

# An Object Oriented Toolkit For Implementation Oriented Modeling Of Real-Time Multimedia Communications

Pallapa Venkataram, Sandeep Kumar and Sankar Singha

Dept. of Electrical Communication Engineering

Indian Institute of Science

Bangalore, 560 042, India

(E-Mail : pallapa, sandeep, sankar@ece.iisc.ernet.in)

## 1 Introduction

A comparative performance evaluation of these protocols and architectures necessitates theoretically evaluating and experimentally comparing different architectures, protocols, topologies and technologies of networks under the same operating conditions. A toolkit providing this facility may therefore be of great importance. In this work a framework of traffic control to accommodate multimedia connections in a network has been proposed. An approach based on Resource Reservation Protocol has been implemented to reserve resources. It follows many novel design principles such as *receiver initiated reservation, separating reservation from packet filtering, maintaining soft states in the network, modularity*, etc.

Corresponding to different traffic models and performance measures many scheduling policies as well as admission control schemes have been proposed. Ferrari and Verma proposed a scheme for establishing multimedia connections [1]. When a connection is to be established, suitable delay upper bounds are assigned to intermediate nodes of the routing path. If the end-to-end cell delay does not go beyond the upper bound of the connection, the connection is accepted. Chen and Liu in [2] suggested a Round-Robin/Priority Scheduling algorithm, involving

time framing strategy, to provide multiple delay bounds in each node. However, time framing strategies suffer from the dependencies they introduce between the queuing delay and the granularity of bandwidth allocation [3]. RCSP queuing [4] overcomes this by separating the rate control and delay control functions. We have adopted the inter and intra class bandwidth regulation principle from [5]. However, in that paper, no lower bound has been specified for the connections. The toolkit has been designed on the lines of the work given in [6, 7]. However, in these works no admission control mechanism has been provided.

The rest of the paper is organized as follows. Section 2 describes about the suggested schemes and protocols. Section 3 discusses the proposed toolkit. Simulation and results have been given in section 4. Finally, we make our conclusions in section 5.

## 2 Suggested Schemes and Protocols used in the toolkit

We, here, briefly describe the following protocols and schemes used in the proposed toolkit.

### 1. The Resource Reservation Architecture

The resource reservation architecture implemented in the simulation toolkit to capture many novel design features of RSVP [8]. It makes receivers responsible for reserving resources according to their specific requirements.

## 2. Bandwidth Regulation

Many video applications are scalable because of the human tolerance to degradation in picture quality, frame loss and end-to-end latency provided the quality of playout is over some perceptual threshold. Scalability can be translated to a reduction in resource requirements for the corresponding connections. Thus tolerance to degradation of audio/video quality allows encoding parameters to be modified to yield lower data rates. Hence, the application can specify a range of bandwidth requirements to the network during connection establishment. The maximum requirement is for best quality while the minimum is for acceptable quality of service.

3. **Delay Based Priority Scheduling** A network architecture that aims at real time applications should incorporate mechanisms for providing required quality of service to these applications. To provide such quality of service the network must reserve a part of its resources for the application. Therefore, it should incorporate a resource reservation architecture. Now just reserving resources does not ensure that the application gets the required quality of service; the network should also ensure that the reserved resources are effectively utilized towards providing the required quality of service to the application.

## 4. Connection Establishment Protocol

In order to set up a connection, a three way handshaking protocol has been suggested. First, the client sends a connection request

control packet (containing all its QOS requirements) to the server. On its way, the packet sets up filter states in the intermediate nodes after passing the bandwidth regulation test, thus reserving resources. The newly calculated service rates (due to bandwidth regulation) of all the connections through these nodes are stored temporarily in the cache to be later picked up by the client refresh packets.

## 3 The Object Oriented Toolkit

An extensive Simulation Toolkit incorporating Resource ReSerVation Protocol (RSVP) and Delay Based Static Priority Scheduling Policy is implemented using Object Oriented Programming Methodology using C++ [9]. The toolkit offers a flexible interface in the form of Network Description Language (NDL) for specifying the network components, component parameters, network topology and simulation commands including logging of various system events and statistics to disk files. The toolkit is fairly easily modifiable and extendible to simulate various network environments with minimal of recoding because of the powerful OOP features such as *data abstraction*, *information hiding*, *inheritance*, *polymorphism*, *code sharing*, etc., effectively utilized in the design.

