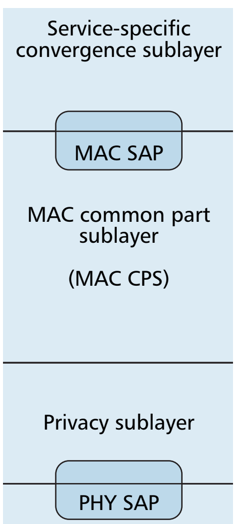
FEEDBACK-BASED REAL-TIME STREAMING OVER WIMAX

MAINAK CHATTERJEE AND SHAMIK SENGUPTA, UNIVERSITY OF CENTRAL FLORIDA
SAMRAT GANGULY, NEC LABORATORIES AMERICA



The authors use commonly used techniques such as forward error correction (FEC) and automatic repeat request (ARQ) to support streaming services over WiMax without violating what has already been standardized.

ABSTRACT

The IEEE 802.16 standard (commonly known as WiMax) has emerged as one of the strongest contenders for broadband wireless access technology. At the same time, with the steady growth of real-time services such as voice over IP and video on demand, supporting delay sensitive streaming data overWiMax is becoming crucial. In this article we use commonly used techniques such as forward error correction (FEC) and automatic repeat request (ARQ) to support streaming services over WiMax without violating what has already been standardized. In particular, we look at the flexible features provided by media access control (MAC) layer of WiMax and exploit them for providing better streaming performance. We use the channel state information to dynamically construct the MAC packet data units. The sizes of these units are thusly determined such that the packet dropping probability is minimized without compromising the goodput. The simulation results presented show the performance enhancements of the proposed ARQ-enabled adaptive algorithm for streaming data.

INTRODUCTION

Today, broadband Internet services are generally restricted to a T1, DSL, or cable-modem-based connection. With the rapid growth of wireless technologies, the task of providing broadband *last mile* connectivity is still a challenge. The last mile is generally referred to as a connection from a service provider's network to the end user — either a residential home or a business facility. Such wireless solutions avoid the prohibitive cost of wiring homes and businesses and allow a relatively faster deployment process. Among the broadband access technologies that are being sought, WiMax (worldwide interoperability of microwave access) is perhaps the strongest contender that is being supported and developed by a consortium of companies [1]. WiMax is a wireless metropolitan access network (MAN) technology that is based on the standards defined in the IEEE 802.16 specification. This standard-based approach is not only simplifying but also unifying development and deployment of WiMax.

WiMax has the capability to deliver high-speed services up to a range of 30 miles, thus posing a strong competition to the existing last mile broadband access technologies such as cable and DSL. WiMax uses multiple channels for a single transmission and provides bandwidth of up to 350 Mb/s [2]. The use of orthogonal frequency division multiplexing (OFDM) increases the bandwidth and data capacity by spacing channels very close to each other and still avoids interference because of orthogonal channels. A typical WiMax base station provides enough bandwidth to cater to the demands of more than fifty businesses with T1 (1.544 Mb/s) level services and hundreds of homes with high-speed Internet access. WiMax's low cost of deployment coupled with existing demands from underserved areas creates major business opportunities. For residential broadband access, WiMax has higher potential, as compared to 802.11 based Wi-Fi technology, due to both range and bandwidth. Even though Wi-Fi based mesh networks are being proposed to extend coverage performance degradation with multiple hops is still a concern, however. Competing with 3G cellular downlink data access, WiMax is geared to support roaming users under the new extension 802.16e that support user mobility.

Nowadays, users do not use broadband Internet just for connectivity and web surfing. Services like VoIP and video on demand are becoming popular in the last mile. The widespread use and bandwidth demands of these multimedia applications are far exceeding the capacity of current 3G and wireless LAN technologies. Moreover, most access technologies do not have the option to differentiate specific application demands or user needs. WiMax is envisioned as a solution to the outdoor broadband wireless access that is capable of delivering high-speed streaming data. WiMax offers some flexible features that can potentially be exploited for delivering real-time services. In particular, though the medium access control (MAC) layer of WiMax has been standardized, there are certain features that can be tuned and made application and/or channel specific. For example, the

1536-1284/07/$20.00 © 2007 IEEE

MAC layer does not restrict itself to fixed-size frames, but allows variable-sized frames to be constructed and transmitted.

