Providing Seamless QoS for Multimedia Multicast in Wireless Packet Networks

Dongyan Xu, Baochun Li, Klara Nahrstedt, Jane W.-S. Liu *

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801

ABSTRACT

To support real-time multimedia applications in wireless packet networks, it is an essential challenge to provide seamless Quality of Service (QoS) to mobile users. In this paper, we address the problem of real-time multimedia multicast in cellular networks, and present our solution to avoid large QoS fluctuations during handoffs. Specifically, during a multicast session, a mobile host may experience varying packet delay, delay jitter, and channel error when it moves from one cell to another. It is thus desirable that these *location-dependent* QoS parameters appear as seamless as possible to mobile hosts. We present protocols to achieve a degree of transmission synchronization among multiple cells, so that the delays and delay jitters of each packet to all subscribing mobile hosts do not vary substantially. In addition, we apply *Forward Error Correction* (FEC) technique to recover the QoS-mandatory packets from wireless channel errors. We show through analysis and simulation that the mobile hosts will experience brief, smooth, and low packet loss rate handoffs.

Keywords: multimedia multicast, wireless packet networks, QoS, handoff, error recovery

1. INTRODUCTION

Wireless multimedia applications are gaining popularity because they provide great convenience to mobile users to perform not only data, but also voice and video communications. In the meantime, the availability of various wireless packet networking technology makes it possible to develop mobile multimedia applications with improved Quality of Service (QoS). One of such applications is wireless multimedia multicast, which we will focus on in this paper. We assume that there is a single source of multimedia data, the source can be either a static host in the wired networks, or a mobile host (MH) connected by cellular networks. The source multicasts to a certain geographic area, which is covered by a set of cells. Any subscribing MH will stay tuned in the multimedia multicast session while they move freely in this covered area. It is desirable that the MHs are provided with constant level of QoS, no matter they are static or mobile, and no matter which cell they are currently in. In particular, during a handoff when a MH moves from one cell to another, the QoS oscillation observed by the MH should be kept as small as possible. The handoff procedure should be brief, smooth, and low in packet loss rate. The goals we describe here are not trivial because of the following problems.

Firstly, since the routes from the source to the destination cells are different, it is likely that MHs in different cells will observe varying packet delays and delay jitters, i.e. for the same packet sent from the source, it (copies of it, to be precise) is received by the MHs not at the same time, because of the different routes in the multicast tree and the different packet scheduling mechanisms used by network switches on the routes. Therefore, in wireless multicast sessions, packet delay and delay jitter are both *location-dependent* QoS parameters, and addition mechanism is needed to make them as seamless as possible to MHs. Otherwise the handoff procedures may not be smooth: suppose a MH moves from cell *c* to cell *c'*, and the end-to-end delays of packets from the source to cell *c* and cell *c'* are *d* and *d'* respectively. If $d > d'$, then the MH will experience packet losses because the packet transmission in cell *c* is *lagging*; if $d' > d$, then the MH will received duplicated packets because the packet transmission in cell *c* is *leading*. For the former case, in order not to lose any packets, the MH may have to listen to both the new channel (in cell *c'*) and the old channel (in cell *c*) for an unboundedly long period of time and re-order the packets received from both cells (from the base stations of both cells, more precisely). Hence, it is necessary to achieve seamless QoS (in terms of delay and delay jitter) so that the handoff procedures can be smooth and brief.

Secondly, the wireless channel error rate is significantly higher than that of the wired networks due to multi-path fading, MH mobility, and interference from other entities such as microwaves etc. In addition, wireless channel errors are usually bursty comparing with the more random packet errors in the wired networks. The frequent and back-to-back packet losses will seriously degrade the QoS of multimedia multicast received by the MHs, and the error recovery is more difficult in wireless channels than in wired networks, due to the fact that the wireless Media Access Control (MAC) mechanism is closely coupled with the packet transport protocol. In fact, the packet error rate is another location-dependent QoS parameter in wireless multimedia multicast: MHs in different cells may experience varying channel error rates, even MHs in the same cell may observe different states of the wireless channel: some observe a bad channel and some observe a good one - both at the same time. During a handoff, if the MH is entering a cell of higher packet error rate than the old cell, the MH should not notice a significant (even short term) increase of lost packets. This is another dimension of requirements in providing seamless QoS.

It is now clear that in order for MHs subscribing to a multimedia multicast session to receive relatively constant and satisfactory QoS, the handoff procedures should be made as smooth, brief, and low in packet loss rate as possible. To achieve this, special mechanisms (in addition to those for multicasting in wired networks) need to be designed to make the location-dependent QoS parameters (delay, delay jitter, and packet error rate) as seamless as possible to the MHs.

