A Simple QoS Signaling Protocol for Mobile Hosts in the Integrated Services Internet

Andreas Terzis
Mani Srivastava
Lixia Zhang
University of California, Los Angeles

Abstract - With advances in packet routing technology, and resource reservation protocols, the Internet is expected to provide ubiquitous integrated transport of speech, audio, video, and other real-time multimedia data in addition to the current best effort data traffic. Such integrated transport will also need to be supported for the increasing number of mobile users who access the internet over wireless access networks, using Mobile-IP to retain continual IP connectivity. In this paper we present a simple Quality of Service (QoS) signaling protocol for mobile users in an integrated service Internet. The protocol works by combining pre-provisioned *RSVP Tunnels* with Mobile IP. Our protocol, even-though simple, captures the essence of QoS provisioning for wireless and mobile networks. The wireless medium provides a completely different medium than wires, and therefore one's expectations from it should be different. Service quality is inherently mobility dependent, and intermittent disconnections are bound to happen. It is not the signaling protocol's task to completely overcome or conceal transient conditions from applications, but rather applications should try to adapt. Our approach can be easily implemented today with minimal changes to other components of the Internet architecture. We also evaluate the application level performance impact of the QoS provisioning delays associated with our protocol on a prototypical packet speech application with various playout buffering strategies, and compare against the performance of the ordinary RSVP protocol suite with Mobile IP.

I. Introduction

There has been much recent research on the provision of Quality of Service (QoS) guarantees in wireless and mobile networks. This increased interest has been fueled by three major advances: the increase in the number of portable devices (laptop computers, PDAs etc.), the increased performance of the portable devices enabling multimedia applications, and the emergence of wireless access technologies that provide high data rates (for example, Lucent's popular WaveLAN modem [6] provides 2Mbps raw data rate, and new models in the near future are expected to provide 8-10 Mbps).

While much of the prior work on QoS in wireless and mobile networks has been done in the context of connection oriented cellular telephone and wireless ATM networks, of increasing importance is supporting QoS for mobile users accessing real-time services over the now ubiquitous Internet. Several pieces of an eventual integrated service Internet architecture capable of supporting mobile users already exist. The RSVP protocol suite [5] allows resource reservation in the Internet for supporting real-time services. Mobile-IP [20], an IP routing protocol supporting host mobility, has been standardized by the IETF to provide continual IP connectivity to mobile users. Over the last two years, several packet scheduling algorithms that address the special characteristics of wireless links while supporting soft notions of quality of service have been proposed [8, 12, 17].

The missing piece, we believe, is a QoS signaling protocol. Although several proposals exist today for QoS in mobile networks in general, we believe that a fundamentally simpler solution exists for the Internet that uses simple extensions of the RSVP protocol and can be deployed easily with minimal changes to the rest of the protocol suite. The key realization is that the time varying wireless medium has unique characteristics that requires one to lower *expectations* regarding services provided and focus on adaptation techniques and class based scheduling to cope with transient conditions [24].

The rest of the paper is structured as follows. In Section II we present RSVP tunnels on which our protocol is based. In Section III we review the prior work on QoS provisioning and resource allocation for mobile users. In Section IV we argue that the right Quality of Service model in wireless networks cannot provide hard guarantees and present the design principles we used in designing our signaling protocol. Section V provides a detailed description of the RSVP tunneling based QoS provisioning protocol, while Section VI evaluates the application level performance impact of our approach. Finally, Section VII summarizes the key points in our approach, and concludes with a discussion about our future work.

II. Background: Supporting RSVP Signaling Over IP-in-IP Tunnels

In this section we describe an extension to RSVP for IP-in-IP tunnels previously proposed by two of the authors. This extension, RSVP *Tunnels*, in combination with Mobile IP forms the crux of the protocol described in this paper for QoS provisioning for mobile hosts.

Tunneling is an integral part of many routing protocols supporting mobile nodes (e.g. Mobile-IP uses IP-in-IP encapsulation [19] to tunnel packets from the home agent to the foreign agent). IP-in-IP tunnels cause problems in the regular processing of RSVP messages, since RSVP messages get "lost" when they cross a tunnel[1]. The RSVP *Tunnels* proposal describes an approach for providing RSVP protocol services over IP-in-IP tunnels. The main idea proposed in [31] is to have a separate RSVP session (in RSVP jargon a session defines one simplex

1. RSVP packets use the Router Alert option to indicate to routers on the path that they require special handling. When RSVP messages are encapsulated with an outer IP header the Router Alert option becomes invisible.

unicast or multicast data flow for which reservations are required), between the tunnel end-points.

The tunnel entry point R_{entry} serves as the sender for the Tunnel session, while the tunnel exit point R_{exit} serves as the receiver. The Tunnel session can exist independently from End-to-End sessions (e.g. created via a management interface), or it's creation can be triggered from End-to-End messages. When an End-to-End RSVP session crosses an RSVP-capable tunnel it is mapped to a tunnel RSVP session. The tunnel RSVP session views the two tunnel end-points as two end hosts. Then a reservation is made from R_{exit} for R_{entry} for the amount of data crossing the tunnel. The original, end-to-end RSVP session views the tunnel as a single (logical) link along the path between the source(s) and the destination(s). *PATH* and *RESV* messages of the End-to-End session are encapsulated at one tunnel end-point and get decapsulated at the other end, where they get forwarded as usual.

