Dynamic Packet Scheduling for cdma2000 1xEV-DO Broadcast and Multicast Services

Kyungtae Kang
School of Computer Science and Engineering
Seoul National University
Seoul, 151-742, Korea
Email: ktkang@cslab.snu.ac.kr

Jinsung Cho
Dept. of Computer Engineering
Kyung Hee University
Youngin, 449-701, Korea
Email: chojs@khu.ac.kr

Heonshik Shin
School of Computer Science and Engineering
Seoul National University
Seoul, 151-742, Korea
Email: shinhs@cslab.snu.ac.kr

Abstract— **cdma2000 1xEV-DO, one of the prominent third-generation mobile communication systems, provides Broadcast and Multicast Services (BCMCS) to meet increasing demand for multimedia data services. Currently, 1xEV-DO schedules broadcast streams using a slot-based static algorithm, which fails to support dynamic environments where broadcast content is added or removed on-line. We propose a dynamic packet-scheduling algorithm that works with a retransmission scheme for the BCMCS scheduler. Integrated with Earliest Deadline First (EDF) real-time scheduling, the proposed algorithm not only adapts to dynamic contexts efficiently but also satisfies the real-time requirements of broadcast streams. Furthermore, by exploiting the fine granular scalable (FGS) characteristics of the MPEG-4 Part 2 standard, our scheme can avoid abrupt degradation of playback quality by protecting the most important data. Extensive simulations have quantitatively validated the efficiency of our approach.**

I. Introduction

As mobile communications systems evolve into the third generation, the focus has shifted away from voice services and onto data services [1]. To follow this latest trend, many countries have adopted cdma2000 1xEV-DO (Evolution-Data only), one of the third-generation systems. 1xEV-DO offers a multimedia capability, support for packet-mode services and more capabilities than second-generation systems. 1xEV-DO devices also provide 'always-on' packet data connections, helping to make wireless access simpler, faster and more useful. A commercial cdma2000 1xEV-DO network has been deployed nationwide in several countries including Korea.

Recently, work has begun, in both the Third Generation Partnership Project (3GPP) and the 3GPP2, on enhancing 3G networks to support multimedia broadcast and multicast services. The 3GPP2 group recently baselined the specification for a cdma2000 high-rate broadcast packet-data air interface (BCMCS[1]) [2][3]. The goal is to design a system that can deliver multimedia broadcast and multicast traffic with minimum resource usage by both the radio access and core networks. In addition, users expect minimum latency when joining or leaving the network, and multimedia streams must be delivered continuously as mobile users move around. A hierarchical design with localized multicasting and local servers is necessary to provide a scalable system. An efficient air-link must also be designed to ensure that the total power required to provide broadcast or multicast services is minimized.

In this paper, we propose a dynamic scheduling algorithm combined with a retransmission scheme for the cdma2000 1xEV-DO BCMCS scheduler. The current static scheduling algorithm incorporated in the BCMCS specification cannot adapt to an environment in which content streams change dynamically. When a new item of content is added, empty slots must be found to service this content. Our proposed algorithm deals with such dynamic contexts by exploiting the inherent attributes of the Earliest Deadline First (EDF) [4] algorithm. Multimedia streams have timing constraints: a video frame is useful only if it arrives at the mobile node before its playback time. This suggests that the EDF real-time scheduling algorithm would be appropriate. Furthermore, the BCMCS currently employs Reed-Solomon coding to correct errors at the cost of air resources. However if the channel condition is good, many slots will be overbooked due to parity information. Additionally, the capacity for error recovery in the Reed-Solomon scheme declines suddenly when a channel condition goes bad. Thus, by adding a retransmission scheme to the EDF scheduling algorithm, we can save many slots when the condition of a channel is good, and reduce the degradation of playback quality by protecting the important packet which is essential for decoding video streams in FGS coding with retransmission when the channel condition is bad.

The remainder of this paper is organized as follows: In Section II, we introduce the cdma2000 1xEV-DO BCMCS, and Section III describes our proposed scheduling algorithm. The simulation results are discussed in Section IV, and we conclude and suggest further studies in Section V.

II. cdma2000 1xEV-DO BCMCS

A. The service architecture

Fig. 1 is a high-level view of the proposed BCMCS architecture in the cdma2000 system [5][6]. BCMCS content originates from the content provider and goes through the BCMCS content server. The BCMCS content provider may be located within the cdma2000 serving network, in a home network, or anywhere in an IP network. If the BCMCS content provider is located in an IP network, the business association, security association, and other related service information

[1]In this paper, the term 'BCMCS' stands for cdma2000 1xEV-DO BCMCS.