### 3.1 Design Principles Of The Toolkit

The toolkit is designed following some clearly defined principles such as:

1. *Implementation Oriented Modeling*

One of our main aims is to know and to explore how the things go about in a real network and how the distributed algorithms work in close cooperation with each other.

2. *Flexible User Interface*

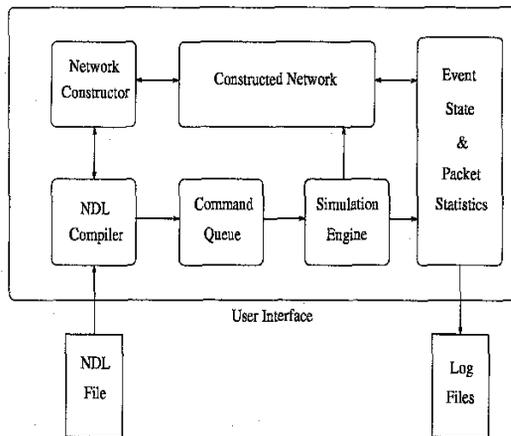


Figure 1: Structure of the toolkit

Providing a flexible user interface was also our main consideration in the toolkit design. It was decided that the user of the toolkit should be given enough flexibility to be able to describe network components, their properties, network topology, simulation command, etc., along with powerful see through capabilities like event, component/system state and packet statistics logging capabilities. Towards all these goals the toolkit offers a flexible language interface called Network Description Language (NDL).

### 3. Easy extensibility

Easy extensibility has been another major design principle, as expected from any good OOP design. The toolkit offers a flexible user interface for describing network properties and topology, but it is not meant for simulating any network.

## 3.2 Structure of the Toolkit

The simulation toolkit is made up of several distinct components each with their specific functions to perform in the overall system. The structure of the toolkit is shown in figure 1. The input to the toolkit is in the form of a NDL file

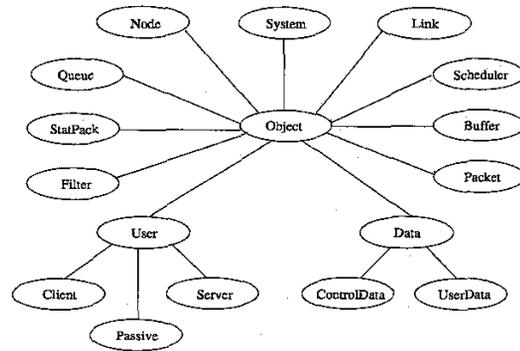


Figure 2: The Object Class Hierarchy

which contains the specifications of the network topology, simulation commands, QOS requirements of the applications, etc. A YACC generated parser compiles the NDL file constructing the various network components giving them the properties as described in the input file, and then it plugs them together as specified by the network topology to form the desired network.

## 3.3 The Network Object Classes

Various network entities are modeled as language objects, that form a well structured class hierarchy is shown in figure 2. A typical node has the structure shown in figure 3. The major classes in the hierarchy and their functions are described in detail in [10].

## 3.4 Network Description Language

The Network Description Language (NDL) provides flexible user interface to the simulation toolkit. It enables the user of the toolkit to describe network components, their properties, network topology, simulation command, etc. along with powerful *see through* capabilities like event, component/system state and packet statistics logging capabilities. The toolkit uses a YACC generated parser as front end to compile NDL file. It constructs the network described

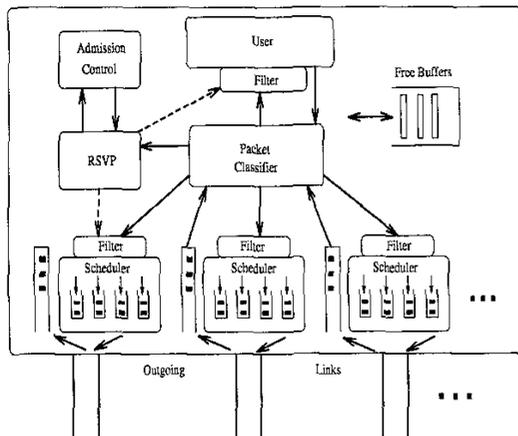


Figure 3: Structure of a node

in the NDL file and then runs the constructed system according to the simulation commands. In NDL there are commands available for specifying various network components, their properties, their interconnections and the simulation commands to be run on the constructed network. There are provisions to run the constructed network for any length of time and log network messages, various events, state of network components and packet statistics as and when required to disk files that can be opened and closed at will through NDL commands. Network Description Language is a context free language based upon LALR(1) grammar. A NDL file is a text file that can be edited using any text editor. NDL is case insensitive and may contain C/C++ style comments. The front end compiler gives extensive diagnostic messages in case of *syntax and network errors* encountered during compiling input NDL file.