Data packets are encapsulated with an IP as well as a UDP header when crossing the tunnel. In this way, packets belonging to diferent flows can be distinguished by routers inside the tunnel, using standard RSVP processing. For more information, the interested reader is encouraged to look into [31] for a detailed description of the protocol extensions.

III. Previous Work on QoS in Wireless and Mobile Networks

Prior work on QoS in wireless and mobile has mostly been done in the context of connection-oriented cellular networks, and more recently wireless and mobile ATM. In mobile and wireless ATM, the QoS problem has been typically been formulated as that of Virtual Connection (VC) rerouting, where the a new path, and associated resource reservations, need to be re-established to the new location of the mobile after it undergoes a hand-off. Many different schemes have been proposed for mobile ATM over the past few years [1, 2, 11, 14, 21, 28, 29], and the signaling protocol for VC rerouting in mobile ATM is now a subject of standardization by the ATM Forum (e.g. [10]).

For connectionless network, Talukdar et al. in [26] have proposed augmenting the Integrated Services architecture to support QoS for mobile nodes. Specifically they propose augmenting the service models with the *mobility* dimension. The mobility is assumed to be predictable and it can be characterized precisely by a *mobility specification* which consists of the set of cells the mobile node is expected to visit during the lifetime of the flow. A node can request a service that is *mobility independent* whereby it will receive the same service in any cell that is within it's mobility profile, or a *mobility dependent* service that is only guaranteed in the current location.

In [27], Talukdar et al. have proposed the MRSVP signaling protocol as part of the augmented Integrated Services architecture presented in the previous section. RSVP, as originally envisioned, cannot support mobility independent reservations. So the MRSVP proposal suggests making resource reservations to all locations where the mobile node is expected to visit during the lifetime of the connection. The mobile node will make an active reservation to its current location but it will also make passive reservations to each of its locations in its mobility specification. The difference between an active and a passive reservation is that in active reservations the data flows through the path where resources are reserved while in passive reservations the resources are reserved but no data from the session flows through that path. Whenever the mobile node moves from one location to another, the active reservation from the sender to its previous location will be turned to a passive reservation and the passive reservation to its new location will be turned into an active reservation.Active and passive reservations for the same session are merged in the same way that usual RSVP requests get merged in routers.

IV. Design Principles Underlying Our Protocol

Mobility indepent service, is overly optimistic in the service it promises to provide and unreasonably wasteful in network resources use. First of all there is the question of whether a mobile node is able to predict its course accurately. The premise is that a node will receive consistent service as long as it stays within its mobility profile. If the node cannot predict accurately it's mobility profile then the perceived service will be mobility dependent!

Furthermore the service received while the mobile node moves across cells depends on other factors that are out of the reach of the signaling protocol. For example, the routing protocol itself may cause discontinuities in the service provided (Mobile-IP and it's various extensions do not make any claims about continuous connectivity). Continuous connectivity also depends on data-link and physical connectivity while the mobile node moves across domains. Even if continuous connectivity at the physical and MAC layer is possible (e.g. well engineered system with cell overlay, radios that can listen for multiple beacons), it cannot be guaranteed and certainly it cannot be predicted in advance.

A second consequence of the mobility independent model, is that resources have to be reserved along multiple paths even though they are not being used. Over-reserving resources of a scarce medium such as wireless bandwidth is grossly inefficient[2].

We think that the wireless medium requires a fundamental change in the *expectations* we have from the service and the level of assurance provided by the network. Applications should request some service from the network but in the case of failure they should be able to adapt or renegotiate the service request and in any case gracefully deal with changing network conditions.

Taking all the above factors in consideration we present, in the paragraphs that follow, the set of basic design principles that define our approach.

1. **Minimum changes to static nodes**. The new QoS protocol for mobile nodes must require the minimum number of changes if it is to be adopted widely.
2. **Support for *soft* QoS guarantees**. We think reservations should provide a *base* layer. Above that, applications should be able to adapt to changing network conditions both in the wireline infrastructure as well as over the wireless link.
3. **Minimum changes to RSVP**. The new protocol should inter-operate with RSVP and it could leverage from the installed

2. Talukdar et al. [26] try to address the issue of under-utilization by "over-booking" resources to mobility-independent and mobility-dependent mobile nodes.

base of RSVP implementations if it requires changes only to the absolute minimum number of places.

4. **Keep RSVP features**. The defining characteristics of RSVP, specifically the soft state of the protocol and the independence from the routing protocol, are most valuable in the highly volatile environment of wireless networks. It is therefore one of our principal goals not to change any of those characteristics but to rather base our protocol on them.
5. **Minimum disruption of service**. If one tried to re-establish the full path of the reservation each time the mobile node changed cells then reservations would have to compete (and possibly fail) for resources over the wireline infrastructure as well as over the wireless medium. This situation would create undue disruptions of service and possibly large latency. We strive towards a solution that would provide the minimum disruption due to node movement without having to pre-allocate resources statically.

After reviewing these principal goals we think that some simple extensions to the RSVP protocol could serve the purpose of providing service differentiation to mobile nodes in the Internet. We continue in the next section by describing the proposed solution.