In this article we use forward error correction (FEC) and automatic repeat request (ARQ) schemes that have been successfully used in other wireless environments to support real-time streaming services over WiMax. We do not consider any particular application but assume that the packets generated by the application must be delivered in a timely fashion. Since the MAC layer of WiMax has a rich set of features, we exploit its flexibility in constructing the MAC layer frames without violating what has already been standardized. We use variable-sized MAC packet data units (MPDUs) that are constructed based on the wireless channel conditions. We design a feedback mechanism at the MAC layer of the receiver which lets the transmitter know about the channel conditions. Depending on the feedbacks, the MAC layer at the transmitting side modifies its MPDU payload size and/or FEC code. The dynamic manner in which the MPDUs are changed to match the channel conditions helps in increasing the packet restore probability, thereby reducing MPDU dropping probability and increasing goodput. The reduction in the number of retransmission of dropped or corrupted packets in turn lowers the delay, which is very crucial for streaming applications. Simulation experiments are conducted to verify our proposed scheme. We assume a four-state Markovian channel model and study the performance with and without feedback. We show that the proposed feedback technique and variable-length MPDUs are effective and significantly increase the goodput and lower the MPDU dropping probability.

The rest of the article is organized as follows. We provide a brief overview on the adaptive techniques that have been proposed to support data/streaming services over wireless channels. We discuss the MAC layer features and consider aggregation and fragmentation of MAC service data units. Feedback design schemes to increase packet restore probability are presented. We present the simulation model and results. Conclusions are drawn in the last section.

Related Work

Streaming of real-time applications over any wireless network (e.g., 3G cellular networks, 802.11-based wireless local area networks, and 802.16-based WiMax) poses many challenges, including limited bandwidth, coping with bandwidth fluctuations, and lost or corrupted data. Due to the growing popularity of streaming services over wireless networks, it has been well researched and many solutions have been proposed that combine audio/video processing techniques with the mechanism that are usually dealt with in the data link and physical layer. These approaches can be broadly classified into two categories — ARQ and FEC. ARQ schemes provide high reliability when the channel is good or moderate. However, for error-prone channels, the throughput drops due to increased frequency of retransmissions. In order to counter this effect, hybrid ARQ schemes are used that combine FEC with ARQ schemes.

The design and performance of a hybrid ARQ with concatenated FEC for wireless ATM was proposed in [3] that was capable of dynamically supporting survivable ATM-based communications. A scheme based on a finite-state Markov channel model was proposed in [4] that used punctured convolution codes for adaptive encoding and decoding. A cross-layer protection strategy was proposed in [5] for enhancing the robustness of scalable video transmission. A combination of MAC retransmission strategy, application-layer FEC, bandwidth-adaptive compression, and adaptive packetization strategies was used. In [6], the transmission and playout policies for streaming media were investigated. The joint problem of transmitter power and playout rate was formulated using a dynamic programming approach and heuristic were proposed that demonstrated significant performance gains over benchmark systems. In [7], the problem of transmission of multimedia data over a wireless LAN was addressed. The wireless LAN was modeled as a packet erasure channel and a solution was presented that efficiently combines ARQ and FEC coding for sending compressed video and audio over the wireless link. Hybrid ARQ error control scheme based on the concatenation of a Reed–Solomon code and a rate-compatible punctured convolution code for low-bit-rate video transmission over wireless channels was proposed in [8]. By using redundancy bits, the error correcting capability of retransmitted packets was enhanced as they were combined with the previous transmission. The problem of unicast and multicast of video steaming over wireless LANs has been addressed in [9]. The wireless channel is modeled as a packet-erasure channel at the IP level. For the unicast scenario, novel hybrid ARQ method has been described that efficiently combines FEC and ARQ. As for the multicast part, the problem has been formalized as an optimization of a cost function of the user space and was solved by combining progressive source coding with FEC.

The research presented in this article differs from the existing work in the sense that instead of proposing a new adaptive link-layer technique, we use the commonly used FEC and ARQ schemes and apply that to the MAC layer of WiMax. These techniques are thusly used so that they do not violate the MAC layer specifications that have already been defined for WiMax.

The research presented in this article differs from the existing work in the sense that instead of proposing a new adaptive link-layer technique, we use the commonly used FEC and ARQ schemes and apply that to the MAC layer of WiMax.

The MAC of WiMax

The MAC layer of 802.16a layer comprises three sublayers which interact with each other through the service access points (SAPs), as shown in Fig. 1. The service-specific convergence sublayer provides the transformation or mapping of external network data, with the help of the SAP. The MAC common part sublayer receives this information in the form of MAC service data units (MSDUs), which are packed into the payload fields to form MAC protocol data units (MPDUs). Privacy sublayer provides authentication, secure key exchange and encryption on the MPDUs and passes them over to the physical layer. Of the three sublayers, the common part sublayer is the core functional layer which provides bandwidth, and establishes and maintains connection. Moreover, as the 802.16a MAC provides a connection-oriented service to the subscriber stations, the common part sublayer also provides a connection identifier to identify which connection the MPDU is servicing.

