

Adaptive Service in Mobile Computing Environments

Songwu Lu Kang-Won Lee Vaduvur Bharghavan
Coordinated Sciences Laboratory
University of Illinois at Urbana-Champaign
1308 W. Main Street, Urbana, IL 61801
Email: {slu, kwlee, bharghav}@crhc.uiuc.edu

Abstract

Emerging communication-intensive applications require significant levels of networking resources for efficient operation. In the context of mobile computing environments, limited and dynamically varying available resources, stringent application requirements, and user mobility make it difficult to provide sustained quality of service to applications. This paper addresses the above issues, and proposes a new service model in cellular mobile computing environments. The major results of this paper are the following:

- An *Adaptive Service* model in mobile computing environments, which enables the network and applications to adaptively (re)negotiate QoS depending on dynamic network conditions
- An algorithmic framework that provides cost-effective resource adaptation in the presence of resource and network dynamics
- A unified implementation architecture for adaptive service support

A brief overview of an ongoing testbed implementation is provided.

1 INTRODUCTION

Recent years have witnessed explosive research in the development of indoor/outdoor mobile computing environments, which seek to provide a mobile user with not only a basic toolchest of applications – editors, compilers, email, ftp, telnet, etc. – but also more advanced applications such as multimedia, WWW browsers, distributed file systems and databases, etc. The latter class of applications is communication-intensive, and will stress the wireless networking component of the mobile computing environments severely. A fundamental problem is that the wireless medium is a scarce, shared resource which has very different characteristics from wireline networks, and needs to be managed efficiently to provide an acceptable Quality of Service (QoS) for communication-intensive applications. Moreover, seamless user mobility constitutes another QoS parameter. Therefore, the network service should be adaptively provided to effectively handle wireless link characteristics and user

mobility. We have proposed a new service model “*adaptive service*” and its implementation on a testbed is ongoing work. This paper provides an overview of the adaptive service and describes our initial experiences with the testbed.

The adaptive service has three key features: (a) a notion of QoS bounds, (b) mobility-related QoS parameters, and (c) an enhanced set of service classes that extend integrated services for wireline networks to mobile computing environments.

To provide adaptive services over mobile computing environments, we have developed an algorithmic approach that seeks to provide (a) admission control and resource reservations for both new and handoff flow requests, (b) a cost-effective and fair resource adaptation among adaptable flows, which also resolves resource conflicts in admission control, (c) predictive advance resource reservation in order to minimize handoff dropping while maintaining efficiency in resource utilization, and (d) a unified framework to support multiple QoS classes for a wide range of applications in mobile computing environments. Initial results were reported in [2, 7]. This paper explores the ideas of adaptive QoS in greater detail and briefly describes an ongoing testbed implementation.

In order to experiment with adaptive QoS, we are building a testbed environment, which has the following features: (a) it treats admission control as a special case of the QoS adaptation, which enables an application to send data packets in a best effort manner before it starts resource reservation, (b) it is readily integrated with the standard TCP/IP protocol stack, and can even be used with unaware servers, (c) it advance reserves resources on next-predicted cell for smooth handoff, (d) it provides soft state flow management at intermediate switches, which is integrated with hard state at the end-hosts, and (e) client-initiated resource reservation for both uplink and downlink reservations.

The rest of the paper is organized as follows. Section 2 describes the issues in providing adaptive service. Section 3 defines the adaptive service model. Section 4 describes our algorithmic approach for supporting adaptive service. Section 5 presents our initial experiences on the implementation, and Section 6 concludes the paper.

2 ISSUES IN QOS SUPPORT FOR MOBILE COMPUTING

The mobile computing environment we consider here has a packet-switched cellular architecture, consisting of a wired backbone component and a wireless cellular component. Base stations are connected to the backbone network and provide wireless networking access to portable computers within *cells*. Two cells are called *neighbors* if a mobile user can handoff between the cells.

In order to provide quality of services in mobile computing environments, the following issues need to be addressed:

- Wireless channel error: the case for *adaptive service with QoS bounds and resource adaptation*

- ‘Seamless’ inter-cellular mobility: the case for *advance resource reservation*
- Maximizing network revenue: the case for *cost-effective adaptation*
- Application adaptation: the case for *adaptive service*

2.1 Wireless Channel Error

The wireless channel medium has two key characteristics that invalidate the assumption of negligible errors: (a) bursty channel errors, and (b) location-dependent channel capacity and errors. Therefore, the network resources can no longer be assumed to remain invariant, and the effective network capacity might vary dramatically over short time scales. Therefore, the resources provided to applications may need to vary *adaptively* depending on dynamic network resource changes; as a consequence, our end-to-end QoS negotiation process results in the network providing QoS within a bounded range, i.e. *QoS bounds*. Within these bounds, the resource adaptation algorithm may conduct adaptation transparent to the applications and vary the QoS level dynamically within the specified bounds.

