AN ADAPTIVE QoS MECHANISM FOR MULTIMEDIA APPLICATIONS IN HETEROGENEOUS ENVIRONMENTS

by

ASHISH N. DESAI

A thesis submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

written under the direction of

Prof. Manish Parashar

and approved by

_____

_____

_____

New Brunswick, New Jersey

October, 2001

ABSTRACT OF THE THESIS

Adaptive QoS Mechanism for Multimedia Applications in Heterogeneous Networks

by ASHISH N. DESAI

Thesis Director:
Professor Manish Parashar

The Internet is a dynamic network of heterogeneous transmission media with widely varying bandwidth capacities and latency characteristics. Given the tremendous popularity of the Internet, Internet traffic has not only increased but also changed in character. New classes of streaming multimedia applications, such as Video on Demand and IP Telephony, are being increasingly deployed. Such media rich applications require time bounded processing and communication. This imposes a need for temporal & spatial guarantees, such as high data throughput and low latency, from the underlying network. These applications can suffer severe performance degradation under conditions of unpredictable delays and losses in the network, as is the case in the Internet today. Consequently runtime Quality of Service (QoS) adaptation is necessary to enable operation of these applications with acceptable performance in spite of potentially insufficient network and end-system resources.

This thesis presents the design, implementation and evaluation of an adaptive QoS mechanism that enables multimedia applications to perform satisfactorily under constrained system and network resource availability. We identify packet loss suffered by the application due to network congestion and processor overload as the crucial parameter affecting application performance. The adaptive mechanism aims to improve the QoS perceived by the end user, by optimizing this packet loss through control of the application data transport characteristics. It performs runtime adaptations using a policy based algorithm. Adaptations investigated include

gradual degradation of media quality and media transformation and are based on user preferences, client system capabilities, and dynamic network and system state.

An experimental evaluation of a Java based implementation of the mechanism is presented. Our implementation provides integrated server side and client side adaptations for a simulated streaming multimedia application. It leverages on the real-time transport support provided by Real-time Transport Protocol (RTP) and real-time feedback support provided by RTP Control Protocol (RTCP). It supports multiple levels of service by using multiple concurrent IP multicast sessions for media transfer. Experimental results indicate that such an adaptation mechanism can significantly improve the performance of networked multimedia applications, if the policy controlling the mechanism is designed with a proper understanding of the application's configuration and requirements.

Acknowledgements

I thank my family for their love and support during my graduate studies in Rutgers University. I am grateful to my advisor Professor Manish Parashar for his invaluable guidance, encouragement and support throughout my stay at Rutgers. I am thankful to Dr. James Flanagan and Dr. Ivan Marsic for their advice and pertinent suggestions regarding my thesis. I acknowledge the suggestions of the committee in developing my technical understanding, research aptitude, thesis writing and presentation skills.

I would like to thank the CAIP and ECE support staff for their prompt and detailed responses and for the excellent facilities that I could avail of in CAIP and ECE labs. Last but not the least I would like to thank my friends of the TASSL lab for their superb company and support without which my stay in Rutgers wouldn't have been so enjoyable.

This work is sponsored in part by the NSF KDI grant and CAIP Center. The CAIP center is supported by the New Jersey Commission on Science and Technology and the center's Industrial Members.

Table Of Contents

Table Of Figures

Chapter 1

Introduction

1.1 Objective

The objective of this thesis is to develop an adaptive QoS mechanism that enables multimedia applications to perform satisfactorily under constrained system and network resource availability. The mechanism should be able to perform runtime application level adaptations assuming only the ubiquitous best effort IP service, i.e., without any network level QoS support.

1.2 Problem Description

The Internet is a network of networks, a mesh of various transmission media with remarkable heterogeneity and dynamism in bandwidth capacity and latency characteristics. Different networks have varying frame sizes requiring additional conversion overheads at the gateways. Furthermore, the networks have different scheduling policies and their interconnectivity results in multiplexing and demultiplexing of traffic. Network delays are imminent in such an environment since the Internet traffic is not only increasing in proportion to available bandwidth, but also changing in character due to emergence of new networked applications. As a result it is difficult to predict peak available data rates at any region of the network at a particular time, and consequently over-provisioning is not a realistic alternative.

Internet Protocol (IP) [2] provides a ‘best effort’ service to applications by routing packets independently (using unique addressing), and ensures seamless delivery over heterogeneous networks (using fragmentation and reassembly). It fundamentally advocates leaving complexity at the edges and keeping the network core simple in line with the ‘end-to-end argument’ [9]. It depends on higher layers of the protocol stack to satisfy other application specific data transfer constraints such as reliability, latency and consistency of data throughput. IP

has proved to be a robust and scalable solution for traditional Internet applications such as email, file transfer and other web applications.

Distributed multimedia applications typically operate in heterogeneous environment such as the Internet where the network resources and the end-host processing capabilities vary significantly. Since the states of the network and end host system are dynamic, distributed applications have to contend with unpredictable resource availability. Additionally, the inherent dynamic nature of the resource requirements of these applications makes it very difficult to optimally define the level of service for these applications. Furthermore, achieving contracted service is not always feasible. Under conditions of unpredictable delays and losses in the network, these applications can suffer severe performance degradation. For example the unpredictable, bursty nature of network traffic often creates transient network congestion causing routing delays and lost packets. The usability of an IP-based telephone service in such an environment is severely limited by the resultant sub-optimal round-trip times. As a result, supporting streaming multimedia applications on best effort IP based Internet poses great technical challenges. Key among these is the need for an uniform application driven integrated multimedia framework that can hide device and network differences, obtain system and network state information and perform runtime adaptations in the application to provide satisfactory QoS for the end user.

1.3 Background

The tremendous popularity of the Internet has led to the emergence of a new generation of applications with widely varying characteristics and requirements. An important class among these is distributed multimedia applications, which have high commercial potential. Video Conferencing, Video-on-Demand and IP telephony are a few of these applications. Deploying these applications on the Internet presents many technical challenges and has evoked much research interest. Traditionally high bandwidth applications, such as Video-on-Demand (VoD) requiring transmission of real-time high quality video signals, are inhibited by the bandwidth

limitation on ‘the last mile’. This is due to low speed connections used by the end user to connect to Internet Service Providers (ISP). However technologies such as cable modem and digital subscriber line (DSL) have removed this hurdle and made such applications feasible on the Internet.

Distributed multimedia applications have time bounded processing and communication requirements, primarily due to the coding and compression techniques involved, impose temporal dependencies on media. Playback procedures typically involve reproduction of multiple media in a tightly synchronized manner. This requires temporal and spatial guarantees from the underlying network. However these applications exhibit a common characteristic that they operate satisfactorily in less than ideal situations by allowing for a tradeoff between certain service requirements. For example one approach taken to leverage this property is to employ low bit rate media coding techniques with standard Internet transport protocols. However, this does not cater to the time varying nature of the communication channel. Furthermore, it does not take care of possible degradation in network performance due to potentially misbehaving sources. An alternative approach is to design new transport protocols and use standard media coding algorithms. But these techniques cannot utilize the strengths of various coding techniques, such as compression efficiency and robustness to transmission errors optimally. Hence, to enable operation of these applications with acceptable performance despite insufficient network and end-system resources, runtime adaptation of service is necessary.

1.4 Adaptive QoS

Quality of Service (QoS) can be broadly defined as the degree of satisfaction of the end user’s requirements. For distributed multimedia applications it can be interpreted as satisfying the application’s system and network resource requirements. Active QoS management over the best effort IP is vital for ensuring some level of quantitative or qualitative determinism in the service provided by the Internet. Note that any QoS assurance is only as good as the weakest link in the

"chain" between sender and receiver. So QoS is fundamentally an end-to-end issue implying that QoS assurances have to be configurable, predictable and maintainable from source to destination. This means that it should be relevant over all architectural layers from source media devices down through source protocol stack, across each network element and finally up through the receiver protocol stack to playback devices. Consequently the issue of QoS can be addressed at different levels of the network protocol stack such as:

- User level by specifying qualitatively/quantitatively user perceivable service parameters.
- Application level by ensuring that the application adapts according to the network and system resource availability.
- Network level by defining traffic models, classification of service disciplines, and resource reservation on a per-flow or flow aggregate basis, to ensure that the application's resource requirement is met.

Presently considerable research effort is directed in handling QoS at the network, application and middleware level. Some of the promising approaches are discussed in the chapter on related work. Adaptive QoS uses runtime adaptations to maintain an optimum QoS under existing resource constraints and can play an important role. The adaptation mechanism however should be sensitive to the needs of the application and should possess knowledge about the network and system resources for maintaining end-to-end QoS for the application. Some guiding principles [10] for designing an adaptive QoS framework are:

- Formulate QoS specifications to capture application level QoS requirements and management policies. The specifications can be used to configure and maintain QoS mechanisms resident in end-system and network.
- Define QoS configurable interfaces to formalize QoS in the end-system & network, thereby integrating QoS control and management mechanisms in the elements.

1.5 Overview of Thesis

This thesis presents an application level QoS management mechanism for distributed multimedia applications. The QoS management mechanism is based on the argument that QoS is ultimately decided by the degree to which the resource requirements of the application are met. At any point of time an application knows its dynamic resource requirements and optimum tradeoffs the best. Hence mechanisms operating at the application level have the potential of being most effective. Furthermore such mechanisms can be implemented over the ubiquitous IP and they can readily leverage any QoS support available at the network and transport level.

In the policy based adaptive QoS mechanism presented in this thesis each of the client systems is guided by a resource management policy that captures degree of QoS adaptation tolerated by the application and the actions to be taken in case of violation. The policy also defines the adaptations in the service provided to the application in response to variations in resource availability and/or application requirement, and is configurable at each of the interacting elements to support deployment in a heterogeneous environment. The adaptive QoS mechanism resident at the client utilizes the knowledge of the local system and network requirements of the application to make optimum tradeoffs in the service parameters.

QoS adaptations investigated in this thesis are based on user preferences, system capabilities and dynamic network and system state. Adaptations may be receiver side or sender side. In receiver side adaptations clients acting as receivers, adapt the application characteristics locally in response to changes in the local system and network resource availability and the statistical transmission information of the remote sender client. In sender side adaptations the source adapts the transmission characteristics based on application state and statistical reception information of the remote receiver clients. Note that the server and client side adaptations are complementary as shown by the experimental results. Adaptations implemented include gradual degradation of media quality and media transformation.

- Adaptation by gradual degradation: The application gradually degrades its service level to match current preferences and network/system state and capabilities. For example, consider a shared image viewing application illustrated in Figure 1. Using progressive image encoding (based on an algorithm developed by P. Meer et. al., CAIP, Rutgers University) each client can view image at different resolutions (and compression ratios), based on current system/network state.

- Information Transformation: Here, the application media type is transformed based on local client preference and capabilities. For example, in the shared image viewing application, if a particular client is visually impaired or is unable to view the image due to excessive packet loss or system resource limitations, the image can be transformed in an alternative medium, e.g. text or speech in this case, and can be presented to the client.

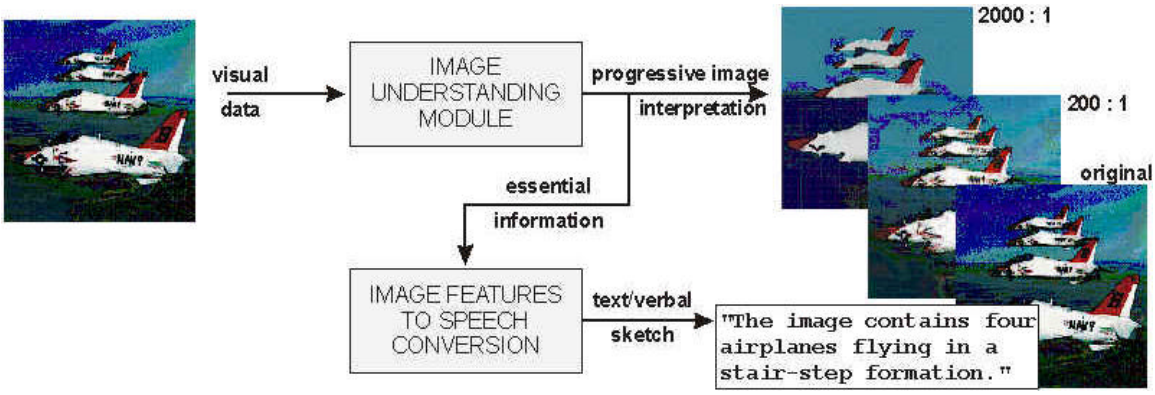

Figure 1 Progressive Image Encoding and Information Transformation (P. Meer et. al.)