V. Protocol Description

The proposed QoS provisioning protocol roughly consists of two parts: modifications necessary to Mobile IP and modifications in RSVP. We describe these parts in the sections that follow.

A. Modifications to Mobile IP

We describe here changes or additions to the Mobile-IP functionality to support resource reservations for mobile nodes. Two of the Mobile-IP functions have to be enhanced to support RSVP capable end nodes: *Agent Discovery* and *Mobile Host Registration*.

Agent Discovery and Registration

Mobile-IP uses the Agent Discovery mechanism for mobility agents to advertise themselves and mobile nodes to identify the network they currently reside. Base stations advertise their capabilities by sending Agent Advertisment messages. The format of these messages is described in [16] and [20].

We propose adding a Q bit in the characteristics bitmap, signifying the fact that the mobility agent can support QoS requests and specifically the protocol we propose here. When a mobile node moves to a new cell and hears an agent advertisement containing the Q bit it knows that the new agent supports QoS requests. In the case where the mobile node has some ongoing QoS sessions and it moves to a cell where QoS is not supported then an error will occur and it should be reported to the other end(s).

As in the case of Agent Discovery above, we propose adding a Q bit in registration requests which signifies that the Care-of Address can handle QoS requests and therefore the Home Agent should forward them.

B. Unicast Case

We describe first the case where we have a unicast RSVP session. The are two scenarios. In the first case the mobile node is the receiver and in the second case the mobile node is the sender. We examine both cases below.

MH Receiver

Figure 1 gives a schematic of how the protocol works in the case where the mobile node is the receiver. Initially the mobile node resides in Cell A. It starts receiving *PATH* messages from *CH*. It then starts sending *RESV* messages with its reservation request. The *RESV* messages set up the reservation state from the *MH* all the way to the *CH*.

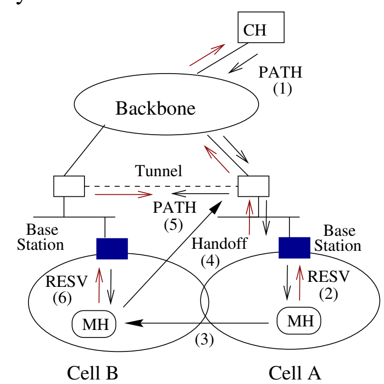

Fig. 1. Mobile Host as a Receiver

When the *MH* moves to Cell B it informs its home agent of its new location. When the HA is informed about the *MH*'s new location it does two things: (1) It sets up a *tunnel* RSVP session between itself and the foreign agent if one does not exist between them. The Tspec in the *PATH* for the tunnel session are the *sum*[3] of all the Tspecs of *End-to-End PATH* messages. (2) It encapsulates *PATH* messages from the *CH* and sends them through the tunnel towards the *MH*'s new location.

Usually reservations are refreshed every 30 seconds. If we require the minimum disruption of service, then the mobile node should send a *RESV* refresh message right after receiving a *PATH* message in the new cell. Note that the mobile node should wait to receive a *PATH* massage before sending its *RESV* refresh. The reason for that is that the receipt of a *PATH* message signifies that the foreign agent has created the corresponding Path State and will forward the request. If a *RESV* arrives at the foreign agent for a session it doesn't know about it will drop the request returning a *ResvErr* message.

When the foreign agent receives a *RESV* message from the visiting mobile node for which it already has Path State, it will go through the following steps.

1. It will send a *RESV* message for the corresponding tunnel session between itself and the *HA*. If this is not the first *RESV* message for the tunnel session it will instead send a refresh message with a modified flowspec. The modified flowspec is the "sum" of all the flowspecs from end-to-end requests mapping to the specific tunnel session.
2. After sending the reservation request, the *FA* waits for a confirmation from the *HA* that the reservation over the tunnel was successful. By doing so, it makes sure that reservations

3. The sum of Tspecs can be defined in many ways either by adding peak rates fashion or using some kind of *equivalent bandwidth* metric, but this is out of the scope of this document

over are reserved before forwarding the request over the tunnel.

3. After receiving the confirmation for the tunnel reservation, the *FA* encapsulates[4] the end-to-end *RESV* message and sends it to the *HA* where it will be decapsulated and forwarded towards the *CH* as needed.

The tunnel between the home agent and the foreign agent could be pre-configured. In this way a level of resources is pre-allocated between the two agents for the mobile nodes that are crossing between the two cells. One can imagine a “threshold” scheme by which the foreign agent initially allocates a small amount of resources for traffic flowing from the home agent. When the resources requested by mobile nodes visiting the foreign agent’s cell surpass the reserved amount, the agent makes a reservation for an additional “chunk” of resources over the tunnel. Using this scheme, the latency of establishing the reservations over the tunnel is reduced and the service provided is “smoother”.

MH Sender

In this scenario the mobile node is the sender of the session. As the sender it starts sending *PATH* messages towards the receiving host *CH*. Once *CH* receives the *PATH* message, it will send a *RESV* message with it’s reservation request.

When the mobile moves to cell B, if we use regular Mobile-IP tunnels, it’s packet will follow a different path towards *CH* that does not necessarily go through the *HA*. The implication is that *PATH* messages sent from the mobile node will now follow a different path up to a *merging point* towards *CH*. Two consequences of the changed path are: (1) the reservation over the new path may fail and (2) it may take longer to get established. Note that the asymmetric routing to and from the mobile node does not affect things in this case since *RESV* messages follow the *reverse* path established by *PATH* messages.

