Multimedia System and Architecture

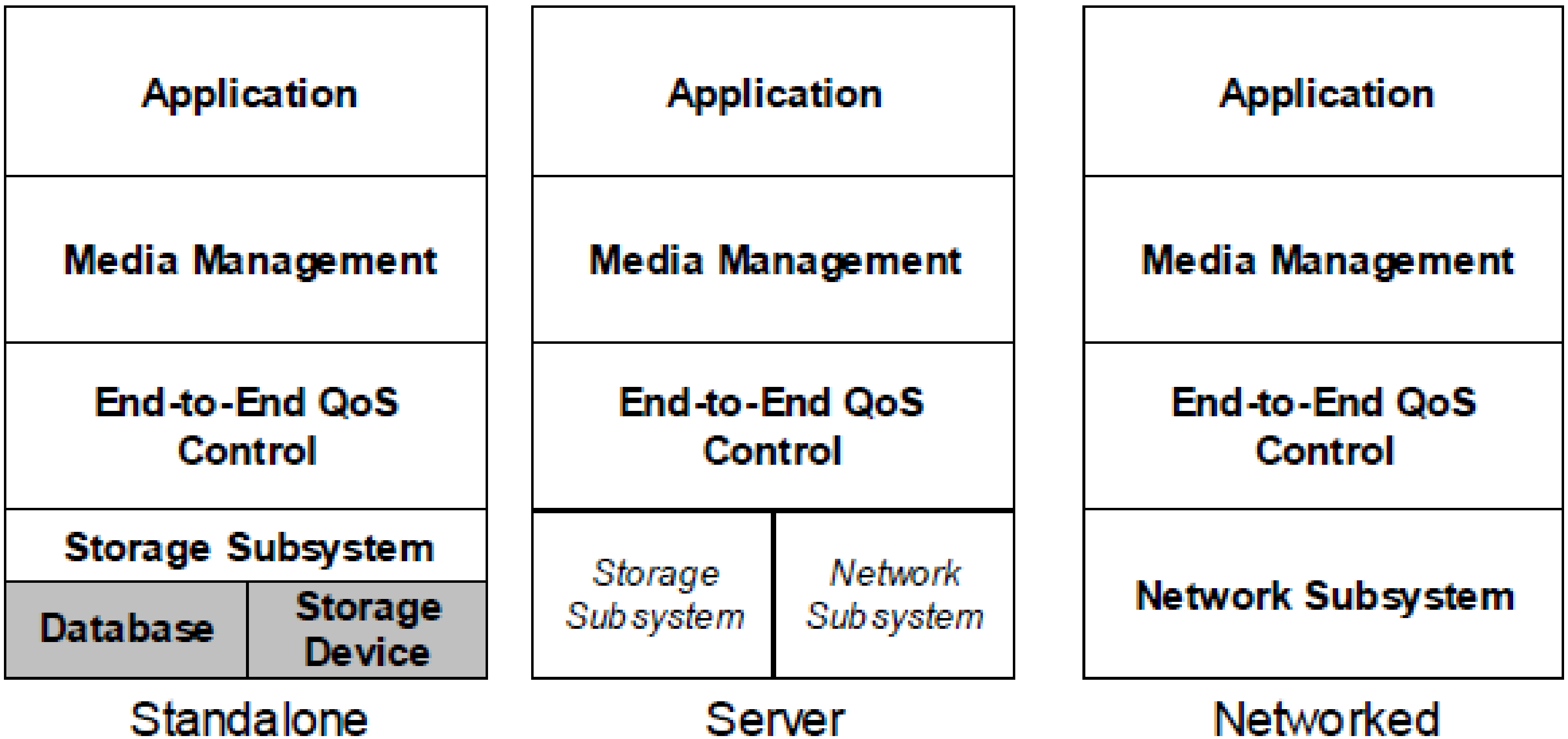
Components of Multimedia System

- Categorised as 'live' or 'orchestrated'

- *Live applications* involve interaction among users. Eg. Video Conferencing
- *Orchestrated applications* reproduces the data that was previously generated and stored in some medium. Eg. Pay per view movies.
- In order to satisfy the requirements of these categories various Multimedia Architectures are used.

- In order to satisfy the requirements of these categories of applications, various Multimedia Architectures are used.
- Multimedia systems can either be standalone , server or networked.
- *Standalone Multimedia System*: consist of storage subsystem , which is a database and storage devices where media content resides.
- *Server Multimedia System*: consist of both storage subsystem and networked subsystem in levels.
- *Networked Multimedia System*: consist of both networked subsystem.

Real Time Multimedia Architecture



Layered architecture

Lower layer provide functionality to upper layers.

- Storage Subsystem and Network Subsystem (Layer 1) :->
  These services are central to multimedia system and is normally part of OS services.

