The Computer Measurement Group, commonly called CMG, is a not for profit, worldwide organization of data processing professionals committed to the measurement and management of computer systems. CMG members are primarily concerned with performance evaluation of existing systems to maximize performance (e.g. response time, throughput, etc.) and with capacity management where planned enhancements to existing systems or the design of new systems are evaluated to find the necessary resources required to provide adequate performance at a reasonable cost.

This paper was originally published in the Proceedings of the Computer Measurement Group's 1995 International Conference.

For more information on CMG please visit http://www.cmg.org

Copyright Notice and License

Copyright 1995 by The Computer Measurement Group, Inc. All Rights Reserved. Published by The Computer Measurement Group, Inc. (CMG), a non-profit Illinois membership corporation. Permission to reprint in whole or in any part may be granted for educational and scientific purposes upon written application to the Editor, CMG Headquarters, 151 Fries Mill Road, Suite 104, Turnersville, NJ 08012.

BY DOWNLOADING THIS PUBLICATION, YOU ACKNOWLEDGE THAT YOU HAVE READ, UNDERSTOOD AND AGREE TO BE BOUND BY THE FOLLOWING TERMS AND CONDITIONS:

License: CMG hereby grants you a nonexclusive, nontransferable right to download this publication from the CMG Web site for personal use on a single computer owned, leased or otherwise controlled by you. In the event that the computer becomes dysfunctional, such that you are unable to access the publication, you may transfer the publication to another single computer, provided that it is removed from the computer from which it is transferred and its use on the replacement computer otherwise complies with the terms of this Copyright Notice and License.

Concurrent use on two or more computers or on a network is not allowed.

Copyright: No part of this publication or electronic file may be reproduced or transmitted in any form to anyone else, including transmittal by e-mail, by file transfer protocol (FTP), or by being made part of a network-accessible system, without the prior written permission of CMG. You may not merge, adapt, translate, modify, rent, lease, sell, sublicense, assign or otherwise transfer the publication, or remove any proprietary notice or label appearing on the publication.

Disclaimer; Limitation of Liability: The ideas and concepts set forth in this publication are solely those of the respective authors, and not of CMG, and CMG does not endorse, approve, guarantee or otherwise certify any such ideas or concepts in any application or usage. CMG assumes no responsibility or liability in connection with the use or misuse of the publication or electronic file. CMG makes no warranty or representation that the electronic file will be free from errors, viruses, worms or other elements or codes that manifest contaminating or destructive properties, and it expressly disclaims liability arising from such errors, elements or codes.

General: CMG reserves the right to terminate this Agreement immediately upon discovery of violation of any of its terms.
ON THE DESIGN AND IMPLEMENTATION OF A CAPACITY MANAGEMENT TOOL FOR LAN ENVIRONMENTS

Daniel A. Menasce
George Mason University
Fairfax, VA 22030

Duane Dregits
Science Applications International Corporation
Reston, VA 22090

Capacity management for LAN environments involves server sizing and location, network connectivity, mapping of applications to servers, and network bandwidth. LAN administrators require the ability to predict the future performance of proposed configurations. They need tools to help in effectively answering several what-if questions in short periods of time. In this paper we describe the design and implementation of CMWLan - Capacity Management Workbook for LAN environments - a tool developed by the authors to satisfy the capacity management needs of a large networked organization. CMWLan uses Microsoft Excel version 5.0 worksheets combined with new functionality developed in Microsoft Excel Visual Basic. The paper presents an object-oriented schema for the objects manipulated by CMWLan, as well as the relational database derived from the schema. The relations are implemented as Excel worksheets. CMWLan's graphical user interface and the underlying analytic performance model are also discussed.

1. INTRODUCTION

CMWLan is a tool that supports the capacity management methodology for LAN environments developed by the authors [Menasce95]. CMWLan uses Microsoft Excel version 5.0 worksheets combined with new functionality developed in Microsoft Excel Visual Basic. Excel was selected because of its Visual Basic feature.

CMWLan covers three key areas of the capacity management methodology: workload characterization, performance modeling and prediction, and cost modeling. Even though CMWLan is not a measurement tool, it provides support to workload characterization by supplying benchmark databases and Rules of Thumb (ROTs) to substitute for missing or incomplete data. In addition, measurements of real workloads can be used when available. Performance modeling and prediction are carried out through built-in analytic models. A cost model is integrated into CMWLan. The cost estimates generated by CMWLan take into account the cost parameters for the different network objects, but also provide ROTs if the user does not have detailed cost figures.

CMWLan is designed to satisfy the following requirements:

- Platform independence: the tool can be used to perform capacity management for network environments consisting of any mix of client workstations, servers, network interconnection devices, and network technologies.
- Application independence: CMWLan can be used to plan the capacity of LAN environments for any mix of applications including office automation, scientific computing, and video conferencing.
- Adaptability to different data gathering requirements: CMWLan can be used in environments where there is very little data gathering capability as well as in situations where detailed data collection facilities are in place. Various benchmark databases and sets of ROTs are integrated into CMWLan to assist in the situations where there are limited data gathering capabilities.
- Ease of use: a user-friendly interface provides extensions to Excel menus and dialog boxes that allow for ease of data entry. The complexities of building and solving queuing network models are hidden from the user of CMWLan.