Let us discuss the common part sublayer and its rich set of features. This sublayer controls the on-air timing based on consecutive frames that are divided into time slots. The size of these frames and the size of the individual slots within these frames can be varied on a frame-by-frame basis. This allows effective allocation of on-air resources which can be applied to the MPDUs to be transmitted. Depending on the feedback received from the receiver and on-air physical layer slots, the size of the MPDU can be optimized. In the later part of this article, we exploit this feature of the common part sublayer that modifies the size of the MPDUs to adapt to the varying channel conditions.

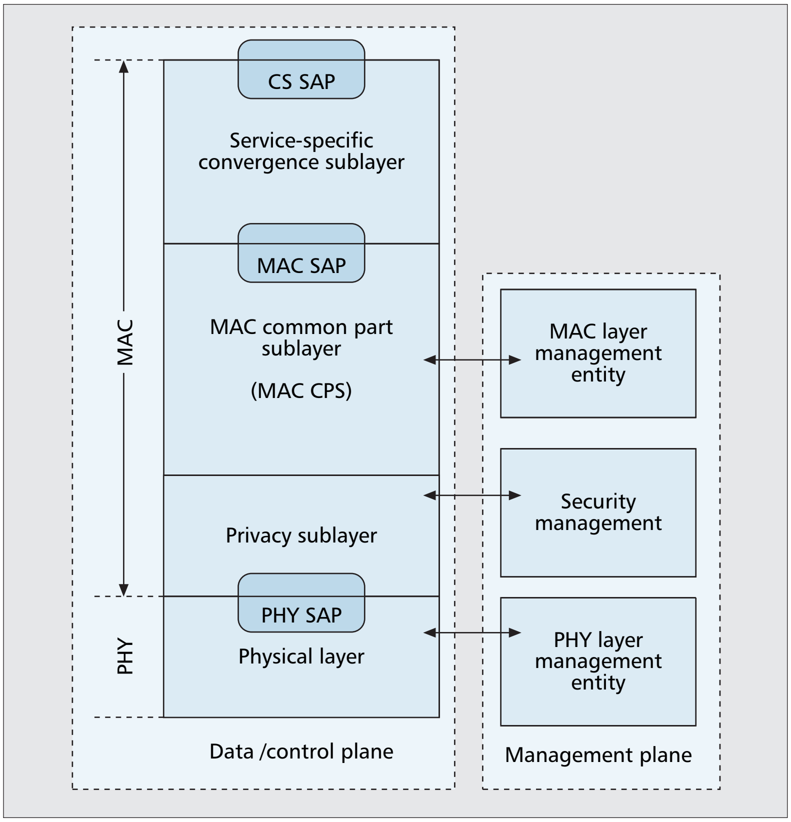


Figure 1. *802.16a MAC layer with SAPs.*

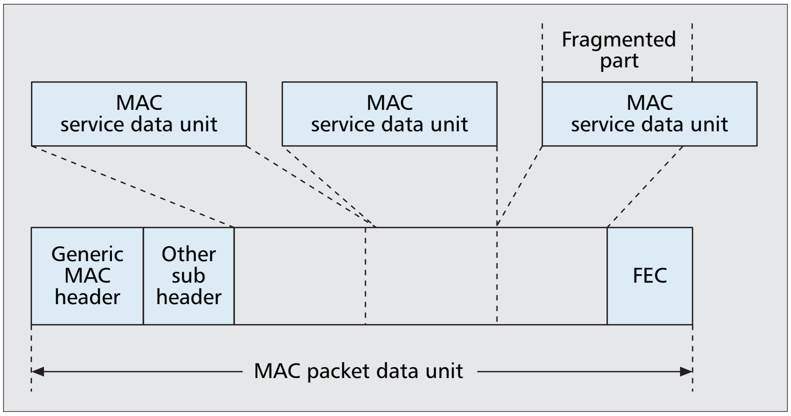


Figure 2. *Multiple MSDUs form a MPDU.*

Packing

The common part sublayer is capable of packing more than one complete or partial MSDUs into one MPDU. In Fig. 2, we show how the payload of the MPDU can accommodate more than two complete MSDUs, but not three. Therefore, a part of the third MSDU is packed with the previous two MSDUs to fill up the remaining payload field preventing wastage of resources. The payload size is determined by on-air timing slots and feedback received from subscriber station.