- End-To-End QoS (Layer 2) :-> This layer deals with maintaining connection between source of the multimedia content and the destination.
  For Standalone system :
  Source -> multimedia database
  For Networked system :
  Source -> multimedia server
  QoS (Quality of Service) comprises of delay, jitter, and packet loss probability.

- Media Management (Layer 3) :->

  It provides general services like media synchronization and media stream management . Also it ensures temporal and spatial relationships between media streams.

- Application Layer (Layer 4) :->

  Interface with the user, to provide access and control to the presentation.

DISTRIBUTED MULTIMEDIA SYSTEM(DMMS)

TABLE OF CONTENTS

INTRODUCTION

- A **distributed system** is designed to support the development of applications and services which have a physical architecture consisting of multiple autonomous processing elements that do not share primary memory but cooperate by sending messages over a communication network.

INTRODUCTION

Distributed multimedia systems consist of multimedia databases, proxy and information servers, and clients, and are intended to for the distribution of multimedia content over the networks.

CHARACTERISTICS

- Delivering the streams of multimedia data
  - Audio samples, Video frames
- To meet the timing requirements
  - QoS (Quality of Service)
- Flexibility (adapting to user needs)
- Availability
- Scalability

REQUIREMENTS OF DMMS

- **Providing support for continuous media types, such as audio, video and animation.**

The introduction of such continuous media data to distributed systems demands the need for continuous data transfers over relatively long periods of time.

High bandwidth is required.

- **The second requirement of distributed multimedia applications is the need for sophisticated quality of service (QoS) management**.

Ensuring that the service requests should be met.

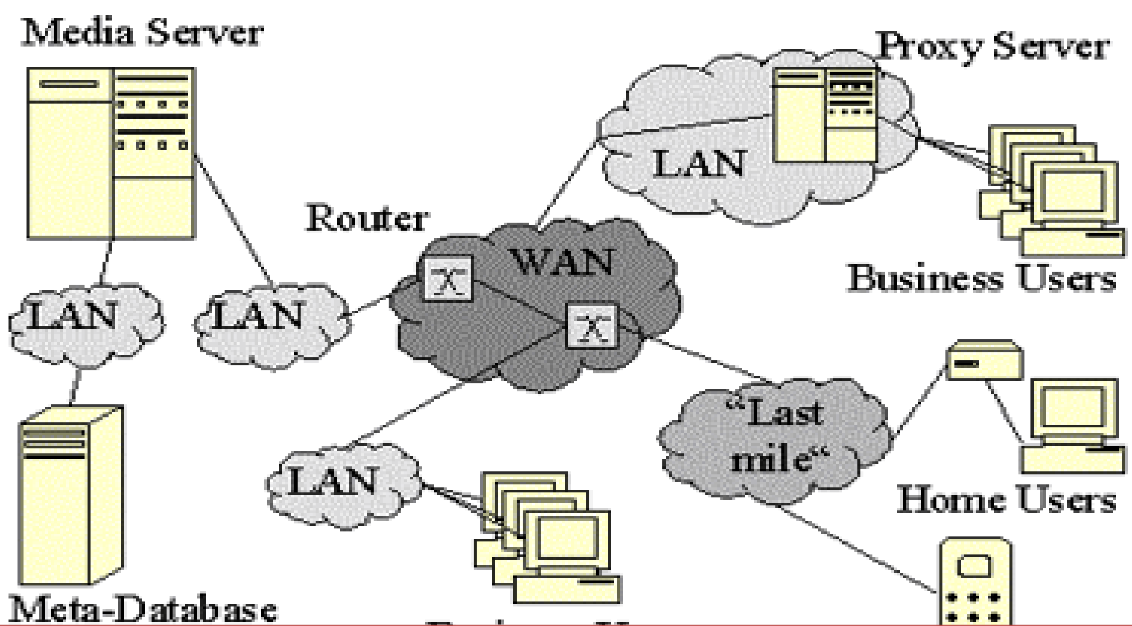
- **Supporting multiparty communications.**

Many distributed multimedia applications are concerned with interactions between dispersed groups of users.

E.g. remote conference application.

BASIC ARTCHITECTURE

- Database
- Proxy/information servers
- Clients
- Wired or wireless networks



FACTORS THAT AFFECT THE SYSTEM

- **Server bandwidth**-Adequate bandwidth of the network to support media streams
- **Cache space**- Sufficient cache space needed for the fast delivery
- **Number of copies**-Adequate number of copies of movies at the multimedia databases (MMDs)
- **The number of clients**- Increasing number of clients increase the server work load

APPROACHES

- Proxy-based approach
- Parallel or clustered servers approach
  - Varies depending on clip duration, number of clients, bandwidth available, etc
- Caching- Caching allows nodes to quickly fetch the required documents without incurring the need to contact the original host.