---

1 CMWLan is currently a client sponsored tool, not a commercial product.
CMWLan since they are automatically calculated from the network and application specifications.

Section 2 presents an object-oriented schema for the objects (e.g., clients, servers, network interconnection devices, network media, and applications) manipulated by CMWLan. This schema is converted into a relational database to support the implementation of the tool as described in Section 3. The relations are implemented as Excel worksheets. The user interface to CMWLan is described in Section 4. Finally, the main components of the underlying performance model are described in Section 5.

2. OBJECT-ORIENTED DESIGN

This section presents an object-oriented design to CMWLan. The tool manipulates several types of objects such as network nodes, network interconnection devices, LAN segments, servers, client workstations, applications and others. Rumbaugh's notation [Rumbaugh91], as described in Fig. 1, is used in Fig. 2 to specify the object-oriented schema for CMWLan.

![Figure 1. Legend for Object-Oriented Schema](image)

Rectangles represent object types or classes. A polygon represents a multiway association type. There are three binary association types (one-to-one, one-to-many, and many-to-many). A special type of association, called generalization or is-a, is used to indicate a hierarchy among classes. The triangle in the is-a association points from a specialized class to a more general class. For example, Server is-a kind of NetNode object and so is Client. NetNode is said to be the parent class of both Server and Client object types. Classes have attributes associated with them. Classes inherit all attributes of their parent class and may have other additional attributes. For example, NetMedia has HeaderSize and Bandwidth as attributes. The MultiDrop class inherits HeaderSize and Bandwidth from NetMedia and has an additional attribute called Length. The only attributes considered here are the ones that affect performance and are thus important to CMWLan.

Figure 2 shows the various object types manipulated by CMWLan. One of the most important types is NetObject that can be further specialized into NetMedia, NetConn, and NetNode. NetMedia and its various specializations-Switched, PointToPoint, and MultiDrop-represent pipes through which packets of data flow. These pipes may be LAN segments, point-to-point links, or switched networks, such as an X.25 Wide Area Network (WAN). NetConn represents network interconnection devices such as routers, bridges, and gateways. NetNode represents the sources and destinations of information of a networked environment. This class specializes into two separate classes: Server and Client. Server represents providers of resources such as files, applications, and compute cycles. These can range from PCs to mainframes. The Client class represents a collection of workstations connected to a NetMedia object (e.g., a LAN segment). The Protocol class specializes into NetProtocol for the network layer protocol and TranspProt for the transport layer protocol. The Application class represents different types of applications such as office automation, scientific applications, database applications, E-mail, and others.

The multiway association called NetPath in Fig. 2 is used to represent network paths consisting of workstations, LAN segments, routers, and servers. Finally, it is important to note all binary associations in the CMWLan schema: IsConnectedTo, IsUsedOn, IsServedBy, Uses, IsUsedBy, Runs, ConsistsOf, and Connects. The next subsection gives a detailed description of each object class.

2.1 Objects

Object Class Name: NetObject

Description: describes a network object. Examples of network objects are LAN segments, bridges, routers, client workstations, servers, and point to point links.

Attributes:
- Objectld: Identification of the network object.
- Objects of different types (see next attribute) may have different naming conventions.
- Type: type of the network object. Possible types are Media, Connector, and Node.

ParentClass: none
Object C/ass Name: NetMedia
Description: describes a type of network media used to connect clients to servers or interconnection devices to other interconnection devices or to LAN segments. Examples of NetMedia objects are LAN segments, point-to-point links, and WAN services.
Attributes:
- HeaderSize: size of the header at the link layer (bytes).
- Bandwidth: media transmission capacity (Mbps).
Parent C/ass: NetObject

Object C/ass Name: MultiDrop
Description: indicates a type of network media in which more than one node is connected to the media. Examples include multiple access LAN technologies (e.g., Ethernet and FDDI), and polled multiplex links.
Attributes:
- Length: length of the physical medium (meters).
Parent C/ass: NetMedia

Object C/ass Name: Ethernet
Description: describes a network media using Ethernet technology
Attributes:
- Type: indicates the type of Ethernet technology. The two possibilities are Slow (i.e., 10 Mbps) and Fast (100 Mbps).
- MinPacketLength: minimum packet length (bytes).
- MaxPacketLength: maximum packet length (bytes).
- MaxNodes: maximum number of nodes per segment.
Parent C/ass: MultiDrop