If one believes that the two implications mentioned above are restrictive then one can set up a *reverse tunnel*[5] ([16]) from *FA* to *HA*. If a reverse tunnel is established, then we will have a similar case as the one described in the previous paragraph. *FA* is the one sending *PATH* messages and *HA* sends *RESV* messages. The two different cases are identified in Figure 2 with two different labels **a** and **b**.

Note that RFC2344 [16], specifying reverse tunneling, provides *selective* reverse tunnels whereby some of the datagrams sent from the mobile node are reverse tunneled while the rest are routed as regular traffic. In our case, this could be used to route packets belonging to reserved sessions over the reverse tunnel while the rest of the mobile’s datagrams would be routed as usual.

C. Multicast Case

Multicast forwarding presents special problems in the case of mobile nodes. First, mobile nodes that wish to receive multicast packets have multiple ways to register via IGMP messages and second multicast routing in general depends upon the IP source address. This means that a mobile node visiting a foreign network might not be able to send multicast packets using it’s home address via regular routing. We investigate the peculiarities of multicast receivers and senders in the two paragraphs that follow.

MH Receiver

A mobile node that wishes to receive multicast packets has to register via IGMP messages. It can either do so via a (local) multicast router on the visited subnet. Alternatively, a mobile node which wishes to receive multicast packets may join groups via a bidirectional tunnel to its home agent. The mobile node tunnels IGMP messages to its home agent and the home agent forwards multicast datagrams down the tunnel to the mobile node.

In the first case, existing reservations should get rerouted over the new path. This can induce extended periods where the reservation is not in place. As an alternative the mobile could establish a *selective reverse* tunnel towards it *HA* and forward IGMP joins over the reverse tunnel. If it does so, multicast packets destined for the mobile are delivered on it’s behalf to the *HA* that encapsulates them and delivers them to *FA* which in term sends them to the mobile. Resource reservation in this scenario is the same as in the unicast case.

MH Sender

As in the unicast case, a mobile node that is sending to a multicast group has two different options. Namely it can either send packets using usual routing at the visiting subnet or it could reverse tunnel it’s packets back to it’s home network. Depending on the choice of forwarding the QoS signaling is done the same way as described in unicast case.

D. Error Cases

Error conditions occur when a mobile node moves to a new cell that does not support QoS or when it’s home agent does not support RSVP tunneling. In both cases the node should be notified and if it has ongoing RSVP sessions the appropriate RSVP error messages should be created.

A mobile that receives an error message during the registration process and had ongoing RSVP sessions should send the appropriate teardown messages to the other end(s). There are two possible cases.

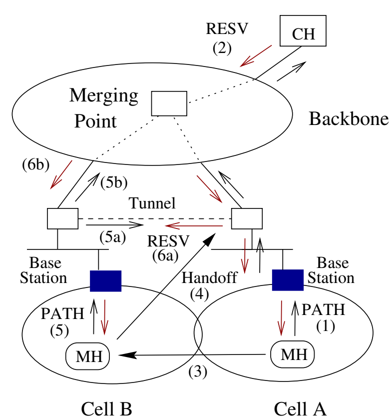

Fig. 2. Mobile Host as a Sender

4. There exists a *reverse* tunnel from the *FA* to the *HA* but this is setup to forward RSVP messages only and not traffic from the mobile node
5. This can be done only if the *FA* supports this mode of operation

1. **MH as sender**. The mobile node it should send *PathTear* for all the sessions it is participating as a sender. RSVP processing rules [6], dictate that if a RSVP daemon receives a *PathTear* for which it does not have Path State it should drop the packet without further forwarding it. Therefore if the mobile node simply sent a *PathTear* it would be dropped by the RSVP daemon[6] at the foreign agent (if one exists there). A possible solution would be RSVP daemon to first check if the message comes from a visiting mobile node and if so to encapsulate the packet and send it to the home agent where it will processed further.
2. **MH as receiver**. The mobile node in this case should send *RevTear* messages for all the RSVP sessions it is participating in. The same rule for *PathTear* messages applies for *ResvTear* messages too. The same solution could be applied in this case.

VI. Evaluation of Impact on Application Performance

The previous sections described the rationale and the details of our approach of providing QoS provisioning to mobile users in an integrated service Internet with minimal changes to the RSVP and Mobile-IP protocol suites. In this section we use simulation to evaluate the effect of signaling overhead induced by our protocol on application level performance, and compare it with plain RSVP with no pre-provisioned RSVP tunnels. The following subsections describe our simulation approach, the application model used, and

A. Simulation Platform

The performance evaluation was done using a RSVP Simulator, written at UCLA's Internet Research Lab using the PARSEC parallel simulation language [18]. The simulator provides an implementation of the full RSVP functionality, together with the proposed RSVP tunnel mechanism, and has a graphical user interface for user interaction.

RSVP routers in our simulators are represented by logical processes that communicate by exchanging time-stamped messages. Each of the logical processes simulating RSVP routers keeps a full log of it's RSVP state available to the user. Moreover a complete log of all data packet exchanges is kept so information such as packet delay and loss are available at the end of simulation.