## 4 Simulation

### 4.1 Simulation Environment

The multimedia applications chosen here, constitute of a number of clients, servers and passive

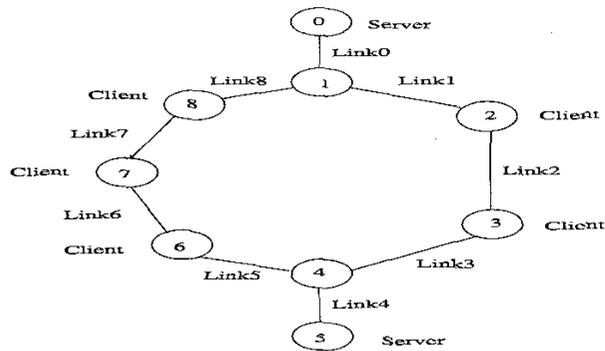


Figure 4: Experimental topology

nodes and involve interaction among these entities. The underlying network is supposed to provide the required throughput and delay guarantees. The application layer entities involved in the simulation try to loosely model the behaviour of the respective entities in a real world environment. Some of them are: **Passive**, which is a user entity does not participate in the ongoing activity, **Client** and **Server** entities.

### 4.2 Network Topology

A ring network topology, as shown in figure 4, has been considered for the experiment. The network and the simulation commands to run on the constructed network are coded in NDL that are used as inputs to the simulation toolkit. Each link has a total capacity of 10Mb/s. Out of this total bandwidth, 0.2Mb/s is used by the resource reservation protocol for sending control signals. The remaining 9.8Mb/s is used to transfer the various stream packets from servers to clients.

### 4.3 Experimental Results and Discussion

The experiment has conducted with two servers and five clients asking for video and audio streams with QOS requirements shown in table 1. As can be seen, link 0 and link 4 are the bottleneck links. Requests from both halves of the

Client Node Number	Video		
	Max Rate (Mb/s)	Min Rate (Mb/s)	Delay Bound (s)
2	4.0	5.0	0.0025
3	3.0	4.0	0.0035
6	0.85	1.0	0.0015
7	3.0	3.5	0.0009
8	4.0	4.5	0.0045

Table 1: QOS requirements of the clients

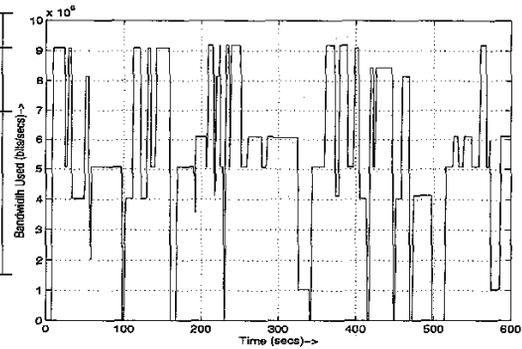


Figure 6: Bandwidth utilization of Link4

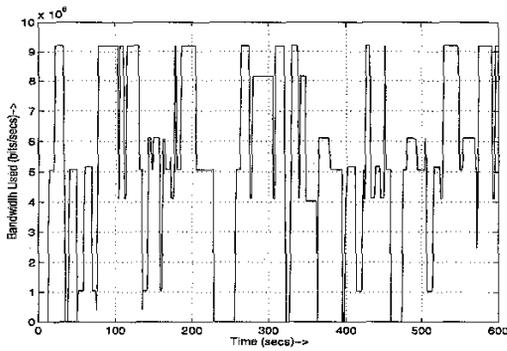


Figure 5: Bandwidth utilization of Link0

ring converge at these links and hence have the chance of getting rejected. Figures 5 and 6 show the bandwidth utilization of these two links.

Figures 8 and 9 shows the end-to-end delays of the stream packets for node 6 and figure 10 demonstrates the connection establishment overhead.