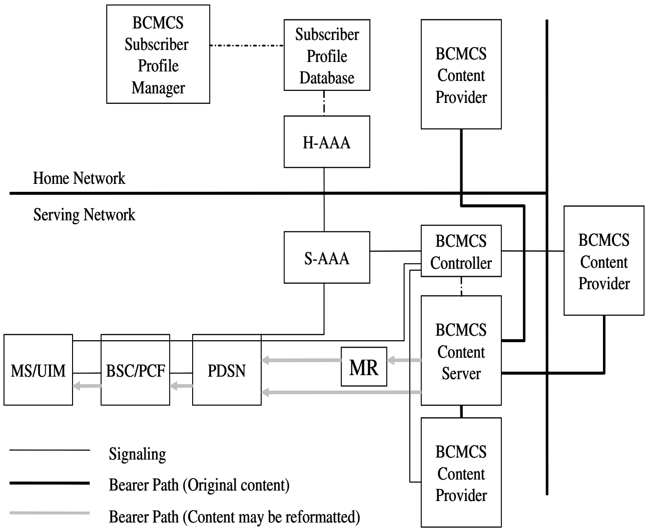


Fig. 1. The cdma2000 BCMCS network architecture.

should be provided by the cdma2000 carrier network and the content provider. The BCMCS content server (CS) is connected to the cdma2000 access network through a Packet Data Serving Node (PDSN) that handles the BCMCS content stream, which it makes available within an IP multicast stream. The BCMCS content server in the serving network is not necessarily the creator or source of the content; it is merely the last application-level entity that manipulates or reformats the content prior to reaching the PDSN.

The server may store and forward content from a single content provider, or merge content from multiple providers. The PDSN is responsible for communication with the Base Station Controller/Packet Control Function (BSC/PCF), including the addition and removal of multicast IP flows. It uses IP multicast protocols to manage bearers supporting multicast IP flows between itself and the nearest router connecting back to the BCMCS content server. The way that it treats multicast IP flows is specified by the BCMCS controller, which is a core network function that is responsible for managing and providing BCMCS session information to the PDSN, the mobile node, and the CS. It also performs authorization using the BCMCS user profile received from the Home Authentication, Authorization, and Accounting (H-AAA) server. Additionally, the BCMCS controller performs security functions such as generating and distributing security keys to the mobile nodes [7].

A mobile node can determine whether a particular multicast IP flow is available and obtain the BCMCS radio configuration information from a base station via the cdma2000 1xEV-DO overhead messages on the forward common signaling channel. If the mobile node cannot obtain the information from the overhead messages, and if the BSC indicates on the overhead messages that BCMCS registration is permitted, the mobile node may request the desired IP flow information by means of a BCMCS registration request.

B. *Packet scheduling in BCMCS*

In a unicast environment, 1xEV-DO employs a time-shared forward link that serves one user at a time in a time-multiplexed manner [8][9][10]. The fundamental timing unit for forward-link transmissions is a 1.67 ms slot that contains the pilot and MAC channels, and a data portion that may contain traffic or a control channel. When a mobile user is being served, each mobile node calculates its signal to interference plus noise ratio (SINR) at every time slot and determines the highest data rate from a list of possible rates that is supportable with the calculated SINR. The mapping between SINR and supportable data rate is given in Table 9.2.1.3.3.2-1 in the cdma2000 standard [8]. The measured data rate is reported to the home base station every 1.67 ms (i.e. every slot) by the mobile node. Using the information reported from mobile nodes, the base station schedules the slot allocation. The scheduled data on the traffic channel can be transmitted at 38.4, 76.8, 153.6, 307.2, 614.4, 921.6, 1228.8, 1843.2 or 2457.5 kb/s. The higher data rates are achieved through a combination of high-order modulation (QPSK, 8PSK, and 16-QAM), forward error-correction coding (the code rate is 1/5 or 1/3), and spreading. Transmission of one encoded packet can occupy from 1 to 16 time-slots. This adaptive rate control uses the full power of the base station to achieve the highest possible data rate for each user in the time-varying channel. Fig. 2 shows forward-link transmission by third-generation 1xEV-DO where a 153.6 kbps packet stream is requested. Four slots are required for the transmission of one encoded packet; these are in pairs, separated by three null slots. This allows enough time for the mobile node to decode the partially received packet and to indicate to the base station whether decoding was successful, using the ACK channel on the reverse link [11].

In a broadcast or multicast environment, however, a fixed transmission rate must be used because it is impossible to reflect the link state of all mobile nodes. BCMCS content streams are delivered to one or more mobile nodes in 1.67 ms time-slots that are allocated statically in advance. In receiving a cdma2000 1xEV-DO broadcast content stream, a mobile node gets the content from the content server in pre-determined slots, but the service parameters are transmitted as an overhead message. When a mobile node requests a BCMCS

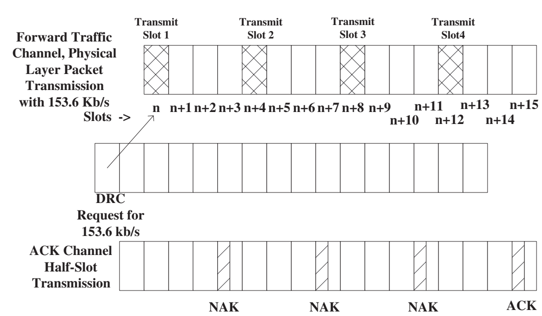


Fig. 2. Timing of forward traffic channel transmissions.

content stream, packets are transmitted from the PDSN to the PCF, which uses a timestamp to make sure that the packet is received simultaneously at all base stations. The mobile node maintains its soft state by registering periodically. Eventually the stream is terminated if no mobile node registers to it.