A Real-time Transport Protocol (RTP) [4] based architecture is defined to enhance the transport functionality of the User Datagram Protocol (UDP) [1] for better real-time support. It uses RTP Control Protocol (RTCP) [4] to obtain real-time feedback about transmission/reception state of clients. The performance of such a framework depends on accurate and timely assessment of resources availability. The architecture integrates server side and client side adaptations for a simulated streaming multimedia application. It supports multiple levels of service for the application by using multiple concurrent IP multicast sessions for media transfer. An

experimental evaluation demonstrates that an adaptive QoS mechanism can significantly improve the performance of distributed multimedia applications in the Internet by enforcing runtime application level adaptations guided by resource management policies defined with a proper understanding of the application's configuration and requirements.

1.6 Contributions

The main contributions of this thesis are:

- Distributed application level adaptive QoS mechanism for a better QoS to end users of multimedia applications in heterogeneous environments

- Integration of a policy based scheme with the standard RTP and RTCP based real time data transport and feedback mechanism for streaming multimedia applications.

- Experimental study of loss based adaptations for streaming applications.
    - Innovative evaluation of an application's network resource utilization efficiency.

1.7 Outline of Thesis

Chapter 2 details the approach pursued to provide QoS to applications assuming only the best effort IP service from the network. It discusses the relative merits of the approach and evaluates it by contrasting against alternate approaches.

Chapter 3 presents related work that discusses various approaches adopted by the research community, and identifies different issues being addressed by various research groups to provide Quality of Service for distributed multimedia applications over heterogeneous networks.

Chapter 4 outlines the design and implementation of the adaptive QoS mechanism. A modular approach is followed to facilitate adding functionality to the mechanism in a phased manner. Responsibilities of the different modules and their interactions are specified.

Chapter 5 describes the experimentation setup and a simulation based evaluation of the adaptive mechanism. The assumptions made for the simulation setup are noted and the results are plotted. The deductions from results obtained are discussed.

Chapter 6 summarizes the research work and discusses directions for future work.

Chapter 2

QoS Management for Multimedia Applications

This chapter presents a discussion on the possible approaches in QoS management for multimedia applications. The approaches are classified as reservation based and adaptation based schemes and the proposed adaptation based scheme is introduced.

2.1 Reservation Based Schemes

Reservation based schemes perform resource management to offer guaranteed service and ensure optimal resource utilization. This is generally done by allocating resources on request and policing application's data transport to ensure resource allocations are not exceeded. A typical resource management scheme incorporates admission control algorithms, reservation protocols, monitoring protocols, and signaling mechanisms. It encompasses process management, memory management, queuing algorithms, traffic shaping and error control algorithms. Resource management can be done at three levels - application, system and network. The quality of service achieved by a multimedia application depends on source/player characteristics, transmission characteristics, media synchronization etc. The system state is determined by communication and operating system parameters such as bits per sec, error rate, processing time and data unit size. The network state is defined by inter-arrival time, latency, bandwidth, jitter; traffic parameters peak data rate and burst length.

From an implementation perspective a standard parameterization allows for flexibility and customization of resource reservations for similar applications. Parameter values determine type of service. Deterministic or statistically bound parameter values imply guaranteed services where as parameter values estimated from past behavior are used to provide 'better than best-effort' service guarantees. QoS requested is initially defined by the application parameters. The resource manager translates these parameters using bi-directional translations. The parameters are

then distributed and negotiated to perform admission and actual reservation. Negotiations are of two types, peer-to-peer and layer-to-layer in the protocol stack. Resource admission test, resource schedulability test for sharing, and spatial test for buffer allocation are performed according to a cost function. Finally resources are reserved and allocated in an exclusive or shared manner.

Reservation can be sender initiated or receiver initiated, depending on the direction in which the reservation actually takes place. Sender initiated reservations are done based on the resource requirements of the application. Receiver initiated reservations can take into account the state and preferences of the client in addition to the application's resource requirement. Hence if different levels of services are supported, receiver initiated reservation can perform better. Furthermore if soft reservations requiring periodic refresh are made, the receiver initiated approach is a more scalable solution. Receiver initiated reservations also obviate the need to maintain per client state at the server. However they involve more signaling and require extensive support form the network level at the intermediate nodes. Reservation can be pessimistic e.g. peak-rate multiplexing or optimistic e.g. statistical multiplexing. Pessimistic reservations provide guaranteed QoS and avoid resource conflicts by allowing for the worst case. However they often lead to underutilization of resources. Optimistic reservations provide high utilization of resources, but overload situations may result in reservation failure and monitor functions may be required to resolve resource conflicts. Reservations can be made with immediate effect, i.e. immediate reservation, or can be effective at a future time, i.e. advanced reservation. Reservation can be performed using a centralized approach with a central reservation server or a distributed approach using reservation brokers. Resource de-allocation can be explicit, performed by the sender, receiver or monitor, or can be implicit using soft states and requiring periodic refresh.

2.2 Adaptation Based Schemes

An adaptation based scheme provides mechanisms for performing run time adaptations to match the application resource requirement with the resource availability. It can be implemented at the network, system or application level. At the network and system level, resource usage is actively monitored and adaptive resource reallocation is performed to maximize utilization. This thesis presents a purely application level adaptation approach with no assumption of any QoS support from the network. However the adaptation mechanism can leverage any QoS support available in any of the above mentioned levels.

Adaptive QoS management approaches can also help multimedia applications to be fair to other applications competing for the limited available resources. Generally multimedia applications use UDP, which lacks any congestion control mechanism and can severely overload the network resulting in bandwidth starved TCP based applications. However adaptation mechanisms can enable the distributed multimedia application to be sensitive and responsive to network congestion. Feedback is an integral part of an adaptation mechanism and feedback rate is an important scaling consideration. If the feedback rate is fixed, the feedback traffic will obviously increase unbounded with the number of clients that connect to the application. It is generally desirable to specify a limit on the feedback traffic as a small percentage of actual application traffic. In our scheme, RTP Control Protocol (RTCP) scaling mechanism decreases the rate proportionally with the increase in multicast group size to keep the RTCP control traffic in check.

Figure 2 shows the heterogeneity existing in the Internet. Application level adaptations for multimedia applications can be performed at the server, client or an intermediate transcoder proxy as shown in the figure. The adaptive QoS approach proposed in this thesis implements integrated server and client based adaptation. As discussed below, there are several advantages

for both of these adaptations and they complement each other very well. The mechanism can be further enhanced by incorporating transcoder based adaptations.

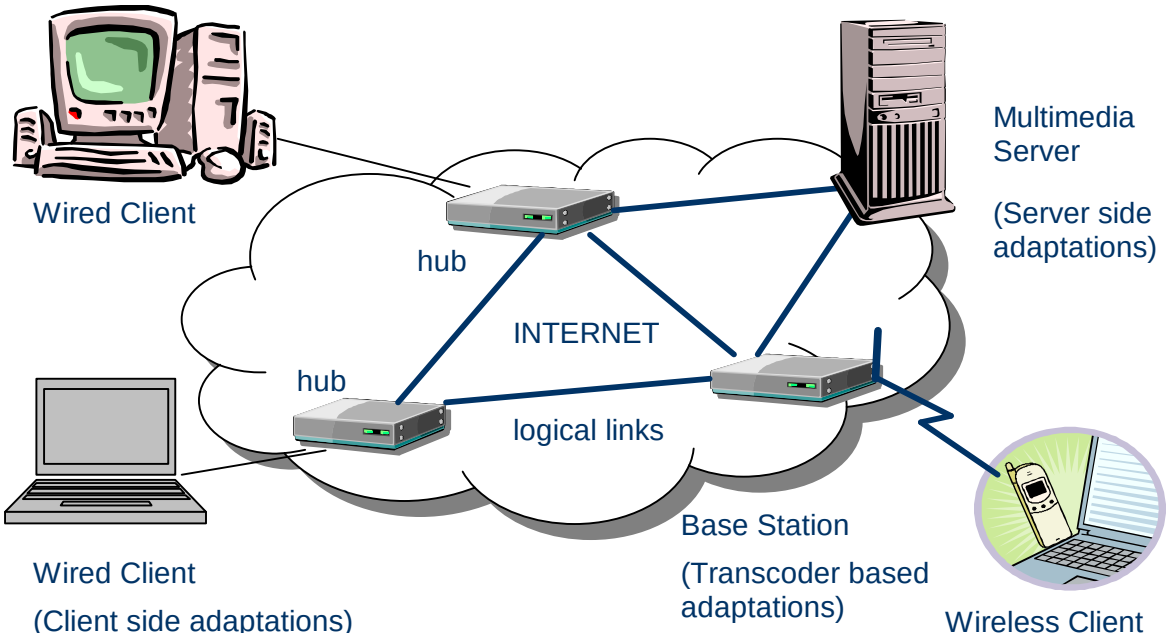

Figure 2 Application Level QoS Adaptations for Multimedia Applications in the Internet

2.2.1 Sender Based Adaptation

A sender can adapt the transmission of application data according to service available from the network. Various feedback schemes can be used to gauge the network state. Buffer based adaptations measure network state based on the occupancy of the transmit buffer. The goal is to maintain the buffer occupancy at a predetermined level. Loss based adaptations use feedback about losses suffered by receivers to indicate the network state. In this case receivers have to periodically transmit their reception state to the sender. Since different receivers may experience widely different degrees of congestion during a multicast session over a heterogeneous network, sender has to devise a mechanism to decide on the levels of service to provide so as to optimize the application's performance.

2.2.2 Receiver Based Adaptation

Receiver side adaptations are based on the encoding and transmission techniques adopted by the sender. Layered encoding with a layered transmission scheme at the sender can be used to enable receiver side adaptations. Layered transmission is implemented by sending each encoded layer to a separate multicast group. Receiver selects appropriate transmission quality by subscribing to certain number of multicast groups. An alternative approach to layered encoding is encoding and transmitting multiple copies of data on separate multicast channels, each independently providing a different level of service. Receiver selects appropriate transmission quality by subscribing to corresponding multicast group. Thus the receiver adapts by independently tuning to a transmitted service level that best fits its need, capabilities and resource availability. The granularity of adaptation is predetermined by the granularity of layered encoding. Layered encoding involves complex encoding algorithms. Although, in most cases encoding can be done off-line, synchronization between decoded streams at the receiver for playback adds to the overall delay. Delay in switching between levels is also a crucial parameter since very short delays can cause oscillations while large delays can result in sub-optimal application performance. Additionally layered transmission can be prioritized such that the base layer has highest priority and subsequently higher layers have decreasing priorities, this enables a better network support for the application. These priorities however, have to be supported from end-to-end in the network.

2.2.3 Transcoder Based Adaptation

In transcoder based adaptation the transcoder acts as a proxy gateway for clients with similar network or device constraints. Such gateways are placed at appropriate locations, e.g. between bordering heterogeneous networks, to deliver different levels of service to network regions with different characteristics. The client connects to the application through the gateway, which then makes requests to the server on behalf of the client. The proxy intercepts the reply

from the server, decides on and performs the adaptation, and then forwards the transformed content back to the client. The transcoder can convert the received media coding, for e.g., from high bit-rate to low bit-rate coding. Alternatively it can invoke a server like adaptive rate control algorithm in response to receiver feedback. Since the deployment of this scheme is at intermediate nodes in the network this adaptation is more difficult to implement and coordinate. A transcoder based architecture makes it easy to place adaptation geographically close to the clients with no modification to existing clients and servers. However since a proxy would potentially handle the content from many servers with widely varying characteristics, it is difficult to determine which modification is most appropriate for each content type. Also for secured or proprietarily encoded content such as Real Networks streaming media, deploying a transcoder will involve coordinating with the service or content provider in order to access the content for performing adaptation.

Client side adaptations provides a truly distributed solution for managing heterogeneity since all clients can locally decide and employ adaptations most appropriate to them without affecting other clients in the session. Since the adaptation overheads are incurred at the client end, rather than the server, this provides a more scalable solution. Application level adaptations, such as content adaptation, are easier to implement at the client side as they can be closely integrated into the application playback procedures. The most up-to-date information about the client system and network state is available at the client and hence on the fly adaptations can be very effective and can be enforced in an isolated manner without affecting the performance of the distributed application. However adaptations that can benefit a group of clients subscribed to the session can be more efficiently implemented at the server or proxy side. Furthermore all of the clients may not be able to implement the content adaptation techniques due to processor and memory resource constraints, and would rather want the server or proxy to perform the adaptations. Finally the

granularity of client side adaptations is dictated by the encoding and transmission techniques employed by the server.