In this paper, we present protocols to deal with the problems in wireless multimedia multicast as described above. An end-to-end approach (from the source to MHs, not just from the base stations to MHs) is taken to make the packet delivery to multiple cells somewhat synchronized, i.e. for the same packet sent from the source, its copies arrive at different cells at approximately the same time. We will later show that the difference in end-to-end delay is bounded. This property guarantees that the handoff procedures can be brief and smooth. Furthermore, if we assume (for the time being) that the wireless channel is lossless, then we can show that the handoffs will be completely lossless, and any MH only has to tune in to both the old and new channels for a short and bounded amount of time. In the presence of wireless channel errors, we adopt the Forward Error Correction (FEC) technique to recover the bursty packet errors or losses. Due to the varying channel error rates in different cells, each base station may have FEC code with a different coding parameters. We then further modify the (error free) handoff procedure and packet scheduling scheme at the base stations to reduce the packet losses during handoffs, especially when a MH is moving between two cells whose packet error rates are not close to each other.

The rest of the paper is organized as follows. Section 2 discusses related work and compares our approach with theirs. Section 3 presents our solution under the assumption that either the wireless channels have zero error rates, or the MHs can tolerate any packet errors during the multimedia multicast session. We lift this simplifying assumption in Section 4, which proposes modifications to the protocols in Section 3 in order to make the handoff procedures low in packet loss rate and to make MHs receive reasonably good QoS. Section 5 shows the performance of our approach through simulation experiments. Section 6 concludes this paper.

2. RELATED WORK

The problem of packet arrival differences in wireless cellular multicast has been studied in previous literature[1].[2] An optimal channel allocation algorithm for multicasting within a cell is presented,[1] which maximizes the transmission throughput to MHs. However, it does not look at the problem in a multi-cell environment. To enforce synchronized broadcast in cellular networks, protocols are also developed[2] to minimize the maximum time difference between the local delivery inception times of a broadcast message by a group of base stations. Handoff procedure is not discussed in detail and is assumed to be a lower priority task during synchronized broadcasts. Both of them[1 2] do not provide an end-to-end solution (focusing on multicast from base stations to MHs instead), and they do not identify the location-dependent QoS parameters (delay, jitter, and loss rate) that should appear seamless to MHs.

Our earlier work[3] addresses the issue of providing seamless delay and delay jitter to achieve smooth, brief, and lossless handoffs. However, the lossless handoff guarantee is based on the assumption that the wireless channels themselves are error-free. The protocols in such an ideal case serve as a starting point for our revision to make the protocols work in the realistic error-prone wireless networks.

Wireless multimedia transmission has been intensively studied[4 5 6 7].[8] Different error control mechanisms are proposed to achieve reliable packet delivery to MHs over error-prone wireless links. However, most of the previous works focus on wireless unicast (at least implicitly), and they do not address the issue of handoff processing. In addition, as will be seen later in this paper, we also propose a differentiation between mandatory and optional packets when performing error control. This will lead to a decrease in wireless bandwidth consumption, while maintaining the acceptable QoS level.

3. BASIC MULTICAST PROTOCOLS - THE ERROR-FREE CASE

In this section we present the basic wireless multicast protocols to provide seamless QoS in terms of packet end-to-end delay and delay jitter. The assumption we make here is that the wireless channels are completely error free. This implies that packet losses can only occur during the handoffs of MHs, due to the varying end-to-end delays of the same packet to different cells (more specifically, handoffs from a 'lagging' cell to a 'leading' cell). Similarly, packet duplications will occur during handoffs from a 'leading' cell to a 'lagging' cell. This simplifying assumption helps us concentrate on the issue of providing approximately identical end-to-end packet delays and delay jitters to MHs in different cells, and thus achieving smooth, brief, and lossless (only under the error free channel assumption) handoffs. In Section 4, we will lift this assumption and examine the more realistic case in which the wireless channels are error-prone.

The network model we adopt in this paper is that of a real-time packet cellular network. The MHs are connected to the base stations via wireless links. The base stations are connected with each other and with the rest of the Internet by wired networks. Besides this, we do not make any other assumptions about how the base stations are organized (flat, clustered, or hierarchical, for example). There is a real-time packet scheduler for each of the network links (wired and wireless). To make our solution generic, we do not assume any particular packet scheduling algorithms for the schedulers. Although there have been many packet scheduling algorithms proposed in literature of this decade, many of them can in fact be categorized into one big class of *Guaranteed Rate* (GR) scheduling algorithms.[9] Well-known algorithms such as *WFQ*,[10] *Virtual Clock*,[11] *Leap Forward Virtual Clock*,[12] *PGPS*,[13] and *SCFQ*[14] belong to the GR class. The GR class formal definition is as follows.[9]

Consider a flow f with a rate of r_f. Let p_f^j and l_f^j denotes the jth packet of flow f and its length. Let $GR^i(p_f^j)$ and $A^i(p_f^j)$ denote the GR tag value and the arrival time of packet p_f^j at switch i. $GR^i(p_f^j)$ is defined as:

$$GR^i(p_f^0) = 0 \tag{1}$$

$$GR^i(p_f^j) = max\{A^i(p_f^j), GR^i(p_f^{j-1})\} + \frac{l_f^j}{r_f}, j \geq 1 \tag{2}$$

A packet scheduling algorithm at switch i belongs to the GR class if it can guarantee that packet p_f^j will be transmitted by $GR^i(p_f^j) + \beta^i$, where β^i is a constant only depending on the scheduling algorithm and the switch.

