A FEDERATION-ORIENTED CAPACITY MANAGEMENT METHODOLOGY FOR LAN ENVIRONMENTS

Daniel A. Menasce
George Mason University
Fairfax, VA 22030

DuaneDregits
Science Applications International Corporation
Reston, VA 22090

Robert Rossin
Science Applications International Corporation
Falls Church, VA 22041

Don Gantz
George Mason University
Fairfax, VA 22030

Large organizations can be composed of several organizational units having a high degree of autonomy with respect to Information Technology management. Each unit can have its own Local Area Network (managed by its own LAN administration staff These staffs can vary substantially in size and skill level, and follow locally defined standards and practices. The LANs are often interconnected through an enterprise-wide backbone, a configuration we call a federation. This paper describes a seven step capacity management methodology for federations that adapts to the idiosyncrasies of the organizational units. Examples and techniques are presented for each step. The authors developed the methodology for a large organization-a federation--of 26 institutes, more than 260 LANs and over 16,000 LAN-based computers.

1. INTRODUCTION

Imagine that your organization has an investment of many millions of dollars in client/server computing environments, and annual costs of millions of dollars for maintaining and expanding this environment. Imagine further that the overall capacity of this environment is unknown and capacity planning and procurement is done without a defined methodology, but in a relatively haphazard fashion, leaving the organization vulnerable to capacity shortfalls in key areas, and unwarranted expenditures for computing resources in others. Imagine further that there is no consistency in the size and skill levels of staff providing LAN support, including LAN capacity planning, and there is heterogeneity in the composition of local area network environments. Furthermore, mature performance measurement tools are missing in the capacity management tools marketplace.

This is not a purely imaginary scenario, but one that represents many, probably most, distributed processing environments.

Upon realizing that a federation of government research institutes mirrored this environment, the Senior Information Resources Management Official of this federation funded the development of the network capacity management methodology that is presented here.

2 METHODOLOGY OVERVIEW

Before one can discuss a capacity planning methodology, it is important to understand what ***adequate capacity*** means in the environment to be analyzed. As indicated in Fig. 1, three key elements define the boundaries, constraints, and meaning of adequate capacity:

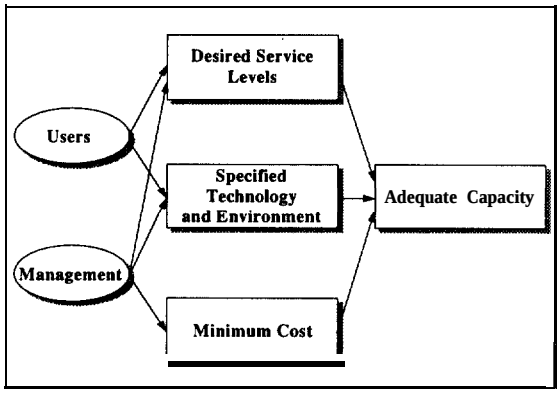

Figure 1. The Defining Elements of Adequate Capacity.

- ***Desired Service Levels:*** Adequate capacity is most strongly defined by acceptable or desirable values for performance metrics such as response time or throughput, for instance, "queries to the database server should take no longer than 2 seconds on average; response times above 5 seconds are unacceptable." In many cases, such service levels are undefined. However, they are

typically implicit in the reactions of users to the service provided. If response time is too long, users complain, and thereby define a boundary level of acceptable service.

- ***Specified Technology and Environment:*** Adequate capacity must be considered in the context of technological and environmental standards serving users. For example, a scientist may require a UNIX workstation with a high resolution monitor for scientific visualization. In this case, providing adequate capacity with a PC-based workstation is not a feasible option.

- ***Cost:*** Adequate capacity requires expenditure to provide acceptable service levels for a given technology and environment and provide desirable room for workload growth. This expenditure includes startup costs and operational costs for a defined period. Startup costs include purchase expenditures for hardware and software, as well as installation costs. Operational costs include hardware and software maintenance, telecommunications costs, and personnel costs required to maintain the system. Amortization costs may be included if they are incurred.

Given the above considerations, ***a computer system is said to have adequate capacity if the service levels are continuously maintained within acceptable levels for specified technology, standards, and growth requirements at minimum cost.***

As an example, consider the scenario of Table 1 where four system configurations (A through D) are analyzed. Configuration B provides acceptable performance at minimum cost, satisfying two criteria of adequate capacity. However, it has no room for workload growth. If workload growth is imminent, C is more appropriate.

Table 1. Example of Adequate Capacity.

Configuration	A	B	C	D
Cost ($)	95K	164K	190K	212K
Performance	Unacceptable	Acceptable	Acceptable	Desirable
Room for Workload Growth	No	No	Yes	Yes

In practice, one needs a cost model, a workload model, and a performance prediction model to support capacity planning. The ***cost model*** accounts for telecommunications, hardware, software, and personnel expenditures. The ***workload model*** captures the resource demands and intensity characteristics for each component of a global workload within a representative time frame. The performance prediction model resolves the performance capabilities of a system to help characterize it as "desirable," "acceptable," or "unacceptable." These models support capacity planning as depicted in Fig. 2.