Unlike the unicast cdma2000 1xEV-DO standard, the broadcast MAC protocol does not use an ARQ-based error control scheme, because the same content will be delivered to many users simultaneously. Instead of handling retransmit requests individually, BCMCS uses a Reed-Solomon forward error-correcting code [3]. Reed-Solomon codes are block-based error-correcting codes with a wide range of applications in digital communications and storage. A Reed-Solomon encoder takes a block of digital data and adds extra 'parity' bits. The encoded data is larger and therefore uses more physical slots, which means using more air resource. On receipt, a Reed-Solomon decoder processes each block and attempts to correct errors and to recover the original data.

TABLE I

PARAMETERS USED IN THIS PAPER.

Parameter	Description
b_i	Bit rate of each BCMCS IP flow (kbps)
p_i	Period of τ_i in units of slots
R_i	Buffer requirement of each BCMCS IP flow
D_i	Relative deadline slot for the periodic multicast video frame τ_i
D_{ak}	Deadline slot for retransmitted packets $\bar{\tau}_k$
F_i	Amount of transmitted data during one period for each video stream
e_i	Number of slots required for the forward video stream during one period
$\bar{e}_r$	Number of slots required to retransmit lost packets of $\bar{\tau}_r$
M_i	Number of bytes that can be forwarded in one slot (data payload)
U_P	Utilization of periodic video frames
τ_i	Periodic broadcast video frame
$\bar{\tau}_k$	Aperiodic retransmitted video frame (packets)
A_k	Arrival time of $\bar{\tau}_k$
N	Total number of mobile nodes in a cell
N_{τ_i}	Number of mobile nodes registered for video flow τ_i

III. THE PROPOSED SCHEME

A. The proposed algorithm

From the service scenario given in Section II, we can infer that current static scheduling algorithms cannot adapt to an environment in which broadcast flows start and stop in part of the system, affecting the resources available. For example, when a new content stream is added, empty slots must be found to schedule this content. Because the transmission rates of content streams differ from each other, the slot periods of these video streams will also differ. Thus, some slots might overlap with more than two content streams, which can cause problems in servicing the streams. Additionally, we can infer that, if all slots are allocated for predetermined video streams, additional video streams or newly created asynchronous video streams (newsflashes or other irregular content) that are not predetermined would never be serviced. This represents inefficient slot utilization, which is exacerbated by the parity data required for Reed-Solomon coding. Finally, it is impossible to change the quality of a video stream within the current static scheduling scheme. All this motivates us to deploy a dynamic scheduling algorithm.

We propose to apply the EDF real-time scheduling algorithm with retransmission to maximize the slot utilization in a BCMCS service. One of the most significant characteristics of broadcast multimedia applications is probably their real-time service requirement: a packet generated at the content source should be received before a specified deadline, and the EDF algorithm endeavors to achieve this. When a mobile node fails to receive packets, it sends retransmission request messages through the unicast channel. Using our proposed scheme in this context has the following advantages: First, we can adapt it to an environment in which the content to be served changes dynamically. Second, it becomes unnecessary to add additional parity information to the original content stream, and so we can use the slots saved flexibly, for instance to retransmit packets or unicast packets. (The performance of our approach will be analyzed from simulation results in Section IV.) Finally, admission can easily be controlled using a simple utilization bound test [12][13][14] when a retransmission request occurs or a new broadcast flow starts. We will discuss admission control in the next section.

When many content servers provide multimedia streams, each BCMCS flow is split into packets which wait to be serviced. If a mobile node cannot receive a packet, it is reported as lost or damaged to the base station, via the reverse unicast channel. Because successful transmission of a new broadcast packet will give satisfaction to more users than a retransmitted packet, the scheduler gives higher priority to broadcast packets than to retransmitted packets. Thus, broadcast packets will be transmitted earlier, giving them more chances of retransmission in case of loss. Also, while accuracy in the base layer is essential for decoding video streams in FGS coding, the enhancement layers are more tolerant of channel errors. Errors in the enhancement layers corrupt video quality only in the current frame and a few subsequent frames, and the deterioration is not as noticeable as that caused by errors in the base layer. Thus, when packets are retransmitted, base-layer packets are scheduled ahead of enhancement-layer packets.

To allow retransmission of lost packets to be requested even when a mobile node has moved to a new service area, every packet must have a sequence number, as in a unicast service.