For a wireless multicast session M, a multicast tree is first established by the routing mechanism (the problem of multicast routing in wireless packet networks is out of the scope of this paper). The leaves of the multicast tree is the MHs, so the last links in the tree are all wireless (since the wireless channel is a broadcast channel, logically and physically there is only one outgoing wireless link from each base station). The multicast session set up process will go ahead and reserve bandwidth resource on each link on the multicast tree. Let r_m be the amount of bandwidth required by M. Then the resource reservation mechanism will reserve r_m bandwidth for the multicast session (again, the issue of resource reservation and rate adaptation is out of the scope of this paper).

Given a multicast session M and its multicast tree with source s, we define the multicasting area of M which is covered by a set of cells C. We assume that any MH can move between any two locations in the area without entering a cell $c \notin C$, and we regard C as *connected*. Function $b()$ maps any cell in C to its corresponding base station, and functions $parent()$, $children()$, and $outlinks()$ map a switch to its parent switch, set of children switches, and set of outgoing links in the multicast tree, respectively.

3.1. Intra-Cell Jitter Control

We first apply the current jitter control mechanism to bound the intra-cell delay jitter, i.e. the delay jitter on the path from the source s to MHs in the *same* cell. It is shown in[9] that the end-to-end (from s to any MH in the same cell c) delay d_f^j of packet p_f^j is bounded by:

$$d_f^j \leq GR^1(p_f^j) - A^1(p_f^j) + (K-1)\frac{l_f}{r_f} + \sum_{i=1}^{K} \alpha^i \tag{3}$$

for simplicity we assume that all packets in f are of the same size l_f. $\alpha^i = \beta^i + \tau^{i,i+1}$, in which $\tau^{i,i+1}$ is the propagation delay from switch i to $i+1$. K is the number of switches on the path from s to the MHs in c.

However the packet arrivals at a MH may still be bursty in spite of the packet delay bound. This is not a major concern in wired networks because the receiver can always buffer the packets for a certain amount of time before delivering them to the application layer. However, in packet cellular networks, the difference in packet arrival time may cause a MH in its handoff procedure to lose packets or to receive duplicated packets. Therefore it is necessary to perform jitter control in the packet cellular environment. By making the scheduling algorithm non work-conserving,[15] we provide a general modification to $GR^i(p_f^j)$ definition to turn a GR class algorithm into a *jitter-controlled* GR class algorithm.

$$GR^1(p_f^0) = 0 \tag{4}$$

$$GR^i(p_f^j) = E^i(p_f^j) + \frac{l_f}{r_f},\ i \geq 1 \text{ and } j \geq 1 \tag{5}$$

$$E^1(p_f^j) = max\{A^1(p_f^j), GR^1(p_f^{j-1})\},\ j \geq 1 \tag{6}$$

$$\begin{aligned} E^i(p_f^j) &= GR^{i-1}(p_f^j) + \alpha^{i-1} \\ &= A^i(p_f^j) + (GR^{i-1}(p_f^j) + \beta^{i-1}) - L^{i-1}(p_f^j),\ i \geq 2 \text{ and } j \geq 1 \end{aligned} \tag{7}$$

The change is that we introduce the *eligible time* $E^i(p_f^j)$ of packet p_f^j, which is the time when p_f^j is put into the ready queue of the scheduler at switch i. From $A^i(p_f^j)$ to $E^i(p_f^j)$, p_f^j has to be buffered. $L^{i-1}(p_f^j)$ in (7) is the actual time the last bit of p_f^j leaves switch $i-1$.

To implement a jitter-controlled GR class algorithm, the packet scheduler should be enhanced with the following operations: (1) timestamp any packet p_f^j with value $(GR^i(p_f^j) + \beta^i) - L^i(p_f^j)$, so that this value can be passed on to the next switch, and (2) extract the timestamp from any p_f^j coming from the previous switch, and hold p_f^j for the amount of time of the timestamp value before putting it into the ready queue at time $E^i(p_f^j)$ (the source scheduler simply holds p_f^j until $E^1(p_f^j)$). We define the scheduler's API for enabling the jitter control operations as follows.

$$enable\_jitter\_control(link\_id,\ additional\_holding\_time)$$

When this function is called at a switch, the scheduler for link $link\_id$ is set to the *jitter control mode*. The second parameter will be explained later in Section 3.2. It is easy to show that if all the link schedulers (from the out-going link of s to the wireless link) work in the jitter control mode, then the end-to-end delay bound of packet p_f^j is still given by (3), and its delay jitter is bounded by $\frac{l_f}{r_f} + \beta^K$.