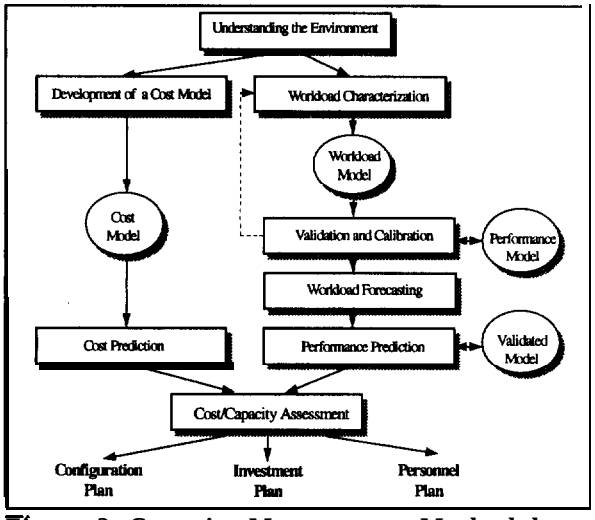

Figure 2. Capacity Management Methodology.

On the surface, this methodology differs little from the capacity planning methodology of traditional environments, however, the support tools and data gathering requirements are quite different. The steps of the methodology are described below, with a focus on how they apply to a IAN environment.

3. UNDERSTANDING THE ENVIRONMENT

The initial phase of the methodology is to learn what kind of hardware, software (OS and applications), and network connectivity are present in the environment. It also involves the identification of peak usage periods, management structures and service level agreements. This information is gathered by various means including user group meetings, audits, questionnaires, help desk records, planning documents, interviews, and other information gathering techniques.

Elements of the Environment

The elements of the environment that must be catalogued and understood are:

- ***Hardware and Basic Software-tie quantities*** and type of hardware platforms and basic software (i.e., operating system related software).
- ***Network Connectivity-the types*** of LANs and

WANs present in the environment and their interconnection structure. Full details of network technologies, devices and bandwidth are represented in a network connectivity diagram.

- ***Network Protocols-all*** protocols being routed in the network environment.

- ***Server Configurations--all*** server types, configurations, and operating systems.

- ***Types of Applications--the*** applications executed in the environment.

- ***Service Level Agreements*** (SL4s)-any SLAs between users and service providers are recorded. SLAS may cover response tune, throughput, availability, and staff responsiveness. When formal SLAs are absent, industry standards are to be used. For instance, a survey of 250 US companies found that response times in JAN environments average 2.7 seconds, throughput averages 7,260 transactions/user/year, and availability averages 92 percent [ITG94]. Such numbers can be used as service level objectives in lieu of established SLAs.

- ***Structure of LAN Management and Support*** Stuff-the LAN management support structure and staffs interactions with users. The main staff functions (e.g., performance and failure monitoring and recovery; capacity planning) should be noted, along with the size and expertise of the support staff relative to the number of users supported.

- ***Procurement Procedures and Justification—*** the elements and constraints of the procurement process, including justification mechanisms (formal and informal) for acquisitions, expenditure limits, authorization mechanisms, and the duration of the procurement cycle.

4. WORKLOAD CHARACTERIZATION

Workload characterization is the partitioning of a computer system's global workload into workload intensity parameters and service demands. This process begins by breaking-down the global workload ***into workload components,*** which are then partitioned ***into basic components*** [Menascé94], such as illustrated in Table 2. The basic components are then characterized by workload intensiiy parameters (e.g., arrival rates) and service demand parameters (i.e., resource consumption in tune units) at each device. This is illustrated in Table 3 for the workload components of Table 2.

Table 2. Workload Partitioning

Workload Component	Basic Components
E-mail	send mail; retrieve mail
Access to Server Files	read block; write block
Mainframe Access Functions	establish remote session; execute remote transaction; close session

Values must be obtained or estimated for these parameters, preferably as measurements with performance monitors and accounting systems. Measurements must occur during peak workload periods and for an appropriate monitoring interval (e.g., one hour).

Consider the "send mail" basic component. Data would be collected relative to all messages sent during the one hour monitoring intervalAssume that 250 messages were sent during this interval. Measurements are obtained for the message size, server CPU time, and server I/O time for each of the 250 messages. The average arrival rate of send mail requests is equal to the number of messages sent (250) divided by the measurement interval (3,600 seconds), i.e., 250/3,600 = 0.07 messages sent/sec. Similar measurements must be obtained for all basic components.

When workload intensity is high, large collections of workload measures can be obtained. Dealing with such collections is seldom practical, especially if workload characterization results are to be used for performance prediction through analytic models [Menascé94]. One should substitute the measured values of all basic components by a more compact representation. This representation is called a ***workload*** model-the end product of the workload characterization process.

If we use average values of send mail measurements to represent the typical send mail command, the send mail component of the workload model would be a tuple such as: 0.07 messages sent/second, 559 bytes per message, 2.9 msec of CPU time per message, 33 msec of I/O time per message. This is a more compact representation of send mail than the 250 tuples of messages sent during the measurement interval.

Table 3. Basic Component Parameters.

Basic Component a n d -	Parameter Type
send mail:	
• no. of messages sent/hr	workload intensity
• avg. message length	service demand
• server CPU time/message	service demand
• server I/o time/message	service demand
retrieve mail:	
• no. of messages retrieved/hr	workload intensity
• avg. message length	service demand
• server CPU time/message	service demand
• server I/O time/message	service demand
read block:	
• no. of read requests/sec	workload intensity
• avg. block size	service demand
• server CPU time/block read	service demand
• server I/O time/block read	service demand
write block:	
• no. of write requests/sec	workload intensity
• avg. block size	service demand
• server CPU time/block written	service demand
• server I/O time/block written	service demand
establish remote session:	
• avg. sessions establishedisec	workload intensity
• server CPU time/session establishment	service demand
• server J/O time/session establishment	service demand
execute remote transaction:	
• no. of transactions executed/sec	workload intensity
• server CPU time/remote transaction	service demand
• server I/O time/remote transaction	service demand
close session:	
• no. of sessions terminated/sec	workload intensity
• server CPU time/session termination	service demand
• server I/O tin&session termination	service demand

A fmal question to ask is whether or not this compact representation is representative of the workload. A workload model is said to be ***representative*** if values of the performance metrics (e.g., response times; throughputs) obtained with the actual workload-all 250 messages-are close to performance values obtained by executing a synthetic workload. The synthetic workload is composed of the same number of basic components as the actual workload but with parameters equal to those of the workload model.