Fragmentation

The common part sublayer can also fragment a MSDU into multiple MPDUs. In Fig. 3, we show how a portion of a single MSDU occupies the entire payload field of a MPDU. Here, the payload field of the MAC packet data unit to be transmitted is too small to accommodate a complete MSDU. In that case, we fragment a single MSDU and pack the fragmented part into the payload field of the MPDU.

We make extensive use of these two features in our algorithm for MPDU construction and transmission discussed in the next section. Since the MPDUs can be of variable size, the natural question that arises is how to determine its optimal size.

Optimal MPDU Size

The optimal size of the MPDU must be matched to the channel conditions so as to obtain a desired level of performance. Since packets often get lost or corrupted during transmission in error-prone wireless channels, ARQ mechanism is usually used to identify and possibly recover the missing frames. In our case, ARQ will play a crucial role in estimating the channel condition and the fate of the MPDUs that have been transmitted. As a result, the round-trip time (RTT) becomes crucial in determining the size of the MPDUs. We define RTT as the time difference between the time the last bit of a MPDU is transmitted and the time the Acknowledgment for that MPDU is received. Moreover, we assume zero time interval between the transmissions of two consecutive MPDUs, that is, the last bit of a MPDU and the first bit of the next MPDU are transmitted back to back.

Let us now show how the RTT affects the size of the MPDUs. If we assume that is the time taken to transmit the MPDU and is the RTT, then the number of MPDUs already transmitted before the acknowledgment of the first MPDU is received is given by $\lceil T/t \rceil$. It can be noted that this depends on the size of the MPDU and thus there is a trade-off between the goodput (information bits/total bits transmitted) and the delay. If a MPDU is large, the transmission time is large, but the overhead due to headers is less, which helps in maintaining a high goodput. If a MPDU is dropped or corrupted due to bad channel condition, the ARQ mechanism will invoke the retransmission of the large MPDU, which will increase the delay in the transmission. Moreover, by the time the MAC common part sublayer receives the feedback (i.e., learns about the channel condition), the transmission of the next MPDU would have already started. If the bad channel condition persists, the probability of the subsequent frame being dropped or corrupted is very high. Thus, there will be more retransmissions of large MPDUs under bad channel condition, resulting in severe degradation of goodput compromising the QoS. To clarify, let us consider a simple example.

An Illustrative Example

Let the size of an MPDU be in bytes (bytes of payload, bytes of FEC code, and bytes of header). If we consider that the base station transmits at Mb/s or more, the transmission time of the MPDU will be msec. The RTT will be on the order of msec when considering a cell of approximately miles radius. Thus, the transmission of the second MPDU will be in progress while the acknowledgment of the first MPDU arrives. As a result, the MAC common-part sublayer can only respond and take appropriate measures before the third MPDU is transmitted. Now suppose that the channel condition is bad and remains so for 2 to 3 msec. Both the first and second MPDUs will be dropped and bytes of retransmission will be incurred.

On the other hand, if the MPDU size is small, say, in bytes (bytes of payload, bytes of code, and bytes of header), the transmission time of each MPDU is msec. With the RTT still being in the order of msec, the Acknowledgment of the first MPDU arrives while the third MPDU is being transmitted. Now appropriate measures can be taken with regard to the fourth MPDU, and even if the loss of all previous three MPDU occurs, still bytes are needed to retransmit, which is almost half the previous example. But the main disadvantage of having small MPDUs is the low goodput due to low payload to overhead ratio. Thus, we observe that both large and small MPDUs have their advantages and disadvantages. We propose to combine the advantages of both by dynamically changing the MPDU size in response to the channel conditions.

Feedback-based Adaptive MAC

Let us first construct the various types of feedback that will be used by the MAC for construction of the MPDUs. Since most encoders classify the raw data into two or more categories that have different priorities, we use a generic two-level classification and do not restrict ourselves to any particular coding methodology. We classify the packets as *important* and *not so important* and propose to treat them differently. For example, MPEG-2 encoders classify video frames into I, B, and P frames. Similarly, data (or frames) obtained from any encoder can always be classified into two levels depending on their importance. The data with different priorities eventually make the MPDUs which are transmitted. These MPDUs may or may not reach the receiver. Even if an MPDU is received, it might not be correct. The exact state in which an MPDU is received depends of the channel condition it experienced. Hence, it is important to identify the possible states that a received MPDU might be in. We propose six types of feedback — each of which depends on the state of the received MPDU and its importance level. We show the different possibilities of feedback in Table 1.