3.2. Inter-Cell Jitter Control

We move on to solve the inter-cell jitter control problem. That is the difference in the arrival time of the same packet at MHs in different cells. Our approach is straight forward: by introducing *additional holding time* (the second parameter in the jitter control enabling function) at switches that have more than one outgoing links in the multicast tree, we make the end-to-end delay bounds of (copies of) the same packet to all cells in C identical. A degree of packet arrival synchronization is achieved among all subscribing MHs in this way. Our protocol SYNM equalizes the delay values on all outgoing links from a switch, so that the end-to-end delay bounds from s to cells in C are synchronized to the largest bound among them. SYNM requires only one bottom-up pass in the multicast tree to set all the link schedulers to the jitter control mode with proper *additional holding time* values. SYNM can also be easily incorporated into *RSVP*,[16] so that it can be performed in conjunction with resource reservation. Protocol SYNM is given is Figure 1. The superscript 0 always indicates a wireless link.

In Figure 1, the *additional holding time* is expressed as $(d_y^{max} - d_y^i)$. When the multicast session begins, the computation of the eligible time $E^{(y,i)}(p_m^j)$ for packet p_m^j should also include the *additional holding time*. Therefore, Equations (6) and (7) are modified as:

$$E^{(s,i)}(p_m^j) = max\{A^s(p_m^j), GR^{(s,i)}(p_m^{j-1})\} + (d_s^{max} - d_s^i),\ j \geq 1 \tag{8}$$

$$\begin{aligned} E^{(y,i)}(p_m^j) =& GR^{Pl(y)}(p_m^j) + \alpha^{Pl(y)} + (d_y^{max} - d_y^i) \\ =& A^y(p_m^j) + ((GR^{Pl(y)}(p_m^j) + \beta^{Pl(y)}) - L^{Pl(y)}(p_m^j) + (d_y^{max} - d_y^i),\ y \neq s \end{aligned} \tag{9}$$

Since we are now considering a multicast tree instead of a linear path, the notation for a link has to be changed: (y, i) denotes a link i from switch y, and $Pl(y)$ denotes the link from $parent(y)$ to y. Correspondingly, the definition of the GR tags is revised as:

$$GR^{(s,i)}(p_m^0) = 0 \tag{10}$$

$$GR^{(y,i)}(p_m^j) = E^{(y,i)}(p_m^j) + \frac{l_m}{r_m},\ j \geq 1 \tag{11}$$

To be executed during multicast channel set up for $\mathcal{M}$:

for each base station x with only the wireless link
  $d_x^0 \leftarrow \frac{l_m}{r_m} + \alpha_x^0$;
  $d_x^{max} \leftarrow d_x^0$;
  send d_x^{max} up to $parent(x)$;
  $enable\_jitter\_control(0, 0.0)$;

for any other switches y (may be a base station)
  if exists $c_y \in C$ such that $y = b(c_y)$
    $d_y^0 \leftarrow \frac{l_m}{r_m} + \alpha_y^0$;
  else
    $d_y^0 \leftarrow 0$;
  receive d^{max} values from switches in $children(y)$;
  for each $i \in outlinks(y)$
    $d_y^i \leftarrow \frac{l_m}{r_m} + \alpha_y^i + (d^{max}$ value received from the reverse of i);
  $d_y^{max} \leftarrow max_{i \in outlinks(y) \cup \{0\}} d_y^i$;
  if exists $c_y \in C$ such that $y = b(c_y)$
    $enable\_jitter\_control(0, (d_y^{max} - d_y^0))$;
  for each $i \in outlinks(y)$
    $enable\_jitter\_control(i, (d_y^{max} - d_y^i))$;
  if $y \neq s$
    send d_y^{max} up to $parent(y)$;

Figure 1. Protocol SYNM

Protocol SYNM guarantees that the arrive time of the same packet from source s to any subscribing MH is well synchronized. Let $A^c(p_m^j)$ be packet p_m^j's arrival time at a MH in cell c. It is easy to show that for any two adjacent cells $c, c' \in C$, it holds that

$$|A^c(p_m^j) - A^{c'}(p_m^j)| \leq \frac{l_m}{r_m} + max\{\beta^{(b(c),0)}, \beta^{(b(c'),0)}\} \tag{12}$$

This means that the arrival times differ at most by a small constant, independent of the paths from s to the cells.

3.3. Handoff Procedure - The Error-Free Case

On the MH side, we present protocol SYNM-MH to achieve smooth (no large jitters), brief, and lossless handoffs, if multicast session M has been set up by SYNM. The handoff procedure can be performed in the following steps: (1) a MH is moving from cell c to a neighboring cell c' (b and b' are the corresponding base stations respectively). It first contacts b', which in turn acknowledges MH with the identifier of the wireless channel for M in c' and the value of $\beta^{(b',0)}$ (the interactions are via a separate control channel). (2) MH can now receive packets on both the old and new channels. Our goal is to make the duration of this transient state as brief as possible.