Object C/ass Name: TokenPass
Description: describes a network media using Token Passing technology
Attributes:
- Type: indicates the type of token passing protocol. Possible types are FDDI, TokenRing4, TokenRing6, and Token Bus.
- MinPacketLength: minimum packet length (bytes).
- MaxPacketLength: maximum packet length (bytes).
- MaxNodes: maximum number of nodes per segment.
Parent C/ass: MultiDrop

Object C/ass Name: PointToPoint
Description: indicates a point to point link.
Attributes: none
Parent Class: NetMedia

Object Class Name: Switched
Description: describes a switched network media.
Attributes:
- MinPacketSize: minimum size of the packet (bytes).
- MaxPacketSize: maximum size of the packet (bytes).
Parent Class: NetMedia

Object Class Name: NetConn
Description: describes a network interconnection device such as a bridge or router.
Attributes:
- Type: type of interconnection device. Example types are bridge, router, brouther, and gateway.
- Latency: fixed latency per packet (msec).
- Cost: cost of the network interconnection device.
Parent Class: NetObject

Object Class Name: Router
Description: describes a router.
Attributes:
- RoutingProtocol: describes the type of routing protocol used. Examples are source routing, spanning tree, and RIP.
Parent Class: NetConn

Object Class Name: NetNode
Description: describes a group of one or more network nodes. Nodes may be client workstations or servers.
Attributes:
- NodeId: identification of the network group of nodes.
- HardType: hardware architecture of the node. Examples are 486 DX4, Pentium 90 MHz, SUN SPARC5, and Mac Quadra.
- NodeOS: operating system of the node. Examples include MS Windows, UNIX, and IBM OS2.
- SpecInt: SPECInt92 and SPECfp92 ratings for the node.
- WindowsPerfIndex: this index provides an aggregate measure of the performance of the client node when running several types of Windows applications. This index is computed from raw data provided in the NSTL reports [NSTL94].
- NetCardRate: network card rate (Kbps).
- Cost: cost of the network node.
Parent Class: NetObject

Object Class Name: Server
Description: describes a server.
Attributes:
- NumDisks: number of disks at the server.
- DiskSeekTime: disk average seek time at the server (msec).
- DiskLatency: disk average rotation time at the server (msec).
- DiskTransferRate: disk average transfer rate at the server (MBytes/sec).
- DiskBlockSize: disk block size at the server (KBytes).
Parent Class: NetNode

Object Class Name: Client
Description: describes a group of client workstations connected to the same NetMedia object (typically a LAN segment).
Attributes:
- Number-Clients: number of client workstations in the group.
Parent Class: NetNode

Object Class Name: UserType
Description: describes a user category in a given NetMedia object (typically a LAN segment).
Attributes:
- Name: name of the user type. Examples are: administrative, scientist, and system administrator.
- Percent Day: percentage of a working day the user is active using his/her workstation.
- ApplProfile: percentage of time each application is executed by the users of this type.
Parent Class: none

Object Class Name: Application
Description: describes a type of application
Attributes:
- Name: name of the application type. Examples include WordPerfect and video conferencing.
- AvgMessagesReq: average number of messages per request.
- AvgMsgSizeToS: average size of messages generated by the application to the server (bytes).
- AvgMsgSizeFromS: average size of messages generated by the server to the application (bytes).
- AvgRequestRate: average rate at which requests are sent to the server for this application (requests/sec).
- ServerCPUDemand: average service demand per request at the server CPU (msec).
3. RELATIONAL DATABASE DESIGN

The objects, their attributes, and their associations were used to derive a relational database design. Each relation was implemented as an Excel 5.0 worksheet. There are three categories of relations:

- Object Relations: these are relations used to represent objects.
- Object Type Relations: these relations are used to represent types of objects.
- Association Relations: these are the relations used to represent associations between objects.

The tables that comprise the relational database are described below with sample contents.

### 3.1 Object Relations

**NetMedia** relation: describes the different network media. For each one, the network medium type and length are provided.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Length (meters)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LANSEG001</td>
<td>Token Ring 4</td>
<td>200</td>
</tr>
<tr>
<td>LANSEG002</td>
<td>Ethernet</td>
<td>150</td>
</tr>
<tr>
<td>Link001</td>
<td>Link56K</td>
<td>1,000</td>
</tr>
</tbody>
</table>

**NetConnectors** relation: describes all network interconnection devices with an indication of their type.

<table>
<thead>
<tr>
<th>Connector Id</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON001</td>
<td>Cisco 4500</td>
</tr>
<tr>
<td>CON002</td>
<td>Cisco 7000</td>
</tr>
<tr>
<td>CON003</td>
<td>Cisco 4000</td>
</tr>
</tbody>
</table>

**Servers** relation: lists all the servers. For each server the following information is given: server type, number of disks at the server, disk seek time, disk latency, disk transfer rate, disk block size, and network card rate.

