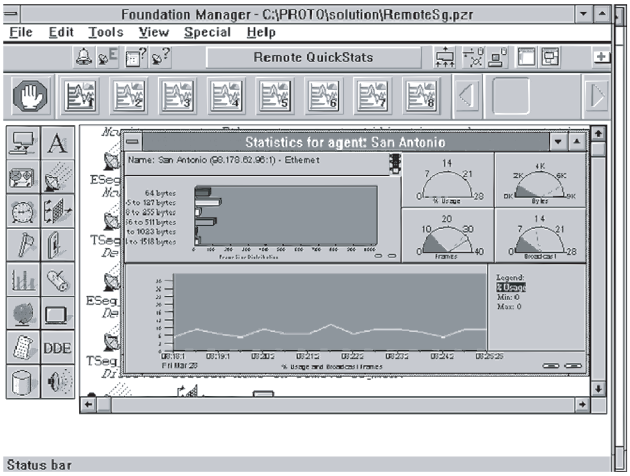

Exhibit 45-6. The Foundation Manager QuickStats display provides users with the ability to visually note important network baseline parameters both in real-time and over a period of time.

---

appropriate icon to the right of the icon with the raised hand in the form of a stop sign.

Each Quick Stats display presents summary information about a monitored network in a similar manner. In examining the statistics display for the network located in San Diego, the upper left display presents a distribution of frame length for the monitored LAN as a horizontal bar chart. The upper right portion of the display contains four gauges that provide a real-time view of network utilization, bytes transmitted, broadcast traffic, and frame rate. The lower half of the display shows a real-time plot over a period of predefined length for any two of the gauge values. Thus, the use of the Foundation Monitor Quick Stats display provides users with the ability to visually note important network baseline parameters both in real-time and over a period of time.

A second interesting feature built into Foundation Manager is its discovery and baselining capability. This capability is available for both local and remotely located networks being monitored, and provides the ability to gather pattern flow information that can be extremely valuable when attempting to determine if the cause of a high level of network utilization results from the activity of one or a few stations on the network.

Exhibit 45-7 illustrates the Foundation Manager local discovery and baselining display as a matrix map of network activity. The first portion of the title of the display, “Discovery,” results from the fact that the probe

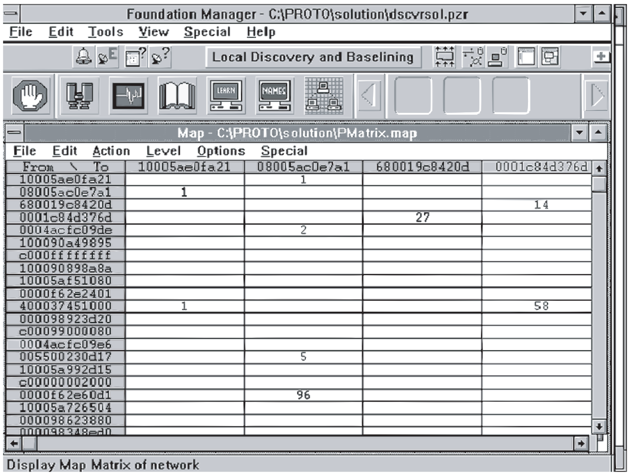

Exhibit 45-7. The local discovery and baselining capability of Foundation Manager enables the flow of data between stations to be identified.

examines each frame flowing on the monitored network and discovers its source and destination by examining the source and destination addresses contained in the frame. The second portion of the title of the display, "Baselining," results from the fact that Foundation Manager extracts information from a matrix table maintained by the probe that denotes the number of frames transmitted from one address to another. Thus, in examining Exhibit 45-7, such numerics as 1, 2, 27, 14, 58, 5, and 96 represent the number of frames transmitted from the address located in the row in the table to the address in the column portion of the table.

When baselining a network, matrix information should be considered as a mechanism to identify the cause of high network utilization. If Quick Stats or a similar display denotes a low level of network utilization, there is no need to use the matrix capability of Foundation Manager or a similar product to identify the actual flow of data between network stations. This is because even if the user can locate a station using too much bandwidth, a modification of the operation of the station will, at best, have a negligible effect upon improving network performance if the network already has a low level of utilization.

CONCLUSION

Baselining is an important process that enables the communications manager and LAN administrator to quantify the status and activities of a network. In doing so, it provides a base of information that enables network trends to be identified, which can allow changes to be made to a network infrastructure prior to network capacity becoming an issue. Thus, network managers and administrators should consider using the tools and techniques described in this chapter as a mechanism to baseline their network infrastructures.