Based on Equation (12), we can design a protocol executed on the MH side that stays tuned in both channels for at most $\frac{l_m}{r_m} + max\{\beta^{(b,0)}, \beta^{(b',0)}\}$ amount of time. Then the MH quits the old channel and the handoff is completed without losing a single packet (note that this is under the assumption that wireless channel is error-free). We describe protocol SYNM-MH in Figure 2. It assumes that the MH has the ability to discard duplicated packets.

Variable $next\_packet\_number$ identifies the next packet *MH* is expecting (at the beginning of the protocol, $next\_packet\_number$ is equal to $j+1$). By examining each case in SYNM-MH, it is easy to see that no packet is lost during the handoff. Although in one case we need to buffer and re-order packets, their actual delivery times to the application layer still fall into their expected arrival intervals.

4. REVISED MULTICAST PROTOCOLS - THE ERROR-PRONE CASE

In this section we lift the assumption of error-free wireless channels and consider the more realistic case in which the channels are error-prone. The channel errors are characterized as bursty and location-dependent: MHs may often experience

Right after a MH has received packet p_m^j from b (denoted as $b.p_m^j$)
via the old channel, begin tuning in to both the old and new channels.
begin a timer with time-out period $\frac{l_m}{r_m} + max\{\beta^{(b,0)}, \beta^{(b',0)}\}$
case the next packet received is:

$b.p_m^{j+1}$: $next\_packet\_number++$;
break;
$b'.p_m^{j-1}$: **break**;
$b'.p_m^{j}$: **break**;
$b'.p_m^{j+1}$: $next\_packet\_number++$;
break;
$b'.p_m^{j+2}$: buffer $b'.p_m^{j+2}$;
wait for $b.p_m^{j+1}$;
deliver $b'.p_m^{j+2}$ after $b.p_m^{j+1}$;
$next\_packet\_number = next\_packet\_number + 2$;
break;
none until time-out: **break**;

quit the old channel and clear the timer

Figure 2. Protocol SYNM-MH

back-to-back packet losses, and different cells may have different lengths of channel error bursts. Proper error recovery techniques should be applied to minimize the packet loss rate, which is yet another location-dependent QoS parameter in a wireless multicast session. Our goal is to let the MHs observe no significant fluctuation in packet loss rate before and after the handoff procedures. However, this is in conflict with the location-dependent nature of the wireless channel errors, because in order to effectively recover the lost packets in each cell, it is natural to adopt a specific error recovery method most suitable for the error characterization in that cell. It is possible that the packet delivery in neighboring cells are no longer in sync, causing further packet losses or duplications.

We first decide the error recovery technique to be used in the wireless multimedia multicast. The commonly available options are ARQ, FEC, or a hybrid ARQ+FEC. ARQ requires the MHs' feedbacks on which packets are lost and should be retransmitted. In wired networks, ARQ can achieve high transmission efficiency. However, in wireless networks, since the packet transmission is closely coupled with the underlying MAC protocol, it is possible that ARQ may on the contrary affect the transmission efficiency: the potentially large group of MHs content with each other (and with the base station) for sending the retransmission requests and thus introduce additional packet transmission latency. If we assume that the MHs cannot hear each other (i.e. all communications to a MH must be via the base station), then the ARQ approach may cause the problem of feedback implosion (otherwise a MH does not have to request a retransmission if it hears that the same request is sent by one of its peers). In addition, it is possible that some of the MHs are simply passive receivers and cannot send packets uplink. For the reasons above, we choose FEC as the error recovery technique for our wireless multicast session.

By introducing some redundancy, FEC allows receivers to reconstruct the original data even though some are lost during the transmission. More specifically, an (n, k) code involves the encoding of k source data packets into $n > k$ encoded packets, and the decoding of *any* k encoded packets to reconstruct the original source data. We choose the Reed-Solomon Erasure correcting code (RSE code) in,[17] because the first k encoded packets are the same as the k source data packets, which are followed by $n - k$ *parity packets*. Therefore, if the first k packets are successfully received, the parity packets are just ignored and no decoding is necessary. We define the n encoded packets derived from a group of k source data packets as a *Transmission Group* (TG).

The second problem is where to locate the RSE function. This is somewhat easier to decide because the base stations seem to be the only reasonable choice. If RSE coding is performed at the source s instead of at the base stations, extra bandwidth will have to be reserved on the entire multicast tree. Also a single RSE code will not perform well in all the cells with heterogeneous channel error characteristics. On the side of the MHs, each MH should be equipped with an adaptive RSE decoder. Being adaptive, the parameters of RSE decoder can be dynamically reset based on the RSE encoder parameters of the current base station.

In order to provide seamless packet loss rate, we bring a degree of uniformity by setting parameter k of the RSE encoders at all base stations equal. However, parameter n is set individually based on the average length of error bursts in each cell: if the average error burst causes h back-to-back packet losses, then we set $n = k + h$. It is easy to show that a $(h + k, k)$ RSE code is sufficient to recover any h back-to-back lost packets (an implicit assumption is that the interval between any two error bursts is at least $h + k$ packets apart). For convenience, in any cell $c \in C$ with the average number of lost packets in an error burst as h^c_m, we denote the RSE code used at the corresponding base station as a (h^c_m, k_m) code.