2.2 User Mobility

In a mobile computing environment, users may have a wide range of mobility patterns, ranging from frequent handoffs between cells to almost static residence in a cell. Obviously, user mobility implies that the QoS negotiated between the application and the network in one cell may not be honored if the user crosses cell boundaries; besides, user mobility incurs dynamic resource changes for both the old cell and the new cell. For a user to be provided the illusion of seamless mobility, the user should not notice perceptible change in QoS upon mobility between cells. This motivates advance reservation of resources in neighboring cell(s). Moreover, the advance reservation should be tailored to the degree of service assurances requested by applications.

Furthermore, users with different mobility patterns should be treated differently. For flows from fast/frequent moving users, the major concern is “seamlessness”, i.e., users care more about the invariance in the QoS level rather than maximizing the QoS level; for flows from slow/infrequent moving users, users care more about maximizing the QoS level in the current cell rather than seamless handoffs between cells.

2.3 Network Revenue

From the network perspective, the service provider needs to maximize its overall revenue in the long term; the goal of resource adaptation performed by a network is to maximize revenue over a time window by dynamically allocating capacity among competing flows. Since the network will incur a cost for each adaptation, the resource adaptation algorithm should guarantee

that any adaptation initiated by the network proactively (rather than as a result of a resource decrease) does not result in a loss of revenue, i.e., the adaptation is initiated only when such an adaptation will increase long term revenue. Thus, frequent mobility will not result in frequent adaptation.

2.4 Application Perspective

Many applications can adapt offered traffic to dynamic network conditions since they either have inherent adaptation mechanisms or can be passed through an adaptive filter [1]. Such applications are ideal candidate for varying QoS with bounds. For example, most video compression standards, like MPEG, JPEG and JBIG, have a notion of ‘progressive mode’ or ‘hierarchical mode’ that can generate variable-rate video streams.

The applications might request several service categories with different service commitments from wireline networks, e.g. *guaranteed service*, *predictive service*, and *best-effort service*. Over wireless mobile computing environments, because the effective resource capacity is no longer invariant due to wireless channel errors and user mobility, the service models defined for wireline networks should be modified and enhanced to effectively handle these issues. This motivates us to define our *adaptive service* model.

3 ADAPTIVE SERVICE IN MOBILE COMPUTING

In order to address the above issues in QoS support, we propose a new type of service model called *Adaptive Service* for mobile computing environments. There are several key aspects of adaptive service: (a) it provides QoS guarantees/assurances *within bounds* (e.g. $[b_{min}, b_{max}]$ for bandwidth); (b) it provides mobility-related QoS parameters; (c) it defines an enhanced set of service classes that extend integrated services for wireline networks to mobile computing environments; and (d) it provides cost-effective resource managements.

3.1 QoS Bound Specification

The QoS bounds specify the QoS parameters within a range. Therefore, the resource contract $Rspec$ is characterized by $\langle [b_{min}, b_{max}], [d_{max}, d_{min}], e_{max}, h \rangle$, where $[b_{min}, b_{max}]$ represents the bounds on rate, $[d_{max}, d_{min}]$ represents the bounds on maximum delay, and e_{max} represents the maximum packet loss probability. h is not a numerical QoS parameter; it takes one of three values: *conservative*, *predictive*, or *best-effort*, depending on the desired advance reservation scheme. These three values correspond to three mobility-related service modes to be described later: conservative reservation for location-independent service, predictive reservation for neighborhood-specific service and cell-specific service.

3.2 Service Classes

(a) Guaranteed versus Predictive

Related literature has pointed out that there are no absolute guarantees in a mobile computing domain, particularly due to two reasons: (a) bursts of wireless channel error may reduce the link capacity below the guaranteed rate, and (b) guarantees which are contracted in one cell may be invalid in another, more heavily loaded cell. Thus, *all guarantees in wireless networking are conditional on currently available system capacity*; On the other hand, several recent measurements taken for state-of-the-art indoor wireless LANs indicate that wireless resources are reliable over a long time scale [3]. Therefore, we take a middle ground: adaptive service promises assurances/guarantees on the QoS lower bound and provides better service beyond that in a best-effort manner.

