An Experimental Architecture for providing QoS guarantees in Mobile Networks using RSVP

Indu Mahadevan[†] **and Krishna M. Sivalingam**[‡*]
School of Electrical Engineering and Computer Science
Washington State University
Pullman, WA 99164
†E-mail: imahadev@eecs.wsu.edu. Phone: (509)-335-1787.
‡E-mail: krishna@eecs.wsu.edu. Phone: (509)-335-3220. Fax: (509)-335-3818

Abstract

Efforts are underway to enhance the Internet with Quality of Service(QoS) capabilities for transporting real-time data. The ReSerVation Protocol(RSVP) provides a signaling mechanism for end-to-end QoS negotiation. The issue of wireless networks and mobile hosts being able to support applications that require QoS has become very significant. Reservation of resources and the maintenance of QoS for the mobile as it moves from one region to another creates a new set of challenges. In our paper, we describe an architecture where a modified RSVP protocol helps provide QoS support for mobile hosts. The modified RSVP protocol will be implemented in an experimental wireless and mobile testbed to study the effects of mobility on QoS, and the feasibility and performance of our approach.

Topic Code: Experiments, Trials and Deployment
Keywords: Mobile Networks, Resource Reservation, RSVP, Wireless Local Data Networks.

1 Introduction

Future wireless and mobile communication networks will be expected to provide resource allocation for the various classes of applications that require Quality of Service(QoS) support. A big drop in service quality when a call hand-off is made as the mobile moves from one region to another may not be acceptable for these applications. It is required to maintain the QoS of these applications, in the presence of user mobility with the use of resource reservation. RSVP [1] is a network management setup protocol designed to help share resource reservations among participating applications. Currently, RSVP is designed to operate in wired networks. In this paper, we will describe the design and architecture of a modified RSVP protocol to guarantee resource reservations to mobile wireless hosts.

In our architecture, a mobile in a region is served by a base station which is connected to the wired network. Resource reservations are made using RSVP between the base station and the mobile. To make sure that a mobile has reservations guaranteed as it moves from one region to another, base stations make reservations with other base stations in all the neighboring regions. These reservations will remain "passive" [2]. That is, the resources may be used by other mobiles until it is needed for this particular mobile. This ensures that resources are not needlessly tied up for potential incoming mobile hosts.

An architecture for using RSVP for Integrated Services Packet Network with mobile networks has been described in [2]. A passive reservation mechanism is suggested in [2] as described above. However,

*Corresponding Author. Part of the research was supported by Air Force of Scientific Research grant F-49620-97-1-0471.

the architecture requires a mobile to know all the subnets it will be visiting. The mobile obtains the identity of the proxy agents, which help with mobile RSVP in all the subnets, using a proxy discovery protocol. The mobile instructs the proxy agent in the region it is currently located to make passive reservations with all the proxy agents in all other regions. Four additional messages are used in addition to the messages already present in RSVP.

The drawback of this architecture is that it assumes that a mobile knows the addresses of all the subnets it is going to move into and which is not always possible. It also places a burden of finding the proxy agents in all these subnets on the mobile. In this paper we will discuss our architecture which does not make these assumptions and show how we can enable RSVP to work with mobile networks so that QoS can be maintained. The work will address not only messages conveying resource requests, but also scheduling at the Base Station(BS) to accommodate resource requests. The experimental testbed under development based on our architecture, consists of two Lucent Wavelan [3] base stations and three laptops equipped with Wavelan cards. We will use RSVP which will be modified to identify "active" and "passive" reservations. Experiments will be done to study the effect on various QoS parameters as a mobile with resource reservations in a region, moves to another region which has some resources reserved passively. Initially, modified Class Based Queueing(CBQ) [4] will be used for traffic classification and scheduling.

The paper is organized as follows. An overview of RSVP and CBQ is given in Section 2. Section 3 gives an outline of our architecture and discusses using RSVP for this problem. We look at QoS specifications in section 4. Section 5 discusses modifications needed for RSVP for use with our architecture and mentions our experimental testbed under development. Conclusions and discussions follow in section 6.

2 An Overview of RSVP and CBQ

The RSVP protocol [1] is a network management setup protocol designed to share resources among participating applications. After a high-level dialogue, the initiator of a flow that wants to use RSVP generates PATH messages to each accepting receiver. The PATH message specifies the upper limits of the flow expected and the message is routed to the receiver(s) using any routing algorithm that is available. The receiving host(s) will then make resource reservation requests using RESV messages along the reverse path to the sender. If at any point along the path the request cannot be supported, that request is blocked. Otherwise, this request is merged with other requests to the same sender in order to better share the bandwidth. Each node (host or routers) that is capable of QoS control needs a packet scheduler and classifier which is handled by Class Based Queueing(CBQ) [4] mechanisms. CBQ consists of a classifier that classifies packets into one of a set of pre-defined classes, an estimator that estimates bandwidth usage of each class and a packet scheduler that selects the next class to send a packet. RSVP is independent of the underlying scheduling mechanism and can be aided by any mechanism like CBQ for QoS control.