If parameter values have a large variance, the average value may not be a useful and representative workload model. Additional workload classes may be needed, possibly one class for long messages, and a second class for short messages. Since short messages demand less resources than long messages, this refined characterization might be more representative than the single class workload model. clustering algoritbms can be used to compute an optimal number of classes of a workload model, and the parameter values that represent each class.

The most common techniques for workload partitioning and class parameter calculation are averaging and clustering ***algorithms*** [Menasc694]. Averaging is an appropriate technique when the workload component is homogeneous in terms of resource consumption, or when accuracy of the workload model is not a big concern. Under opposite circumstances, more precise, but more time-consuming techniques, are called for. These techniques include clustering algorithms such as the k-means or minimal spanning trees IMenasc6941.

Data Collection Issues

ln ideal situations, performance monitors and accounting systems are used to determine the parameter values for each basic component. In reality, the tool base required for integrated network and server data collection is still emerging. LAN environments are generally heterogeneous in configuration and there is very little integration among existing tools. The problem is compounded by the fact that most monitoring tools provide aggregate measures at the device levels (e.g., total number of packets transmitted on a LAN segment, or total server CPU utilization). These measures must be apportioned to the basic components.

Due to a lack of monitoring tools, and typical shortages of skilled staff to use them, it may be infeasible to collect data through measurements. Benchmarks and rules of thumb (ROTS) may be needed to apportion aggregate measures to basic components in lieu of real measurements. Pig. 3 illustrates the range of data collection possibilities that exist for a capacity manager.

In the organization for which we developed this methodology, there is a lack of measurement tools and human resources for data collection purposes. Here, benchmark data and ROTS will be extremely important until tools and practices grow in maturity. This is a chronic problem for many organizations.

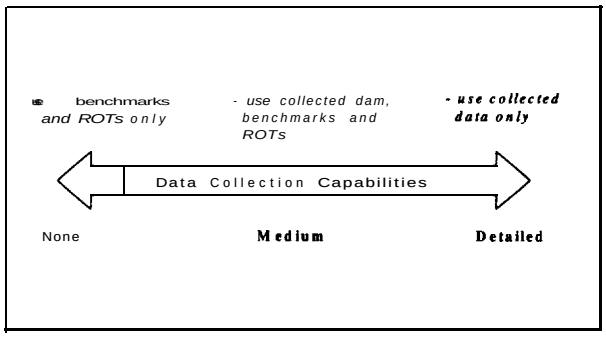

Figure 3. Data Collection Alternatives for Workload Characterization

If proper monitoring tools are available for workload characterization, then measurements should be obtained on the basis of Table 4. The goal should be to use monitoring and accounting tools to support workload and performance modeling as described in Section 6.

Table 4. Data to Collect per System Elements.

System Element	Data to be Collected
Server	CPU utilization utilization of the disks, page fault rates, and disk cache hit ratio.
Client-server *interactions*	Arrival rate of client requests for each typo of request, message size distribution for requests and replies, and average disk visit ratios per type of request.
LAN Segment	Number of packets transmitted, distribution of racket sizes. and mediumuti.l&ion.
Bridges, Routers, and Gateways	Number of rackets filtered and number of packets processed per LAN segment.
Point-to-point connections	Line utilization and average packet size.
WANS	Arrival rate of packets and packet size distribution.

In the environment in question, it was possible to detect a fairly limited number of applications (e.g., wordprocessing, spreadsheet, and e-mail) that accounted for significant portions of resource usage. Workload measurements were taken for a subset of these applications in a controlled environment. These measurements were made separately at the client and at the server for scripts representing typical users using each of the applications mentioned above. These measurements were aimed at obtaining service demands at the CPU and storage devices at the client and server using Osrm2 Lite Resource Monitor Kc) 1993, 1995 C.O.L. Systems Inc.]. Benchmarks are being used to translate the results obtained for a specific type of client and server to other clients and/or servers. For this purpose, we define an index, called Windows Performance Index (WPI), to scale up or down resource usage figures on 486 and Pentium machines running Windows 3.1 applications. WPI is a geometric mean of raw data given in the National Software Testing Laboratories (NSTL) benchmarks [NSTL94] suite. To scale up or down measurements for RISC servers, SPECmarks are used.

5 .

Workload forecasting is the process of predicting how system workloads will vary in the future. They answer questions such as: How will the number of Email messages handled per day by the server vary over the next six months? Will the arrival rate of transactions to the remote mainframe increase by more than five percent over the next year?

Answering such questions involves evaluating an organization's workload trends if historical data are available and/or by analyzing the business or strategic plans of an organization, then mapping these business plans to changes in business processes (e.g., staff increases and paperwork reduction initiatives will yield 50 percent more E mail and Internet usage; 80 percent more imaging).