Predictive service bases its QoS assurances (reliable estimates as opposed to guarantees) on measured values as opposed to a priori worst-case characterization of flows [9, 4]. In most related work, measurements have typically pertained to the flows, while the channel capacity is typically assumed to be invariant. Over a wireless link, we can apply measured values to both *flows* and *link capacities*. Thus, there are four modes of QoS specification for a wireless link within our adaptive service model:

1. *Absolute Guaranteed mode*: Using a priori values for channel capacity and a priori values for flows: e.g. guaranteed service for wireline networks, real time guarantees for mission-critical tasks in the wireless domain [8].
2. *Flow-Conditioned Predictive mode*: Using a priori values for channel capacity and measured values for ongoing flows: e.g. predictive service in wireline networks.
3. *Channel-Conditioned Guaranteed mode*: Using measured values for channel capacity and a priori values for flows: e.g. conditional guarantees in wireless links.
4. *Channel/Flow-Conditioned Predictive mode*: Using measured values for channel capacity and measured values for ongoing flows: e.g. predictive service in wireless links.

(b) Mobility-Related QoS Parameter

We consider three possible service specifications to account for mobility:

1. *Location-independent service*: The service specifications are valid for a flow independent of its location in the network. This service would require advance reservation in all cells the user might visit (as specified via e.g. the mobility profile) [5].
2. *Neighborhood-specific service*: The service specifications are valid for a flow in its current cell and its neighbors [7, 11]. As a user moves, advance reservations in neighbors are made and canceled dynamically. We consider two

flavors of advance reservation: (a) conservative reservation in all neighbors [11], and (b) predictive reservation in only the next-predicted cell [7].

3. *Cell-specific service*: The service specifications are valid only in the current cell.

In our model, we define a special QoS parameter h to support all the above three service modes. However, the provision for location-independent services is likely to incur an enormous resource overhead; hence, we focus on neighborhood-specific and cell-specific services.

3.3 Cost-Effective Adaptation

In mobile computing environments, in order to shield flows from frequent adaptation due to rapid resource fluctuation (caused by time-varying effective wireless link capacity and user mobility), we adopt a cost-effective approach for rate adaptation. Essentially, the network associates an adaptation cost for each flow, and will not initiate new adaptation for a flow unless the revenue gain of a flow would be able to compensate its adaptation cost upon detection of a network resource change. Through this way, the network defines a dynamic *adaptable set*, and designs its adaptation policy for flows within this set so that it maximizes its overall revenue subject to resource and flow demand constraints. It turns out that the optimal adaptation policy is a max-min fair rate allocation among the flows in the adaptable set for many scenarios.

4 ALGORITHMS FOR ADAPTIVE SERVICE

In this section, we provide an overview for three key resource management algorithms and the interaction between them.

4.1 Admission Control and Resource Conflicts

Admission control converts end-to-end QoS requirements into per-hop requirements and tests for the availability of resources at intermediate nodes [6]. It is conducted for both a *newly arriving* flow and a *handoff* flow. For flows with QoS bound specification, the network is only committed to provide assurance for the QoS lower bound, and provides better service beyond the lower bound depending upon the dynamic availability of the network resources. The major difference of admission control policies for predictive and guaranteed services is whether they use measured values or a priori worst-case characterizations. Admission control is the same for handoff and new flows except that handoff flows may already have advance reserved their minimum requested resources.

The introduction of QoS bounds introduces a new problem for admission control, i.e. *resource conflict*. The problem of resource conflict arises when the network cannot accept a new flow without reducing currently allocated resources (within pre-negotiated QoS bounds). The resolution of resource con-

flicts readjusts the resources allocated to each flow but does not violate the pre-negotiated bounds of existing flows. In our framework, we provide a unified approach for resource conflict resolution and resource adaptation.