B. Admission control

Admission control and resource allocation are tightly related to the packet scheduling algorithm. In our scheme, the admission of newly created video frames and retransmitted packets is allowed if sufficient resources for the adequate delivery of data to a mobile node are available. In the EDF scheduling algorithm, resource availability is checked by a utilization bound test. In our scheme, admission is also controlled by a utilization bound test, as follows: Suppose there are n BCMCS flows currently being serviced in the cell and the bit-rate of each BCMCS flow is b_i kbps. Then the scheduler must periodically transmit at b_i kbits per second, since the BCMCS flow corresponds to a periodic task in a real-time system. Thus, each flow has the period p_i. If the buffer requirement (R_i) to guarantee continuous playback is $K * p_i$, then the relative deadline slot is $K * p_i/1.67$. Using the transmission scheme in cdma2000 1xEV-DO, e_i is determined by $\lfloor b_i/M_i \rfloor$. Thus the admission of a new multicast video stream (τ_j) is decided using the following equation [4]:

$$U_P = \sum_{i=0}^{n-1} \frac{e_i}{Min(p_i, D_i)} + \frac{e_j}{p_j} \leq 1 \ . \qquad (1)$$

There may also be packets for retransmission, which are assumed to represent a set of aperiodic tasks. Suppose the aperiodic task set $Q(t) = \{\bar{\tau}_0, \bar{\tau}_1, \bar{\tau}_2, ...\bar{\tau}_k, ...\}$ and the periodic task set $P = \{\tau_0, \tau_1, \tau_2, ...\tau_{i-1}\}$ are to be scheduled by the EDF algorithm. For $\bar{\tau}_k \in Q(t)$, $\bar{e}_k(t)$ is defined as the remaining execution time of $\bar{\tau}_k$ at time t, and we define the set S to be the union of $Q(t)$ and P. For all $\bar{\tau}_k \in Q(t)$, $S(t, \infty)$ is schedulable if and only if $\forall k, U_{S(t,\infty)}(\bar{\tau}_k) \leq 1$, where

$$U_{S(t,\infty)}(\bar{\tau}_k) = \frac{\sum_{i=0}^{n-1} e_i(t) + \sum_{\bar{\tau}_r \in Q((t,\infty),HP(D_{ak}))} \bar{e}_r(t)}{D_{ak} - t}. \qquad (2)$$

The term $HP(D_{ak})$ represents 'higher priority': if $\bar{\tau}_k$ is included in the base-layer packets, then $\bar{\tau}_j$ are tasks which are included in the base-layer packets and have an earlier deadline than D_{ak}. If $\bar{\tau}_k$ is included in the enhancement-layer packets, then all packets for retransmission corresponding to the base layer or the enhancement layer, with deadlines earlier than D_{ak}, are included in $HP(D_{ak})$. Thus, base-layer packets always have higher priority than enhancement-layer packets. If two packets correspond to the same layer, the packet with the earlier deadline has the higher priority. Using Equation (5), we might now expect to be able to determine the feasibility of a new retransmission request. However this test is actually impossible because an infinite number of tasks ($Q((t, \infty), HP(D_{ak}))$) would have to be considered. So we use the following recursive method to calculate an upper bound on the utilization [12][13][14][15].

A task set $S(t, \infty)$, $Q(t) = \{\bar{\tau}_1, \bar{\tau}_2, \bar{\tau}_3, ..., \bar{\tau}_{k-1}, \bar{\tau}_k, ...\}$ is sorted by priority. Thus, $\bar{\tau}_{k-1}$ has a higher priority than $\bar{\tau}_k$, and an aperiodic task $\bar{\tau}_k \in Q(t)$ is schedulable when $\bar{U}(\bar{\tau}_k) \leq 1$. The virtual finish time f_k is defined as the sum of the execution times (number of slots required) of all periodic tasks and of the aperiodic tasks that have a higher priority than $\bar{\tau}_k$ during the time period $[A_k, D_{ak}]$. Thus

$$U(\bar{\tau}_k) \leq \bar{U}(\bar{\tau}_k) \ , \qquad (3)$$

$$\bar{U}(\bar{\tau}_k) = \frac{max\{f_{k-1} - A_k, 0\} + \bar{e}_k + U_P * (D_{ak} - D_{a(k-1)})}{D_{ak} - A_k}, \qquad (4)$$

$$f_{k-1} = A_{k-1} + U(\bar{\tau}_{k-1}) * d_{k-1} \ . \qquad (5)$$

By using these equations, while maintaining the value of U_k and F_k, the schedule can be analyzed in constant time. Hence, we can ignore the admission control overhead.

IV. Performance evaluation

A. Simulation model

Our simulator consists of a video frame generation module, an admission control module, a scheduler, and an error generation module. This is shown in Fig. 3, in which the dotted rectangles on the left correspond to the base station and those on the right correspond to the mobile node. Each module is triggered by an event, which is generated at each slot cycle by an event generation module. In the base station, the video frame generation module generates BCMCS video streams. We conducted experiments using the Foreman testbench video sequences streamed at 30 frames per second with a total of ten thousand frames. Each video stream is handled with our reference MPEG-4 FGS codec [16][17][18], which is derived from the framework of the European ACTS Project Mobile Multimedia Systems (MoMuSys) [19] and modified for our experiment. It consists of a 100 kbps base-layer bit-rate and a 120 kbps enhancement-layer bit-rate with 120 levels (1 kbps step). All video streams are sent through channel coding and packetized before being transferred via a cdma2000 1xEV-DO physical slot. In the experiment, we used QPSK modulation with a 1228.8 kbps data-rate forward channel. Other simulation parameters such as deadline, period and execution time are then calculated as described in Section III-B. At each slot cycle, the scheduler selects a packet from the waiting queue using the EDF algorithm and transmits the packet to the mobile node.