<table>
<thead>
<tr>
<th>Server Id</th>
<th>Server Type</th>
<th>No. of Disks</th>
<th>Disk Seek Time (msec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SV0001</td>
<td>Hi-End Pentium</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>SV0002</td>
<td>486-100 DX4</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>SV0003</td>
<td>SUN MP670</td>
<td>2</td>
<td>11</td>
</tr>
</tbody>
</table>

### Attributes

- **BaseServer**: server type at which the value of ServerCPUDemand was obtained.
- **ClientCPUDemand**: average service demand per request at the client CPU (msec).
- **BaseClient**: client type at which the value of ClientCPUDemand was obtained.
- **IOVolume**: total number of bytes read or written per request at the server per request (KBytes).
- **Benchmark**: type of benchmark (e.g., Windows Performance Index, SPECint92, SPECfp92) to be used when computing Performance Speedup Factors for this application.
Clients relation: lists all clients, indicating for each of them their type, network card transfer rate, and number of clients in the group of nodes designated by a given client id. Note that all clients of the same client group have to be connected to the same network media.

### Table 4. Clients Relation

<table>
<thead>
<tr>
<th>Client Id</th>
<th>Client Type</th>
<th>No. of Clients in the group</th>
<th>Net Card Rate (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC0001</td>
<td>486-33 W3.1</td>
<td>45</td>
<td>10</td>
</tr>
<tr>
<td>PC0002</td>
<td>486-66 W3.1</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>UNIX001</td>
<td>SPARC5</td>
<td>20</td>
<td>10</td>
</tr>
</tbody>
</table>

Applications relation: for each application, provides the following information: average number of messages generated per request to the server, average rate at which requests are sent to the server, average size of the messages sent to the server, average size of the messages received from the server, average CPU service demand per request at the server, base server for which service demand is given, average CPU service demand per request at the client, total number of bytes read or written at the server, and the benchmark (e.g., Windows Performance Index, SPECint92, or SPECfp92) to be used when computing a Performance Speedup Factor (see Section 5) for the application.

### Table 5. Applications Relation

<table>
<thead>
<tr>
<th>[columns 1-4]</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Name</td>
<td>Avg. Msgs. per Request</td>
<td>Avg. Request Rate (req/sec)</td>
<td>Avg. Msg to Server</td>
</tr>
<tr>
<td>E-mail</td>
<td>1.2</td>
<td>0.001</td>
<td>240</td>
</tr>
<tr>
<td>Video</td>
<td>45</td>
<td>0.0002</td>
<td>10000</td>
</tr>
<tr>
<td>Conferencing</td>
<td>4.2</td>
<td>0.001</td>
<td>348</td>
</tr>
</tbody>
</table>

### Table 6. NetProtocol Relation

<table>
<thead>
<tr>
<th>Network Protocol</th>
<th>Max Packet Size (bytes)</th>
<th>Overhead (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>1000</td>
<td>20</td>
</tr>
<tr>
<td>IPX</td>
<td>1200</td>
<td>18</td>
</tr>
</tbody>
</table>

3.2 Object Type Relations

NetMediumType relation: lists all types of network media. The following information is given on each of them: bandwidth, overhead, minimum packet length (data plus overhead), maximum packet length (data plus overhead), and maximum number of nodes in the network medium. The data sources for the NetMediumType include [Albert94, Gohring92, Held94, Nemzow93, Nilausen94, Stallings90, Stallings94].

### Table 7. NetMediumType Relation

<table>
<thead>
<tr>
<th>[columns 1-3]</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Min Packet Length (bytes)</td>
<td>Max Packet Length (bytes)</td>
<td>Max. Nodes</td>
</tr>
<tr>
<td>53</td>
<td>53</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>60</td>
<td>1511</td>
<td>512</td>
</tr>
<tr>
<td>15</td>
<td>4500</td>
<td>512</td>
</tr>
</tbody>
</table>

ConnectorType relation: describes all connector types. The following information is given on each of them: maximum throughput, packet size for which the maximum throughput is measured, routing protocol used, and cost.

### Table 8. ConnectorType Relation

<table>
<thead>
<tr>
<th>[columns 1-3]</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector Type</td>
<td>Max Throughput (packets/sec)</td>
<td>Max Throughput Packet Size (bytes)</td>
<td></td>
</tr>
<tr>
<td>Cisco 4000</td>
<td>100,000</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>Cisco 4500</td>
<td>120,000</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>Cisco 7000</td>
<td>300,000</td>
<td>64</td>
<td></td>
</tr>
</tbody>
</table>
ServerType relation: describes all server types. The following information is given on each server type: operating system, Windows Performance Index, SPECInt92 rating, SPECfp92 rating, and cost.

### Table 9. ServerType Relation

<table>
<thead>
<tr>
<th>Server Type</th>
<th>Operating System</th>
<th>Windows Performance Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pentium P90</td>
<td>Windows NT</td>
<td>207</td>
</tr>
<tr>
<td>486-1 00 DX4</td>
<td>Netware 4.0</td>
<td>149</td>
</tr>
<tr>
<td>486-1 00 DX4</td>
<td>LAN Manager</td>
<td>149</td>
</tr>
</tbody>
</table>

ClientType relation: describes all client types including the Windows Performance Index (WPI) for PC type clients, the SPECfp92 for UNIX clients, and the client cost.