During workload characterization, workload components are resolved for business processes and their workload intensity and service demand parameters determined. Using this information, business plans can be mapped to changes in workload parameters, and subsequently, the adequacy of a system for future workloads can be determined. For example, assume the capacity of a hospital health care system is at issue because an annex to a clinical center is going to be built, adding new workloads. Up to 600 new beds will be distributed into different wards such as heart and lung diseases, maternity and intensive care. Each ward will have different business requirements in terms of number of research physicians, nurses, and resident doctors per bed. These are ROTS for the organization. ROTS for computerized patient monitoring and alert systems per bed are also known, along with workstation ROTS per bed for nurses, doctors and administrative staff. From information in the strategic plans about bed increases, one can determine the number of workstations needed to support the annex. Total

beds in ibis example is a ***key value indicator @M,,*** or ***forecasting business unit (FSU)*** [Menasc694].

A workload forecasting methodology adapted t?om lMenasc6941 is as follow:

- ***Selection of Applications:*** Select applications that account for most of the utilization of network and server resources and use them as a basis for workload forecasting. The strategic plan may contain new applications to be developed. Include these as part of the workload forecasting process.

- *Identification* ***of*** *Forecasting Business Units:* Identity the FBUs associated with applications whose growth will be forecast.

- ***Collection of Statistics on FBUs:*** Collect statistics on the number of FBUs. Statistics about the past are important, if they are unavailable, obtain this through interviews.

- ***Forecasting of Number of FBUs:*** Use forecasting techniques to determine how tbe number of FBUs will vary in the future. Several pattern types may be identified, including trend, stationary, seasonal, and cyclical, using forecasting techniques such as linear regression and moving averages. If reliable, historical data is unavailable, use FBU evolution and the business and organizational projections in strategic plans.

6 PERFORMANCE PREDICTION

An important aspect of capacity management involves predicting if a system will deliver performance metrics (e.g., response time and throughput) that meet desired or acceptable levels. This section discusses this aspect for LANs.

6.1 Performance Prediction Techniques

Performance prediction is the process of estimating performance measures of a computer system for a given set of parameters. Typical performance measures include response time, throughput, device utilization, and device queue length. Parameters are divided into the following categories:

- ***System parameters-the*** characteristics of a system that affect performance. Examples include operating parameters such as levels of multiprogramming and scheduling disciplines.

- ***Device parameters-the*** characteristics of devices that affect performance. Examples include disk seek times, latency, transfer rates, CPU speed ratings (e.g., MIPS; MFLOPS), network bandwidth, and router latencies.

- ***Workload parameters-parameters*** derived from workload characterization and divided into:

  – ***Workload Intensity Parameters-the*** parameters indicating number of workload units of work that will contend for system resources. Examples are transaction arrival rates and think times at workstations.

  – ***Workload Service Demand Parameters-*** for a workload unit of work (e.g., transactions), the total time spent by this unit of work at that device. For example, the service demand of a DB transaction at the CPU of the SQL server is the total CPU time of the database transaction. Complete workload characterization specifies a workload's service demands at each device.

Performance prediction requires the use of models. Two types of models may be used: simulation models or analytical models. Analytic models for LAN environments are based on ***queuing networks (QNs)*** lMenasc6941, a QN being a set of interconnected ***devices.*** A device-also called a ***queue*** in queuing network terminology-is composed of one or more service-providing elements and a waiting queue. Devices are used to model different elements of an actual system: server CPUs, server disks, routers, IAN segments, point-to-point links, and workstations. Devices are characterized by their average service rate, or equivalently, by the average customer service time (the inverse of the service rate) at the device. A special class of devices, called delay ***device,*** is used to model situations when no queue arises at the device. Workstations in a client/server environment are usually modeled as delay devices since a reply to a request finds the submitting workstation always available. Devices may be further classified into load independent and load dependent. ***Load independent*** devices are those for which the service rate is constant, i.e., it does not vary with the number of customers present at the device. ***Load dependent*** devices have a service rate that varies with the number of customers at the device. These devices are useful to model LAN segments where service rate is a function of the traffic as in an Ethernet segment.

In queuing network terminology, the unit of work that flows from one device to another is called a customer. Customers may be jobs, processes, database transactions, or remote tie service requests. A customer is the model representation of the workload unit of work. Since a system may contain many different types of workloads contending for system resources, a QN model may contain many different classes ***of customers.***

Customer classes may be of two types: open and closed. ***Open classes are*** those in which the total number of customers at any given time in the QN is not bounded. Open classes are used to model situations where the arrival rate of customers to the system is independent of the number of customers in the system. An on-line transaction processing (OLTP) system would typically be modeled by open classes. A ***closed class*** is one in which the ideal number of customers of that class is constant. A system where customers submit a request and wait for the reply before submitting a new request (e.g., time-sharing terminals) could be modeled by a closed class. A QN where all classes are open is called an open QN, a Q N which all classes are closed is called a closed QN; and a QN where some classes are open and some are closed is called a mixed class QN.

Customer classes are characterized by the service demand at each device. The service demand D at a particular device is defined as the total time spent by a customer getting service at the device. This is equal to the average number of visits ***V to*** the device multiplied by the average service time S at tbe device per visit. Thus, D = ***V S.*** Closed classes are further characterized by the total number of customers in the class ***(customer population)*** and open classes by the ***average*** arrival rate of customers.

Once all the parameters of a QN are determined, one can solve for the average throughput, average response time, device utilizations and queue length using quite efficient algorithms [Menascé94].

6.2 Performance Prediction in LANs

Before we describe performance prediction techniques in LAN environments, it is important to understand the elements of these environments that affect performance.