At the mobile node, whether the video packet is received successfully is determined by the error generation module, which uses the simple threshold model suggested by Zorzi [20][21]. When an error occurs, the error generation module may subsequently request the retransmission of packets, by means of a retransmit event generated by the event generation module. These packets are then inserted into the waiting queue by an event handler in the base station, and wait to be serviced by the scheduler. Eventually, these synthetic packet errors are inserted into the real transport stream. In our simulation environment, the air channel is simulated by a two-state Markov model, which can simulate the error sequences generated by data transmission channels: these errors occur in clusters or

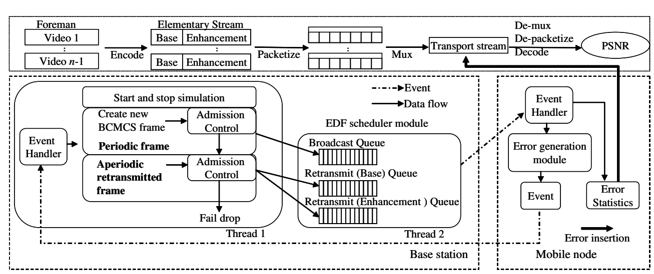


Fig. 3. Simulation architecture.

bursts with relatively long error-free intervals between them. In the following equations, 1-α (or β) is the probability that the i-th packet block is successfully transmitted, given that the (i-1)-th block was successful (or unsuccessful), and fading in the air channel is assumed to have a Rayleigh distribution. The steady-state error rate ε is then obtained as follows:

$$\varepsilon = \frac{\alpha}{\alpha + \beta} \ . \tag{6}$$

When the Rayleigh fading margin is F, the average physical layer packet error rate can be calculated as

$$\varepsilon = 1 - e^{\frac{1}{F}} \ . \tag{7}$$

We can now derive the values of α and β from Equation (6) and from Equations (9) and (10) below. F is the *fading margin* and the parameter $f_d N_{BL} T$ (the Doppler frequency normalized to the data rate) is taken as 0.02, where N_{BL} is the packet-block length. $Burst$ is the average length of packet errors and is given by $1/\beta$, where

$$\beta = \frac{Q(\theta, \rho\theta) - Q(\rho\theta, \theta)}{e^{-\frac{1}{F}} - 1}, \text{ where } \theta = \sqrt{\frac{2/F}{1-\rho^2}} \ . \tag{8}$$

The term ρ, where $\rho = J_0(2\pi f_d T)$, is the correlation coefficient of two samples of the complex Gaussian fading process, and $Q(\cdot,\cdot)$ is the Marcum-Q function [22]. Thus, the relationship between the physical layer packet error rate and the Markov parameter can be represented as

$$\beta = \frac{1-\varepsilon}{\varepsilon}[Q(\theta, \rho\theta) - Q(\rho\theta, \theta)] \ , \tag{9}$$

where

$$\theta = \sqrt{\frac{-2\log(1-\varepsilon)}{1 - J_0^2(2\pi f_d N_{BL} T)}} \ . \tag{10}$$

We compare our retransmission model with the current Reed-Solomon-based BCMCS error correction scheme. A Reed-Solomon code is specified as a tuple (N, K, R) and we use the value (16,12,4) in our experiments [2][3].

B. Simulation Results

We derived the average playback quality using the peak signal-to-noise ratio (PSNR) value and the slot utilization, both for our proposed scheme and for the current BCMCS scheme. We varied the number of mobile nodes and the average physical layer packet error rates ($PER_{Physical}$) of the mobile nodes, using values of 1%, 3% and 5%. We also adjusted the maximum retry count to find an optimum. For convenience, we denote U_{BCMCS} as the slot utilization when frames are encoded by Reed-Solomon, and $U_{Proposed}$ when the proposed retransmission scheme is used. As Figs. 4 and 5 show, when the utilization (U_{BCMCS}) is around unity, no slots are left after serving the BCMCS flow. However, when we use our retransmission scheme instead of Reed-Solomon coding, about 25% of the total slots in use can be reserved for retransmitted packets, space previously used for parity information. Our proposed scheme shows a lower average packet error rate when the physical layer packet error rate is high. Additionally, when there are fewer than 40 nodes, the slot utilization of our scheme always drops below 1, which means that slack slots are left. We are now able to reduce the packet error rate, and the resulting slack periods allow slots to be used more efficiently. As can be seen from Fig. 4, the performance of the Reed-Solomon-based scheme does not depend on the number of mobile nodes, but that does affect the retransmission scheme. The average lost packet count suddenly increases when the number of mobile nodes reaches the slot utilization at which $U_{Proposed} = 1$, but still shows a lower packet loss rate than the Reed-Solomon.