One question might arise as to how the MAC common-part sublayer distinguishes the impor-

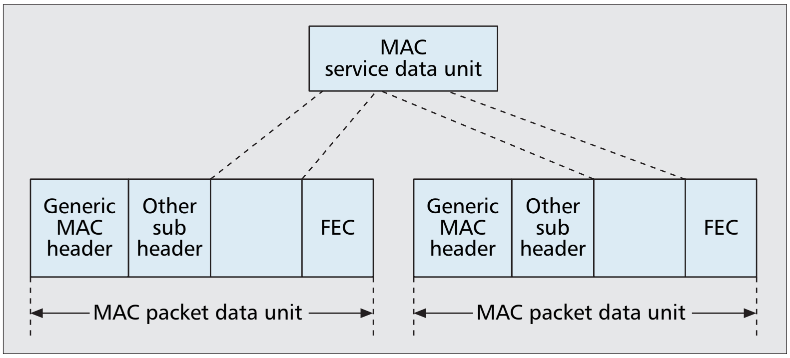


Figure 3. *Single MSDU forms multiple MPDUs.*

Feedback type	Feedback classification
1	MPDU received correctly
2	MPDU received with errors, but correctable
3	MPDU received with errors, and uncorrectable
4	MPDU dropped, timeout in receiver MAC occurred
5	Receiver MAC buffer full, last stored frame is important
6	Receiver MAC buffer full, last stored frame is not so important

Table 1. *Feedback possibilities.*

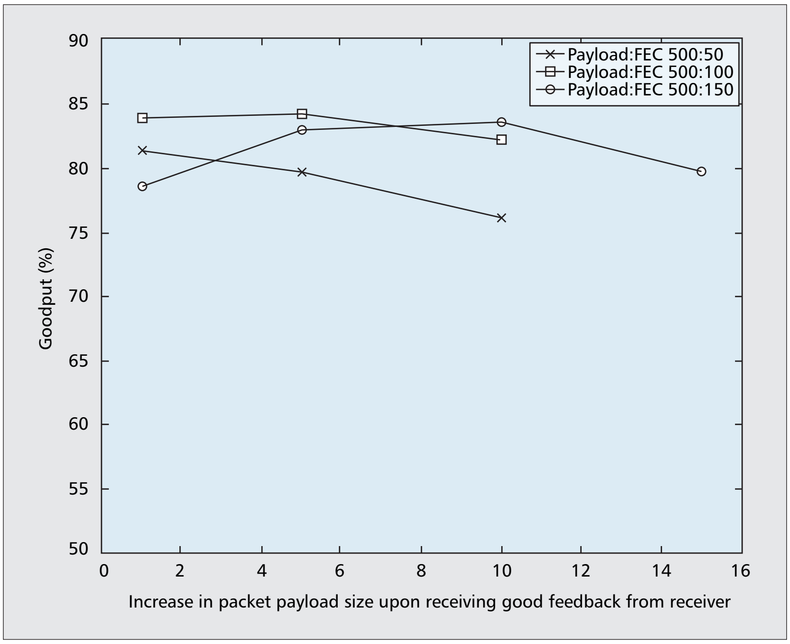


Figure 4. *Goodput with payload increment.*

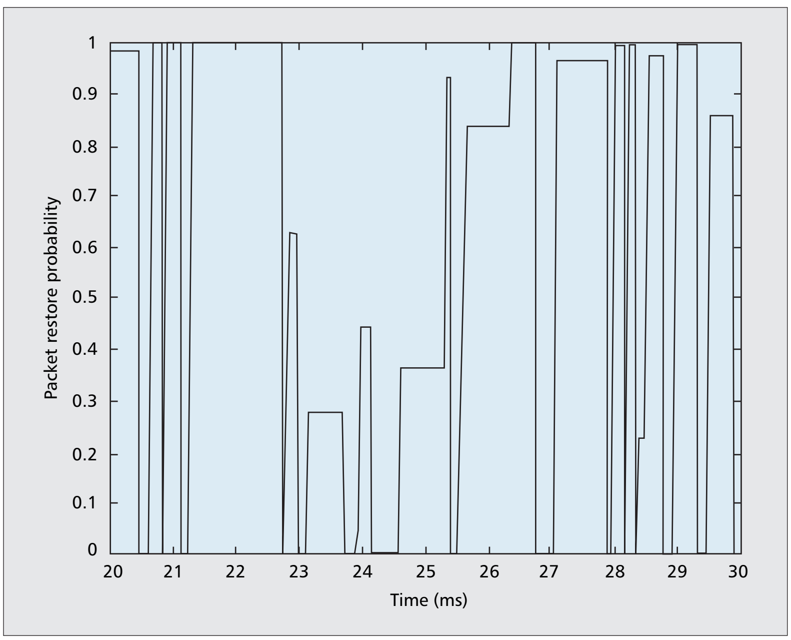
tant from the not so important frames being transmitted. This can be done by using the application headers where the type of the frame can be set. We suggest that a single bit in the "other subheader" field of MPDUs can be used to differentiate the two priority levels of MPDUs. To represent the six types of feedback, only three bits would be sufficient. As our mechanism depends on the retrieval of the feedbacks, we assume that these bits suffer no loss due to robust coding.