## 5 Conclusion

In our work we have proposed a framework of toolkit with many novel design principles such as *receiver initiated reservation, separating reservation from packet filtering, maintaining soft states in the network, modularity, etc.*

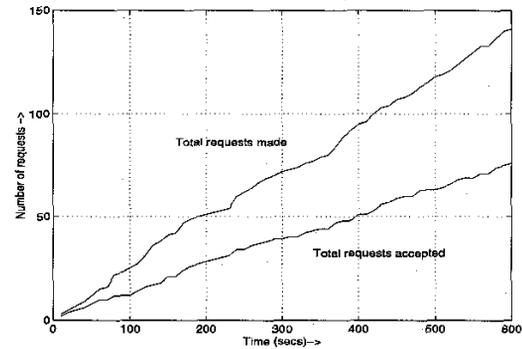


Figure 7: Request statistics

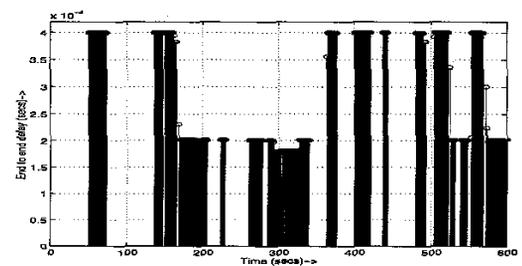


Figure 8: End to end delay of video packets for client6

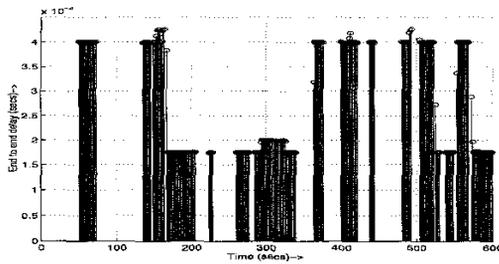


Figure 9: End to end delay of audio packets for client6

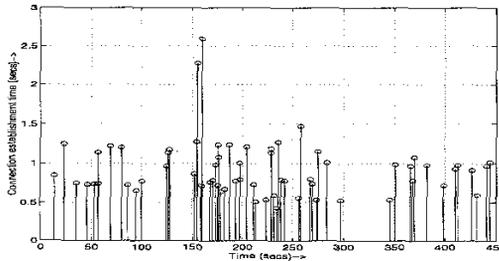


Figure 10: Connection establishment overhead

The results were found to be quite encouraging. The proposed schemes and protocols work satisfactorily towards their design goals of supporting real time multimedia applications by providing guaranteed throughput and bounded delay. We also found that the connection establishment time is bounded within a range of values even when the network load fluctuates.

## References

- [1] Domenico Ferrari and Dinesh Verma, "A scheme for real-time connection establishment in wide-area networks", In *IEEE Journal on Selected Areas in Communication*, Vol-3, number-8, Apr, 1990, pp. 368-379.
- [2] Uan-Jiun Lju and Wen-Tsuen Chen, "A feasible framework of traffic control on an ATM Wide-Area Network", In *Computer Networks and ISDN Systems*, Vol-27, 1994, pp. 67-84.
- [3] K. Sriram, "Methodologies for bandwidth allocation, transmission scheduling and congestion avoidance in broadband ATM networks", In *Proc. GLOBECOM'92*, pp. 1545-1551.
- [4] Hui Zhang and Domenico Ferrari, "Rate-Controlled Static-Priority Queueing", In *Proc. INFOCOM'93*, pp. 227-236.
- [5] Ian F. Akyildiz and Debapriya Sarkar and Jorg Liebeherr, "Bandwidth Regulation of Real-Time Traffic Classes in Internetworks", In *Computer Networks and ISDN Systems*, Vol-28, 1996, pp.855-872.
- [6] Sandeep Kumar and P. Venkataram, "An Object Oriented Network Simulation Testbed for Real-Time Multimedia Applications", In *Proc. 12th International Conference on Computer Communication, 1995*, Vol. 1, August, 1995.
- [7] Sandeep Kumar, "An Object Oriented Network Simulation Testbed for Real-Time Multimedia Applications", In *IISc. M.E. Project Report*, 1995.
- [8] Lixia Zhang and Scott Shenker and Darnial Zappala, "RSVP: A new Resource ReSerVation Protocol", In *IEEE Network Magazine*, September, 1993, pp. 8-17.
- [9] Stanley B. Lippman, "C++ Primer", In *2nd Edition, Addison-Wesley Publishing Company*, 1995.
- [10] Sankar Singha, "An Object Oriented Toolkit for Implementation Oriented Modeling of Real-Time Multimedia Communications", In *IISc. M.E. Project Report*, June, 1997.