QUALITY OF SERVICE (QoS)

- DMMS are real-time systems as data must be delivered on time
- "Acceptable" service is measured by:
  - **Bandwidth** (Throughput)
  - **Latency** (Access time)
  - **Data Loss Rate** (Acceptable loss ratio)

QoS MANAGEMENT

- “QoS Management”
  - Process of managing resources to meet the Acceptable service criteria.
- Resources include:
  - CPU / processing power
  - Network bandwidth
  - Buffer memory(on both ends)
  - Other factors affecting communication

WHY WE NEED Qos

1. We may get the wrong video
   – Bad search facilities
2. We may get the wrong quality-variation
   – E.g too high quality level for a PDA
3. We may get the proper variation in bad quality
   – E.g jitter is too high
4. We may get the video at the wrong time

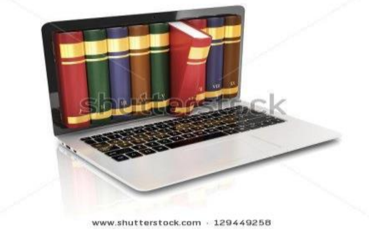
QoS MANAGERS

- Software that runs on network nodes which have two main functions:
  - *QoS negotiation*: get requirements from apps and checks feasibility versus available resources.
  - *Admission control*: If negotiation succeeds, provides a "resource contract" that guarantees reservation of resources for a certain time

WAYS TO ACHIEVE QoS

- Buffering (on both ends)
  Compression
  - More load on the nodes, but that is fine
- Bandwidth Reservation
  Resource Scheduling Traffic
  Shaping
- Flow Specifications
  Stream Adaptation

APPLICATIONS OF DMMS



DIGITAL LIBRARIES

DISTANCE LEARNING

**CONFERENC
E**

AUDIO STREAMING

VIDEO STREAMING

**E-
COMMERCE**

AND

CONCLUSION

Let us realize the fact that there is still a long way to go in DMS (Distributed Multimedia Systems). I presented an overview of DMS, discussed the major requirement of QoS Management.

Thus summarized all the primary issues leading to the basics of DMS.

QoS architecture for multimedia

Outlines

- Motivation and problem statement
- Goals
- Current QoS architectures
    - Static
    - Adaptive
- Comments on the previous work
- The proposed approach
- Experimental results
- Conclusion

Motivation and Problem statement

- A feedback mechanism is a tool that provides the system with information about the state of the network for adaptation and management purposes .
- Why do we need a feedback mechanism?
  - To detect the state of the network
  - To detect the status of the current applications
  - To decide wither a new flow should be accepted to the media or not
  - To control the congestion level
  - To guarantee the required QoS

Motivation and problem statement (2)

- The characteristics of Multimedia:
  - QoS renegotiation
  - Tolerate higher delay, higher packet drop rate, and lower bandwidth
  - They do not tolerate high jitter.
- In a client-server based applications, the server might generate hundreds or southlands of packets to support the client request. Therefore, using packet-based feedback mechanism is not accurate

Goals

- Designing a QoS-aware feedback mechanism which allows multimedia applications to renegotiate their required QoS as a respond to network conditions.
- Desired features:
  - Accommodates multimedia applications.
  - Improves QoS.
  - decreases control overhead and management complexity.
  - Improves service revenue

Current QoS approaches

- Static:
  1. DiffServ: Differentiated service Networks
  2. IntServ: Integrated service networks

Dynamic:
  1. SWAN: Stateless Wireless Ad-hoc Networks
  2. AQuaFwain:

DiffServ

- Features:
  - Per-hop and stateless architecture
  - It is a priority-based architecture
  - service aggregation and classification
  - End-to-end load control
- Advantages:
  - Scalable
  - Simple and has less control overhead
- Disadvantages:
  - Not an end-to-end architecture
  - Does not guarantee required QoS

IntServ QoS

- Features:
  - Per-flow state management
  - Resource-reservation
  - End-to-end load control
- Advantages:
  - Better QoS guarantee
  - Has call admission and rejection capabilities
  - Can treat flows individually
- Disadvantages:
  - Non-scalable
  - Excessive resource management

IntServ QoS (2)

- Example of Load control schemes:
  - Using utilization target to control the traffic load
  - Using information about incoming flow bandwidth
  - Using the probability that the needed resources exceeds the available resources to decide wither a flow should be admitted or not

SWAN

- Stands for Seamless Wireless ATM Networks
- supports multimedia applications over wireless ATM networks
- Connects heterogeneous networks.
- Uses MAC delay information as a feedback mechanism.

2- AQuaFWiN

- a QoS architecture which supports multimedia applications over wireless networks
- Uses packet probing to capture the path characteristics and send them to applications as a feedback.