The objective of this thesis is to develop an adaptive QoS mechanism to enhance the performance of multimedia applications over heterogeneous networks and client systems with limited resource availability. An application level adaptive QoS approach is proposed as an efficient, scalable and effective solution. The key contribution of this approach is integration of a distributed client-server policy based adaptation scheme with the standard RTP and RTCP based real time data transport and feedback mechanism for streaming multimedia applications.

Chapter 3

Related Work

Providing service guarantees to multimedia applications has evoked much research interest and many different approaches have been proposed. The approaches can be broadly divided into network level protocols, reservation based schemes, adaptation based schemes and optimization schemes proposed to enhance QoS support. Network level protocols provide QoS support by interpreting the application's requirements in terms of network parameters and enhancing the network switches to service application flows or flow aggregates according to the assigned service levels. Reservation based schemes reserve network/system resources based on the application's requirements and are typically accompanied by admission control schemes to check if the application's request can be serviced with the existing resource availability. Adaptation based schemes utilize the adaptive behavior of applications that do not require hard service guarantees, to provide them with a better than best effort service by performing application aware active resource management with runtime adaptations. Various optimization schemes propose techniques for monitoring, integrating application processing with data transport and embedding QoS state information in application data to achieve a better performance for applications requiring service guarantees.

3.1 Network Protocols with QoS Support

The Internet Engineering Task Force (IETF) has addressed the issues in building a QoS support infrastructure to be deployed over IP on the Internet. Several working groups have proposed many promising alternatives, however considering the scale and the growing heterogeneity of the Internet, a single solution is still elusive. The most prominent of the proposals are discussed here.

3.1.1 Integrated Services

The Integrated Services (Resource Reservation) approaches reserve network resources for individual application flows based on explicit resource requests. This mechanism, as implemented in Resource Reservation Protocol (RSVP) [5], provides hard service guarantees, high granularity of resource allocation and a detailed feedback mechanism. It involves an explicit reservation setup and teardown phase. In the setup phase, a "PATH" message, carrying traffic specification, is sent by sender to receiver and is used to establish a "path-state" at intermediate RSVP enabled routers. A reciprocal "RESV" message, carrying flow descriptor (request specification and filter specification), is sent by receiver to sender and is used by routers to reserve resources. RSVP uses a token-bucket model for traffic shaping to characterize its input/output queuing algorithm. Soft reservations requiring periodic refresh are made in each router. Reservations are receiver-based to handle heterogeneous multicast receiver groups. In a multicast scenario, reservations are merged at traffic replication points. This mechanism is highly complex involving elaborate signaling mechanisms, and imposes considerable overheads on applications and network elements. In principle this is a significant deviation from the highly successful and scalable best effort IP. Furthermore, non-RSVP routers in traffic path can be weak links degrading QoS provided. Consequently this mechanism is ill-suited for some applications, specially those that provide scope for adaptability in the resource requirements, and can operate more efficiently with mechanisms providing a simpler and less fine tuned QoS support.

3.1.2 Differentiated Services

Differentiated Services (diffserv) [6] classifies the network traffic and allocates the network resources to flow aggregates based on a management policy. Service classes are created with different QoS guarantees and flows are assigned these classes. In the differentiated services domain, Service Level Agreements (SLA) are setup between adjacent networks. SLA establishes policy criteria, and defines the traffic profile to be adhered by independently managed domains.

Bandwidth Brokers (BB) are identified in each diffserv domain to manage and negotiate network resources based on SLAs. Traffic is policed and smoothed at network egress points according to the SLA. Per-Hop Behaviors (PHB) are applied by the traffic conditioner to traffic at a network ingress point according to pre-determined policy criteria. This mechanism provides a comparatively simple and coarse mechanism for QoS support. It exhibits greater flexibility and is able to allocate resources efficiently while still providing service guarantees. Consequently this approach is well suited for providing network level adaptive QoS support to distributed applications operating in heterogeneous networks.

3.1.3 Multi Protocol Label Switching

Multi Protocol Label Switching (MPLS) [8] is a traffic engineering protocol that provides resource management for flow aggregates via network routing control according to fixed length 'labels' in packet headers. Like Differentiated Services it marks traffic at the ingress network boundary, and un-marks it at egress points. The MPLS-enabled router, Label Switching Router (LSR), routes efficiently, using the fixed length label to determine the next hop. Distribution and management of labels among MPLS routers is done using a complex algorithm, Label Distribution Protocol (LDP), to ensure the various labels have a uniform meaning. MPLS is a protocol independent mechanism resident in network level switches with no application control. Hence higher layer QoS protocols such as Differentiated services can readily leverage on the management support provided by MPLS.

3.1.4 Support Services

Support services such as Policy Management, Accounting/Billing are also essential to the success of the deployment of QoS protocols. For example managing peering arrangements between various Internet Service Providers (ISPs) is important. Common Open Policy Service (COPS) [7] has been proposed for distributed policy management. COPS can also be used for

dynamic inter-domain policy exchange. Bandwidth Brokers can be third parties that manage SLAs and billing and accounting for various ISPs and enterprises.

3.2 Reservation Based Schemes

Intuitively, the most straightforward method for assuring service guarantees is to reserve resources according to an application's request. However, reservation based schemes entail admission control algorithms, reservation protocols, monitoring protocols, and signaling mechanisms. They are generally complex and require extensive network support. This affects their scalability and robustness and makes it difficult to deploy them on a large scale in the Internet. Nonetheless a number of innovative approaches have been proposed that can provide solutions to some pertinent technical problems towards the larger goal of supporting QoS in the Internet.

A Resource Broker model is presented in [15] as an enhancement to a user level Dynamic Soft Real Time (DSRT) CPU scheduler. The resource broker integrates reservation mechanisms for advanced or immediate reservations within the scheduler. The resource broker is responsible for the processing of brokerage requests (reservation, modification, allocation, release) and for the actual allocation. It maintains reservation state in a reserve graph, implemented as a multilevel two dimensional array, in coordination with a reservation server. Modification requests are serviced by releasing resources or performing admission control to grant additional resources and are accompanied by appropriate updates to the reservation graph. It is shown that the resource broker enhances DSRT performance by achieving fast and constant response time for all brokerage requests of DSRT. The speedup is obtained by separating scheduling work from brokerage to eliminate brokerage delays due to processing of scheduling events.

A multi-resource reservation algorithm [16] utilizes the resource broker model for an integrated approach to reserving and scheduling the resources with low resource contention. It adopts a component-based approach with Resource Brokers, QoSProxies and service components

as the main entities. The end-to-end QoS provided to the client is determined by the service quality achieved by each individual service component. Input and output qualities of each service component are represented as vectors of multiple QoS parameters. A dependency graph is generated with service components as its nodes and their inter-dependencies as its edges. The algorithm computes a resource reservation plan to reserve a minimum amount of bottleneck resources, i.e. resources with maximum conflicting requests, while deciding appropriate levels of input and output quality for each service component. Simulations with uniform and varying average request arrival rates indicated that the proposed algorithm works better, in terms of reservation success rate than a random reservation path selection algorithm.

An adaptive resource allocation (ARA) model [20] [22] is proposed to describe an application's adaptation capabilities and the runtime variation of its resource needs. Applications are structured as multiple interacting components with either sequential or parallel tasks. Processing is represented by a communication graph of a fixed set of components, with fixed precedence constraints. An application is modeled using a resource usage model that defines computation and communication needs, and an adaptation model that defines acceptable configurations and adaptation overheads. ARA is a middleware component that mediates between resource providers and clients to provide reservation based services. Performance is determined by allocation correctness and the latency of the decision mechanism. It is reported that ARA mechanisms benefit if the application characteristics and its current state is considered for detection and allocation decision. To capture the adaptation model of complex applications, a hierarchical adaptation model [22] is used which describes an application as hierarchical entities on top of its functional components. However for this mechanism to work detailed information about each application's adaptation capabilities is required. The decision overhead can cause a significant delay in the application reaching a steady state performance. Also per request

invocation of all concerned service components affected by a particular reservation allocation can be result in large overheads especially during adaptations.

An intelligent network architecture based on service brokers [27] attempts to provide application oriented reservations. Service brokers perform resource discovery and optimization of resource usage using application domain knowledge. An application oriented signaling protocol that handles the complete set of flow of an application is proposed for actual resource allocation. A hierarchical brokerage and resource management structure is suggested to handle resource allocation and sharing.

3.3 Adaptation Based Schemes

An alternative approach to reservation based schemes is to enable an application to perform satisfactorily given the existing resource constraints. Such a scheme is feasible only for applications that provide for adaptations such as multimedia applications. An adaptation can be complex, however since the complexity is generally at the middleware or application level at the end-hosts, the scalability of the approach is not adversely affected. Furthermore since it does not require network level support it can be deployed without considerable difficulty. However adaptation schemes can seldom provide hard service guarantees.

The Task Control Model [13] is a middleware layer of software components with translation mechanisms and applies control theoretical approaches. It monitors network traffic, as well as the resource and QoS demand dynamics at the end systems, and utilizes the measured samples for adaptations. It also quantitatively analyzes the stability and responsiveness properties of adaptive algorithms. This approach allows for global control of multiple concurrent applications, enabling coordination and resolution of conflicting resource requirements. It facilitates implementation of theoretical algorithms with strictly proved stability, fairness and adaptation agility properties. However the approach is not a very scalable solution due to its inherent centralization of control since the overheads of adaptation for all concurrent applications

add up. Furthermore the mechanism has to be comprehensive enough to cater to constantly evolving applications, while at the same time be light-weight given the limited resource availability. The utility of this model was demonstrated using a distributed visual tracking application. The adaptation model was able to maintain high precision tracking by trading off the image frame size. Without adaptation howeve, when the network throughput degrades to a certain degree, errors accumulate rapidly causing the tracking to lose the object.

COMM^{adapt} [19] is a communication infrastructure for on-line adaptation of a protocol's resource usage according to application requirements and resource availability. A software implementation offering resource management mechanisms necessary for dynamic configuration is evaluated. The issue of estimating runtime resource usage of complex applications with numerous communication streams is addressed. Different communication configurations were tested under varying load & resource condition to establish that no single configuration can provide for 'the best' performance under variable application requirements. Auto-configuration can lead to improved processing and network resource utilization as compared to user-specified configurations and connection-time configurations. Furthermore better compliance to specific application needs can be achieved by using customized configuration algorithms.

An end-to-end communication layer [24] [25] is proposed for service adaptation in the context of a single data stream. This layer configures the communication protocol based on application resource requirement and the network resource availability. The communication layer utilizes a "payoff based" technique for evaluating benefits of various tradeoffs to the application to select a communication configuration that is suitable for the current state of application. Payoff functions are designed to enables application to place a quantitative value on communication parameters relative to the costs incurred to satisfy the parameters. Hence application requirements are characterized by payoff curves while the network behavior is characterized by load-loss/service availability curves. Given these inputs, the communication layer is able to perform

adaptations to optimize resource usage and the total payoff achieved by the application. Experiments conducted for a virtual environment application indicated that payoff based adaptation is able to achieve a higher aggregate payoff than a static configuration by performing tradeoffs between reliability and transfer rate.

Network aware applications require feedback about resource availability, which can be achieved implicitly or explicitly. Implicit feedback is periodic and provides for incremental feedback. It can be sometimes difficult to interpret. Explicit feedback can be periodic or event-driven. However since explicit feedback requires network support, it can be used to complement implicit feedback for QoS adaptations decisions. A layered architecture with an adaptation layer for implementing the implicit feedback is proposed in [28] for developing a network aware architecture. The adaptation layer has a unified API for application level, transport level and network level feedback mechanisms for network service quality feedback.

Three adaptation strategies, Model based, Performance based and Feature based, are studied in [30] for network aware applications. In Model based adaptation, an application models its performance as a function of parameters characterizing its runtime environment. Performance based adaptation involves monitoring of application performance to trigger adaptation. In Feature based adaptation a particular critical feature of application is monitored for adaptation decisions. Model based adaptations rely on explicit feedback from network while both performance and feature-based adaptations work well with implicit feedback. Feature based adaptations are easier to implements since probing for multiple parameters is not required. However it is difficult to identify a feature to represent application state. Comparison between these adaptation models indicates that performance and feature based adaptations are similar with medium to high robustness, medium to high adaptation delay and suited for local incremental adaptations. Model based adaptations on the other hand have low robustness, low adaptation delay and are better suited for local/remote coarser adaptations.