4.2 Predictive Advance Resource Reservation

To provide neighbor-specific predictive advance reservation, our algorithm predicts the next cell for a mobile user and makes advance resource reservation based on portable/cell profiles. We distinguish *static* portables from *mobile* portables. A portable is labeled ‘mobile’ if it has performed a handoff within a specified time window, otherwise it is labeled ‘static’. A flow is labeled static if both its end hosts are static, otherwise it is labeled mobile. For a *mobile* flow, the minimum acceptable resources are reserved in its next-predicted cell(s) of its *mobile* end-host(s). Otherwise, bandwidth is not reserved in advance in its *neighboring* cells on a flow/portable-specific basis; instead, each base station in the *neighboring* cells sets aside a dynamically adjustable fraction of resources to accommodate “unforeseen” events (e.g. sudden mobility of static portables). This common reserved resource pool may also handle wrong predictions for mobile movements. A detailed discussion is presented in [2].

4.3 Resource Adaptation

Adaptive Service provides QoS within bounds; this implies that flows can be accepted so long as their lower bounds can be satisfied. However, in order to increase the utilization of the network and provide better service to applications, excess resources (after the satisfaction of the lower bounds of ongoing flows) need to be distributed effectively among the flows.

In order to perform resource adaptation, we propose a new adaptation algorithm which (a) guarantees no loss of revenue for networks due to adaptation, (b) maximizes network revenue and the adaptation policy achieves max-min fair allocation among the adaptable flows, (c) exploits only the current information on available resources, and does not make predictions for resource changes in the future, (d) shields wired backbone flows and static flows from frequent resource fluctuations due to mobility of other flows, (e) presents an efficient distributed framework which accomplishes the adaptation using only local information and converges fast, and (f) maintain feasibility at all times during the adaptation process.

Specifically, when a network link performs adaptation, it first determines its current *adaptable set*, which consists of flows that will adapt at this time. A flow is considered *adaptable* only if its revenue gain via adaptation has paid off its adaptation cost. For flows within the adaptable set, a network link attempts to maximize its total revenue over all adaptable flows sharing it. It can be shown that the optimal solution allocates the excess resources in a max-min fair sense among contending flows within the adaptable set. Our implementation of the cost-effective adaptation algorithm imposes a ‘con-

control tree' structure on the graph, for propagating the state information and computing max-min bandwidth allocation. As a result, the implementation propagates information through the control tree in an effective manner and achieves max-min adaptation with few steps and propagating limited state, while maintaining feasibility all the times during adaptation. Details of the adaptation algorithm are beyond the scope of this paper. Readers may refer to [2] for a preliminary description.

4.4 Interaction among Resource Management Algorithms

Provision of adaptive services in mobile computing environments has involved the following algorithms to interact: (1) service specification, (2) routing, (3) resource reservation, (4) admission control, (5) scheduling, (6) resource adaptation and resource conflict resolution, and (7) advance reservation based on mobility prediction. The interactions among them are illustrated below via steps involved in flow establishment or handoff.

When a new unicast flow request is made, the following actions occur:

- (1) The initiating host specifies the QoS bounds and the service mode.
- (2) The routing algorithm computes a route for this flow.
- (3) The two end hosts are subjected to the static/mobile test.
- (4) The resource reservation algorithm is invoked along the route for the new flow. At each intermediate switch, a local admission test is performed in the forward phase. All admission tests for both static and mobile flows are performed only with respect to the lower bound of requested resources.
- (5) An admission test may return one of three results: *flow accept*, *conflict detection*, and *flow reject*. If the admission test at each intermediate switch in the forward pass is either flow accept or conflict detected, then in the reverse pass, the new flow is admitted, and a conflict resolution algorithm is performed (if required).
- (6) Resource reservation for a newly admitted flow may end up performing two tasks: (a) conflict resolution, and (b) excess resource allocation. Both invoke the cost-effective adaptation.
- (7) For a mobile flow, the next cell of the mobile end-host is predicted, and the lower bounds of flow resources are reserved in this next cell. The predictive advance reservation is only applied when $h = \text{predicted}$ in the *Rspec*; when $h = \text{conservative}$, a brute force advance reservation algorithm is performed over all neighboring cells; when $h = \text{best-effort}$, no advance reservation is performed.
- (8) For mobile flows that turn static and join the adaptable set, resource adaptation is performed (i.e. Step 6) after the change is detected.

For the case of flow handoff, the steps are similar.