Other examples for feedback approaches

- Using link utilization as a feedback information
- Using packet drop rate
- Mobility aware routers use mobility information as a feedback

Shortcomings of the previous work

- Using information for feedback:
  - The majority of packets might be generated by the server not the client
  - It will send millions of feedback messages comparing to flow-based feedback
  - If a given session or flow is broken into 2500 packets, that will require 2500 accounting records or a ten-minute call might require thousands of records.
- Using link information as a feedback does not reflect the actual processed traffic

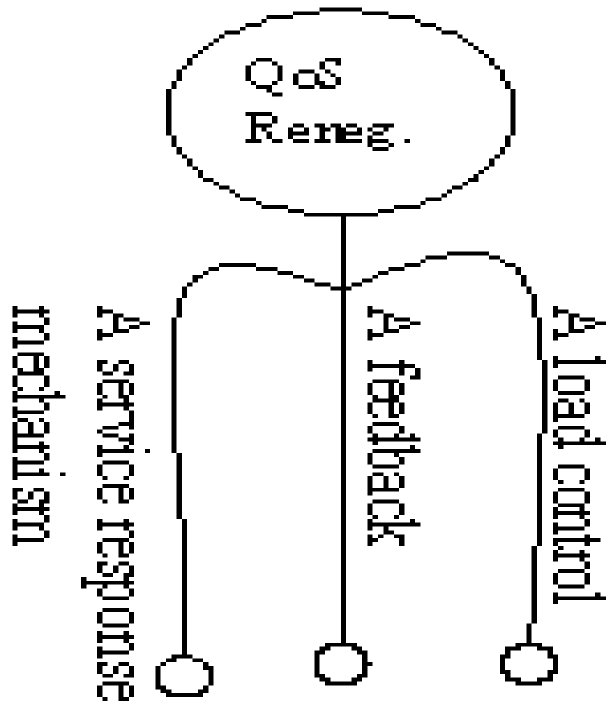
The proposed adaptive QoS scheme

- Features:
  - Uses call rejection notification as a feedback mechanism to capture the path characteristics.
  - Dynamic and allows QoS renegotiation.
  - It uses periodical resource estimation.
  - It simplifies traffic monitoring and QoS management.

The proposed adaptive QoS scheme (2)

- This work emphasizes the importance of flow-based traffic management and monitoring

Components of a QoS renegotiation architecture



Definitions

- **Feedback Mechanism:** using information about packet delay, packet drop rate, and link utilization level to detect the system sate.
- **Load control schemes:** the network load is controlled by using a control tool such as utilization targte, bounded delay, or packet drop target
- **Service response schemes:** services respond to system status by encoding, compression, or renegotiating the required bandwidth.

Assumptions

- Multimedia applications can renegotiate bandwidth:
  - From 8KbPS to 128kbps for voice.
  - from 29kbps to 1500kbps for video.