In [35] [36], S. Bhatti et al. discuss a network model that allows application flow state and network QoS to interact. The model produces reports in the form of QoS summaries indicating which flow-states are currently supportable by the network. This compatibility between application's flow-states and the network QoS, can be used by applications to make adaptation decisions. The adaptation is based on relative rather than absolute compatibility using state compatibility values (SCV), thereby opting for best fit rather than trying to achieve a particular QoS level. An application adaptation function evaluates suitability of a QoS state based on SCV. This approach means that application need not totally rely on the user to set correct preferences and can utilize compatibility values to adapt. However the large number of possible parameters of which only a small dynamic subset of which is relevant to an application, requires the production of customized reports to provide relevant information to application with minimum overhead.

In [44] F. Chang et al. present an application-independent adaptation framework with a tunability interface and a virtual execution environment. The tunability interface exposes adaptation choices as application configurations while encapsulating core application functionality. The virtual execution environment emulates application execution under diverse resource availability enabling off-line collection of information about resulting behavior. It tries to alleviate the burden on application developers of providing explicit specification of both resource-utilization profiles and application adaptation behaviors. The tunability interface is exposed using language-level annotations that allow a preprocessor to generate monitoring and steering agents. The virtual execution environment is implemented using API interception that emulates different availability of system resources by continuously monitoring and controlling application requests for system resources. The active visualization application, a client-server application for interactive image viewing, is used to test the framework. Control parameters exported by the tunability interface were image resolution, image size and compression technique. Performance of framework was evaluated using transmission time and response time.

The advantage of this approach is that application developers are required only to expose the adaptation structure of the application using the tunability interface. The execution system obtains application behavior profiles using the virtual execution environment, and incorporates them into adaptation decisions at run time.

3.4 Other Optimization Schemes

Several optimization schemes have been proposed to enhance the Internet infrastructure for better QoS support. Such schemes involve monitoring tools for getting the network and system resource state in real time, integrating media encoding and transport for better adaptability support, considering QoS state information for routing decisions etc.

In [14] K Nahrstedt et al. propose an integrated data compression algorithm with a transmission protocol to transport the compressed stream. This integrated approach tries to address the mismatch between properties of encoder/decoders and transport protocols. It dynamically tracks the channel state and feeds the encoder with inputs on the channel state. The suggestions include modification to TCP/IP for delay-constrained traffic, loss tolerant algorithm for compressing video data and rate control algorithm for binding the coder and transport. Modifications to TCP proposed include elimination of retransmission, using partial ACKs and using RTP headers for time stamping and sequencing packets. The ability of the controller to track changes in channel states and the quality of video signals delivered allows it to utilize the Internet bandwidth intelligently to transfer higher quality of video signals.

A hierarchical QoS routing algorithm has been proposed in [17]. This work deals with QoS state aggregation, representation and end-to-end routing based on this information. The concept of scalable hierarchical routing with inter-domain and intra-domain routing is enhanced for QoS support using a delay bandwidth plane. This plane is divided into areas reflecting the difference in admissibility. Routing is done based on ticket-based probing (TBP). The algorithm is compared with flooding (FD) and shortest path (SP) algorithm and is found to perform better

than SP but worse than FD in terms of success ratio with respect to bandwidth and delay. However TBP is able to keep the number of ticket probes to a much lower level than FD.

A generic application component model proposed in [18] splits the QoS management into resource management and service management. QoS specifications in the form of application configurations are translated using a QoS compiler and embedded into the middleware running at each host. The middleware performs adaptations in application configuration with different granularities for maintaining QoS. The middleware system has to interact with underlying OS and network QoS support to provide QoS support.

The Interactivity Layer Infrastructure (ILI) [21] is proposed to support applications such as a distributed laboratory with frequent dynamic client arrivals and departures and variable end user needs. It adapts to changes in the behavior of the computational instruments and end users across a shared and dynamically changing set of computational nodes and networks. ILI models applications as sets of tasks communicating via events. The flow of events from task to task represents task linkages and forms a task linkage graph. Workgroups, i.e. sets of linked tasks, are mapped to computational units such that QoS specifications are met. Task to workgroup assignments are constructed dynamically and reassigned as task linkage graphs change. ILI's online adaptation heuristics are applied cyclically in three phases, detect state to monitor for QoS constraint violations, predict next state to determine appropriate reconfiguration if necessary and shift state to implement reconfiguration. Configuration attempts to achieve load balancing by maintaining event rates in accordance to QoS specifications. The task reassignment during reconfiguration has to be done carefully since if it is done too often it can lead to thrashing.

Remos [29] [31] [33] is a query-based interface for network aware applications to gain information about the execution environment. It provides statistical reliability and variability measures to add system awareness to application. Remos supports flow based queries and queries about network topology. It categorizes applications based on flow characteristics into fixed flows,

variable related flows and independent flows for resource sharing. It provides a portable interface for network resource monitoring across different network architectures to support explicit feedback to network aware applications. A hierarchical SNMP based data collector [33] is proposed for collecting raw network information to be used by Remos. A master collector breaks the application query, distributes it to data collectors on relevant sub-nets and aggregates the information to provide a network independent resource information access to applications. This collector is able to extract information more concisely, but comes with an overhead on the routers and limits the scalability of this approach. Further, relatively high latencies of SNMP queries might limit the use of information for some real-time applications. However developing a uniform API for a variety of diverse applications on heterogeneous network environments and protocols, all of which are constantly evolving, raises many technical challenges, only a subset of which can be realistically handled by such an interface.

In [34] [37], A. Watson et al. discuss methods to determine perceived QoS in multimedia conferencing applications. The complex relation between perceived quality and quantifiable parameters is analyzed at different packet loss rates, and various parameters affecting perceived quality are studied. Intelligibility and frame rate are identified as the biggest determinant in perceived speech and video quality respectively. It is further observed that both these parameters are closely related to packet loss rate. However this study is conducted by comparing a particular coding technique against packet repetition mechanism, and it has to be validated by conducting experiments with different speech encodings with a variation in material and loss patterns.

In [45], K Nagao et al. presents a semantic transcoding technique based on external annotations for content adaptations according to user preferences. Typically user profiles are insufficient for transcoders to recognize and modify fundamental document features. Annotating Web data using XML formatting can provide transcoders with necessary and sufficient information to perform efficiently. Three approaches for annotating documents, linguistic,

commentary and multimedia, are proposed. Linguistic annotation is to make text in web page understandable using a new tag set. Commentary annotation annotates nontextual elements like images and sound with additional information such as tagged text. Multimedia annotation refers to annotation techniques for multimedia data such as text summarization of video. A transcoding proxy interacts between the web server and annotation server for servicing a client request. It is responsible for maintaining user preferences, gathering and managing annotation data and activating and integrating transcoders.

3.5 Discussion

This chapter discussed in detail various approaches proposed to address the issue of providing Quality of Service support to multimedia applications in the Internet. The scale and heterogeneity of the Internet makes it difficult to deploy network level protocols for end-to-end QoS support. Furthermore, most of the proposed network protocols have their strengths and weaknesses and are best suited in specific application domains. However network level QoS support is vital for providing hard service guarantees for distributed applications. This implies that these protocols will in all probability coexist. Arriving at a standard scheme for their interoperability though is proving to be a painfully slow process. Reservation based schemes typically assume some form of network support and utilize application information to develop resource reservation algorithms. However due to the vast number of issues involved in admission control, signaling, coordinated resource allocation and de-allocation, resolving reservation conflicts, these schemes become unwieldy and prove to be an overkill for most applications that do not require hard service guarantees. Hence reservation based schemes are suited for mission critical applications and will have a limited deployment in the Internet. Adaptation based schemes utilize the adaptive behavior of applications to make the application operate in a configuration that can be best supported given the existing resource availability. Runtime adaptations are enforced based on information gathered by actively monitoring local/remote network and system

state. These schemes are implemented in the application or middleware level and are comparatively lightweight. Adaptive QoS significantly improves the performance of applications over best effort service, however no service guarantees are made to the application. If implemented with proper application knowledge it provides an efficient and effective solution for adaptive distributed applications.

Chapter 4

Application Level Adaptive QoS Management

4.1 Architecture

An architectural overview of the application level adaptive QoS management mechanism for multimedia applications is presented in Figure 3. The framework uses RTP implemented over UDP and IP-multicast to transport application data. It uses RTCP to monitor and exchange remote client information and statistical transmission / reception information. Local system and network state information can be monitored with the System/Network Monitor using standard protocols such as Simple Network Management Protocol (SNMP) [8], that hides the heterogeneity in client device and network connection. The Application Interface enables the Manager to enforce appropriate adaptations to enable the application to operate optimally under dynamic resource availability. The manager, guided by a resource management policy, is able to make intelligent decisions to adapt the application's traffic profile based on the local and remote information.

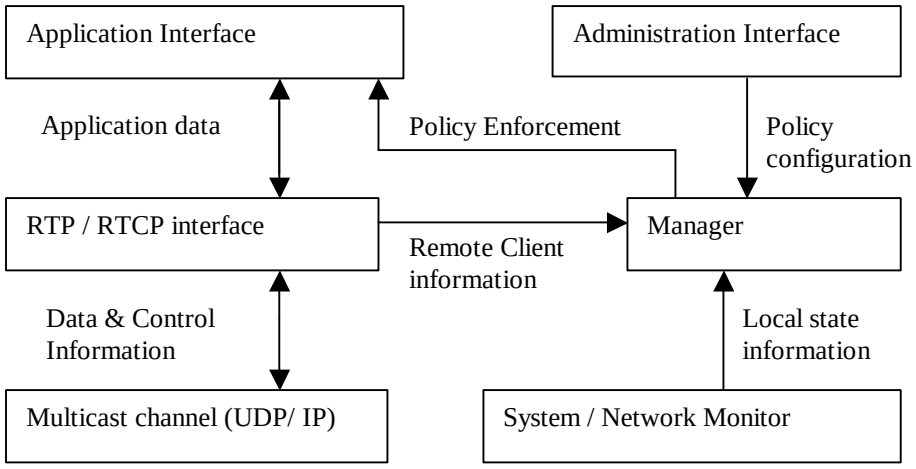

Figure 3 Policy Based Adaptive QoS Mechanism

The policy is defined with a good understanding of the application's configuration and requirement. It is configured at the client using the Administration Interface. This is essential for

two reasons. Firstly the policy is application specific while the adaptation framework is generic and hence the policy effective at the client systems has to be configurable. Furthermore to support deployment in a heterogeneous environment different policies may be required at different client systems connected to the same application.

4.2 Implementation

The adaptive QoS management mechanism is integrated into the two main application entities, the streaming server and the multimedia client. Figure 4 depicts the adaptive streaming multimedia application model consisting of the media server, the multimedia client and the integrated adaptation algorithm. The RTP channel is used for actual application data transport while the RTCP channel is used for exchanging transmission/reception statistics. Rate based adaptations are performed at the server for each of media type in the streaming application. Modality based adaptations are performed at the client in an independent isolated manner.

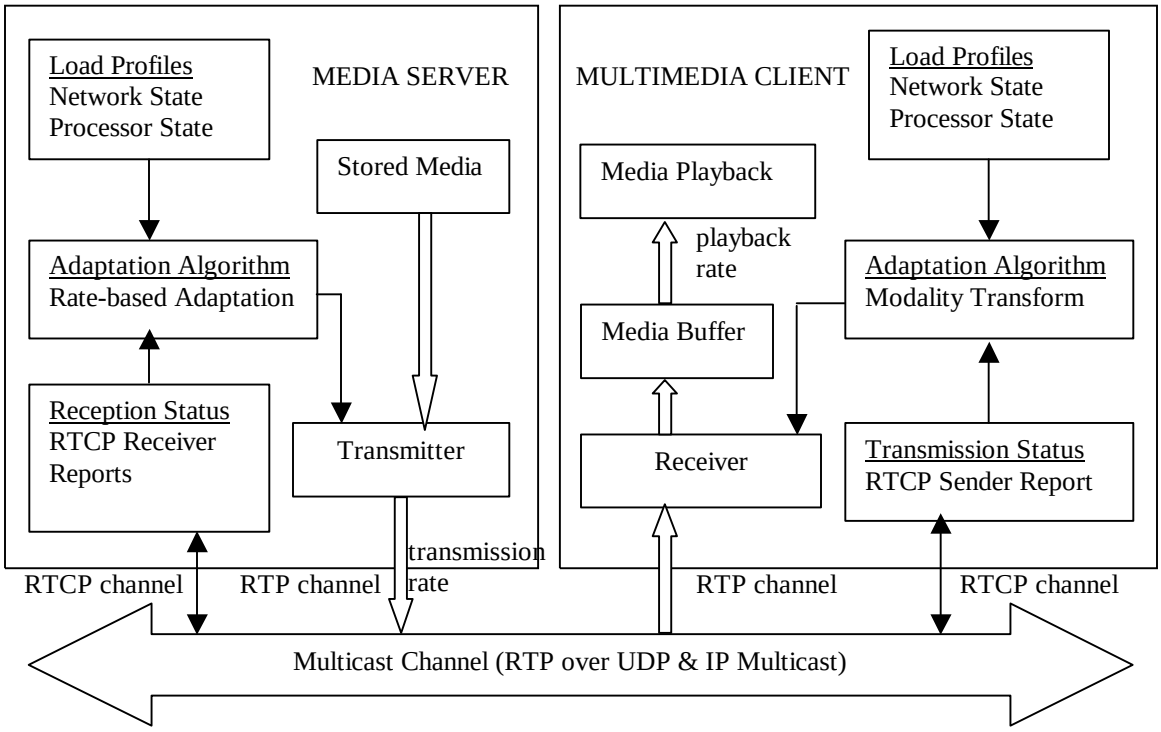

Figure 4 Adaptive Streaming Multimedia Application Model

4.2.1 Media Server

A multithreaded implementation is adopted for the media server to stream multimedia data using RTP over UDP and IP-multicast. The server initiates multiple RTP sessions one for each media type - video, audio and text. Handling each RTP sessions involves execution of threads to perform the following functions

- Transmitting application data via RTP,
- Generating RTCP Sender Report and transmitting to clients subscribed to the media.
- Receiving and processing RTCP Receiver Reports to maintain aggregate client status.