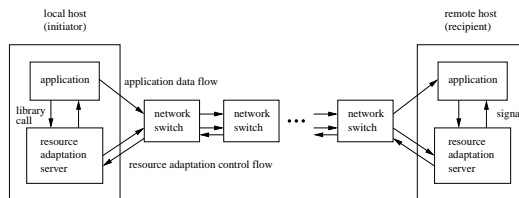


Figure 1 Four components and their interaction

5 IMPLEMENTATION

In order to experiment with adaptive service, we are building a testbed implementation of a mobile computing environment. The testbed consists of Gateway P6-200s (network ‘software’ switches and base stations) and TI Travelmate P-120s (mobile hosts), all of which run Linux 2.0.30. The wired backbone is a point to point switched Ethernet with P6-200s functioning as switches. The wireless network is WaveLAN. The Ethernet device driver ‘de4x5.c’ and WaveLAN device driver ‘wavelan_cs.c’ were modified to implement WRR(Weighted Round Robin) scheduling, admission control, and resource reservation at intermediate switches.

There are four main components in our resource adaptation implementation; *application programs(client/server)*, *resource adaptation server*, *runtime library*, and *intermediate switches*. Figure 1 represents the interaction between them. Currently only two QoS parameters have been implemented: *data rate* and *delay*. *Error* and *handoff* QoS parameters are on-going work.

The main features of our implementation are as follows:

- Data transmission and resource reservation are decoupled. Hence it is readily integrated with the standard TCP/IP protocol stack; In fact it is possible to support QoS even with ‘unaware’ applications. In our implementation, admission control is treated as a special case of QoS adaptation; Thus an application can send data packets before resource reservation in a best effort manner. Only when the application wants QoS guarantees, does it start resource reservation.
- It uses *soft state* extensively in intermediate switches for resource reservation, advanced reservation, flow management, and mobility support. At the end host, applications explicitly register, deregister, and make QoS request with the adaptation server. Using soft state, the network switches periodically free resources unless explicit refresh messages are exchanged. This enables us to handle mobility easily, since resources on defunct routes are automatically freed upon a timeout.
- The resource reservation is initiated by an application client for both the uplink and the downlink. The reservation process finishes in 2-way handshake whether the resource reservation is successful or not.

5.1 Adaptation Server

The resource adaptation server is a simple stateless server, which manages end-to-end QoS. Typically it resides at the ‘aware’ end hosts and base stations. The main functionality of the resource adaptation server is to perform end-to-end QoS adaptation and resource reservation on behalf of the application.

QoS parameters, *delay* and *rate* are associated with flows. A flow is identified by the following parameters: <src addr, src port, dest addr, dest port, protocol>, which constitute the flow id.

Applications register/deregister flows with the adaptation server. When the server receives a QoS request from an application, it sends a QoS adaptation request message to the peer server of the flow. As the message traverses intermediate hops, each switch performs reservations for uplink and downlink flows on the downstream and upstream interfaces respectively. For a mobile flow, the adaptation server may communicate with the neighboring base stations in order to initiate advance resource reservation. Section 5.3 describes an example.

A runtime library (`register`, `deregister`, `send_QoS_request`, `retrieve_QoS`) is provided to applications as an interface to the adaptation server.

5.2 Network Switches

Each network switch implements a hierarchical *Weighted Round Robin* (WRR) scheduling algorithm. In addition, the modified Ethernet and WaveLAN device drivers perform admission control and resource reservation upon a QoS request. The modified device drivers have a hierarchy of three queues: (a) *FIFO queue* for control packets, (b) *WRR queue* for reserved flows, and (c) *Round Robin queue* for best effort traffic.

If an adaptation request on the uplink/downlink can be satisfied by the switch, it reserves the resources on device drivers at the uplink/downlink for the flow. Otherwise it sets the flag in the control message to inform the failure to other switches and the adaptation servers.

Resource reservation in intermediate switches involves a basic 2-way handshake. The client initiates resource reservation request for both the uplink and the downlink. On the forward pass, bandwidth admission control/resource reservations are performed. On the reverse pass delay admission control/resource reservations are made. If either admission control fails, then no refresh packets will be sent by the adaptation server at the client and reserved resources will be freed eventually. Note that since messages are exchanged between adaptation servers, the application server can be ‘unaware’ of QoS.

5.3 Typical Resource Adaptation Steps

Figure 2 shows typical resource adaptation steps. For simplicity, only one network switch is shown in the figure.