B. Application Model

The design of network control policies is ultimately driven by the traffic characteristics and the QoS requirements of the applications that use the network. We have chosen to evaluate the effectiveness of our RSVP-tunnel based QoS provisioning protocol on the performance seen by a model application that is typical of real-time UDP applications being deployed for real-time services on the Internet. The model application attempts to capture the characteristics of a large class of applications that generate traffic according to some periodic pattern and require *bounded-delay* transmission. These applications can often tolerate a small amount of packet loss by employing loss-concealment techniques. Examples include packet telephony, teleconferencing, interactive video games etc.

Speech traffic source

The traffic source for the model application is a 2-state Markov chain with two states, TALKING or SILENT. This model, based on the one used in [9], simulates a speech source with silence detection, with one state corresponding to talkspurts and the other state corresponding to silences and gaps in speech. During talkspurts, a Constant Bit Rate (CBR) data stream is generated, and packets are injected at periodic intervals into the network. During silences and gaps, no packets are sent. All talkspurts and silence periods have statistically independent exponentially distributed durations.

In our simulation, speech is divided into frames of 16 ms. During talkspurts the compressed speech corresponding to each frame is packetized and sent as a single UDP/IP datagram. The speech payload is 64 bytes / frame. Together with a 8 byte UDP header and a 20 byte IP header, one gets a total packet size of 9bytes. The model parameters (Markov chain transition probabilities) are chosen so that the average durations of talkspurts and silences correspond to empirical measurements: 1.0s for talkspurts, and 1.35s for silences [9].

Adaptive playout schemes

The receiver implements a playout process that determines when an incoming packet is ready for presentation. The goal of this playout process is to compensate for delay jitter introduced by the network and restore the inter packet timing pattern as seen at the sender. Packets that arrive after their playout deadline are discarded.

We examined various policies for determining the playout time of packets. A simple policy would be to buffer the first arriving packet for a fixed time interval and then schedule a playout time for each subsequent packet at intervals of *1/r*, where *r* is the transmission rate during talkspurts expressed in packets per unit time. Such a scheme is suitable only for networks which guarantee bounds on the maximum variation in the transit delay suffered by packets. Such is not the case with the Internet, particularly when mobile users hand-off. Therefore, we examined the performance of our protocol with two *adaptive* playout policies, in which the playout time is varied over time using estimates of the mean and variance of the per-packet delay. The playout time of the first packet in a talkspurt is decided according to the adaptive algorithm, and the playout times of the subsequent packets in the talkspurt are scheduled at intervals of *1/r*. Such adaptive playout policies have been successfully used in the Internet where the delay values vary over considerably over the lifetime of a connection, depending on the degree of network congestion [3]. The two adaptive schemes implemented in our simulation were originally proposed in [22] and differ in how they estimate the delay of packets in transit.

The first scheme uses a low pass filter to estimate the mean delay and delay variance from the measured per-packet delay, n_i. This scheme is identical to how retransmission timers are set in TCP and a slight variant of this scheme is used by the Internet voice tool *vat*[7]. These schemes are typically designed to ignore

6. One might ask why the foreign agent has RSVP since it does not support QoS. We present here the worst case scenario for completeness, even-though we think this is unlikely to happen.

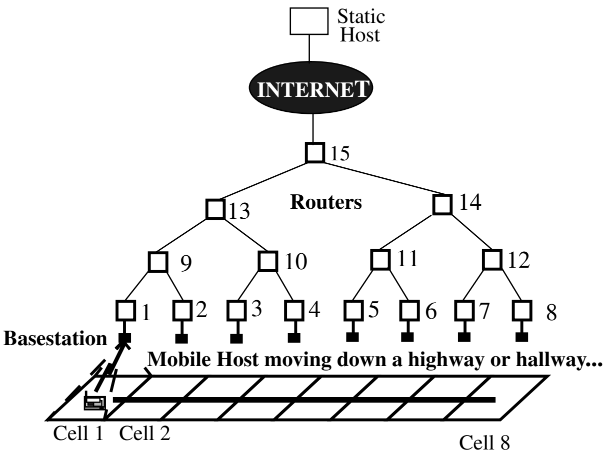

Fig. 3. Network Configuration Used for the Simulation Study

noise in the measurements while retaining the ability to track long-term fluctuations in the delay values.

The second adaptive playout scheme is based on the empirical observation that there can sometimes be *delay spikes* in the transit time of packets going over multi-hop paths in the Internet. Therefore, this scheme tries to track delays more accurately by detecting the beginning and end of delay spikes and by using different delay estimation algorithms in the *normal* and *spike* modes. A delay spike is flagged when the difference in the transit delays of two consecutive packets exceeds a threshold value. In the normal mode, a low pass filter is used as in the previous scheme. However, in the spike mode, the estimator tracks the actual measured values exactly. The variance estimation and playout time computation is the same as in the previous scheme. We chose to implement and study this scheme because intuitively the effect of a handoff would be to cause exactly such a delay spike if the network were to buffer (and/or salvage) packets during the path rebuild. The details of this scheme are presented in [22].