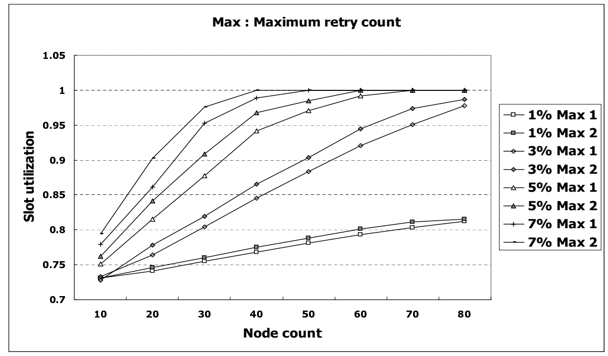
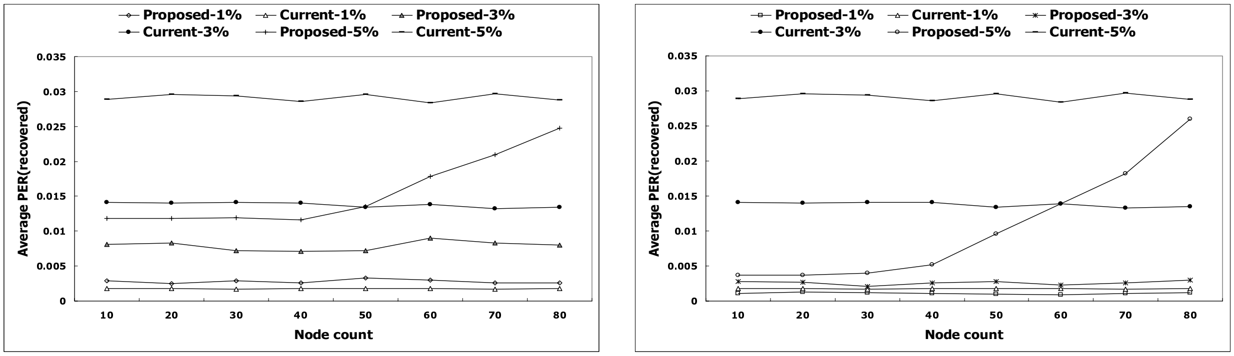


Fig. 5. Comparison of slot utilization in different cases.

The graph in Fig. 4 shows the average packet loss rate for all mobile nodes after recovering lost packets using error recovery schemes, as the average physical-layer packet error rate and the number of mobile nodes increases. When the average physical-layer packet error rate is increased to 3% and 5%, our service model shows better results in terms of the packet loss rate. In these cases, the average packet error rate of all mobile nodes increase greatly. However our error recovery scheme can at least save the important packets, and in particular the base-layer packets, even when the channel condition is extremely bad. Fig. 4(a) shows the case when a mobile node can make a maximum of one retransmit request and Fig. 4(b) shows the effect of allowing up to two retransmit requests. In both cases, the number of retransmit requests allowed for base-layer packets is unlimited because these packets are required to avoid abrupt quality degradation.

When the maximum retry count is set to 2, the average rate of recovery of lost packets is improved, compared to a retry count of 1. However, when the channel condition deteriorates, adjusting the maximum retry count achieves little reduction in the packet error rate. In this case many retransmitted packets are timed out while waiting, because the queue of packets for retransmission to mobile nodes is now much longer. This is also true in the case when the maximum retry count is set to 3. Thus, we set the maximum retry count to 2 in our experiments.

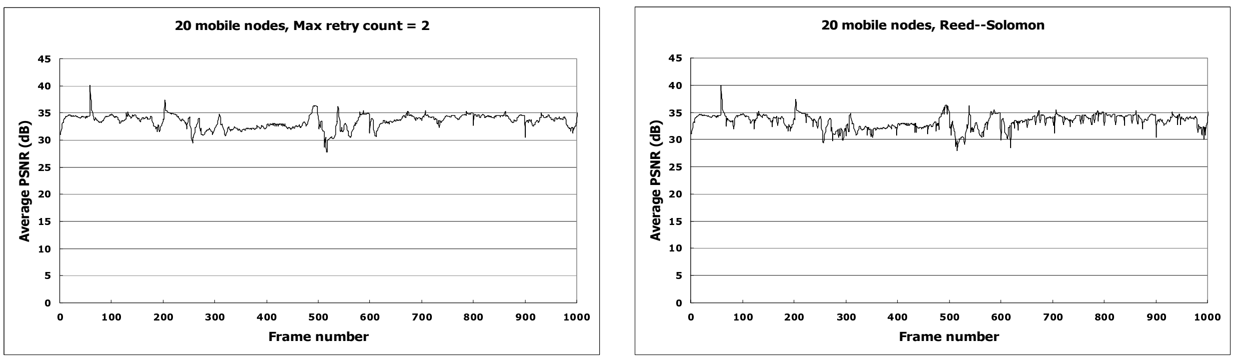
In Fig. 5, U_{BCMCS} equals 1 and the graph shows the utilization (U) in three simulation cases, with average physical layer packet error rates of 0.01, 0.03 and 0.05 respectively. When there are fewer than 40 mobile nodes, unused slots are