The third problem is how to modify the wireless link scheduler at the base stations so that the packet delivery to different cells are still approximately in sync. In addition, the use of RSE code introduces redundancy in packet transmission, but the wireless link bandwidth is a relatively scarce resource. Therefore, it is desirable to retain a reasonable transmission efficiency while providing packet loss recovery. We exploit the fact that multimedia data transmission can tolerate some degree of information loss, and the QoS level can be adapted to the loss as long as the minimum acceptable QoS is maintained. A typical example is the MPEG video which is encoded in multi-layer format with different levels of importance. The source s of the multimedia multicast classifies and tags all source data packets as either *mandatory packets* or *optional packets*, which carry the basic QoS data and enhanced QoS data respectively. At the base stations, only the mandatory packets are RSE encoded while the optional packets are transmitted directly.

To make the packet arrivals at different cells still approximately the same, we further modify the wireless link schedulers at the base stations. The basic idea is to transmit any TG (a TG consists of mandatory data packets and h^c_m parity packets) within the time to transmit only the data packets using the original bandwidth r_m (note that the number of data packets in a TG may be fewer than k_m because there may not be enough mandatory packets to encode. In that case, the encoder assume some 'dummy' packets containing all zeros. However the dummy packets are not transmitted). Therefore, there are two rates that are associated with the multicast session at each base station: r_m, the original rate to transmit the optional packets, and r^c_m, the rate to transmit the TGs:

$$r^c_m = r_m \frac{k_m + h^c_m}{k_m} \tag{13}$$

The definition of the GR tag values at the wireless link schedulers also has to be changed. If packet p^j_m is a mandatory packet or a parity packet, then

$$GR^{(y,0)}(p^j_m) = E^{(y,0)}(p^j_m) + \frac{l_m}{r^c_m} \tag{14}$$

If packet p^j_m is an optional packet, then

$$GR^{(y,0)}(p^j_m) = E^{(y,0)}(p^j_m) + \frac{l_m}{r_m} \tag{15}$$

The rest of the GR tag definition remains unchanged.

Finally we are ready to present the revised SYNM-MH protocol. In Figure 3, the duration of the handoff procedure is bounded by $k_m \frac{l_m}{r_m} + max\{\beta^{(b,0)}, \beta^{(b',0)}\}$, the time to transmit a TG plus the constant $max\{\beta^{(b,0)}, \beta^{(b',0)}\}$. When the multicast source tags an optional packet, it also puts the sequence number of the next optional packet in its $next\_Op$ field, which will be used in the revised SYNM-MH to decide whether the handoff can be completed earlier when there are no mandatory packets being transmitted (in that case the handoff procedure is completed within $\frac{l_m}{r_m} + max\{\beta^{(b,0)}, \beta^{(b',0)}\}$). Before an MH executes the revised SYNM-MH, it has already got the RSE code parameters $(h^{c'}_m, k_m)$ and its decoder is reset. Therefore the MH will ignore the parity packets from cell c and only buffer those from cell c'. However, to improve reliability, the MH buffers the mandatory packets received from both cells. After the timer expires, the MH performs RSE decoding and delivers the source data packets to the application layer.

In summary, our approach provides the subscribing MHs with seamless packet delay, delay jitter, and packet loss rate when they move freely from one cell to another. The QoS of the multimedia multicast appears relatively smooth and undisturbed to the MHs in the entire session, even during their handoff procedures.

5. EXPERIMENTAL RESULTS

In this section we present results of our simulations to show the performance of the protocols (SYNM and the revised SYNM-MH). We use the same experiment setup as in our earlier work.[3] However, we introduce bursty channel errors in each cell, and RSE coding/decoding functions in the base stations/MHs. Figure 4 shows the area of multicasting which is covered by cells numbered 1 to 12. s is the source. Although the base stations are directly connected to each other in our experiment,

When the MH receives strong enough beacons from both b and b', begin tuning in to both the old and new channels.
begin a timer with time-out period $k_m \frac{l_m}{r_m} + max\{\beta^{(b,0)}, \beta^{(b',0)}\}$
while (**not** (time_out **or** $DONE$)) {
  case the next packet received p_m^j is:
    mandatory from b: **if** $not\_buffered(p_m^j)$ **then** $buffer(p_m^j)$;
      break;
    optional from b: **if** $next\_Op \leq j$ **then** { deliver p_m^j; $next\_Op \leftarrow p_m^j.next\_Op$; }
      break;
    parity from b: **break**;
    mandatory from b': **if** $not\_buffered(p_m^j)$ **then** $buffer(p_m^j)$;
      break;
    optional from b': **if** $next\_Op \leq j$ **then** { deliver p_m^j; $DONE \leftarrow$**TRUE**; }
      break;
    parity from b': $buffer(p_m^j)$;
      break;
}
quit the old channel and clear the timer;
if there are buffered mandatory and parity packets **then**
  perform RSE decoding and deliver the source data packets;