### Table 10. ClientType Relation

<table>
<thead>
<tr>
<th>Client Type</th>
<th>WPI</th>
<th>SPECfp92</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>486-66 DX4</td>
<td>125</td>
<td>-</td>
<td>$3,000</td>
</tr>
<tr>
<td>486-100 DX4</td>
<td>149</td>
<td>26</td>
<td>$3,900</td>
</tr>
<tr>
<td>SUN SPARC5</td>
<td>-</td>
<td>-</td>
<td>$7,800</td>
</tr>
</tbody>
</table>

UserType relation: lists all types of users and the client id they are associated with. It also indicates the percentage of a working day that a user is active using his/her workstation. Note that the user type-client association is one-to-one. Therefore, different user types may have to be created to indicate similar user types associated with different client ids. For example, one cannot have associations such as (administrative, PC0001) and (administrative, PC0002). Instead, one should have (administrative1, PC0001) and (administrative2, PC0002).

### Table 11. UserType Relation

<table>
<thead>
<tr>
<th>User Type</th>
<th>Client</th>
<th>Percent Day (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrative</td>
<td>PC0001</td>
<td>40</td>
</tr>
<tr>
<td>Scientist</td>
<td>UNIX001</td>
<td>50</td>
</tr>
<tr>
<td>Programmer</td>
<td>PC0002</td>
<td>80</td>
</tr>
</tbody>
</table>

3.3 Association Relations

NetPath relation: this relation describes all network paths. A network path describes a path followed by requests from a client to a server going through a sequence of one or more network media and zero or more network interconnection devices. The routing frequency indicates the fraction of requests that leave a client and follow the specific path.

### Table 12. NetPath Relation

<table>
<thead>
<tr>
<th>Net Path Name</th>
<th>Application</th>
<th>Routing Frequency</th>
<th>Client CPU Service Demand (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path 1</td>
<td>E-mail_PC</td>
<td>0.7</td>
<td>0.003</td>
</tr>
<tr>
<td>Path 2</td>
<td>E-mail-PC</td>
<td>0.3</td>
<td>0.003</td>
</tr>
<tr>
<td>Imaging</td>
<td>Imaging</td>
<td>1.0</td>
<td>0.010</td>
</tr>
</tbody>
</table>

Server-ApplAssociation relation: describes all associations between servers and the applications served by them.

### Table 13. Server-ApplAssociation Relation

<table>
<thead>
<tr>
<th>Server</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>SV0001</td>
<td>Video Conferencing</td>
</tr>
<tr>
<td>SV0002</td>
<td>E-mail</td>
</tr>
<tr>
<td>SV0003</td>
<td>Wordprocessor</td>
</tr>
</tbody>
</table>

UserType-ApplAssociation relation: lists the relationships between user types and the applications they execute. The percent application column gives the percent of the time the user is active and executing a given application.

### Table 14. UserType-ApplAssociation Relation

<table>
<thead>
<tr>
<th>User Type</th>
<th>Application</th>
<th>Percent Application (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrative</td>
<td>E-mail</td>
<td>20</td>
</tr>
<tr>
<td>Administrative</td>
<td>Word-processor</td>
<td>80</td>
</tr>
<tr>
<td>Scientist</td>
<td>Video</td>
<td>100</td>
</tr>
<tr>
<td>Programmer</td>
<td>Conferencing</td>
<td>20</td>
</tr>
</tbody>
</table>
NetMedium-NetProtocolAssociation relation lists the network protocols used in each network medium.

<table>
<thead>
<tr>
<th>Network Medium</th>
<th>Network Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>LANSEG0001</td>
<td>IPX</td>
</tr>
<tr>
<td>LANSEG0002</td>
<td>IP</td>
</tr>
<tr>
<td>LANSEG0003</td>
<td>IPX</td>
</tr>
</tbody>
</table>

**Table 15. NetMedium-NetProtocolAssociation Relation**

4. CMWLan USER INTERFACE

The menu system of Excel 5.0 was extended to provide efficient user access to functions for manipulating CMWLan objects and associations, solving the cost and performance model, and viewing the results of the model. Figure 3 shows the extensions to the Insert pull-down menu. The two bottom options allow the user to manipulate CMWLan Associations and CMWLan Objects.

The CMWLan Associations submenu (see Fig. 4) allows the user to associate a network medium-with a network protocol, to create network paths, to associate a server with an application type, and to associate a user type with an application type.

The CMWLan Objects submenu (see Fig. 5) provides the user with options to manipulate applications, clients, client types, network connectors, network connector types, network media, network media classes, network media types, network protocols, servers, server types, and user types.