3 The Network Architecture

This paper assumes a microcellular network architecture, with a geographical region divided into cells. Each cell has a Base Station (BS) serving all mobiles within its coverage region and connected to the wired network. When a mobile moves to another cell, it is handed off to the base station serving that cell.

In our network architecture, we use RSVP along with Class Based Queueing(CBQ) to reserve resources in the network. The sender or receiver of an application can be a mobile, the base station or any host on a wired network. In the discussion here, we emphasize on the communication between the base station and the mobile. Reserving resources is required to make sure an application gets the required QoS. The problem with wireless mobile networks is to make sure we maintain the QoS as the mobile moves from one cell to another. This means we need to make sure that the mobile has some form of resource reservation anywhere it goes. One way of guaranteeing QoS as a mobile moves from one cell to another is to reserve resources with all the base stations in the neighboring cells because the mobile might move into one of them. This would be a waste of limited wireless resources. Alternatively, we can make reservations that can be used by other mobiles in the cells till the mobile moves into that cell. These reservations made in the neighboring cells will be "passive" and can be used for other applications till the mobile actively starts using them [2]. An architecture to make such resource reservations using RSVP is discussed below.

3.1 A Scenario with Reservations for Mobiles

Fig. 1 aids the discussion in this section. The cells are denoted by A, B, C, etc. BS represents the base station and M represents the mobile. The solid line represents an active reservation while the dotted line indicates a passive reservation. We assume that a base station knows the addresses of the base stations in all the neighboring cells.

In a wireless environment we need to distinguish between the two kinds of reservations that need to be made: a) between the sender and various base stations in neighboring cells in the wired network, and b) between the base station and the mobile in the wireless region. In our example, the mobile M is initially in cell A. BSa is the base station in this cell and resource reservations must be made between BSa and M. When the mobile moves, it could move into any of the other six cells. At this point we will need to make two kinds of resource reservations that will remain "passive": a) one between the "current" base station BSa and all the other base stations namely BSb, BSc, BSd, BSe, BSf and BSg, and b) the other where the base stations BSb, BSc etc. make a passive resource reservation on their wireless interfaces to accommodate a mobile that may enter their cells. In the example provided above, the base stations are assumed as the end points of an application. This need not be the case. If another host is the end-point, a reservation needs to be made between that host and the base station in the wired domain using regular RSVP requests.

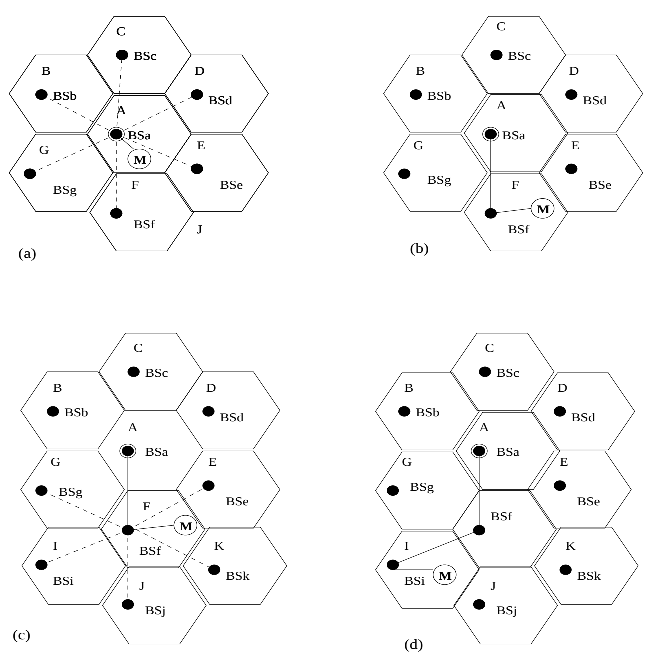

Figure 1: Overview of a Mobile Environment

In fig. 1(b) we see that the mobile M has moved into cell F. At this point the resources that were labeled passive in the wired environment between BSa and BSf are made active and the resources made on the wireless interface of BSf is also activated and is used for communication between BSf and M. All other resources reserved passively can be deleted now. This scenario further continues as shown in figures 1(c) and (d). At some point, we could make a re-routing decision so that the sender BS directly connects to the BS in a cell where a mobile is currently located instead of going through all the intermediate BSs. This decision will be a trade-off between the re-routing cost and the cost of reserving resources in all the intermediate BSs. Resource reservations in our architecture is made with the help of RSVP and CBQ. More details are provided in the following section.