C. Configuration of the Simulated Network Scenario

Figure 3 shows the representative network scenario used in our simulation. The topology models an indoor wireless microcellular Local Area Network (LAN) connected through intermediate routers to the wide area Internet. The LAN topology models an office corridor type layout with basestation routers with radio ports laid out at regular distances. There is a two-level router hierarchy above the basestations[8]. For simplicity we use an arrangements of the basestations, in which the physical layout of the basestation matches their logical interconnection. The links between routers were modeled to have 1 ms delays and a 1 Mbps capacity - the precise values are not important since the intent is to obtain relative numbers.

The mobile host moves from Cell 1 to Cell 2 to Cell 3... and so on until it reaches Cell 8. It therefore undergoes seven hand-offs. The hand-offs are spaced 10 seconds apart.

The goal of the simulation study is to evaluate the reduction in signalling overhead due to pre-provisioned RSVP tunnels used by our QoS signaling protocol as compared to using plain RSVP. The only delay that we are interested in is the signalling delay - other delays, such as for radio level hand-off or for route updates by Mobile-IP are common to both the cases, and not of direct interest. Therefore, the other less relevant aspects of the performance were abstracted away in our network scenario. In particular, (i) radio level hand-off is assumed to take place immediately, and (ii) overhead of Mobile IP was set to zero - in other words, the sender host is modeled to immediately know the new location of the mobile host after the handoff.

Another abstraction that we made for simulation purposes relates to the packet scheduling and buffering within the routers. RSVP is just a signaling protocol, and works together with packet scheduling and buffer management within the routers. As articulated above, our focus is on the disruption induced due to signalling, so that details of packet scheduling and buffer management are not directly relevant. However, within the simulator we do need to model the different treatment of a flow by a router when that flow has resource reservation vs. when the flow does not have resource reservation. We adopted a simple model of the routers whereby it gives the desired fixed bandwidth to the speech flow when a reservation exists for that flow, and gives the flow a fixed and much reduced bandwidth (thus modeling the impact of cross traffic) when there are no resources reserved for the flow, as would be the case soon after a hand-off.

Finally, to study the performance impact of our scheme in the common case where the RSVP tunnels are already provisioned, we arranged for another mobile to move ahead of the first mobile, leading by 10 seconds. This leading mobile results in the RSVP tunnels being pre-configured for the mobile understudy, and any subsequent mobiles.

D. Impact on Application Performance

Figures 4 and Fig. 5. respectively show the performance of the application with plain RSVP for QoS provisioning, and our proposed scheme which in the common case has pre-provisioned RSVP tunnels for lower latency. Each of the figure shows the performance for both the adaptive playout schemes mentioned previously. Each of the graphs plots two curves: one is the time at which the packets arrive at the receiver, and the other is the playout deadline as calculated by the playout scheme. Note that the X-axis shows the progression of time in terms of the sequence number of speech packets. Since the speech traffic has periods of silence, the rate of increase of packet numbers as a function of time is not constant: the correspondence between packet numbers and time is not linear. As mentioned earlier, each packet represents 16 ms. However, time also advances during the silence periods even when no packets are being sent. This is why the hand-off events, while equally spaced in time at 10 s intervals, are not uniformly spaced in terms of packet numbers.

A packet may not be played for one of two reasons. First, the packet may be lost in the network. This happens because of the service degradation due to loss of resource reservations during

7. Vat does not maintain a delay estimator. Instead, it updates the variance estimator in Equation 1 by replacing $\hat{d}_i$ by n_{i-1}. It also uses additional knobs to modulate the rate of change of the playout interval.

8. A similar hierarchy is common in many organizations with departmental machines on the same Ethernet segment and multiple Ethernet segments connected through a hub.

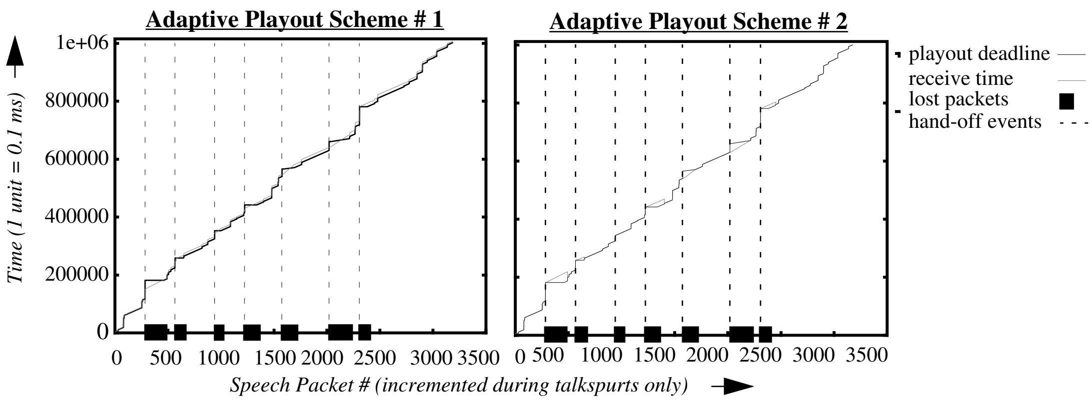

Fig. 4. Performance of adaptive real-time speech application with QoS provisioning using plain RSVP