Since each media is handled in a separate session, transmission statistics for each media type are maintained independently. This facilitates adaptation since each media type potentially has different resource requirements, and hence allows for different adaptations, which can be handled independently. However sometimes different media types, though transmitted in different media channels, have to be played back in synchronization. This is handled by identifying the different media types with the same synchronization source in the RTP message header.

The server aims to optimize the overall performance of the streaming application for individual client groups by controlling the frame rate to minimize the average packet loss. It monitors the state of the clients subscribed to each media type using the RTCP receiver reports corresponding to the media type, and maintains a running average of the packet loss suffered by each client group. It then uses this information to perform rate-based adaptations that affect the quality perceived by the entire group.

4.2.2 Multimedia Client

The multimedia client subscribes to the multicast channel corresponding to the media channel of interest determined by user preferences. The client also has a multithreaded implementation with independent threads performing the following tasks for each RTP session

- Receiving application data via RTP.
- Generating RTCP Receiver Report to inform the server about the reception status.
- Receiving and processing RTCP Sender Reports to maintain the transmission statistics.

Local client state is defined by two parameters, processing load and network congestion. Each media type's resource requirements are also specified in terms of its processor and network resource requirement. The policy-based manager determines whether a media type is supported at the client and performs run-time adaptations to switch to the best supported service level. This is done periodically to ensure that the playback rate of the media is maintained to provide acceptable performance.

Quality of Service perceived by the end user is determined by the packet loss incurred due to the processing overload and the network congestion. Application and media specific thresholds are specified for processor and network bandwidth. Processor overload threshold is determined by the processing requirement of the application's selected media type. Network congestion threshold is determined by the bandwidth requirement of the selected media type. These processing and network bandwidth requirements are converted to a fraction of the processing capability and network bandwidth of the client respectively. They are then added to the processor load profile and network load profile, which are also expressed as fraction of the processing capability and network bandwidth of the client respectively. If the sum exceeds the capacity available, corresponding threshold is violated and packet drop results. For example in our simulation setup a Video on Demand application while operating in best resolution video mode loads the desktop terminal to about 30 percent and consumes 40 percent of a LAN's bandwidth capacity. On the other hand the Distance Learning application while operating in the best resolution mode loads the desktop terminal to about 20 percent and consumes 15 percent of a LAN's bandwidth capacity.

The media buffer is implemented as a synchronized queue at the client. Two threads, RTP receiver thread and Application thread have synchronized access to the buffer. The RTP receiver thread is fed with a network load profile to simulate network state. If the network resource requirements of the media type violate the threshold, given present network state, the receiver thread drops the packet to simulate loss due to network congestion. Else it adds the packet to the media buffer corresponding to this media type. The Application thread is fed with a processor load profile to simulate the processing load at the client. It processes packets from the buffer at a uniform play back rate corresponding to the media type. However if the processing overhead of the media type violates the threshold, packets are not processed during that playback interval. This results in packets being queued up in the buffer thereby increasing delay, adding jitter and decreasing the Quality of Service perceived by the client. If this processing overload persists for a significant period it will result in the media buffer being filled up and result in packet loss. It is important to note that since we are considering real-time multimedia data, the data that has been queued for the longest period, head of the queue, is the most outdated and hence of least value for the application. In our design this packet is dropped instead of tail drop where the latest arriving packet is dropped.

Modality transformation is implemented by defining a threshold on the cumulative packet loss suffered by the application. Violation of the threshold implies that the media type can no longer be supported by the client with existing resource constraints. The client then subscribes to the multicast channel corresponding to a media type with the next lower requirements in terms of processing overhead and network bandwidth. In absence of such a mechanism the user would have experienced periods when the application would stall due to excessive packet loss from network congestion, CPU overload or both. The adaptation mechanism decreases the packet loss and results in a more efficient utilization of network and processing resources. This also translates

in to a better Quality of Service to the end user. The assumption is that a modality transformation is possible.

4.3 Operational Overview

The operation of the adaptation mechanism is explained in terms of the interactions at the multimedia client and the streaming server depicted in Figure 5 and Figure 6.

4.3.1 Client Side Interactions

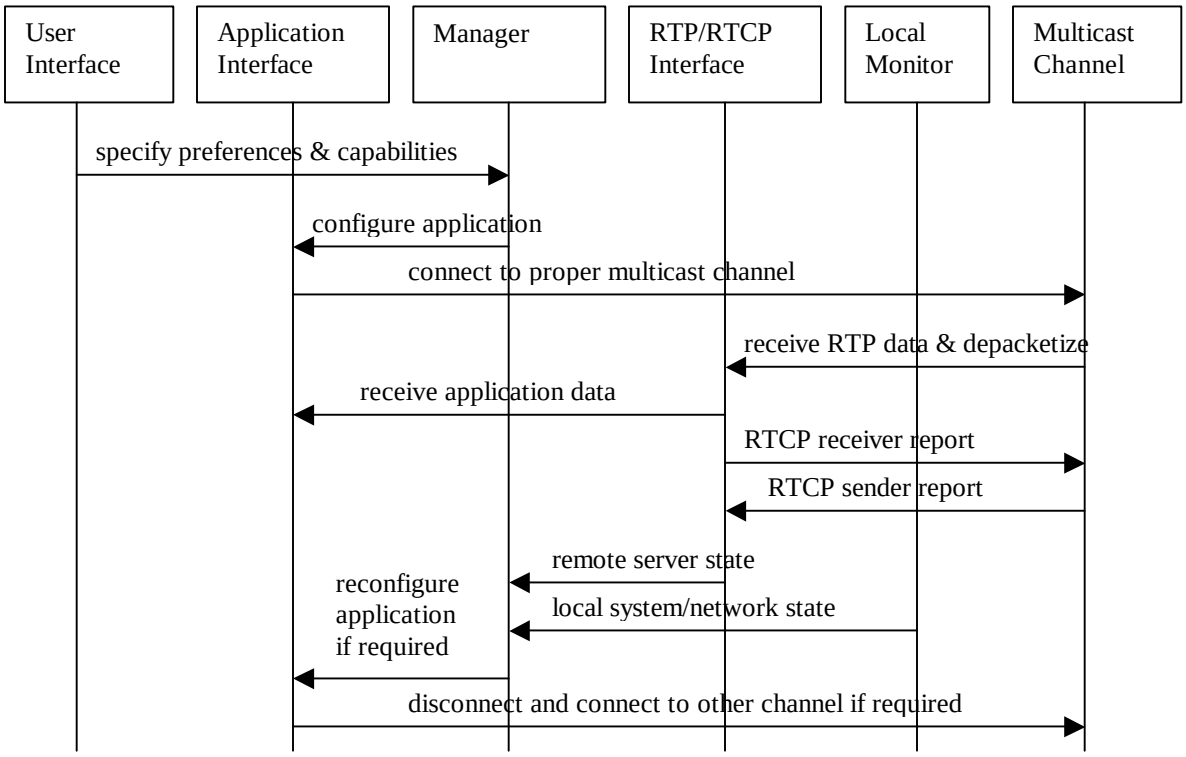

Figure 5 Interaction Model for Multimedia Client

Clients connecting to the multimedia application indicate their preferences and capabilities in terms of desired media type and device specification using the User Interface. The client's desired media type determines the target service level that the Manager tries to achieve and maintain. The Manager selects the appropriate media channel and configures the application to connect to the corresponding Multicast Channel using the Application Interface. It receives transmission statistics from the RTCP Interface in the form of RTCP Sender Reports that are

periodically transmitted by the server on the control channel. The Local Monitor provides the local system/network state information to the Manager. The Manager uses the local network congestion, processor load and the resource requirements of the selected media type while making adaptation decisions.

The network and processing load generated at the client due to the streaming application depends on the media type that the client is subscribed to and the rate at which the media is streamed to the client. Threshold values are defined for network and processor utilization to identify network congestion and processor overload. The operating states for the client are categorized as Normal, Congested and Overload. A client is considered to be operating in Normal state if the client is able to accept and process the media data while remaining within the threshold limits of processor and network utilization. A client operates in the Congested state if the traffic generated by the media stream that the client is subscribed to, causes network utilization threshold to be exceeded indicating network congestion at the client end. A client is categorized as operating in the Overload state if the processor utilization or both the processor utilization and the network utilization exceed their thresholds.

4.3.2 Server Side Interactions

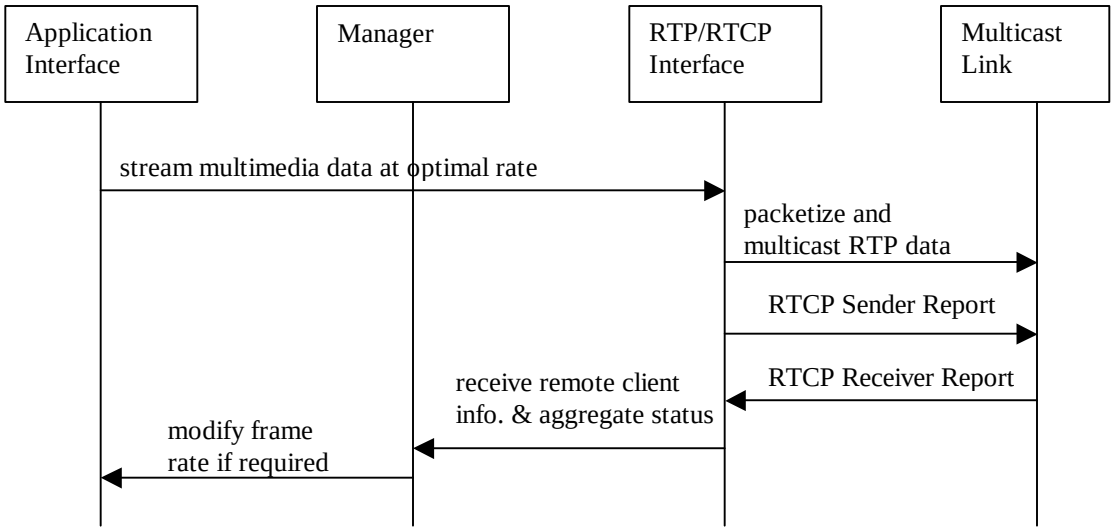

Figure 6 Interaction Model for Media Server

The media server streams audio, video and text media on three media channels. The permissible values for the resolution / fidelity and the playback rate for the media types enables the server to perform rate-based adaptations. The server has multiple concurrent media channels each of which operates independently as shown in Figure 6. The Manager receives reception statistics from the RTCP Interface in the form of RTCP Receiver Reports that are periodically transmitted by the clients subscribed to this media channel and maintains an aggregate status of this client group. It packetizes the media frame and transmits it at the optimal transmission rate on the Multicast Link. If the Manager observes that the client group is suffering from an excessive packet loss, it senses that the network congestion is on a shared link and reduces the transmission rate within acceptable limits using the Application Interface. This provides a “pseudo” congestion control and at the same time helps to keep the application operational. In the absence of such a mechanism, application operation would have worsened the network congestion thereby causing a degradation of performance as well as that of other applications sharing the network resources. Thus the media server adjusts the rate of data transfer to obtain an optimal aggregate performance for the clients subscribed to that media channel.