(a) Max retransmit count = 1

(b) Max retransmit count = 2

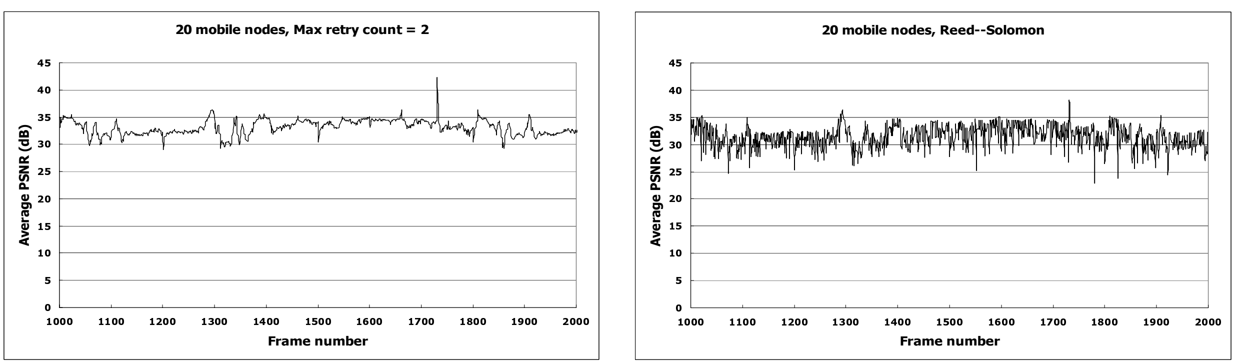
Fig. 4. Comparison of average packet loss rate with the current BCMCS.



(a) Proposed retransmission scheme

(b) Current Reed-Solomon scheme

Fig. 6. Comparison of average PSNR with the current BCMCS when the average $PER_{Physical}$ is 1%.



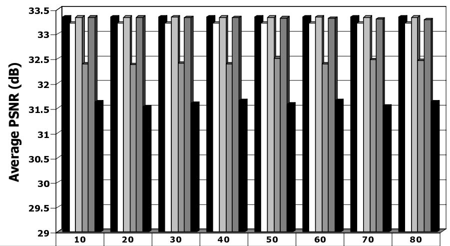
(a) Proposed retransmission scheme

(b) Current Reed-Solomon scheme

Fig. 7. Comparison of average PSNR with the current BCMCS when the average $PER_{Physical}$ is 5%.

still available in the system after all packets have been dealt with: either by successful delivery, or by attempted redelivery, up to the specified retry count of 2. This means that our scheme can reduce the number of slots required for delivering broadcast packets, as well as the number of lost packets. Generally, the number of mobile nodes in a cell is less than 50, which means that many more slack slots are available. They can be flexibly redeployed to service unicast packets or to retransmit failed packets more frequently, improving the average playback quality of FGS video streams.

Fig. 6 and Fig. 7 compare the average playback quality of the proposed scheme and the current BCMCS scheme when the average physical layer packet error rate is 1% and 5% respectively. In both cases, there are a total of twenty mobile nodes and each mobile node can request the retransmission of lost packets once or twice. We sampled from Frame 1000 to Frame 2000 in the former case and from Frame 1 to Frame 1000 in the latter case. Using the proposed retransmission scheme, the delivery of base-layer packets is guaranteed. Thus the average PSNR does not fall below 25 dB, which means that



Node count	10	20	30	40	50	60	70	80
Proposed-1%	33.34065	33.34018	33.34071	33.34087	33.34114	33.34157	33.34108	33.34047
Current-1%	33.21386	33.21546	33.22157	33.21796	33.21624	33.21372	33.22265	33.21654
Proposed-3%	33.3368	33.33742	33.33811	33.33672	33.33652	33.3381	33.33722	33.33657
Current-3%	32.39257	32.37927	32.40612	32.39975	32.51321	32.38969	32.47554	32.45976
Proposed-5%	33.33521	33.33479	33.33381	33.33233	33.32185	33.314	33.30406	33.29007
Current-5%	31.62733	31.52557	31.59731	31.65469	31.57675	31.6559	31.54097	31.62202

Node count

Fig. 8. Comparison of average PSNR with the current BCMCS with regard to the number of mobile nodes.

the proposed error-recovery scheme prevents abrupt quality degradation. In Reed-Solomon scheme, however, the average PSNR falls below 25 dB when an error occurs in the base layer. The fluctuations seen in Fig. 6(b) come from this failure to deliver base-layer packets, which is exacerbated by the poor error recovery of Reed-Solomon as the physical layer packet error rate increases, as shown in Fig. 7(b).

The bars of the graph in Fig. 8 indicate the average PSNR for the full ten thousand frames, when the maximum retry count is set to 2, as the number of mobile nodes changes. When the average physical layer packet error rate is 1%, the two schemes show similar PSNR values, even when the number of mobile nodes increases to 80. This is because the slot utilization is below unity for all node counts. In the Proposed-5% cases, the playback quality degrades smoothly as the slot utilization approaches unity. However, the extent of the degradation is small if we compare it with the cases Current-3% and Current-5%, when the average quality degradation reaches about 0.91 dB and 1.72 dB respectively. From these results, we can see that our error recovery scheme is much more efficient than the current Reed-Solomon scheme.