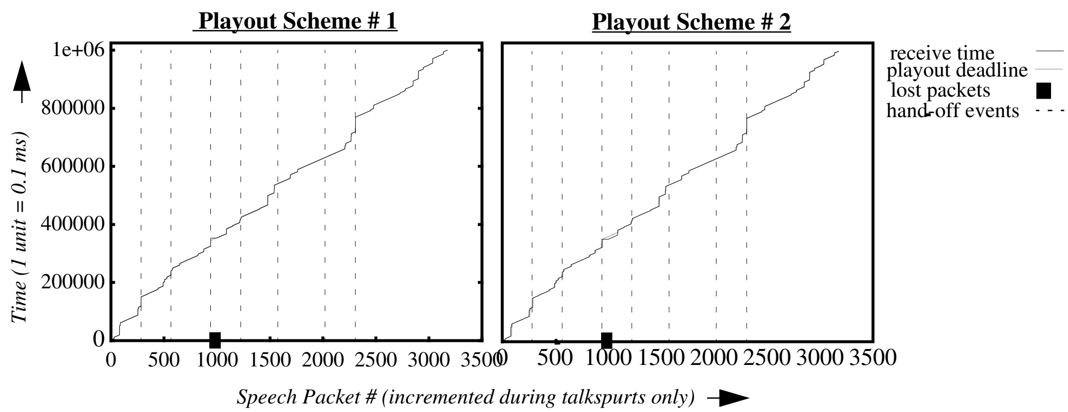

Fig. 5. Performance of adaptive real-time speech application with QoS provisioning using proposed protocol based on RSVP tunnels

	dynamic provisioning		pre-provisioning	
	Scheme #1	Scheme #2	Scheme #1	Scheme #2
Late Packets	120	181	20	47
Lost Packets	654	654	37	37
Packets played out	2401	2340	3118	3091
Total	3175	3175	3175	3175
Average Disruption on Hand-off (in lost packets)	110.6	119.3	8.1	12

TABLE I: COMPARISON OF PACKET LOSS STATISTICS

the interval from the time mobile moved to the new location till the signaling protocol is able to re-establish new resource reservations. Second, the packet may suffer an increased delay during the hand-off, and arrive after its playout deadline has past. In such a case the playout algorithm will drop it. In Figures 4 and Fig. 5., the packets that are dropped in the network are indicated by the solid line segments along the X axis, while the packets that are not lost but are late for playout are indicated by regions where the second curve (playout deadline) is higher than the first curve (arrival time at the receiver). The total disruption at the application level due to the signaling overhead is the sum of the two: lost packets and late packets.

Visually, it is clear from Figures 4 and Fig. 5. that the ability of our protocol to pre-provision RSVP tunnels that can be used by mobiles is highly beneficial, and results in much reduction (almost total elimination) of packet losses as well as late packets compared to using plain RSVP which does dynamic QoS provisioning on every hand-off. Indeed Table 1, which quantitatively summarizes the simulation results, verifies this. The overall application level performance disruption is reduced by large factors (about x10-15) by the ability of our protocol to pre-provision RSVP resource reservations. In fact, the hand-of disruption is almost eliminated with our protocol in nearly all the hand-off events: the network losses are reduced, as are the late arrivals.

Finally, while it does not detract from our overall result, an apparently puzzling observation from Table 1 is that the adaptive playout algorithm #2, which uses spike following to track sudden increases in network delays, actually does worse than algorithm #1. A detailed analysis of our simulation logs show that actually the scheme does indeed, as expected, do better during the delays spikes encountered in hand-offs. However, likely because of poorly tuned control algorithm parameters, overall algorithm #2 declares more packets to be late. There were many "narrow misses" of the playout time resulting in an excess number of late packets.

VII. Conclusions

We have presented here a new QoS signaling protocol for wireless and mobile networks. The protocol works by encapsulating end-to-end RSVP messages over the tunnel connecting the home agent with the foreign agent and by recursively applying RSVP for the connection between those agents. The RSVP "pipe" connecting the agents serves the aggregate traffic crossing the two cells and can be administratively created in advance or dynamically created. The size of the reserved "pipe" can shrink or increase depending on the reserved traffic between the two agents, triggered by end-to-end RSVP messages.

Our protocol, while not supporting all the proposed services to mobile nodes (e.g mobility or location independent QoS guarantees), is realistic and easily implementable. Furthermore it is not inefficient with scarce network resources and requires minimum changes to the network architecture and static hosts. The low overhead due to the simplicity of our protocol, together with the use of RSVP tunnels which can be pre-provisioned, leads to much lower service disruptions during hand-offs for typical internet real-time applications that use receive-end playout buffering to cope with packet delay jitter and losses.

Acknowledgments

Andreas Terzis would like to thank Constantinos Nikoloudakis and Lan Wang for their original work on the RSVP simulator.