6.2.1 LAN Components and Their Impact on Performance.

A IAN may be characterized by workstation components, server components, communications medium components, protocol components, and LAN interconnection components. Workstation components include the workstation's processor and disks. Server components include the server's processor(s), the server's disks and disk cache. The communications medium is the conduit through which packets of data flow among the various workstations and between workstations and servers, such as coaxial and fiberoptic cable.

The protocol components include the complete protocol stack used to control the communication between workstations and servers. These protocol stacks are organized in layers and run on top of the medium access protocols, such as Ethernet, Token Ring, or F DDI. Typical examples include TCP/IP and Novell's Netware IPX.

The LAN interconnection components include repeaters, bridges, routers, brouters, and gateways. A ***repeater*** connects two similar IAN segments, such as two Ethernets to overcome distance and station count limitations. It operates at the physical layer and has a negligible effect on LAN performance [Johnston92]. In contrast, a ***bridge*** connects LANs at the data link layer and is used to filter traffic A bridge can become a performance bottleneck if it is not able to receive frames and determine in a timely fashion if they should be passed ***to another LAN. Routers are*** used to connect LANs at the network layer (layer 3 of the IS0 OS1 Reference Model). They determine the paths to be followed by packets as they go from one LAN to another. They provide robust error recovery and flow control and f+equently are used to interconnect technologically dissimilar LANS, but of the same protocol. Routers can easily become a performance bottleneck since routing imposes a larger computational demand on a device than bridging. ***Brouters*** are a combination of bridges and routers. They route the protocols that account for most of the network traffic (e.g., TCP/IP, IPX) and bridge any unrecognized protocols. ***Gateways*** perform protocol conversion and have the lowest performance of all network interconnection devices. They also incur the greatest interconnection cost [Jobnston921, so LAN administrator often avoid them in favor networks that run multiple protocol stacks simultaneously in network servers, a practice that increases the load on the server.

6.2.2 APerfonname Prediction Methodology for LAN Environments

Performance prediction is generally concerned with applications in a LAN-based environment. Applications do not run on a standalone machine, but partly on a client workstation and partly on one or more servers linked to the client by the network. The network may be as simple as a single IAN segment, or it may be a myriad of LAN segments connected by bridges, routers or even WANs.

Predicting the performance of applications ii client/server environments entails using models that can take into account the delays incurred by an application in the various parts of the system. There is a wide range of modeling alternatives. As more system elements are represented in greater detail, model accuracy increases. Data gathering requirements also increase, as shown in Pig. 4. R is important that a reasonable balance be made between model accuracy and ease of use to allow for the analysis of many alternatives with little effort and in very little time. Analytic models are quite appropriate for the performance prediction component of any capacity management/planning study. In this section we explore how QN models can be used to model LAN environments composed of multiple LAN segments.

Fig. 5 displays two LAN segments connected by an FDDI ring. Each segment has its own file server. Most user requests for file service are served locally, the other requests are served by the remote file server. The work performed by users at their workstations is characterized by periods of local processing of average duration equal to Z seconds alternating with requests to the file server. Let p12 be the fraction of file requests originated at LAN segment 1 and directed to the file server in LAN segment 2. Similarly, let ***p21*** be the fraction of tile requests originated at LAN segment 2 and directed to the file server in LAN segment 1. Assume that both file servers have identical disks with seek time equal to 9 msec, average rotational delay equal to 8.3 msec and transfer rate equal to 10 Mbytes/sec. Each tile request requires 10 msec of processing at the server CPU. Each bridge imposes a delay equal to 1 msec per packet. Each LAN segment is an Ethernet running at 10 Mbps. The FDDI ring has a bandwidth equal to 100 Mbps.

A file access request sent from a workstation to the file server has 100 bytes on average while a reply from the file server to the workstation has 4500 bytes on average. Each file request requires 2.5 I/O operations, on average, from the file server disk. On average, 4 packets are exchanged between a workstation and a server for each request.

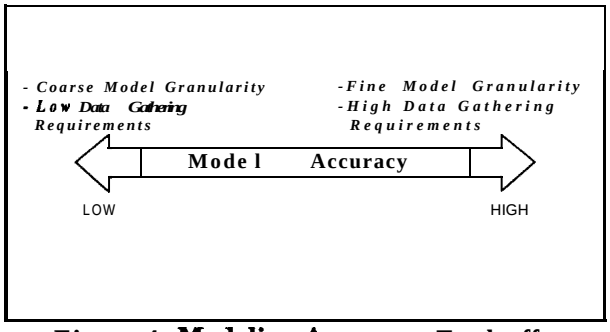

Figure4. Modeling Accuracy Tradeoffs.

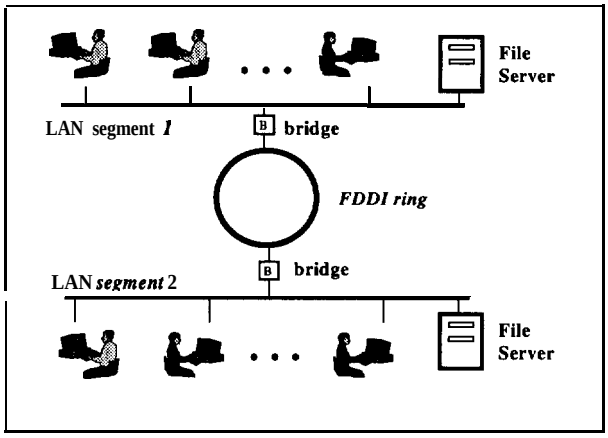

Figure 5. LAN Example.