Figure 3. Revised Protocol SYNM-MH

our approach is generic to work in other cellular network architectures (base station cluster for example). Each wired link has a bandwidth of $10Mbps$, and each wireless link has a bandwidth of $2Mbps$. The wireless link errors are simulated as the following: the average number of back-to-back lost packets during an error burst in a cell is uniformly distributed in interval $[3, 6]$. The RSE parameter h_m^c for each cell c has the same value. We set the other RSE parameter k_m to 12. The link scheduling algorithm is *Virtual Clock*[11] with the jitter control mode. The uniform packet size is $1KB$.

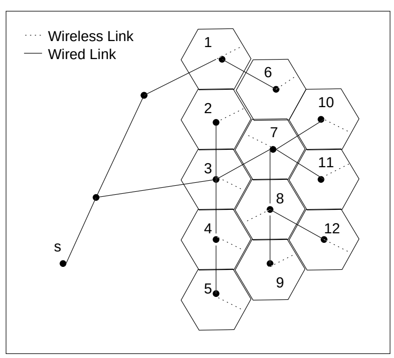


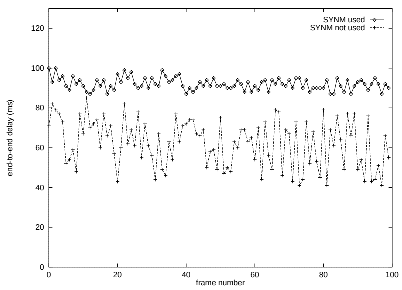
Figure 4. Experiment Setup: A Multicast Session

A short MPEG video trace with a frame rate of $15fps$ is used as the multimedia multicast source data. The GOP pattern of the video trace is IBBPBBPBBPBB. For the basic QoS, we define the mandatory packets as those carrying the I frame and the second P frame in each GOP, and the optional packets as those carrying the other frames. The original bandwidth r_m is $1.2Mbps$, r_m^c for each cell c is computed using (13).

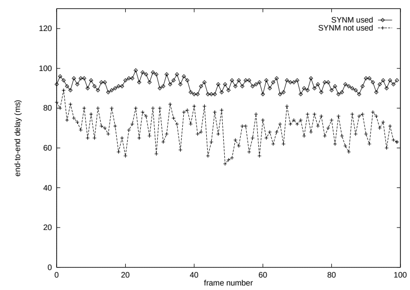
We have performed two sets of experiments: one assuming error-free wireless channels and the other with error-prone wireless channels. The purpose of the first set of experiments is to evaluate the provision of seamless packet delay and delay jitter using protocol SYNM. The second set examines the QoS received by an MH as it moves freely, and thus shows the overall performance of our approach, i.e. seamless delay, delay jitter, and packet loss rate.

5.1. Results in The Error-Free Case

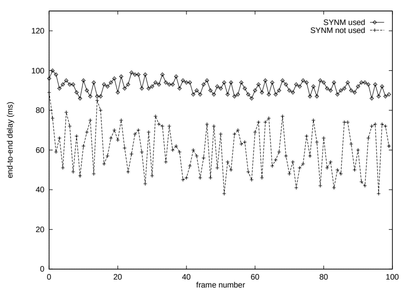
Figure 5 shows the end-to-end delays of 100 MPEG frames (i.e. the delays of the last packet in each frame), observed by MHs in four of the cells (#4,5,8, and 9), respectively. To show the effect of SYNM, we also plot the delays when SYNM is not executed (i.e. the link schedulers do not work in the jitter control mode). It is obvious that the packet (frame) arrivals are well synchronized if the multicast session has been set up by SYNM. The MHs in different cells receive the same frame at approximately the same time (as reflected in the Figure, the curves of the four cells are very much ’alike’). On the contrary, without SYNM the packet (frame) arrival times vary substantially.



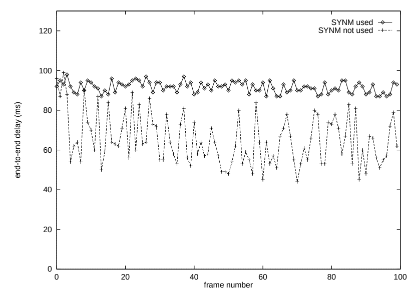
(a) Cell #4



(b) Cell #5



(c) Cell #8



(d) Cell #9

Figure 5. End-to-End Delays of 100 MPEG Frames in 4 Cells

5.2. Results in The Error-Prone Case

We simulate a moving MH whose handoffs are controlled by the revised SYNM-MH. The MH stays in one cell for some time and moves to a neighboring cell, then it stays there for some time and moves on... We divide the time into intervals numbered 1,2,3... Each odd number interval represents the time the MH stays in a cell, and each even number interval represents the 10-second period right before and after a handoff. Figure 6 shows the percentage of mandatory and optional packets successfully received by the MH during these time intervals. The mandatory packets have lower loss rate than the optional packets. More importantly, the mandatory packet loss rate does not change significantly over these time intervals (when the MH is in different cells, especially before and after handoffs). Therefore the mandatory packet loss rate appears relatively seamless to the MH. For the optional packets, since there is no error control, their loss rate still varies in different intervals or cells. However, this is a trade-off between the QoS and wireless bandwidth requirement.