The action taken by CMWLan for each object depends on whether one is dealing with an object or object type. For example, if an object type is chosen (e.g., Net Medium Type), a worksheet of network media types is activated and the user can enter a new medium type or update existing ones. If an object such as Net Medium is chosen, the worksheet of network media is activated. To enter a new network medium, the user has to specify its type. If the Insert ➔ CMWLan Object ➔ Net Medium Type sequence is selected, a dialog box is activated. The user can then choose from a list box the network medium type of interest, click the OK button, and the selected option will appear in the proper position of the network media worksheet. See Fig. 6 for an example of the dialog box mentioned above.

The View menu was extended with options to display the results of the Cost and Performance models and to view the existing network paths (see Fig. 7). The performance results can be displayed as tables and graphs. The performance result tables include: response times per workload and per device, network media results, device utilization report, waiting time factors per workload and per device, and aggregated workload results.

---

**Figure 3. Extensions to Insert Pull-down Menu**

**Figure 4. CMWLan Associations Submenu**

**Figure 5. CMWLan Objects Submenu**

**Figure 6. Network Medium Type Dialog Box**

**Figure 7. Existing Network Paths View Menu**
Figure 7. Extended View Menu

Figure 8 shows an example of the aggregated workload results indicating how the response time per workload is broken down per type of device.

<table>
<thead>
<tr>
<th>Client Type</th>
<th>Aggregated Response Time Per Type of Device (msec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrative Email</td>
<td>0.048</td>
</tr>
<tr>
<td>Administrative Email</td>
<td>0.141</td>
</tr>
<tr>
<td>Administrative Workstation</td>
<td>0.173</td>
</tr>
<tr>
<td>Administrative Workstation</td>
<td>0.173</td>
</tr>
<tr>
<td>Scientific Imaging</td>
<td>0.182</td>
</tr>
<tr>
<td>Scientific WWW</td>
<td>0.182</td>
</tr>
<tr>
<td>Administrative Email</td>
<td>0.048</td>
</tr>
<tr>
<td>Administrative Email</td>
<td>0.141</td>
</tr>
<tr>
<td>Administrative Email</td>
<td>0.141</td>
</tr>
</tbody>
</table>

Figure 8. Aggregated Response Times

5. PERFORMANCE PREDICTION MODEL

The performance prediction model that supports CMWLan is an open multiclass queuing network (QN) model [Menasce94] automatically built from the information provided by the user on the various network objects. Each class in the QN model is associated to a pair (user type, application). For example, assume that there are three user types: scientist, administrative, and data-entry clerk. Each user type is associated to one and only one group of client workstations on the same network segment. Assume that the applications executed by user type Administrative are E-mail, spreadsheet, and wordprocessor. Then the following three classes will be generated by CMWLan: (Administrative, E-mail), (Administrative, Spreadsheet), and (Administrative, Word-processor).

A customer in this QN model is a transaction generated by users of a certain type when executing a specified application. The definition of a transaction varies according to the type of application. For an E-mail application, a transaction could be defined as sending or receiving a message. A basic building block of such a QN is a network path that consists of a client, a sequence of network media connected by network connectors, and a server. Figure 9 shows the portion of the QN that represents a network path. The user represents the source of requests to the client. The client is represented in the QN as a delay device since client nodes are assumed to be single user computers.

QN models require two types of parameters: workload intensities and service demands [Menasce94]. CMWLan computes these parameters from the relational database (see Section 3) that supports the tool. Ideally, the values in the database have been obtained through actual service demand measurements of applications on the network under consideration. However, taking measurements in a LAN environment can be challenging due to the heterogeneous nature of the environment and the lack of proper measurement tools. For this reason, CMWLan's database provides service demand measures obtained by the developers in a controlled benchmarking environment for various typical applications; for instance, Word Perfect 6.0 running on a typical Intel 486 platform. These benchmark measures are selectable as surrogate service demands when actual measurements are unobtainable. These values are adjusted by CMWLan, with the use of proper benchmark scaling figures (e.g., SPECmarks, Windows Performance Index) stored in CMWLan's database, to account for varying performance characteristics of various client and server platforms. When actual measurements are possible, they should be obtained and used in model building to assure the most valid and accurate results. Surrogate measures are viewed as acceptable substitutes until measurement capabilities are implemented.

Similarly, industry standard cost rules of thumb for installing, maintaining, and managing a network are used by CMWLan to compute startup and operational costs when actual cost data on the various system components and support staff are unknown to the user.