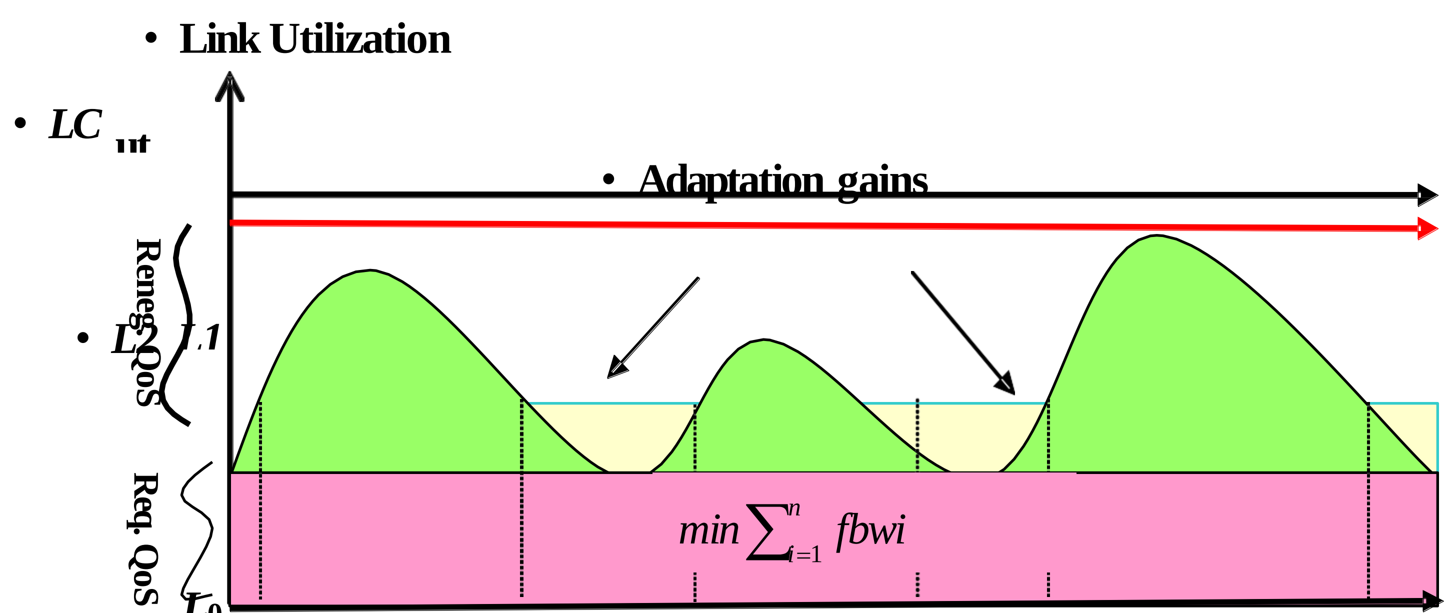
The impact of system traffic

- If the incoming traffic is less than the available resources, QoS renegotiation is not required and it is required otherwise.

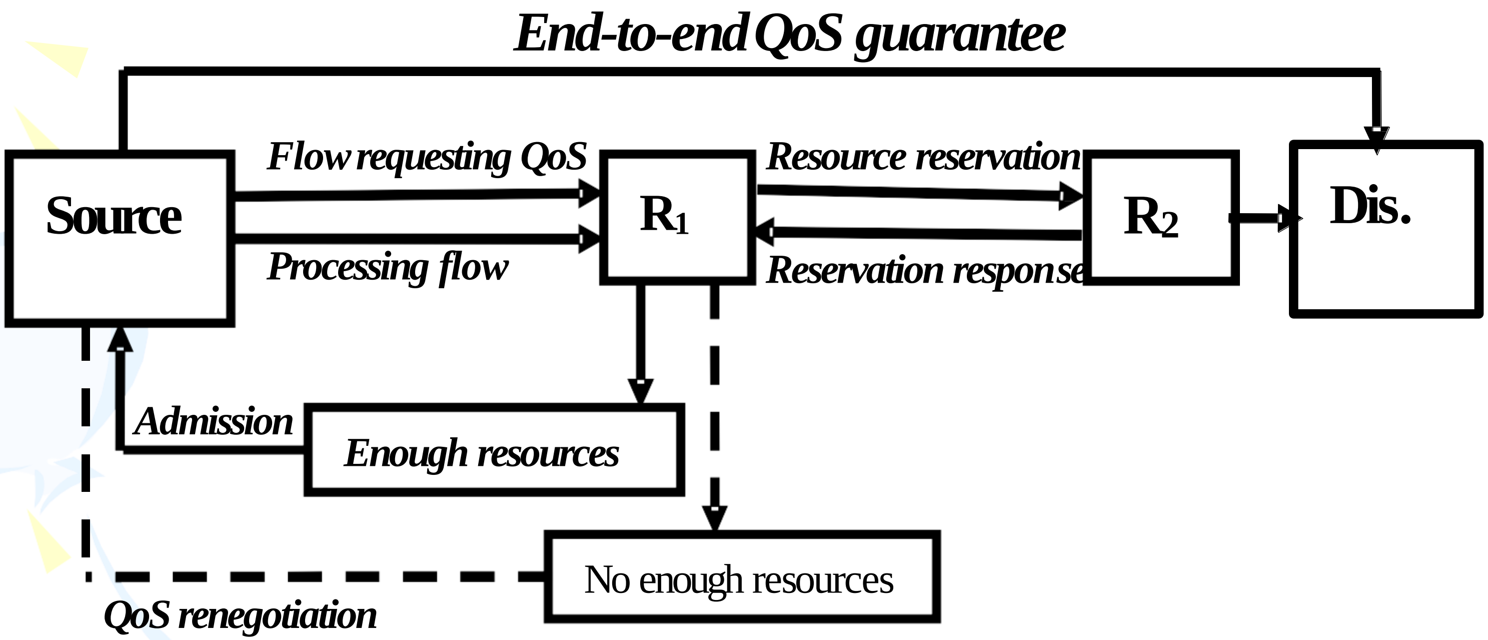
$$\sum_{i=0}^{n} in-flow_i BW > \sum_{i=0}^{n} out-flow_i BW$$

An inside look at the link and QoS renegotiation

- Requested QoS is granted in periods t1, t3, and t5.
- QoS is renegotiated in periods t2, t4, and t6.
- More calls get admitted because of enforcing the resource sharing concept.