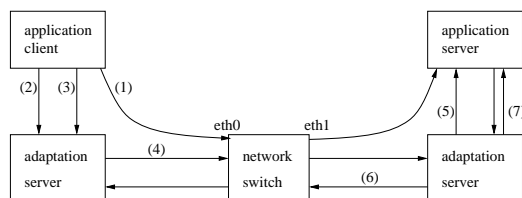


Figure 2 Typical interaction between resource adaptation servers

1. An application client opens a flow and sends/receives data packets in a best effort manner.
2. When the application wants to associate QoS with the flow, it registers the flow to the local adaptation server.
3. Then it sends a `QoS_Request` message to the adaptation server.
4. The local adaptation server sends a `QoS_Request` to the remote adaptation server. During this request pass, bandwidth adaptation for both the uplink and the downlink is done at the intermediate switches.
5. The remote adaptation server updates its QoS entry table then signals the application server. An unaware application server may ignore the signal.
6. The remote adaptation server sends a `Response` message back to the local adaptation server. During this response pass, delay adaptation is performed for both the uplink and the downlink. Note that if the bandwidth adaptation were unsuccessful, then delay adaptation will not be performed. Also note that if delay adaptation were unsuccessful, the reserved resources at intermediate switches will be freed after timeout.
7. At any time, an application client/server can query current the QoS for a flow by invoking the `RetrieveQoS` function to contact its local adaptation server.
8. Since resource reservation at intermediate switches use soft state, the adaptation server sends `KeepAlive` packets periodically to refresh the flow resources.
9. For advance reservation, the base station of the next-predicted cell is notified to initiate a `QoS_request`. This request terminates at the nearest branch point, and is identified as a secondary flow.
10. Upon handoff, the adaptation server in the previous cell ceases to send a `KeepAlive` message. Consequently resources along this path are released after timeout.

6 CONCLUSION

Four factors motivate the work in this paper: (a) user mobility and wireless channel error motivate the definition of adaptive service model, (b) the resource dynamics introduced by wireless links and user mobility motivates resource adaptation, (c) a cost-effective resource management motivates our algorithmic framework for resource adaptation, (d) the enhanced set of mul-

multiple QoS classes motivates our unified implementation structure to provide adaptive services.

This paper has proposed a new service model “adaptive service” for mobile computing environments, and describes an ongoing implementation for providing adaptive service. The ultimate goal is to build an integrated services cellular network that provides quality of services for a wide range of applications.

REFERENCES

- [1] V. Bharghavan, “Challenges and solutions to adaptive computing and seamless mobility over heterogeneous wireless networks,” to appear on *International Journal on Wireless Personal Communications*.
- [2] S. Lu and V. Bharghavan, “Adaptive resource management algorithms for indoor mobile computing environments,” *Proceedings of ACM SIGCOMM’96*, Stanford, California, August 1996.
- [3] D. Eckhardt and P. Steenkiste, “Measurement and analysis of the error characteristics of an in building wireless network,” *Proceedings of ACM SIGCOMM’96*, Stanford, CA, August 1996.
- [4] S. Jamin, P. Danzig, S. Shenker and L. Zhang, “A measurement-based admission control algorithm for integrated services packet networks,” *Proceedings of ACM SIGCOMM’95*, 1995.
- [5] A. K. Talukdar, B. R. Badrinath and A. Acharya, “On accommodating mobile hosts in an integrated services packet network,” to appear on *Proceedings of IEEE INFOCOM’97*.
- [6] C. Parris, H. Zhang and D. Ferrari, “Dynamic management of guaranteed performance multimedia connections”, *Multimedia Systems Journal*, vol. 1, 1994.
- [7] V. Bharghavan and M.P. Jayanth, “Profile-based next-cell prediction in indoor wireless LANs,” *Proceedings of IEEE Singapore International Conference on Networking*, April 1997.
- [8] S. Shenker, C. Partridge and R. Guerin, “Specification of guaranteed quality of service,” *draft-ietf-interv-guaranteed-svc-06.txt*, IETF Draft, August 1996.
- [9] D.D. Clark, S. Shenker and L. Zhang, “Supporting real-time applications in an integrated services packet network: architecture and mechanism,” *Proceedings of ACM SIGCOMM’92*, 1992.
- [10] L. Zhang, S. Deering, D. Estrin, S. Shenker and D. Zappala, “RSVP: A new Resource reSerVation Protocol,” *IEEE Network*, September 1993.
- [11] K. Lee, “Adaptive network support for multimedia,” *Proceedings of IEEE MOBICOM’95*, Berkeley, California, November 1995.
- [12] D. A. Levine, I. F. Akyildiz and M. Naghshineh, “The shadow cluster concept for resource allocation and call admission in ATM-based wireless networks,” *Proceedings of ACM MOBICOM’95*, November 1995.