5.1 Parameter Derivation for CMWLan

The notation $R[k, c]$ is used to refer to the value in the row of database relation $R$ selected by using "$k$" as the value of the primary key and in column "$c$". For example, considering the example relations of Section 3, it follows that $\text{UserService [Scientist, Client]} = \text{UNIXOOI}, \text{NetPath [Path2, Routing Frequency]} = 0.3,$ and $\text{NetMediumType [Ethernet-S, Bandwidth(Mbps)]} = 10$

The derivation of workload intensity parameters and service demand parameters is given in the following subsections.
5.1.1 Workload Intensity Parameters

Let a class (workload) \( r \) be defined by the pair \( <u, a> \) where \( u \) is a user type and \( a \) is an application. Then, the client id, \( Cid (u) \), associated with user type \( u \), is given by

\[
Cid (u) = \text{UserTypes} \ [u, \text{Client}]
\]

The average rate \( L(a) \) at which application \( a \) generates requests is given by

\[
L(a) = \text{Applications} \ [a, \text{Avg. Request Rate}]
\]

The number of client workstations, \( NC(u) \), associated with user type \( u \) is

\[
NC (u) = \text{Clients} \ [Cid(u), \text{Number in the Group}]
\]

Finally, the arrival rate \( ArrRate (r) \) of requests of class \( r = <u, a> \) can be computed as

\[
ArrRate (r) = NC (u) \times \text{UserType-AppIAssociation}[u, \text{Percent Day}] \times L(a)
\]

Using the example database of Section 3, the following values are obtained:

\[
Cid (\text{Administrative}) = \text{PC0001}
\]
\[
L (\text{E-mail}) = 0.001 \text{ req/sec}
\]
\[
NC(\text{Administrative}) = 45
\]
\[
\text{UserType-AppIAssociation[Administrative, Percent Day]} = 0.5
\]
\[
\text{ArrRate (<Administrative, E-mail>)} = 45 \times 0.5 \times 0.001 = 0.0225 \text{ req/sec}
\]

5.1.2 Service Demand Parameters

The service demand \( D(i, r) \) for class \( r \) at device \( i \) is computed by CMWLan from the values in the relational database. The basic principle for obtaining \( D(i, r) \) for all devices except the CPU at the server is the relationship \( D(i, r) = V(i, r) \times S(i, r) \) where \( V(i, r) \) is the average number of visits that requests of class \( r \) make to device \( i \) and \( S(i, r) \) is the average service time of requests of class \( r \) per visit to device \( i \). To understand how the values of \( V(i, r) \) and \( S(i, r) \) are obtained, it is necessary to know the role that a network path plays in satisfying a request of a given class. A network path consists of a sequence of one or more network media and zero or more network connectors. A class \( r = <u, a> \) can be associated with a client by the function \( Cid (u) \). On the other hand, application \( a \) may be served by one or more servers in the set \( S(a) = \{S_1, \ldots, S_n\} \). Figure 10 gives an example of the relationship between user types, applications, servers, and network paths. In this figure, an Administrative user type is associated with client Client001. The administrative user runs several applications including E-mail and Spreadsheet shown in the figure. The E-mail application is served by SVO01 that acts as the mail server and the Spreadsheet application can be served by either SVO01 or SVO02. There are two network paths: Client001 \( \Rightarrow \) LANSEGO01 \( \Rightarrow \) CONN001 \( \Rightarrow \) LANSEGO05 \( \Rightarrow \) SVO02, and Client001 \( \Rightarrow \) LANSEGO02 \( \Rightarrow \) CONN002 \( \Rightarrow \) LANSEGO07 \( \Rightarrow \) SVO01. According to the routing frequencies shown in Fig. 10, 40 percent of the traffic leaving Client001 traverses the first path, while the remaining 60 percent follows the second path.

Let \( P (c, a) = \{P_1, \ldots, P_m\} \) be the set of paths from client \( c \) to the servers in \( S(a) \). The routing frequency \( RFreq (p) \) of a path \( p \) is given by

\[
RFreq (p) = \text{NetPath} \ [p, \text{Routing Frequency}]
\]

The computation of the service times \( S(i, r) \) varies according to the type of device as described later. The computation of the visit ratios \( V(i, r) \) are better understood with the help of Figs 9 and 10. The average visit ratio \( V(m, r) \) of class \( r = <u, a> \) to network medium \( m \) in a path \( p \) from client \( Cid (u) \) to a servers in \( S(a) \) is given by

\[
V (m, t) = 2 \times RFreq (p)
\]

since each request has to visit the network medium twice (first on its way to the server and then when the reply is sent from the server back to the client). Note that the customer in the QN model is a request. Clearly, a request may generate several messages and each message may have to be broken up into several packets transmitted by the network medium. This is considered in the...
calculation of the service times at the various network media and network connectors.