Packet Restore Probability

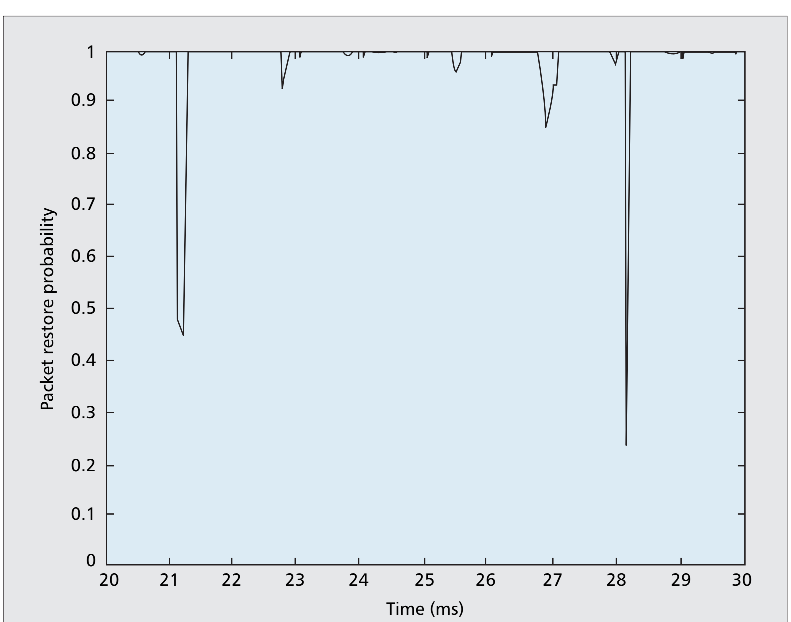
If a receiver gets a corrupted a packet or a video frame, it is in no position to correct the errors. However, if some redundant bits in the form of FEC are applied before transmission, then there is a probability that the receiver would be able to detect and possibly correct the errors. The correction capability of these codes will depend on the kind and the length of the code used. Since this article does not attempt to propose new coding techniques, we will just use the simplest of codes — *block codes*. In block codes, M redundancy bits are added to the N information bearing bits. (Note that these extra bits are generated using a generator matrix operating on the bits.) If we consider such an MPDU the resulting bit loss probability is given by [10],

$$b = \sum_{i=M+1}^{M+N} \binom{M+N}{i} b_p^i (1-b_p)^{M+N-i} \frac{i}{M+N} \qquad (1)$$

where b_p is the bit loss probability before decoding and b is the decoded bit error probability. The restore probability of such an MPDU with payload size N bits and M code bits is given by, $p = (1 - b)^{(M+N)}$. We apply the following three schemes to manipulate this packet restore probability.



■ **Figure 5.** *20–30 ms window for the nonadaptive scheme.*



■ **Figure 6.** *20–30 ms window for adaptive scheme.*

Decreasing Payload Keeping Code Fixed — Let b be the resulting bit loss probability after decoding of an MPDU with payload size N and code size M. Now, if we decrease the payload size to N' ($N' < N$) keeping the code size fixed, then the resulting bit loss probability after decoding is given by

$$b' = \sum_{i=M+1}^{M+N'} \binom{M+N'}{i} b_p^i (1-b_p)^{M+N'-i} \frac{i}{M+N'} \qquad (2)$$

It can be argued that with the decrease in payload with the code fixed, $b' < b$. Now, let p' be the new packet restore probability. Then p' is given by

$$p' = (1 - b')^{(M+N')} \qquad (3)$$

As b' and b are close to 0, $(1 - b)$ and $(1 - b')$ are close to 1. Without loss of generality, it can be said that, for $N' < N$, $p' > p$, that is, with a decrease in payload, packet restore probability increases.