The algorithm description



Outlines

```
• Do periodical resource estimation ( ) {
•   Get_flowi( BWi)
•    If (BWi + CurentLoad ≤ linkbandwidth)
•                          Admit (flowi)
•      Else {
•              reject (flowi )
•                        If rejected ( flowi )
•                                  BWi = mBWi
•             if (mBWi + CurntLoad ≤ linkbanwdth)
•                        Admit ( flowi)
•             Else
•                   reject (flowi ) } }
```

Advantages

- No receiver or lower layers involvement.
- Low control overhead.
- In congested periods, you can identify the source of a specific flow and its type.
- Flow-based management has a course a granularity level that decreases the control and management overhead.

Experimental results

- Four approaches have been simulated:
  - Utilization target based IntServ.
  - Incoming flow bandwidth based IntServ.
  - IntServ based on the probability that the required resources exceed the available resources (PNREARBI)
  - The proposed QoS renegotiation approach.

Experimental results

- IntServ module in ns-2 version 2.26 is used.
- The algorithm implemented for single source and single distention.
- Two QoS parameters are measured：
  - PDR：reflects the application's requirments
  - Utilization target：reflects the service provider's goal.

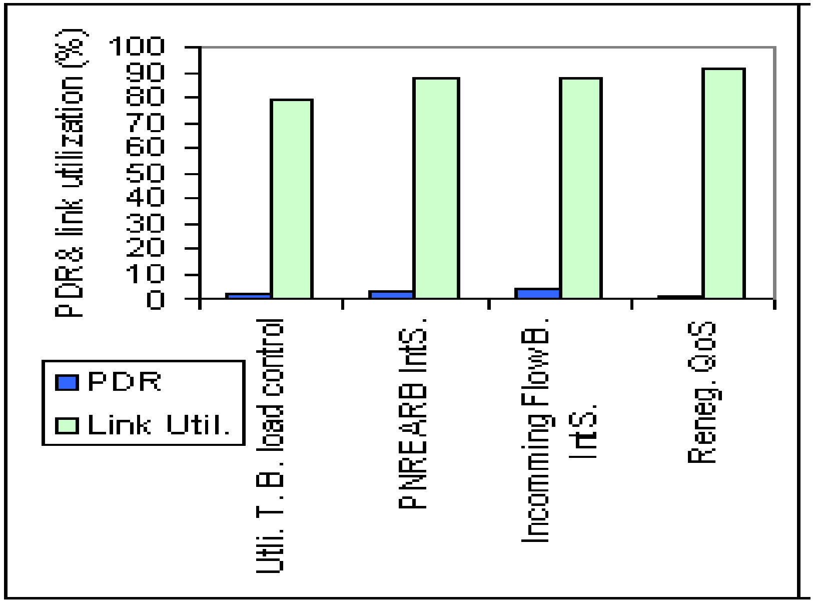
Experiment 1: Improving QoS

AVE(%)	PDR	Link Util.
Utli. T. B. load control	2.5	79
PNREARB Int S.	3.5	87
Incomming Flow B. Int S.	4.3	88
Reneg. QoS	1.2	91

- Link utilization and PDR

Experiment 1: improving QoS, Cont.

- The proposed approach provides the highest link utilization and the lowest PDR



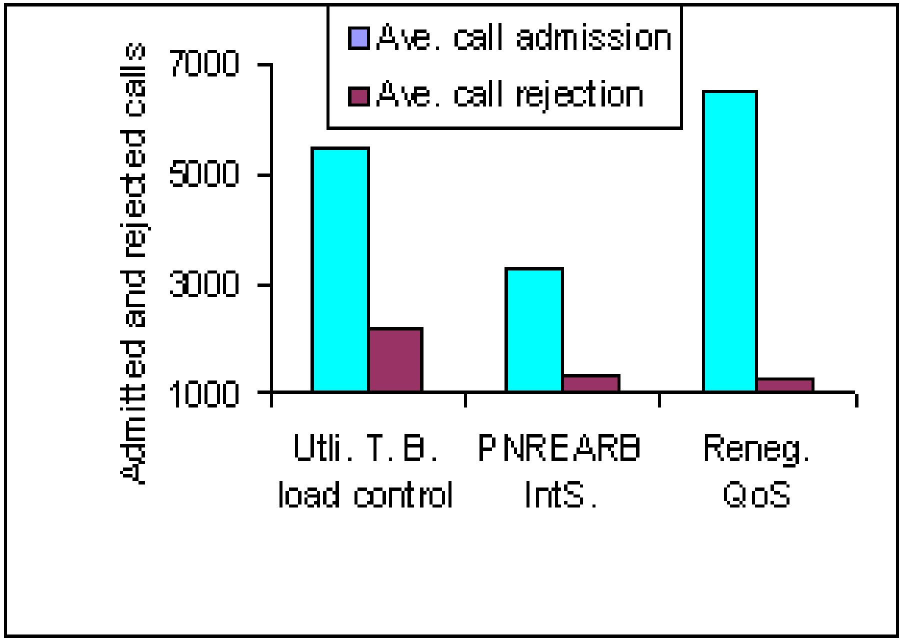
Experiment 2: Average call success versus average call rejection