3.2 Using RSVP and CBQ to reserve resources

In this section we take a closer look at the problem with regard to the RSVP protocol. The problem is discussed using fig. 2 with two neighboring cells. We assume, for simplicity of discussion, that a base station(BS) is the sender of an application. We use the RSVP protocol, modified to recognize passive and active reservations. The mobile is currently in the cell in which the sender/BS resides. The BS sends PATH messages to the mobile and since the reservations are going to be used it is indicated as an active reservation denoted in Fig. 2 as (1). The mobile responds with a RESV message (active one) if it can accept the call. After this point, the sender of the application has made sure that resources are reserved for this application. The BS now has to make passive reservations with all the BSs' in the neighboring cells. The traffic specifications of this passive reservation are the same as those used by the active reservation. We concentrate on only one neighboring cell in our discussion. In fig. 2 the "current" BS 1 sends PATH messages denoting it is a passive reservation to BS 2(3) and the BS 2 will send a "passive" RESV message to BS 1(4). BS 2 also needs to make reservations on the wireless interface(5) for the wireless link which the mobile may use.

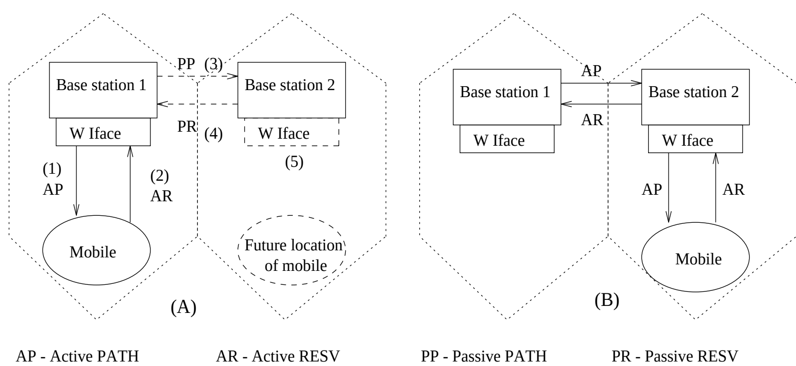

Figure 2: RSVP Messages for Reservation

Once the mobile has moved to the neighboring cell, both the mobile and the BS will know that a hand-off has been made. At this point, PATH and RESV messages denoting an active reservation are exchanged between BS 1 and BS 2 and between BS 2 and the mobile. Reservation between BS 1 and the mobile can be deleted by sending TEAR message or by just letting the the reservation to be deleted by the lack of refreshing PATH and RESV messages. In the discussion above when we say that we have a reservation, we mean that RSVP has done the end-to-end QoS negotiation and the data sent from an application will now use CBQ for packet scheduling and classification. CBQ will now make forwarding decisions on the outgoing interface which will help in achieving the promised QoS on the particular link-layer medium used by that interface. In the shared wireless environment like ours, CBQ makes decisions on the outgoing interface for the wireless downlink protocol between the BS and the mobile.

3.2.1 Features of Our Architecture

One of the major features in our architecture we need to emphasize is that we do not assume that a mobile knows where it is going. This kind of information may not be available all the time. This is unlike the architecture in [2] where a mobile requires to know the subnets it is going to visit beforehand. Our architecture assumes that a base station knows the addresses of the base stations in all the neighboring cells. This is a more reasonable assumption to make because it is easier to get this information on a wired network and this information will not change frequently. Also to be noted is the fact that a base station needs to keeps track of a fixed number(six) of addresses only. In the architecture in [2], a lot of burden is placed on the mobile host. The mobile host not only has to know the addresses of all the subnets it is moving into, which may not be possible or may be a very large number, it must also use a “proxy discovery protocol” to know the proxy agents in all these subnets.

4 A look at QoS Specifications

In this section we take a general look at the various QoS parameters that need to be taken care of in a mobile environment and how they can be specified and handled. QoS parameters for real-time services include parameters like packet delay, packet loss rate, delay jitter and minimum and maximum bandwidth which are specified and handled by Integrated Service classes supported by RSVP and CBQ. Integrated Services offers QoS based on three service classes: Guaranteed Service which provides a firm bound on data throughput and delay along the path, Controlled Load Service where probabilistic promises are made to provide some service and Best Effort Service with no performance guarantees. Apart from the above mentioned QoS parameters, there are certain parameters that will help deal with problems which are unique to a mobile computing environment [5]. One of the parameters is the loss profile which gives the applications an opportunity to choose whether a bursty loss or a distributed loss is preferred in case of an overloaded situation. This will be based on the nature of the application. The second QoS parameter is the probability of seamless communication which defines the nature of breaks that can be allowed in the service.