Fig. 6 displays a QN model of the Pig. 5 LAN. The workstations at each LAN segment are characterized by delay devices W1 and W2 and their service demands as *Z*. Bridges B1 and B2 are represented by queuing devices B1 and B2, respectively. The CPUs of the servers are represented respectively by queuing devices Cl and C2. The queuing devices D1 and D2 represent the disks at file server 1 and file server 2, respectively. Delay devices L1 and L2 represent LAN segments 1 and 2, respectively. Finally, the FDDI ring is modeled by a delay server due to its high bandwidth. The model has two customer classes representing requests originating at IAN segment 1 (class 1) and LAN segment 2 (class 2). The average visit ratios (V), average service times (S), and service demands (D) at each of the QN devices are given in Tables 7a and 7b as a function of *Z*, *p12*, and *p21*.

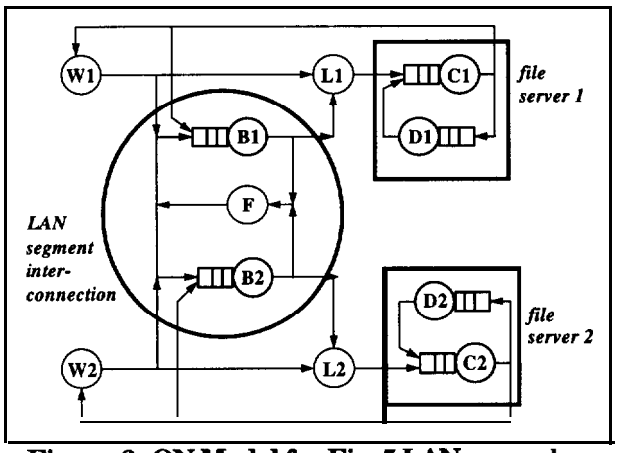

Figure 6. QN Model for Fig. 5 LAN example.

Table 7a. Class 1 Serv. Demands on Fig. 6 QN.

	Class 1		
Device	V	S	D = V S
W1	1	Z	Z
W2	0		0
B1	1	4	4
B2	D12	4	4 p12
F	P12 L	0.46	0.46 p12
C1	(1-p12)	10	10 (1-p12)
D1	2.5 (1-p12)	21	52.5(1-p12)
C2	p12	10	10 p12
D2	2.5p12	21	52.5p12
L1	1	4.6	4.6
L2	p12	4.6	4.6 p12

Table 7b. Class 2 Serv. Demands on Fig. 6 QN.

	Class 2		
Device	V	S	D = V S
W1			
W2	1	Z	Z
B1	p21	4	4 p21
B2	1	4	4
F	p21	0.46	0.46 p21
C1	p21	10	10 p21
D1	2.5 p21	21	52.5 p21
C2	(1-p21)	10	10 (1-p21)
D2	2.5 (1-p21)	21	52.5 (1-p21)
L1	p21	4.6	4.6 p21
L2	1	4.6	4.6

If Z is set to 0.1 seconds, *p12* and *p21 are* set to 0.2, and the queuing network model is solved, the average response time per request as a function of the number of users in each LAN segment is obtained as depicted in Fig. 7. It is assumed that both segments have equal numbers of users.

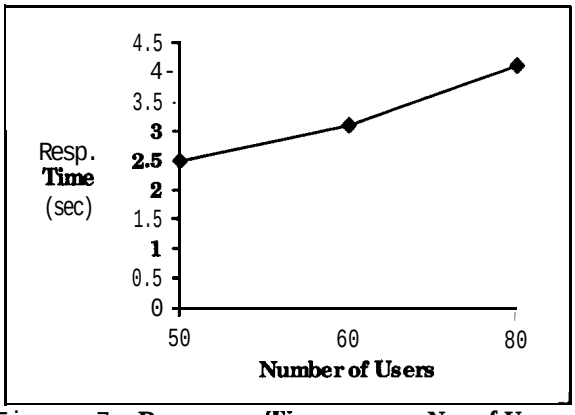

Figure 7. **Response Time versus No. of Users.**

7. VALIDATION OF MODELS

In building any model, abstractions of the reality being modeled are made for simplicity, ease of data collection and use, and the computational efficiency of the modeling process. The abstractions compromise the accuracy of the model, so the model must be validated within an acceptable margin of error, a process called ***model*** validation. If a model is deemed invalid, it must be calibrated to render it valid. This is called ***model calibration***

In capacity management, as indicated in Fig. 2, there are two models to be validated the workload model and the performance model, as follows.

7.1 Workload Model Validation

Validating workload models entails running a synthetic workload composed of workload model results and comparing the performance measures thus obtained with those obtained by running the actual workload. If the results match within a 10 to 20 percent margin of error, the workload model is considered to be valid. Otherwise, the model must be refined to more accurately represent the actual workload. This process is depicted in Fig. 8.

Tbis canonical procedure of workload model validation may be too time consuming at times. An alternative, albeit less accurate, procedure could take place by means of interviews with users about their workload profiles. For instance, do they spend 60 percent of their workday doing WordPerfect-based word-processing, mostly between 10 AM to noon and 2 PM to 4 PM? Does this result in 40 new pages of text on average?