❖ Ave. Admitted calls:

Renegotiation B. IntServ		PNREAR B IntServ		Util. T. b. IntServ	
time	Adm calls	Prob.	Adm. calls	Util. T.	Adm. calls
1	4473	20	3272	98	5181
2	4742	15	3211	95	5265
3	5293	14	3244	92	5319
4	6414	12.5	3256	90	5385
5	7180	11	3213	88	5423
6	7300	10	3237	85	5503
7	7389	9	3921	83	5574
8	7463	8	3338	80	5636
9	7463	5	3247	78	5666
10	7463	1	3237	70	5891
AVE	6518	AVE	3317	AVE	5484
%	1		1.96		1.18

Experiment 2: Call success rate versus call rejection, Cont.

- Ave. Admitted and rejected calls for the different approaches:



Experiment 2: Call success rate versus call rejection, Cont.

❖ Rejected calls:

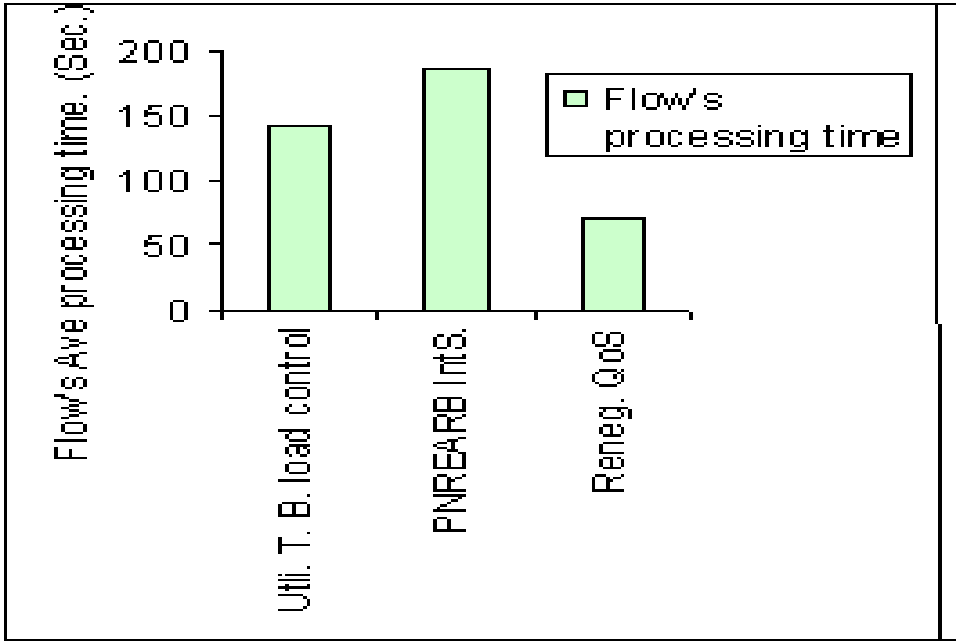
Reneg. B. Int.Serv		PNREAR B. Int.Serv		Utilization T. B. Int.Serv	
time	Rej. calls	Prob.	Rej. calls	Util. T	Rej. calls
1	3175	20	1292	98	2466
2	2906	15	1319	95	2383
3	2355	14	1293	92	2329
4	2355	12.5	1333	90	2263
5	468	11	1341	88	2225
6	348	10	1334	85	2145
7	259	9	1302	83	2094
8	185	8	1286	80	2012
9	185	5	1356	78	1982
10	185	1	1324	70	1767
Ave	1242	AVE	1318	AVE	2166
%	1		1.06		1.74

Experiment 3: Call processing time

Adaptive IntS.		PNREAR B IntServ		Util T.b. IntS	
Time	F. proc T.	Prob.	F.P.T.	Util T.	F. P.T.
1	205	20	190	98	164
2	191	15	184	95	158
3	156	14	174	92	152
4	85	12.5	172	90	148
5	25	11	168	88	144
6	20	10	164	85	142
7	9	9	162	83	138
8	4.5	8	205	80	133
9	4.5	5	212	78	129
10	4.5	1	209	70	114
AVE	70	AVE	185	AVE	142
%	1		2.6		2.02

- The proposed approach provides the lowest average processed time for flows.

Experiments 3: Call processing time



- The proposed approach provides the lowest average flow processing time.

Conclusions

- Implementing a flow-back feedback mechanism for real-time applications which captures the characteristics of multimedia applications and heterogeneous networks.
- Improving the performance of IntServ.
- Decreasing control overhead for load controlled networks by using flow information as a feedback mechanism.
- The concept of adaptation gains has been introduced

Conclusions (2)

- The proposed feedback mechanism:
  - Improves IntServ performance by admitting more calls for the same amount of resources.
  - Achieves more fairness by decreasing the average rejected calls.
  - Decreases control overhead.
  - Improves QoS parameters.