![Diagram of Figure 10: Relations of User Types, Clients, Applications, Servers, and Network Paths.](image)

Similarly, the average visit ratio $V(c,r)$ of class $r = <u,a>$ to network connector $c$ in a path $p$ from client $Cid(u)$ to a server in $S(a)$ is given by

$$V(c,r) = 2 \times RFreq(p)$$

Let $SDisk(s)$ be the disk at server $s$. Then, for each server $s$ in $S(a)$, the average visit ratio $V(SDisk(s),r)$ for class $r = <u,a>$ is given by

$$V(SDisk(s),r) = \text{Applications}[a, \text{No. Bytes Read/Written in Kbytes}] / \text{Servers}[s, \text{Disk Block Size}]$$

The average service time $S(SDisk(s),r)$ at disk $SDisk(s)$ at server $s$ is computed as follows:

$$S(SDisk(s),r) = \text{Servers}[s, \text{Disk Seek Time}] + \text{Servers}[s, \text{Disk Latency}] + \text{Servers}[s, \text{Disk Block Size}] / \text{Servers}[s, \text{Disk Transfer Rate}]$$

Let $SCpu(s)$ be the CPU at server $s$. Then, for each server $s$ in $S(a)$, the service demand $D(SCpu(s),r)$ for class $r = <u,a>$ is given by

$$D(SCpu(s),r) = \text{Applications}[a, \text{CPU demand at server}] / PSF(s)$$

where $PSF(s)$ is a performance speedup factor used to scale the performance of one server relative to the server taken as the basis for measurements in the application. This factor may use the Windows Performance Index (WPI) for Windows applications or SPECmarks for scientific applications. An example of the computation of the performance speedup factor using WPI is given by

$$PSF(s) = \frac{\text{ServerType}[s, \text{Windows Performance Index}]}{\text{ServerType}[\text{BaseServer}(a), \text{Windows Performance Index}]}$$

where $\text{BaseServer}(a)$ is the base server for application $a$, given by

$$\text{BaseServer}(a) = \text{Applications}[a, \text{Base Server}]$$

To compute the service demands at the network media, we need to compute the average arrival rate of packets to the medium as well as the average packet size. The following sequence of calculations lead to these values. Each request of class $r = <u,a>$ generates a number $NMessages(r)$ to the server computed as

$$NMessages(r) = \text{Applications}[a, \text{Avg. No. Messages per Request}]$$

Let $M(p) = \{M_1, \ldots, M_k\}$ be the set of network media in path $p$ and let $\text{Conn}(p) = \{C_1, \ldots, C_n\}$ be the set of connectors in the same path. The maximum packet size $\text{MPS}(m)$ on network media $m$ is:

$$\text{MPS}(m) = \text{NetMediumType}[\text{NetMedia}[m, \text{Net Medium Type}], \text{Max Packet Length}]$$

Each message sent from a client $Cid(u)$ to a server $S_j$ along a path $p$ may have to be broken up into smaller sized packets to be transported by the network media in the path from the client to the server. The average number of packets $NP(x,m)$ generated by a message of size $x$ on network medium $m$ is given by

$$NP(x,m) = \lceil x / \text{MPS}(m) \rceil$$

where $\lceil z \rceil$ stands for the smallest integer greater than or equal to $z$. So, the average number of packets $NPtoS(a,m)$ generated to the server by a request of application $a$ on medium $m$ is

$$NPtoS(a,m) = NP(\text{Applications}[a, \text{Avg. Msg. Size to Server}], m)$$

and the average number of packets $NPfromS(a,m)$ generated from the server by a request of application $a$ on medium $m$ is

$$NPfromS(a,m) = NP(\text{Applications}[a, \text{Avg. Msg. Size from Server}], m)$$
So, the average arrival rate $PArrRate(r,m)$ of packets of class $r = \langle u, a \rangle$ to network medium $m$ is:

$$PArrRate(r,m) = ArrRate(r) \cdot NMessages(I) \cdot (NPToS(a) + NPfromS(a))$$

The average packet size $AvgP(x,m)$ generated by a message of size $x$ on network medium $m$ is:

$$AvgP(x,m) = x \cdot NP(x,m)$$

The average packet size $AvgPW(r,m)$ for class $r$ on network medium $m$ is given by

$$AvgPW(r,m) = \frac{AvgP(Applications[a, Avg. Msg. Size to Server], m) \cdot NPToS(a,m) + AvgP(Applications[a, Avg. Msg. Size from Server], m) \cdot NPfromS(a,m)}{(NPToS(a,m) + NPfromS(a,m))}$$

Finally, the service demands for all devices for all classes are computed as follows:

For every class $r = \langle u, a \rangle$ do

For every path $p$ such that the client for $p$ is the client associated to user $u$ do

compute network media service demands for all media in $p$

compute network connectors service demands for all connectors in $p$

compute client service demands for the client in $p$

compute server cpu service demand for the server in $p$

compute disk service demand for the server in $p$

EndFor

EndFor

6. CONCLUDING REMARKS

Object-orientation is a powerful tool for the design and abstraction of performance tools (see [Salsburg94]). This paper presented the object-oriented design, database design, user interface, and performance model of CMWLan—a tool for capacity planning in LAN environments. CMWLan automatically builds an analytic model of a LAN environment from the network specifications. The tool was developed by the authors to support the implementation of a capacity planning methodology for LAN environments in a large government organization. The tool is fully operational and is nearing deployment.

References