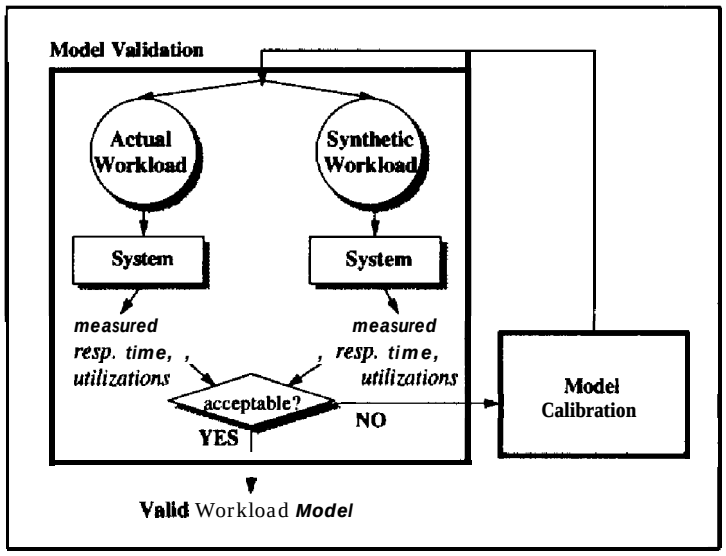

Figure 8. Workload Model Validation Steps.

7.2 Performance Model Validation

A performance model is said to be valid if the performance metrics (e.g., response time, device utilizations, and throughputs) calculated by the model match with measurements of the actual system, and within a certain acceptable margin of error. Accuracies from 10 percent to 30 percent are acceptable in capacity planning [Menascé94].

Fig.9 illustrates the various steps involved in performance model validation. During workload characterization, measurements are taken for service demands, workload intensity, and also for performance metrics such as response time throughput, and device utilization. The same measures are computed by means of the performance model. If the computed values do not match the measured values within an acceptable level, the model must be calibrated. A detailed discussion on calibration techniques is given in [Menascé94]. Otherwise, the model is deemed valid and can be used for performance prediction.

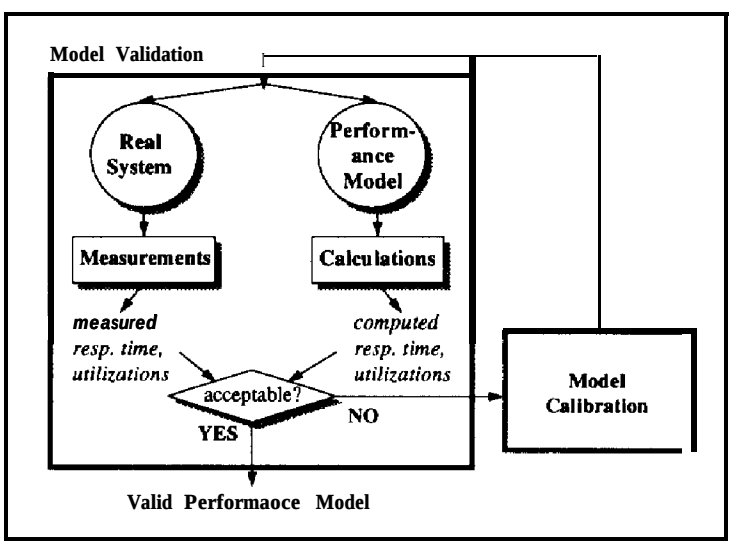

Figure 9. Performance Model Validation Steps.

8. DEVELOPMENT OF A COST MODEL

A capacity planning methodology requires the identification of major sources of cost as well as the determination of how costs vary with system size. Costs are categorized into startup and operational costs. Startup costs are those incurred in setting up the system, while operational costs are the annual expenses incurred to maintain the system and provide upgrades in hardware and software to avoid obsolescence. Startup costs apply to hardware, software, infrastructure and initial installation charges. Operational costs are related to hardware and software maintenance and upgrades, personnel costs, training telecommunications services and consulting fees.

Determining Costs in LAN Environments

The highly distributed nature of LAN and client/server environments is frequently responsible for a lack of knowledge of the costs incurred in maintaining these environments. A recent survey of 250 companies showed that fewer than five percent of US corporations Quantitjr their expenditures of PCs and LANs [ITG94]. The same study indicates that costs associated with LAN connected PCs range from $5,000 to $10,000 per user, making PC/LAN usage the largest undocumented IS cost in most corporations.

Personnel costs are a major source of costs in LAN environments. Data from a recent survey [ITG94] indicates that personnel costs in IAN environments account for about 60 percent of the total cost. Only 22 percent of costs are due to hardware and 11 percent to software.

One can be as detailed as required when accounting for costs. This increases the accuracy of the cost assessment at the expense of the time and effort needed to obtain all needed information. Alternatively, one can use more aggregated cost models that account for most of the cost elements with a reasonable degree of accuracy.

Table 8 presents an example of Cost Model results. This table contains all the elements of a cost model suggested here. The column labeled ROT stands for Rules of Thumb. The ROT cells show typical values computed as a function of the number of workstations. Some examples of cost ROTS are (see Bufnagel941 and [ITG94]):

- Hardware and software upgrade is 10 percent of the purchase price per year.

- System administrator costs lie between $500 and $700 per client per month.

- TxAning costs range from $1,500 to $3,500 per technical staff person per year.

- 40 percent of personnel costs are in the resource management category, 40 percent are in applications development and maintenance, and 20 percent in other personnel.

Table 8. Cost Model Results.