Multimedia Framework (MMF) Architecture

- MMFramework is an open multimedia framework which may be used for dynamic creation of various multimedia applications and which could be extended by new multimedia devices.
- The proposed framework's architecture consists of six layers.
- Each layer consists of a collection of components which are characterized by similar functionality.
- The structure and goals of the layers are the following:

- The first layer called **MMHardware and System Software Layer** consists of multimedia hardware and software provided by vendors.
- This layer is represented by a wide spectrum of devices such as: video cameras, computers, audio/video encoders/compressors, media servers, etc.
- These devices are usually equipped with proprietary control software.

- The second layer - **MMHardware CORBA Server Layer** packs up the vendor-provided software by CORBA interfaces.
- This layer introduces a uniform abstraction defined by an interface specified in IDL and standard communication mechanisms provided by the IIOP protocol (Internet Inter-ORB **Protocol)**
- The IDL interfaces defined in this layer support all operations provided by the native software.
- The main goal of introduction of this layer is to establish a common base for the system development.

- The third layer - **A/V Streams Control Layer** is dedicated to multimedia streams creation, control, and destruction.
- This layer implements the OMG specification and provides software objects which expose functionality of the lower layer CORBA servers in standard form most suitable for audio and video streams control.
- It provides an abstraction of a stream encapsulated in the form of a CORBA object which represents its parameters and control operations.
- This layer provides also mechanisms for streams parameters negotiation between source and destination multimedia devices and provides streams addressing and QoS control.

- The fourth layer - **Presentation Layer** resolves the problem of different data types used for the parameters of multimedia devices and streams representation.
- The main goal of this layer is to translate the parameters types from their actual values to CDF (Common Data Format).
- This format is used above the Presentation Layer to simplify presentation of the system's state and to provide a uniform view of the system components for control and visualisation purposes.
- This layer supports users with a powerful mechanism of forms that makes system configuration simple and less susceptible to errors.

- The **Management and Access Control Layer** provides a uniform view of the MMF components' state and a set of functions for their manipulation and accessing (e.g involving security or providing statistics).
- Each component which is an object with its own interface and notification mechanism represents the state of a single connection or a device.
- The items from the repository provide the following general functionality:provide operations of two following categories:
  - reading actual values of attributes - state of the system component represented by the given item,
  - changing values of attributes - these operations may involve also a call of suitable operations on the lower layers.
- act as an event producer and sender to interested receivers - the push model of the events notification has been chosen. The message may be a result of internal or external event in the system.

- A top layer of the MMF architecture is called Application Layer.
- The entities of this layer are collection of user interfaces that provide access to control and visualization of the system state in the most convenient(usually graphical) form.
- The objects defined on this level act as observers of the system components and combine them in the given application scenario.
- They may also perform the MMF clients' role actively changing the system's state by operations invocations on the devices and connections abstraction provided by the lower layer.

- MMFramework has been constructed taking into account the distributed system scalability. The conventional request/reply synchronous client-server paradigm has been replaced, where appropriate, with efficient event-driven asynchronous communication. The publish/subscribe patterns are widely applied with unicast and reliable multicast communication protocols when a device state or property changes have to be reported to a group of clients. This style of information dissemination and event notification has been implemented with the support of CORBA Event Service and Notification Services. Resulting MMFramework has been structured as collection of observable distributed objects what is the characteristic feature of the proposed architecture.

- The novel aspect of MMFramework is the definition of mobile multimedia devices.
- The background of this study originates from location-aware computational systems such as Active Badge next generation (ABng).
- This system is a CORBA-compliant implementation of the Active Badge System developed at Olivetti & Oracle Research Laboratory (ORL).
- ABng allows to locate people and equipment within a building determining the location of their Active Badges.
- These small devices worn by personnel and attached to equipment periodically transmit infra-red signals detected by sensors which are installed in the building.
- Hence, video or audio stream may be attached logically to a locatable user and follow him.

- MMFramework has been also equipped with integrated graphical interfaces built in Java that represent in a compact, user-friendly form configuration, state and control of complex distributed systems.
- The system exploits Java Applets communicating via IIOP protocol with suitable CORBA servers of the framework.
- The graphical elements of these interfaces may by connected in run-time to call-back functions which generate suitable events or perform control activity.
- A lot of effort has been put into invention of a uniform graphical form of the system components' representation for the global system state visualization.

- The system has been designed with existing CORBA Services and OMG specifications related to multimedia applications in mind.
- The multimedia streams control has been implemented based on an OMG document using own implementation of the specification.
- The system has been integrated using Name Service.
- The multimedia devices and streams are characterized by a large number of parameters what justified the Property Service usage.