References

1. A. S. Acampora, and M. Naghshineh. An architecture and methodology for mobile-executed handoff in cellular ATM networks. *IEEE Journal on Selected Areas in Communications*, vol.12, (no.8), Oct. 1994. p.1365-75.
2. B. A. Akyol and D. C. Cox. "Rerouting for Handoff in a Wireless ATM Network.", ICUPC 96, Sep. 1996.
3. J-C Bolot. "End-to-end packet delay and loss behavior in the Internet." in *Proceedings of ACM SIGCOMM '93*, San Francisco, California, pp. 289--298, September 1993.
4. R. Braden and L. Zhang. Resource ReSerVation Protocol (RSVP), Version 1 Message Processing Rules. *RFC 2209*, September 1997.
5. R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP), Version 1 Functional Specification. *RFC 2205*, September 1997.
6. A. Claessen, L. Monteban, and H. Moelard. The AT&T GIS WaveLAN air interface and protocol stack. *Proceedings of the 5th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'94)*, 1994..
7. G. Dommety, M. Veeraraghavan, and M. Singhal. Route Optimization in Mobile ATM Networks. Proceedings of MobiCom'97, pp. 43-54, Budapest, Hungary, September 1997.
8. C. Fragouli, V. Sivaraman, M. B. Srivastava. Controlled Multimedia Wireless Link Sharing via Enhanced Class-Based Queuing with Channel State Dependent Packet Scheduling. *Proceedings of IEEE INFOCOMM '98*, March 1998.
9. D. J. Goodman, and S. X. Wei. Efficiency of packet reservation multiple access. *IEEE Transactions on Vehicular Technology*, Feb. 1991, vol.40, (no.1, pt.2):170-6.
10. C. Kalmanek, P. Mishra, and M. Srivastava. ATM Virtual Circuit Routing to Support Mobile Host Roaming. ATM Forum contribution #97-0155, February 1997, San Diego, California.
11. K. Keeton, B. Mah, S. Seshan, R. Katz, and D. Ferrari. Providing Connection-Oriented Services to Mobile Hosts. In *Proceedings of the USENIX Symposium on Mobile and Location-Independent Computing*, pp. 83-102, Cambridge, Massachusetts, August 1993.
12. S. Lu, V. Bharghavan, and R. Srikant. Fair Scheduling in wireless packet networks. *Proceedings of ACM SIGCOMM '97*, Cannes, France, Sep 1997.
13. S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driver layered multicast. *Proc. ACM SIGCOMM '96*, pages 117--130, Stanford University, CA, Sep 1996.
14. P. Mishra and M. Srivastava. Call-Establishment and Rerouting in Mobile Computing Networks. *AT&T Bell Laboratories Technical Memorandum 11384-940906-13TM*, September 1994.
15. P. Mishra and M. Srivastava. Effect of Connection Rerouting on Application Performance in Mobile Networks. IEEE Transactions on Computers, vol. 47, no. 4, April 1998.
16. G. Montenegro. Reverse Tunneling for Mobile IP. *RFC 2344*, May 1998.
17. T. S Eugene Ng, Ion Stoica, and Hui Zhang. Packet Fair Queueing Algorithms for Wireless Networks with Location-Dependent Errors. *Proceedings of IEEE INFOCOMM '98*, March 1998.
18. PARSEC: Parallel Simulation Environment for Complex Systems. *http://may.cs.ucla.edu/projects/parsec/*.
19. C. Perkins. IP Encapsulation within IP. *RFC 2003*, October 1996.
20. C. Perkins. IP-Mobility Support. *RFC 2002*, October 1996.
21. B. Rajagopalan. Mobility Management in Integrated Wireless-ATM Networks. *Proceedings of MobiCom'95*, Berkeley, California, November 1995.
22. R. Ramjee, J. Kurose, D. Towsley and H. Schulzrinne. "Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks." In *Proceedings of IEEE Infocom '94*, pp 680-688, Toronto, Canada, June 1994.
23. R. Ramjee and T. La Porta. "Experimental Evaluation of Connection Rerouting Schemes for ATM-based Wireless Networks." AT&T Bell Laboratories Technical Memorandum 113450-960122-01TM, January 1996.
24. M. Srivastava and P. Mishra. On Quality of Service in Mobile Wireless Networks. *Proceedings of Network and Operating System Support for Digital Audio and Video (NOSSDAV '97)*, St. Louis, Missouri, May 1997.
25. R. Steinmetz, "Human Perception of Audio-Visual Skew," in Architecture and Protocols for High-Speed Networks, pp. 235--252, Kluwer Academic Publishers, 1994.
26. A.K. Talukdar, B.R. Badrinath, and A. Acharya. On Accommodating Mobile Hosts in an Integrated Service Packet Network. *Proceedings of IEEE INFOCOM '97*, 1997.
27. A.K. Talukdar, B.R. Badrinath, and A. Acharya. Admission Control Algorithms and Reservation Protocol for Supporting Mobile Hosts in an Integrated Services Packet Network. Technical report, Dept. of Computer Science, Rutgers University, 1998.
28. C-K Toh. "The Design and Implementation of a Hybrid Handover Protocol for Multi-Media Wireless LANs." In *Proceedings of the First International Conference on Mobile Computing and Networking (MobiCom '95)*, pp. 49-61, November 1995.
29. M. Veeraraghavan, M. Karol and K. Eng, "Implementation and Analysis of Mobility Management in a Wireless ATM LAN." In *Proceedings of 8th IEEE Workshop on Local and Metropolitan Area Networks,* Berlin, Germany, August 1996.
30. O. Yu and V. Leung, "Extending B-ISDN Architecture to Support User Terminal Mobility over an ATM-Based Personal Communications Network," Globecom'95, Dec. 1995.
31. L. Zhang, J. Wroclawski, J. Krawczyk, and A. Terzis. RSVP Operation Over IP Tunnels. *Internet-Draft, work in progress*, February 1998.