CostModel

Startup Costs (year 1)	Unit Cost	ROT	No. units	Total
Hardware Cost per Workstation	$ 1,500	$1,700	50	$ 75,500
Software Cost per Workstation	$ 750	$...800	50	$ 37,500
Total Cost per Workstation	$ 2,250	$2,500	50	$112,500
Server Hardware Cost	$ 5,500	$6,000	4	$ 22,000
Server Software Cost	$ 2,500	$2,500	4	$ 10,000
Total Server Cost	$ 8,000	$8,500	4	$ 32,000
Printers	$ 4,500			$ 4,500
Comm. Equip. (bridges, routers, etc.)	$ 6,000			$ 6,000
Total Startup Cost				$155,000
Avg. Startup Cost per WS	$ 3,100			
Ongoing Costs per Year		**ROT**		**Total**
Hardware Maintenance per WS	$ 200	$ 150	50	$ 10,000
Software Maintenance per WS	$ 80	$ 75	50	$ 4,000
Server Hardware Maintenance	$ 650	$ 550	4	$ 2,600
Server Software Maintenance	$ 300	$ 250	4	$ 1,200
Total Maintenance Cost				$ 14,950
Avg. Maintenance Cost per WS	$ 356			
Personnel Costs			%	
Resource Managers	$250,000	$300,000	67%	
AppLDevelope&Maint.enanee	$ 80,000	$ 72,800	21%	
Other	$ 45,000	$ 37,442	10%	
Total Personnel Cost	$375,000	$288,269		$375,000
Personnel Cost per WS	$ 7,500			
Training Costs	$ 65,000	$ 53,500		$ 65,000
WAN Communications Costs	$ 10,000	$ 14,413		$ 10,000
Total Operational Costs	$ 464,950			
Avg. Operational Cost per WS	$ 9,299			

9. CONCLUDING REMARKS

The methods that have served capacity planners well in traditional mainframe environments, including workload characterization, workload forecasting, and performance prediction can still serve capacity managers well in client/server, distributed processing environments. (See [Nyman941, [Domanski941 and [Grunumř 94] for additional discussions of client/server issues and approaches.) However, reliable and effective measurement tools and estimation methods for workload characterization must become available for the newest architectures. In lieu of quality tools, objectively derived ROTS and benchmark data can serve as inputs to analytical modeling. Such measurement "surrogates" are integral parts of the methodology presented here, and enablers of the earliest analytical modeling activities when real measurements may be unavailable. It was the goal of the underlying project initiative to develop a methodology that helps a loosely integrated federation make cost-effective decisions about a heterogeneous computing environment. By giving LAN capacity managers an adaptable and easy means of modeling performance and costs, this goal will be met. To help assure the methodology's success, its development has been accompanied by the creation of a supporting capacity planning tool called CMWZan-Capacity Management Workbook for IAN environments, an analytical modeling tool that supports internaJ

ROTS and benchmark databases, and also accepts inputs of workload and performance measurements from monitoring tools. Analytical models are prepared automatically by CMWLan using inputs designated by the user for assessments of different configuration alternatives. This new tool has been successfully demonstrated with the methodlogy to members of the federation's network capacity management working group in February 1995.

10. REFERENCES

[Bell92] Bell, T. E. and Falk, A. M., Measuring Application Performance in Open Systems, Proceedings of the 1992 Conference of the Computer Measurement Group, December 1992, pp. 277-284.

[Domanski94] Domanski, B., Capacity Management for Client/Server Architectures, Techniques & Systems Management Issues, Proceedings of the 1994 Conference of the Computer Measurement Group, December 1994, pp. 136-144.

[Grummitt94] Grummitt, A., Performance Management of Client/Server Systems (Open & Proprietary), Proceedings of the 1994 Conference of the Computer Measurement Group, December 1994, pp. 290-299.

[Hajare91] Hajare, A. R., Performance Modeling of a LAN Bridge, Proceedings of the 1991 Conference of the Computer Measurement Group, December 1991, pp. 426-431.

[Hajare93] Modeling a Multiple Segment LAN, Proceedings of the 1993 Conference of the Computer Measurement Group, December 1993, pp. 818-828.

[Ho92] Ho, E., Performance Management of Distributed Open Systems, Proceedings of the 1992 Conference of the Computer Measurement Group, December 1992, pp. 742-751.

[Hufnagel94] Hufnagel, E., The Hidden Costs of Client/Server, Your Client/Server Survival Kit, supplement to Network Computing, Vol. 5, 1994.

[ITG94] Information Technology Group, Cost of Computing, Comparative Study of Mainframe and PC/LAN Installations, Mountain View, CA, 1994.

[Johnston92] Johnston, J., Performance Implications of LAN Interconnection Options, Proceedings of the 1992 Conference of the Computer Measurement Group, December 1992, pp. 1053-1062.

[Mayer94] Mayer, J. H., Risc vs. Cisc: No Longer a No-Brainer Decision, Servers, *Client/Server Computing*, February 1994, pp. 80-83.

[Menascé94] Menascé, D. A., V. A. F. Ahneida, L. W. Dowdy, *Capacity Planning and Performance Modeling: from mainframes to client-server systems*, Prentice Hall, Englewood Cliffs, NJ, 1994.

[Nemzow93] Nemzow, M. A., *LAN Performance Optimization*, Windcrest/McGraw-Hill, 1993.

[NSTL94] National Software Testing Laboratories, High Performance 486 DX2 and Pentium Systems, PC Digest Ratings Report, Volume 8, Number 2, February 1994.

[Nyman94] Nyman, M., Methodology for Assessing Capacity Needs in an Intelligent Workstation/Local Area Network (IWS/LAN) Environment, Proceedings of the 1994 Conference of the Computer Measurement Group, December 1994, pp. 165-176.

[Vicente93] Vicente, J., Network Capacity Planning, Proceedings of the 1993 Conference of the Computer Measurement Group, December 1993, pp. 758-768.

[Waldner92] Waldner, R., Client/Server Capacity Planning, Proceedings of the 1992 Conference of the Computer Measurement Group, December 1992, pp. 1046-1052.