The two parameters – loss profiles and probability of seamless communication – can be specified along with the Traffic and Flow specifications provided with the RSVP PATH and RESV messages. Loss profiles can be guaranteed by changing the way packet dropping is done at the queues by CBQ. CBQ can drop every “1 in n” packets for a distributed loss and drop few packets is a row for a bursty loss. Seamless communication can be obtained by making sure that when a mobile moves from one cell to another, the packets that could be delayed or lost during the call hand-off are already buffered in the new cell. This can be done by multicasting some data to the neighboring cells ahead of time. The amount of buffering that needs to be done is based on the value of the probability of seamless communication parameter.

In view of using RSVP in a mobile environment, another parameter could be useful. Since a BS has to make passive reservations in advance with all the neighboring BSs there is a possibility that some of these reservations may be turned down because of lack of resources. A parameter denoting a factor by which the original resource request can be reduced in case a reservation does not go through is useful. Such a mechanism will make sure atleast some resources are reserved. We call this parameter the “ resource reduction factor”.

5 Testbed and Work in Progress

We are developing a testbed to implement and test our proposed solution. Wavelan PCMCIA and ISA cards are used on a FreeBSD platform. FreeBSD Wavelan driver supports roaming and is used in conjunction with WavePOINT. Currently Mobile IP [6] will be used for forwarding messages as the mobile moves from one cell to another. Changes are being made to RSVP and CBQ for our use. The current CBQ architecture only supports Controlled-Load Service for RSVP. Controlled Load Service provides no firm quantitative guarantees. Currently only the bandwidth parameter is in scope of this class and it is used to map the RSVP flow to a CBQ class.

Reserving resources passively needs a mechanism that lets these resources be used by other applications till the application that this reservation was intended for wants to use it. This is supported by CBQ. The CBQ mechanism allows a class of applications to borrow bandwidth from another class if the present class is going over its allocated bandwidth and also if the other class is identified as a class that the present class can borrow from. A passive reservation can be identified as a class from which other classes can borrow data. Bandwidth can be reserved for the wireless class of applications and other classes of applications may be allowed to borrow from this class until the intended user wants it.

Various changes need to be made to RSVP and CBQ and some major ones we are addressing are listed here:

- Make RSVP aware of passive reservations. RSVP is being modified such that a flag in the PATH and RESV messages indicates whether a reservation is passive or active. The rate parameter for passive reservation is also a part of the PATH and RESV messages. The average passive rate is merged separately from the regular rate parameter as the RESV messages travels from the receiver to the sender.

- Find a way to keep track of all the base stations in the neighboring cells to make passive reservations.

- Modify CBQ to make it aware of passive reservations.

- Merge requests as done in RSVP taking into account both the active and passive reservations. Scheduling needs to be done based only on the active reservation. As discussed in first listing above, passive requests are merged separately. At any node, the difference between the merged rate parameter and the passive rate parameter gives the bandwidth for the active connections. CBQ uses this bandwidth for scheduling.

- Add the loss profile and probability of seamless communication parameters onto the traffic specifications of RSVP along with the resource reduction factor parameter. These parameters are added to RSVP and are conveyed along with the other traffic parameters while sending the PATH and RESV messages.

- Make CBQ able to drop packets based on the loss profiles parameter which means we must map this parameters from RSVP traffic specification to CBQ. A loss profiles parameter is being added to CBQ. RSVP conveys this parameter to the kernel-based queueing module through an ioctl call.

6 Conclusions and Discussions

In this paper we have proposed an architecture which uses a modified RSVP protocol for resource reservations in a mobile environment. The highlight of our architecture is that we do not make assumptions about knowing in advance that path a mobile is going to follow. Resource reservations are addressed and taken care of in the network.

In all the discussions here we talked of resource reservations that has been made from the wired to the wireless networks. Reservations in the opposite direction(uplink) are also very important. For complete QoS guarantees in a mobile environment, the base station must be able to schedule the way in which it receives packets from the mobiles in addition to scheduling what is sent as discussed in this paper. This could be possibly done by some kind of polling or reservation-based mechanism. This is a topic of further study and is beyond the scope of this paper.

References

[1] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A New Resource ReSerVation Protocol," *IEEE Network*, vol. 7, pp. 8–18, Sept. 1993.

[2] A. K. Talukdar, B. R. Badrinath, and A. Acharya, "MRSVP: A Reservation Protocol for an Integrated Services Packet Network with Mobile Hosts." Tech report TR-337. Rutgers university.

[3] "Wavelan Cards." http://www.wavelan.com.

[4] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 365–386, Aug. 1995.

[5] S. Singh, "Quality of Service guarantees in mobile computing," *Computer Communications*, vol. 19, pp. 359–371, Apr. 1996.

[6] "MobileIP implementation on FreeBSD." ftp://ftp.cs.pdx.edu/pub/mobile/mip-July97.tar.gz, July 1997.