6. CONCLUSION

In this paper we present our approach to provide MHs with seamless QoS in wireless multimedia multicast sessions. The QoS parameters considered are packet delay, delay jitter, and packet loss rate. By introducing intra-cell and inter-cell jitter control mechanisms, we are able to make the arrivals of the same packet at different cells approximately in sync, therefore the MH handoff procedure is brief, smooth, and if error-free wireless channels are assumed, lossless. For the realistic case of

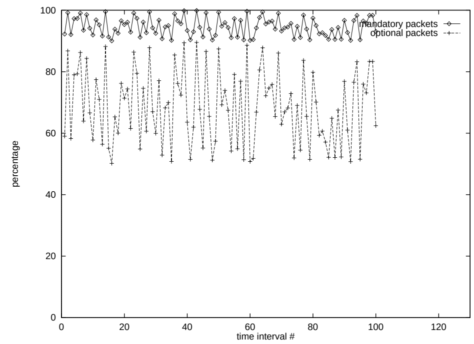


Figure 6. Percentage of Packets Successfully Received by the MH during the Experiment

error-prone wireless channels, we adopt the FEC technique to recover from bursty packet losses without feedbacks from the MHs. Furthermore, to save the scarce wireless bandwidth at the same time, we differentiate between mandatory and optional packets. Therefore, we can provide low and seamless mandatory packet loss rate to the MHs. The simulation results indicate that our protocols have the expected good performance.

REFERENCES

1. K. Brown and S. Singh, "The Problem of Multicast in Mobile Networks," *Proceedings of IEEE 5th International Conference on Computer Communications and Networks* , Oct. 1996.
2. D. Awduche and A. Ganz, "Synchronized Broadcast in Cellular Networks," *Proceedings of 2nd Telecommunications R&D Conference in Massachusetts* , Mar. 1996.
3. D. Xu and B. Li, "Synchronized End-to-End Multicast in Real-Time Packet Cellular Networks," *Proceedings of 1998 IEEE Military Communications Conference* , Oct. 1998.
4. E. Ayanoglu, P. Pancha, A. Reibman, and S. Talwar, "Forward Error Control for MPEG-2 Video Transport in a Wireless ATM LAN," *Mobile Networks and Applications* **1**, Jan. 1996.
5. J. Gomez-Castellanos and M. Naghshineh, "Link Error Impact on MPEG Video over Wireless Broadband Networks," *Proceedings of IWQoS '97* , May 1997.
6. H. Liu and M. E. Zarki, "Transmission of Video Telephony Images over Wireless Channels," *Wireless Networks* **2**, Aug. 1996.
7. J. Moura, R. Jasinschi, H. Shiojiri, and J. Lin, "Video Over Wireless," *IEEE Personal Communications* **6**, Feb. 1996.
8. G. Rajugopal and R. Hafez, "Adaptive Rate Controlled, Robust Video Communication over Packet Wireless Networks," *Mobile Networks and Applications* **3**, June 1998.
9. P. Goyal and H. Vin, "Generalized Guaranteed Rate Scheduling Algorithms: A Framework," *IEEE/ACM Trans. Networking* **5**, Aug. 1997.
10. A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queuing Algorithm," *Proceedings of SIGCOMM '89* , 1989.
11. L. Zhang, "VirtualClock: A New Traffic Control Algorithm for Packet Switching Networks," *Proceedings of ACM SIGCOMM'90* , Aug. 1990.
12. S. Suri, G. Varghese, and G. Chandranmenon, "Leap Forward Virtual Clock: A New Fair Queuing Scheme with Guaranteed Delays and Throughput Fairness," *Proceedings of INFOCOM '97* , Apr. 1997.
13. A. Parekh and G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks," *IEEE/ACM Trans. Networking* **1**, June 1993.
14. S. Golestani, "A Self-Clocked Fair Queuing Scheme for High Speed Applications," *Proceedings of INFOCOM '94* , Apr. 1994.
15. H. Zhang, "Providing End-to-End Performance Guarantees Using Non-Work-Conserving Disciplines," *Computer Communications: Special Issue on System Support for Multimedia Computing* **18**, Oct. 1995.
16. L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A New Resource ReSerVation Protocol," *IEEE Network Magazine* , Sept. 1993.
17. J. Nonnenmacher, E. Biersack, and D. Towsley, "Parity-Based Loss Recovery for Reliable Multicast Transmission," *Proceedings of SIGCOMM '97* , Sept. 1997.