Chapter 5

Experimental Evaluation

5.1 Simulation Models

A simulated heterogeneous device/network test-bed and streaming application are modeled to evaluate the performance enhancement achieved by the adaptive QoS management approach presented in this thesis. The experimentation setup consists of a multimedia streaming server and multiple multimedia clients. The clients are heterogeneous with respect to their network connection bandwidth, processing speed, and memory size. Load profiles are generated at each client to simulate the processing and network load condition. Application characteristics, client device and network heterogeneity are simulated by developing traffic models to simulate applications typically observed on the Internet.

5.1.1 Application Models

Two streaming multimedia applications, Video On Demand and Distance Learning application, are modeled and simulated to test the performance of our adaptation mechanisms. Video On Demand (VOD) requires higher resolution video and higher (compact disc like) sound quality for optimum performance. However in constrained resource conditions a low- resolution video with stereo sound would be acceptable. The distance learning application streams pre-recorded lecture series to geographically distributed students and is in effect a video broadcasting (VBcast) application. It provides acceptable performance with low resolution video and stereo to mono audio quality.

Frame size and frame rate are identified as the parameters defining the application's data transport requirements. For video, the resolution (pixels per frame) and encoding (bits per pixel) define the frame size. Frame rate for video can be typically altered form 30 frames per second for very good quality video to as low as 15 frames per second for an acceptable quality. The bit rate

encoding for the audio signal determines its fidelity and frame size. Audio signal are highly sensitive to delay and jitter and hence have a smaller range of permissible frame rates. Collectively, frame size and frame rate modification translates into a range of bandwidth requirement for each media type for acceptable performance. The typical values for the applications considered are listed in Table 1.

Quality		Good	Fair	Acceptable
Appln	Media	Typical Bandwidth Requirements		
VOD	Video	4Mbps	1.5Mbps	256Kbps
	Audio	128Kbps	96Kbps	64Kbps
VBcast	Video	1.5Mbps	512Kbps	128Kbps
	Audio	64Kbps	32Kbps	16Kbps

Table 1 Application Bandwidth Requirements

5.1.2 Client Device Models

The heterogeneity in devices employed by end-users, for Internet access and network computing, is simulated by modeling different device types. Device types are identified by their processing speed and memory size. These parameters are important as they define the capability of the client to support the application's buffering and media processing requirements. Three representative device types with typical parameter values are listed in Table 2.

Device type	CPU speed	Memory size
Desktop	800MHz	256MB
Laptop	500MHz	128MB
Palmtop (PDA)	200MHz	64MB

Table 2 Device Specifications

5.1.3 Client Network Models

Numerous network access technologies with varying access bandwidths coexist in the Internet today. The prevalent types considered in our experiments with their typical bandwidth values are listed in Table 3.

Connection type	Bandwidth
LAN	10Mbps
Cable modem	4Mbps
DSL	1.5Mbps
ISDN	128Kbps
X2 modem	56Kbps
Wireless	19.2Kbps

Table 3 Network Connection Specifications

Each media type is characterized by its requirements in terms of the processing resources, network bandwidth and memory buffer size. The processing and network bandwidth requirements of a media type depends on the frame size and the playback rate, parameters that are determined by the selected media type and the quality level of the media type. On a client device the media buffer is allocated as fraction of the available capacity, and hence the device type determines the number of packet that can be buffered at client. To simulate the processor and network load at the client, processor and network load profiles are generated at the client. The load profiles are specified as a percentage of the maximum processor and network resources, which in turn depends on the device and connection type of the client.

5.2 Adaptation Mechanism

The adaptation mechanism the packet loss at a client to decide whether the client is able to support the current media type to which it is subscribed. Packet loss can be caused by network congestion and processor overload. Packets lost due to processor overload are packets that are delivered to the client but are discarded as they could not be processed in time. This packet drop also results in wasted network resources as these packets are discarded after reaching the client. To capture this loss we define a “application efficiency” as the fraction of the total packets delivered to the client that are actually processed by the client. It should be noted that this application efficiency figure is not valid when network losses dominate the packet loss figure.

The adaptation mechanism defines “upgrade” and “downgrade” media operations to implement modality transformations. Upgrades imply switching from text to audio media or from audio to video media. Downgrades imply switching from video to audio media or from audio to text media. A sliding window (fixed time interval) algorithm is used to calculate packet loss that is compared against a fixed loss threshold to trigger downgrade adaptations. If for one complete window time interval no packet loss is detected and the application is operating in a downgraded configuration an upgrade adaptation is effected. The time interval and the loss threshold are application specific parameters and can be assigned values based on the dynamism and loss tolerance of the application. For example the video media is highly loss tolerant and less susceptible to distortion due to latency and jitter than the audio media. Hence in our simulation setup for the Video on Demand application the video media has a loss threshold of fifty packets tracked with a time window size of ten seconds. The audio media on the other hand has a loss threshold of twenty packets tracked with a time window size of five seconds.

5.3 Experimental Setup and Verification

Two experiments were performed for evaluating the application level adaptive QoS management mechanism using the two applications described above. In both experiments three heterogeneous clients subscribe to the application. The clients differ in their device types and the type of connection to the network. Different network and processor load profiles are used in the two experiments. Application performance is measured in both the optimized (i.e. with the adaptation mechanism) and unoptimized case (i.e. without the adaptation mechanism). The packet drop profile and the application efficiency observed at the clients are plotted. The packet drop profile is plotted on a logarithmic scale while the application efficiency is plotted on a linear scale. The points of modality transformation are marked on the plots. An ‘upgrade’ operation is marked with a ‘⬮’ and is numbered as U1, U2, … and so on. A ‘downgrade’ operation is marked with a ‘■’ and is D1, D2, … and so on.

5.3.1 Experiment 1

The first experiment has three clients connecting to a Video on Demand application. The clients configurations are:

- Client 1: Desktop terminal on the same LAN as the media server.
- Client 2: Laptop device connected to the Internet via a cable modem connection.
- Client 3: Desktop device connected to the Internet via a DSL connection.

Figure 7 and Figure 8 are the plots of the results observed at the three clients for the first experiment.

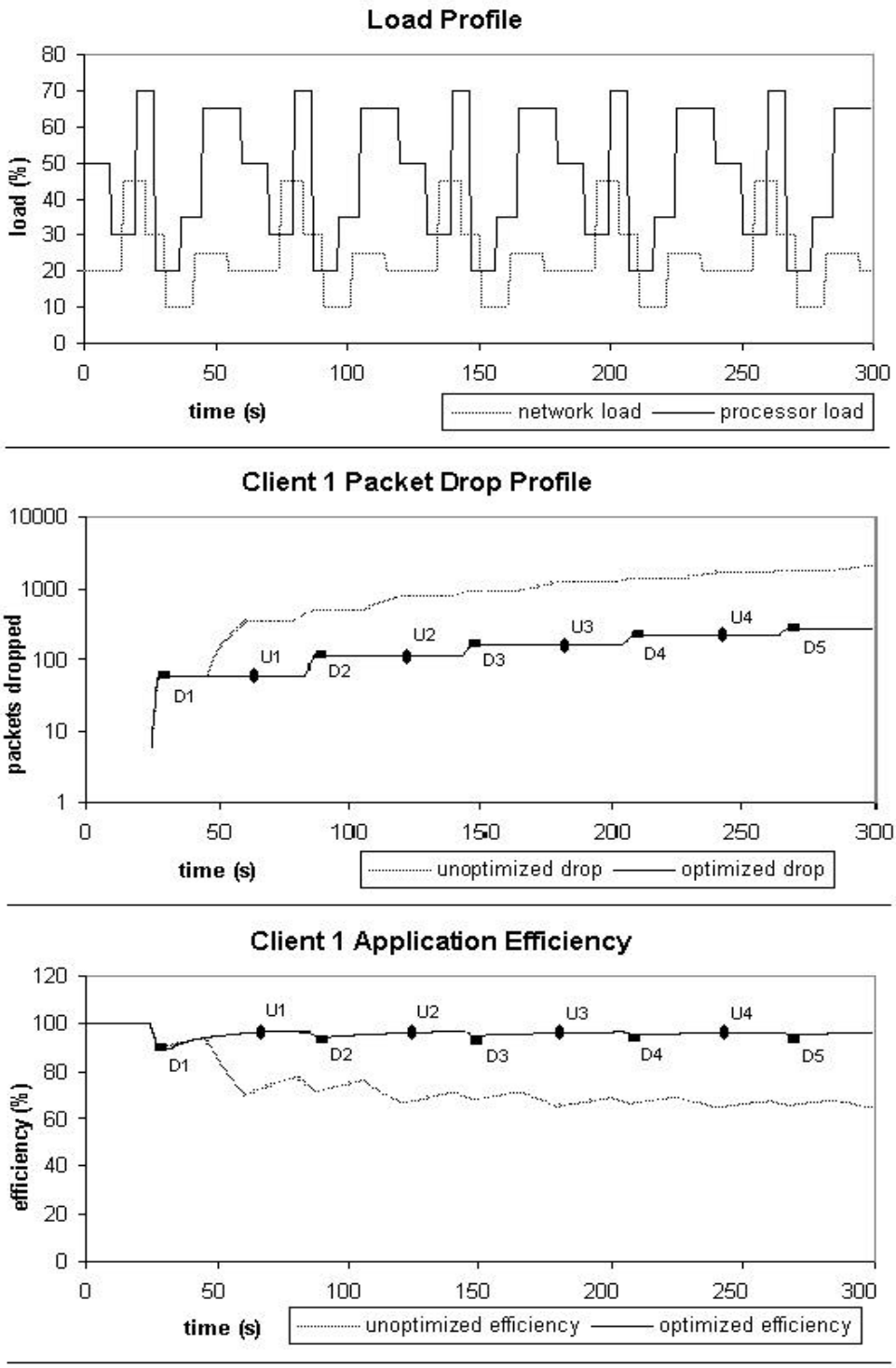

Figure 7 Performance of Client 1 in Experiment 1

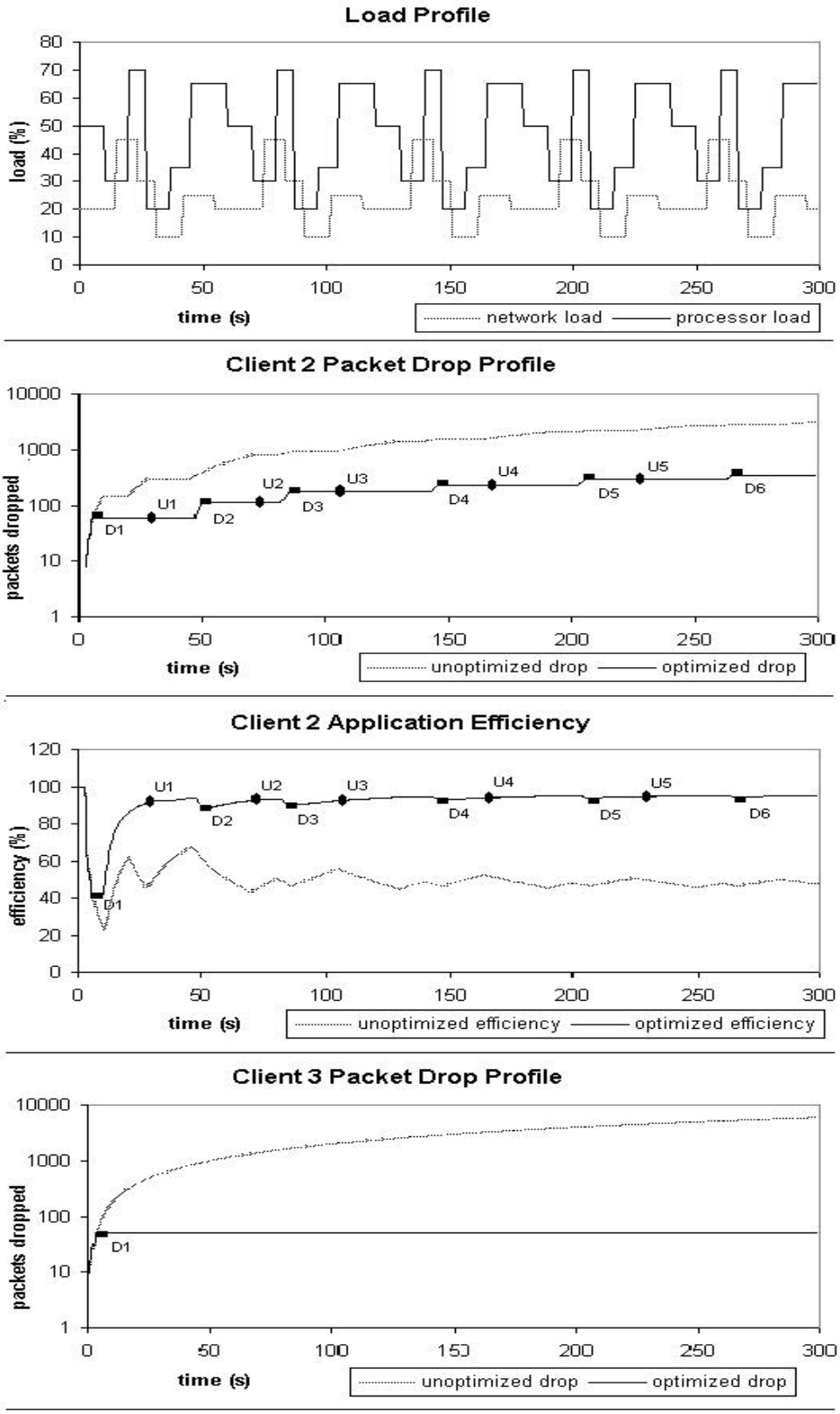

Figure 8 Performance of Client 2 and Client 3 in Experiment 1

It can be observed from the packet loss plots for the three clients, drawn on a logarithmic scale, that using QoS adaptations all three clients have a significant decrease in the packet loss. This is because the adaptation mechanism tracks resource availability and media resource requirements and can enable the application to operate in the best supported configuration for most of the execution period. After the initial sharp rise in packet loss and corresponding drop in application efficiency a downgrade transition is triggered at D1 as seen for all client. At U1 an upgrade transition occurs for clients 1 and 2 since these client are supporting the downgraded configuration without incurring packet loss. As can be seen when overload condition recurs the application is again downgraded at D2 for client 1 and 2.