Increasing Code Size Keeping Payload Fixed — Similarly, it can be argued that if the code is increased keeping the payload fixed, the resulting bit loss probability decreases and packet restore probability of MPDUs increases.

Increasing Both Payload and Code — If the MPDU to be transmitted is "important" and its payload cannot be decreased because of playout frame rate restriction at the receiver, the best scheme would be to increase both the payload and the code. As we know, increasing payload only will increase the resulting bit error probability, so we must also increase the code to compensate for the increased payload.

Connection Setup and Transmission

Let us now discuss how a new connection is set up and how the MPDUs are transmitted as per the six types of feedback.

Phase 1: *Subscriber station requests connection request*: The subscriber station that wants a streaming service from the base station transmits the ranging request (RNG-REQ) packet, which enables the base station to identify the initial ranging, timing, and power parameters. Service flow parameters requests (bandwidth, frequency, peak, or average rate) are sent next and variable-length MSDU indicators are turned on.

Phase 2: *Base station confirms connection*: After receiving a connection request from a subscriber station, the base station transmits a ranging response which provides the initial ranging, timing, and power adjustment information to the subscriber station. Service-flow parameters are agreed upon and a basic connection-ID is provided to the subscriber station.

Phase 3: *Base station starts transmission of MPDUs*: MSDUs obtained from the MAC convergence sublayer are converted to MPDUs. As needed, MSDUs can either be packed or fragmented to form the desired sized MPDUs. Since no feedback is received at the start of transmission, the payload and code size agreed at the time of connection establishment is maintained. When a feedback is received, the next awaiting MPDU is formed depending on the type of feedback received. Upon the reception of each of the six feedbacks, the payload and code sizes are changed. It can be noted that the increase or decrease in payload and code will depend on the ratio of the payload and code. The exact values are found through experimentation (discussed next). In qualitative terms, we mention the action taken by the base station for the respective feedbacks.

- **Feedback 1::Action 1**: increase MPDU payload; decrease code size for not so important MPDU.
- **Feedback 2::Action 2**: increase code size for important MPDU; keep payload and code fixed for not so important MPDU.
- **Feedback 3::Action 3**: decrease payload of MPDU; increase code size of MPDU.
- **Feedback 4:: Action 4**: same as Feedback 3, but the increment/decrement is more.
- **Feedback 5::Action 5**: stall transmission until further request received.
- **Feedback 6::Action 6**: skip transmission of not so important frames; important frame(s) is/are transmitted.

Simulation Model and Results

We conducted simulation experiments to evaluate the improvements achieved by the proposed technique. Evaluations for adaptive and nonadaptive schemes were done under the same channel conditions for a fair comparison.

Channel Model

We assumed a four-state Markov model for the channel. Four states were used to have more granularities in the channel conditions. Each state was characterized by a certain bit error probability (BER): *good state* had a BER of 0.045, the *fair state* had a BER of 0.06, the *medium state* had a BER of 0.07, and the *bad state* had a BER of 0.085. By setting appropriate transition probabilities among these four states, we are able to model different channel conditions for our simulation.

Feedback type	Payload (imp)	Payload (not-imp)	Code (imp)	Code (not-imp)
1	+1 byte	+1 byte	+0 byte	–1 byte
2	+0 byte	+0 byte	+2 bytes	+0 byte
3	–50 bytes	–50 bytes	+5 bytes	+1 byte
4	–75 bytes	–75 bytes	+10 bytes	+1 byte