V. Conclusion

We have proposed a dynamic scheduling algorithm based on EDF to handle the situation where the broadcast channel environment change dynamically. Our algorithm adapts to changing link states and circumstances such as handoff, and meets the real-time requirement of broadcast streams. In addition, it reduces the playback quality degradation of broadcast streams by deploying a retransmission scheme in unicast services instead of the Reed-Solomon coding used in the current BCMCS. Our proposed retransmission scheme prevents abrupt playback quality degradation by protecting the base layer, assuming the use of FGS coding as specified in MPEG-4 part 2. Extensive simulation has shown the efficiency of our scheme compared with the current Reed-Solomon coding. When the channel condition is good, we can save slots and increase playback quality significantly by repeatedly retransmitting lost packets. The saved slots can be used flexibly to service other broadcast or unicast packets. Also, when a mobile node enters into an area with a bad channel condition, the proposed scheme shows better error-recovery results than the current scheme, with respect to packet error rate and average playback quality. In future work, we will consider using a scalable media stream to improve the average video quality, in cases where the spare slack is insufficient to service all video streams. This may well be based on a dynamic scheduling algorithm.

References

[1] D. N. Knisely, S. Kumar, S. Laha, and S. Nanda, "Evolution of wireless data services: Is-95 to cdma2000," *IEEE Communications Magazine*, vol. 36, no. 10, pp. 140–149, October 1998.

[2] P. Agashe, R. Rezaiifar, P. Bender, and QUALCOMM, "Cdma2000 high rate broadcast packet data air interface design," *IEEE Communications Magazine*, vol. 42, no. 2, pp. 83–89, February 2004.

[3] 3GPP2 C.S0054 v1.0, "cdma2000 high rate broadcast-multicast packet data air interface specification," February 2004.

[4] F. Cottet, J. Delacroix, C. Kaiser, and Z. Mammeri, *Scheduling in Real-Time Systems*, Wiley, October 2002.

[5] 3GPP2 X.P0019 v0.1.3, "Broadcast-multicast services (bcmcs) framework draft document," August 2003.

[6] J. Wang, R. Sinnarajaj, T. Chen, Y. Wei, E. Tiedemann, and QUALCOMM, "Broadcast and multicast services in cdma2000," *IEEE Communications Magazine*, vol. 42, no. 2, pp. 76–82, February 2004.

[7] 3GPP2 S.R0083-0 v1.0, "Broadcast-multicast services security framework," October 2003.

[8] 3GPP2 C.S0024 v3.0, "cdma2000 high rate packet data air interface specification," December 2001.

[9] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi, "Cdma/hdr: A bandwidth-efficient high-speed wireless data service for nomadic users," *IEEE Communications Magazine*, vol. 38, no. 7, pp. 70–77, July 2000.

[10] A. Jalali, R. Padovani, and R. Pankaj, "Data throughput of cdma-hdr a high efficiency-high data rate personal communication wireless system," *Proc. 51st Vehicular Technology Conference*, vol. 3, pp. 1854–1858, May 2000.

[11] E. Esteves, "The high data rate evolution of the cdma2000 cellular system," *Multiaccess, Mobility and Teletraffic for Wireless Communications*, vol. 5, pp. 61–72, December 2000.

[12] T. F. Abdelzaher, V. Sharma, and C. Lu, "A utilization bound for aperiodic tasks and priority driven scheduling," *IEEE Transactions on Computers*, vol. 53, no. 3, pp. 334–350, March 2004.

[13] B. K. Choi, D. Xuan, R. Bettati, W. Zhao, and C. Li, "Utilization-based admission control for scalable real-time communication," *Real-Time Systems*, vol. 24, no. 2, pp. 171–202, March 2003.

[14] M. Spuri and G. Buttazzo, "Scheduling aperiodic tasks in dynamic priority systems," *Real-Time Systems*, vol. 10, no. 2, pp. 179–210, March 1996.

[15] J. Park, M. Ryu, and S. Hong, "Utilization demand analysis: An integrated admission control approach for a mixed set of periodic and aperiodic tasks," *Proc. KISS Spring Conferences*, pp. 391–393, April 2004.

[16] ISO/IEC 14496-2, "Coding of audio-visual objects-part2:visual," May 2004.

[17] W. Li, "Overview of fine granularity scalability in mpeg-4 video standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 301–317, March 2001.

[18] H. M. Radha, M. van der Schaar, and C. Yingwei, "The mpeg-4 fine-grained scalable video coding method for multimedia streaming over ip," *IEEE Transactions on Multimedia*, vol. 3, no. 1, pp. 53–68, March 2001.

[19] A. Pearmain, A. Carvalho, A. Hamosfakidis, and J. Cosmas, "The momusys mpeg-4 mobile multimedia terminal," *Proc. 3rd ACTS Mobile Summit Conference*, pp. 224–229, June 1998.

[20] M. Zorzi, R. R. Rao, and L. B. Milstein, "Error statistics in data transmission over fading channels," *IEEE Transactions on Communications*, vol. 46, no. 11, pp. 1468–1477, November 1998.

[21] M. Zorzi and R. R. Rao, "On the statistics of block errors in bursty channels," *IEEE Transactions on Communications*, vol. 45, pp. 660–667, June 1997.

[22] J. G. Proakis, *Digital Communications*, McGraw-Hill, 2nd edition, 1989.