Configurations of client 1 and 2 are generally able to support the video media type, and switch QoS levels only because of intermittent network congestion or periodic overload. Client 3, connected via a DSL link, is not able to support the high resolution video media. This can happen if the user unwittingly tries to subscribe to an application configuration that cannot be supported by the device/network connection. This can result in a stalling of the application as well as severe network congestion at this client, and can affect other networked applications that this client might have subscribed to. The adaptation mechanism averts this by automatically switching to a lower application configuration to avoid subsequent packet loss, as indicated in the plot.

Plots in Figure 7 and Figure 8 show that the application efficiency for clients 1 and 2 improves while using the adaptation mechanism. This means that the application is using the network resources more efficiently and processes a higher fraction of packets received at the client system. This also implies that this application shares network resources in a fair manner with the other applications by implementing “pseudo congestion control”. For client 3 the network losses dominate the packet loss figure. Application efficiency in this case is no longer valid as not many packets are getting through to the client system to evaluate how efficiently they

are being processed. The plots confirm that the adaptation mechanism is able to control the packet loss and maintain a high efficiency for the entire duration of the simulation.

5.3.2 Experiment 2

The second experiment uses three clients connecting to a Distance Learning application. The client configurations are as follows:

- Client 1: Desktop terminal connected to the Internet via a cable modem connection.
- Client 2: Laptop device connected to the Internet via a DSL connection.
- Client 3: Palmtop device connected to the Internet via a ISDN link.

Results for this experiment are plotted in Figure 9 and Figure 10

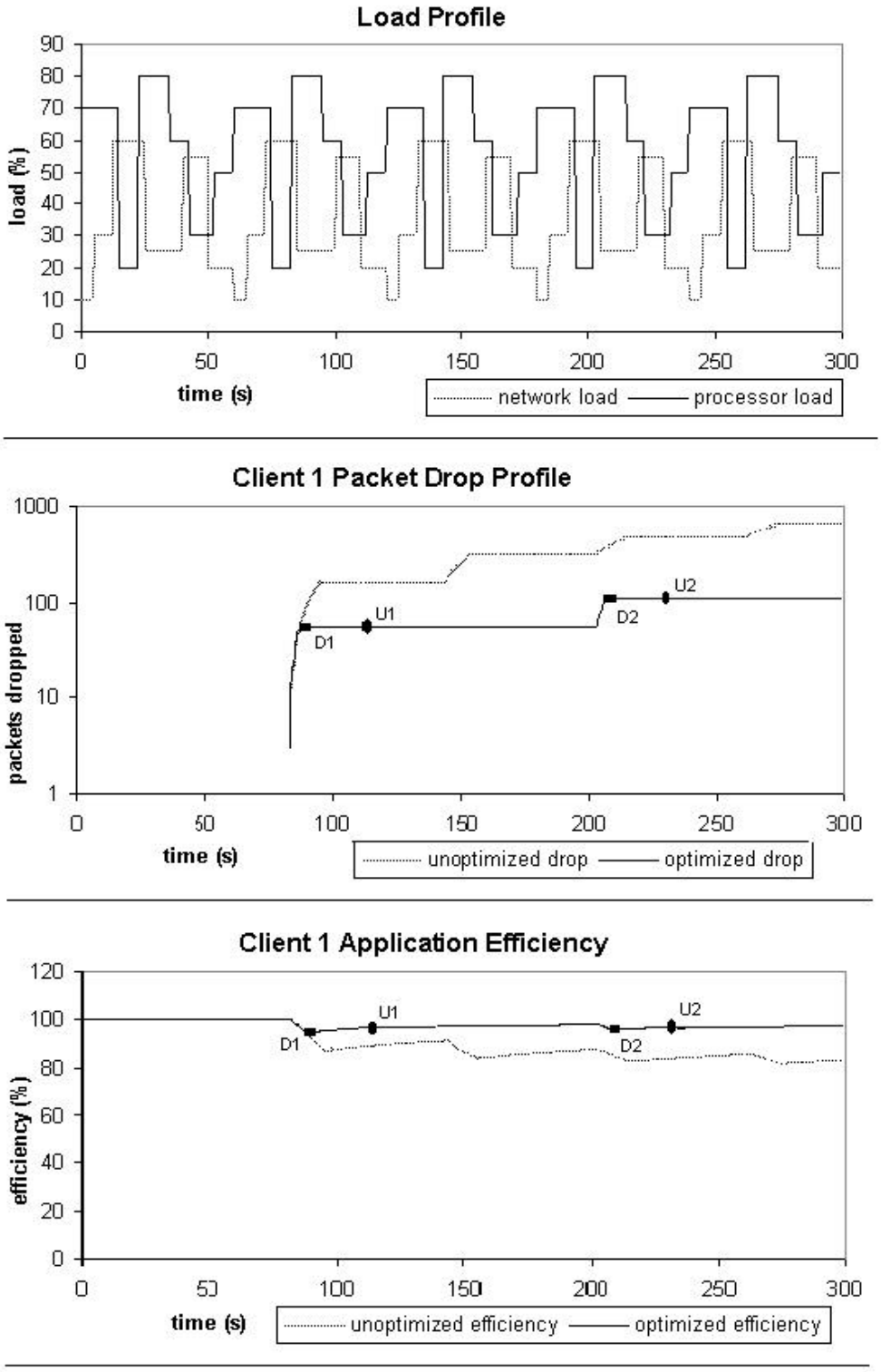

Figure 9 Performance of Client 1 in Experiment 2

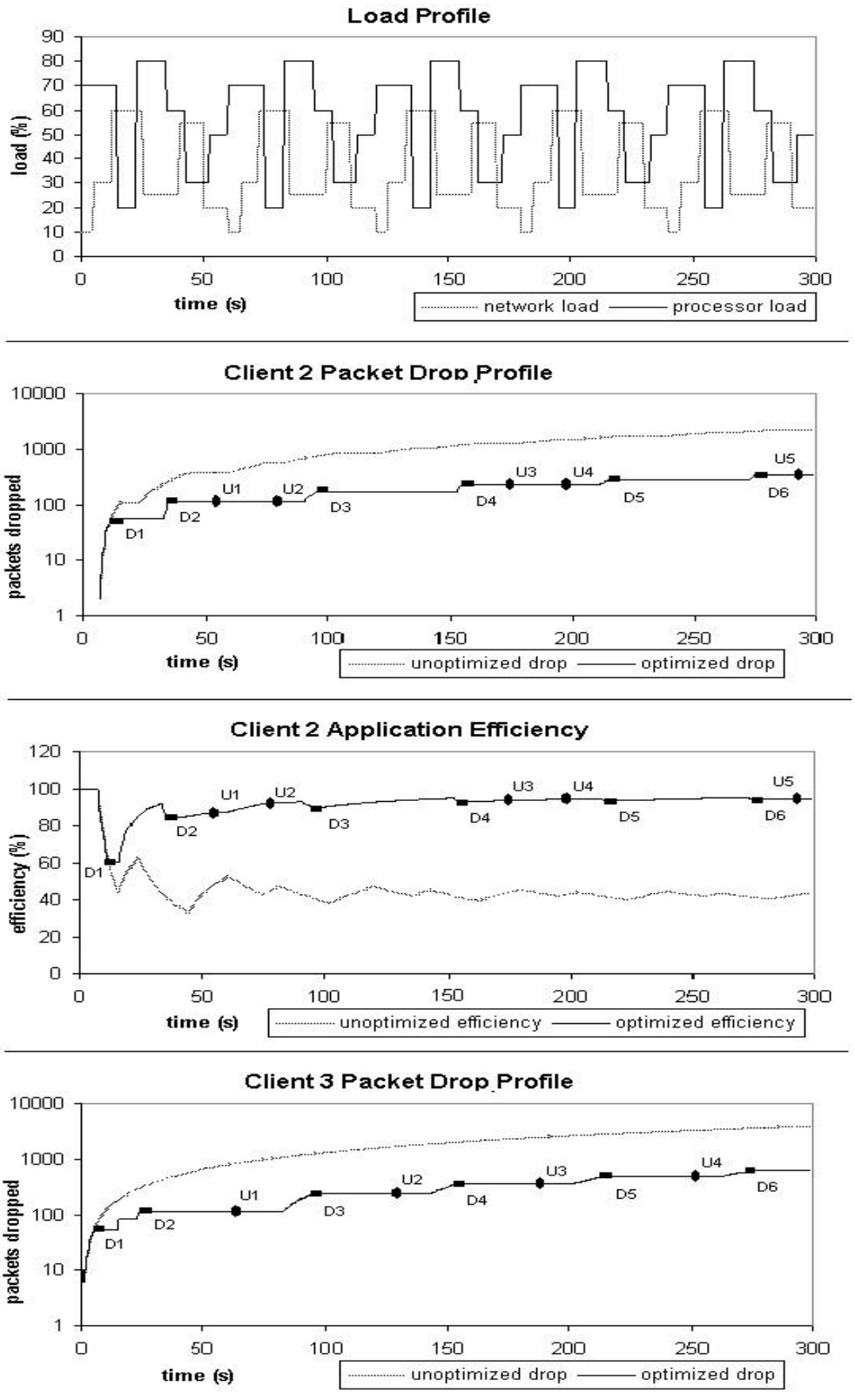

Figure 10 Performance of Client 2 and Client 3 in Experiment 2

In this experiment the packet loss plots again indicate that all three client have a better overall performance while using the adaptive QoS management mechanism. In the experiment client 1's configuration best supports the desired service level under the existing network congestion and processing load conditions. This is indicated by the small number of adaptations required. More importantly the complementary effect of server side adaptations is clearly evident in plots for Client 1. Between points U1 and D2 client 1 is receiving the video media in both optimized and unoptimized case. In optimized plot the packet loss doesnot increase between these points, however an increase in packet loss is seen in the corresponding region of the unoptimized plot. This is explained by the fact that the since all three clients suffer loss while they are initially subscribed to the video media the server drops the video frame rate from "good" quality to "fair" quality (refer Table 1). Client 1 in optimized case doesn't suffer packet loss since it is able to support "fair" video though it cannot support "good" video as is evident in the unoptimized case.

Client 2's configuration is just about able to support the desired service level and results in high number of switches between service levels. Note that the switching in this case is between three service levels as indicated by two consecutive downgrades followed by two consecutive upgrades in the plots. Frequent adaptations can be undesirable and a heuristic algorithm for calculating delay before upgrading might perform better than a fixed delay used in this algorithm. In contrast client 3 settles to the lower service level, best supported by its configuration, and operates with fewer number of adaptations triggered by the dynamic congestion levels and processing loads. As client 3 is connected via a low bandwidth ISDN link, the network losses dominate its packet loss and hence application efficiency becomes irrelevant. However application efficiency at client 1 and client 2 improves using the adaptive QoS management mechanism.

The above experiments confirm that application level adaptive QoS management mechanisms can significant improve the performance and resource utilization efficiency of

multimedia applications. Furthermore client and server side adaptation mechanisms operate in a complementary manner to provide an efficient and scalable distributed solution. Loss based adaptations presented in this thesis leverage on the real time support provided by RTP and RTCP to optimize the application's data transport characteristics and can be readily deployed on best effort IP based Internet.

Chapter 6

Conclusions and Future Work

6.1 Summary and Conclusions

Recently a phenomenal growth has been recorded in the number of Internet users and industries that are opting for Internet based solutions. However the Internet Protocol (IP) based Internet does not provide the infrastructure to support the diverse service requirements of these evolving applications, specially over the emerging networks and end systems that are increasingly heterogeneous. This thesis presents the design, implementation and evaluation of an adaptation mechanism to provide Quality of Service for distributed multimedia applications. The mechanism builds on the real time data transport support provided by RTP/RTCP and is designed to operate over the best effort Internet Protocol with no network QoS support. It adopts a purely application level adaptation approach and adaptations are performed independently at the clients and the server and are guided by an application specific policy. Distributed client side adaptations are integrated with centralized server side adaptations such that they complement each other to provide an improved service as perceived by the end-user. The coordination between these distributed entities is handled using the RTCP feedback mechanism. The server maintains an aggregate status of clients subscribed to each media type and performs rate-based adaptations. This provides a fine-grained control over the quality of service experienced by the entire client group. Furthermore this is a scalable solution since the server has to maintain state only of the order of the multicast channels used to transmit media data, which for most applications would be reasonably limited, and not on the order of the number of clients subscribed to the application. The clients independently perform modality transformations by switching between media channels to receive the level of service best supported by the local resource availability and optimize client resource utilization.

An evaluation of the adaptive mechanism shows that it significantly improves the performance of certain multimedia applications if the policy controlling the mechanism is designed with a proper understanding of the application's client configuration and requirements. The other important point to be noted is that an application based approach reduces the overhead on shared resources such as the network, thereby providing a scalable solution. Such a loss based approach has the added advantage of implicitly effecting congestion control thereby allowing other competing applications to have a fair share of shared resources. Enhancing the utilization efficiency of shared resources with availability constraints is another desirable feature of this mechanism. By defining the application specific information using a standard policy interface, such an application level mechanism can be a more general solution. However a truly general solution is still elusive, given that applications are always evolving and newer applications with diverse requirements rapidly emerging.

6.2 Contributions

The main contributions of this thesis are:

- Distributed application level adaptive QoS mechanism for a better QoS to end users of multimedia applications in heterogeneous environments

- Integration of a policy based scheme with the standard RTP and RTCP based real time data transport and feedback mechanism for streaming multimedia applications.

- Experimental study of loss based adaptations for streaming multimedia applications.
    - Innovative evaluation of an application's network resource utilization efficiency.

6.3 Future Work

Distributed multimedia applications with real-time delay and bandwidth constraints raise many technical problems in the present ubiquitous best-effort IP based Internet. Given the basic adaptation framework, simulation model and evaluation technique developed in this thesis, future

work will concentrate on defining generic adaptation techniques, such as content adaptation techniques, and on integrating them into the adaptation mechanism. Another area that needs investigation is design of a generic policy interface to enable easy integration of the adaptive QoS management mechanism within different classes of applications. This will also facilitate testing of the adaptation mechanism with real applications without being concerned about the application structure and implementation.

References

[1] J. Postel, User Datagram Protocol (UDP), RFC 768, Aug 1980.

[2] Internet Protocol (IP), RFC 791, Sep 1981.

[3] J. Case, M. Fedor, M. Scho-Stall and C. Davin, Simple network management protocol (SNMP), RFC 1157, May 1990.

[4] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, RTP: A Transport Protocol for Real-Time Applications, RFC 1889, Jan 1996.

[5] Resource ReSerVation Protocol (RSVP), RFC 2205, Sep 1997.

[6] Differentiated Services (diffserv), RFC 2475, Dec 1998.

[7] Common Open Policy Service (COPS), RFC 2748, Jan 2000.

[8] Multiprotocol Label Switching Architecture (MPLS), RFC 3031, Jan 2001.

[9] J. Saltzer, D. Reed, D. Clark, "End to End Arguments in System Design", *ACM Transactions in Computer Systems*, Vol. 2, No. 4, pp. 277-288, November 1984.

[10] C. Aurrecoechea, A. T. Campbell and L. Hauw, "A Survey of QoS Architectures", *ACM/Springer Verlag Multimedia Systems Journal, Special Issue on QoS Architecture*, Vol. 6, No. 3, pp. 138-151, May 1998

[11] K. Nahrstedt, R. Steinmetz, "Resource Management in Multimedia Networked Systems", *IEEE Computer*, Vol. 28, No. 5, pp. 52-65, May 1995.

[12] K. Nahrstedt, "Challenges of Providing End-to-End QoS Guarantees in Networked Multimedia Systems", *ACM Computing Surveys Journal*, Vol. 27, No. 4, pp. 613-616, December 1995.

[13] B. Li, D. Xu, K. Nahrstedt, J. W. S. Liu, "End-to-End QoS Support for Adaptive Applications Over the Internet", *SPIE International Symposium on Voice, Video and Data Communications*, pp. 147-161, November 1998.

[14] S. Servetto and K. Nahrstedt, "Video Streaming over the Public Internet: Multiple Description Codes and Adaptive Transport Protocols", *IEEE International Conference on Image Processing (ICIP'99)*, pp. 226-235, Kobe, Japan, October 25-29, 1999.

[15] K. Kim and K. Nahrstedt, "A Resource Broker Model with Integrated Reservation Scheme", Proceedings of *IEEE International Conference on Multimedia and Expo 2000 (ICME2000)*, pp. 859-862, New York, July 31 - August 2, 2000.

[16] D. Xu, K. Nahrstedt, A. Viswanathan and D. Wichadakul, "QoS and Contention-Aware Multi-Resource Reservation", Proceedings of the *9th IEEE International Symposium on High Performance Distributed Computing (HPDC-9)*, pp. 318-327, Pittsburgh, PA, August 2000.

[17] K. Lui, K. Nahrstedt and S. Chen, "Hierarchical QoS Routing in Delay-Bandwidth Sensitive Networks", Proceedings of *IEEE LCN 2000*, pp. 176-189, Tampa, FL, November, 2000

[18] K. Nahrstedt, D. Xu, D. Wichadakul and B. Li, "QoS-Aware Middleware for Ubiquitous and Heterogeneous Environments", to appear in *IEEE Communications Magazine*, 2001.

[19] D. Ivan-Rosu and K. Schwan, **"**Improving Protocol Performance by Dynamic Control of Communication Resources", *Second IEEE International Conference on Engineering of Complex Computer Systems*, pp. 291-304, Montreal, October 1996.

[20] D. Ivan-Rosu, K. Schwan, S. Yalamanchili, and R. Jha, **"**On Adaptive Resource Allocation for Complex, Real-time Applications", *Real-Time Systems Symposium*, pp. 249-262, San Francisco, IEEE, Dec. 1997.

[21] V. Martin and K. Schwan, **"**ILI: An Adaptive Infrastructure For Dynamic Interactive Distributed Applications", *International Conference on Configurable Distributed Systems*, pp. 118-131, May 1998.

[22] D. Ivan-Rosu and K. Schwan, **"**FARA - A Framework for Adaptive Resource Allocation in Complex Real-Time Systems", *IEEE Real-Time Technology and Applications Symposium*, pp. 324-337, June 1998.

[23] R. Kravets, K. Calvert, and K. Schwan, "Payoff Adaptation of Communication for Distributed Interactive Applications", *Journal of High Speed Networks*, Special Issue on Multimedia Communications, Vol. 7, pp. 143-157, July 1998.

[24] R. Kravets, K. Calvert, and K. Schwan, **"**Payoff-Based Communication Adaptation based on Network Service Availability" *IEEE Multimedia Systems '98 (ICMCS)*, pp. 33-42, Aug. 1998.

[25] R. West and K. Schwan, **"**Dynamic Window-Constrained Scheduling for Multimedia Applications", *6th International Conference on Multimedia Computing and Systems (ICMCS'99)*, Vol. 2, pp. 87-91, Florence, Italy, June 1999.

[26] R. West, K. Schwan, and C. Poellabauer, **"**Scalable Scheduling Support for Loss and Delay Constrained Media Streams", *IEEE Real-Time Systems and Applications Symposium (RTAS)*, pp. 376-389, June 1999.

[27] P. Chandra, A. Fisher, C. Kosak and P. Steenkiste, "Network Support for Application-Oriented Quality of Service", *Sixth IEEE/IFIP International Workshop on Quality of Service*, pp. 187-195, Napa, May 98.

[28] J. Bolliger and T. Gross, "A Framework-Based Approach to the Development of Network-Aware Applications", *IEEE Trans. Software Engineering (Special Issue on Mobility and Network-Aware Computing)*, Vol. 24, No. 5, pp. 376-390, May 1998.

[29] B. Lowekamp, N. Miller, D. Sutherland, T. Gross, P. Steenkiste, and J. Subhlok, "A Resource Query Interface for Network-Aware Applications", *Proceedings of the 7th IEEE Symposium on High-Performance Distributed Computing*, IEEE Computer Society, pp. 189-196, Chicago, Illinois, July 1998.

[30] P. Steenkiste, "Adaptation Models for Network-Aware Distributed Computations", *3rd Workshop on Communication, Architecture, and Applications for Network-based Parallel Computing (CANPC'99)*, pp. 16-31, Orlando, January 8, 1999.

[31] T. Gross, P. Steenkiste, and J. Subhlok, "Adaptive Distributed Applications on Heterogeneous Networks", *8th Heterogeneous Computing Workshop (HCW '99)*, pp. 219-225, April 1999.

[32] B. Lowekamp, D. O'Hallaron, and T. Gross, "Direct Network Queries for Discovering Network Resource Properties in a Distributed Environment", *8th IEEE Symposium on High-Performance Distributed Computing* (Redondo Beach, California), IEEE Computer Society, pp. 38-46, August 1999.

[33] N. Miller and P. Steenkiste, "Collecting Network Status Information for Network-Aware Applications", *Infocom'00*, pp. 641-650, Tel Aviv, March 2000.

[34] A. Watson and M. A. Sasse, "Multimedia Conferencing via Multicast: Determining the Quality of Service Required by the End User", *AVSPN '97 - International Workshop on Audio-Visual Services over Packet Networks*, pp. 189-194, 15-16 September 1997, Aberdeen, Scotland.

[35] S. N. Bhatti and G. Knight, "Notes on a QoS information model for making adaptation decisions", *HIPPARCH'98 - 4th International Workshop on High Performance Protocol Architectures*, pp. 76-89, UCL, London, UK, 15-16 June 1998.

[36] S. N. Bhatti and G. Knight, "QoS Assurance vs. Dynamic Adaptability for Applications", *NOSSDAV'98 - 8th International Workshop on Network and Operating System Support for Digital Audio and Video*, pp. 146-154, New Hall, Cambridge, UK, 8-10 July 1998.

[37] A.Watson and M. A. Sasse, "Measuring Perceived Quality of Speech and Video in Multimedia Conferencing Applications", *ACM Multimedia '98*, pp. 55-60, 12-16 September 1998, Bristol, England.

[38] S. N. Bhatti and G. Knight, "Enabling QoS adaptation decisions for Internet applications", *Computer Networks*, vol. 31, no. 7, pp. 669-692, March 1999.

[39] White Paper – "The need for QoS", www.qosforum.com/white-papers/Need_for_QoS-v4.pdf.

[40] White Paper – "Introduction to QoS policies", www.qosforum.com/white-papers/qospol_v11.pdf.

[41] White Paper – "QoS protocols and architectures", www.qosforum.com/white-papers/qosprot_v3.pdf.

[42] X. Wang and H. Schulzerinne, "Comparison of Adaptive Internet Multimedia Applications", *Institute of Electronics, Information and Communication Engineers (IEICE) Transactions on Communications*, Vol. E82-B, pp. 806-818, June 1999.

[43] N. Shiratori, T. Kinoshita, T. Suganuma and G. Mansfield, "Towards Application-Centric Flexible Network Operation and Management", *IEICE Transactions on Communication*, Vol. E82-B, No.6, pp. 800-805, 1999.

[44] F. Chang and V. Karamcheti, "Automatic Configuration and Run-time Adaptation of Distributed Applications", *Ninth IEEE Intl. Symposium on High Performance Distributed Computing (HPDC)*, pp. 11-20, August 2000.

[45] K. Nagao, Y. Shirai and K. Squire, "Semantic Annotation and Transcoding: Making Web Content More Accessible", *IEEE MultiMedia*, Vol. 8, No. 2, pp. 277-289, April-June 2001.