Table 2. *Change in payload and code.*

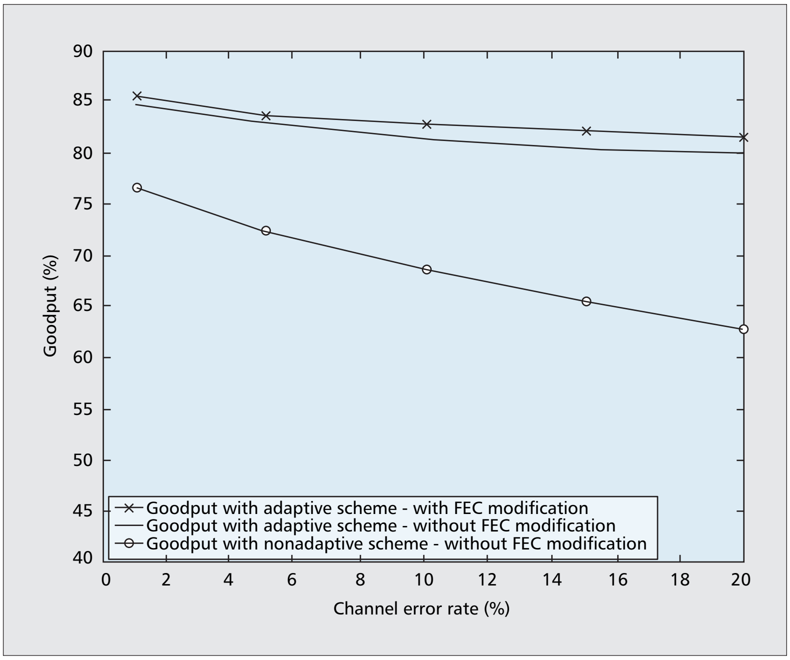


Figure 7. *Goodput comparison.*

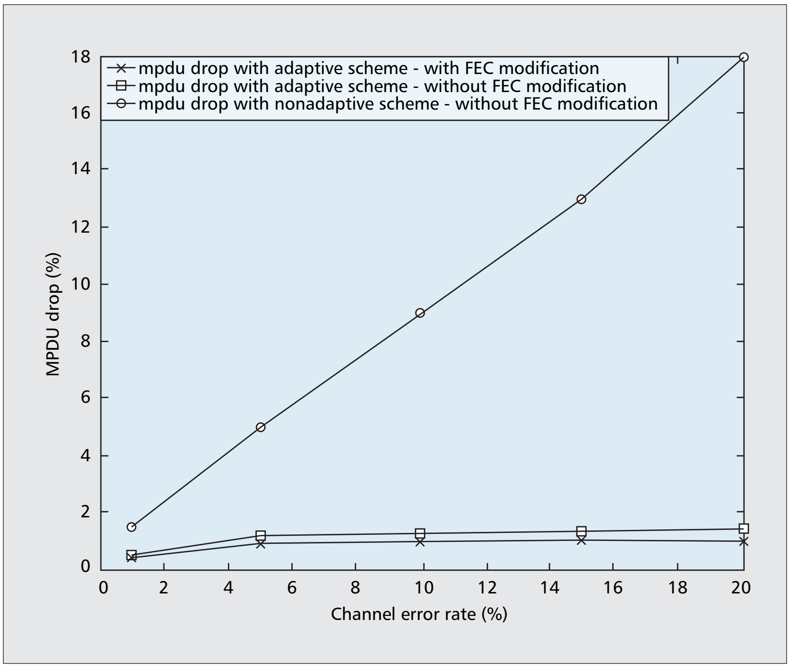
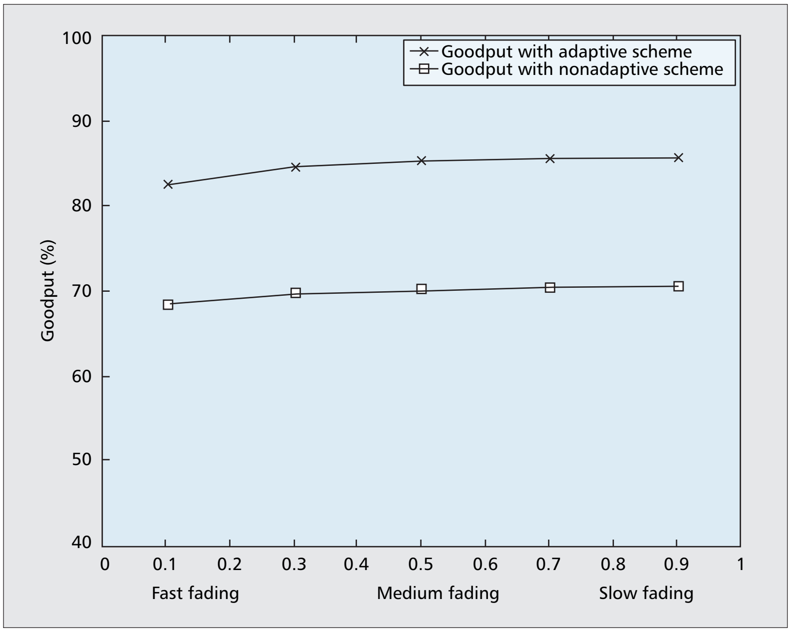


Figure 8. *MPDU drop comparison.*

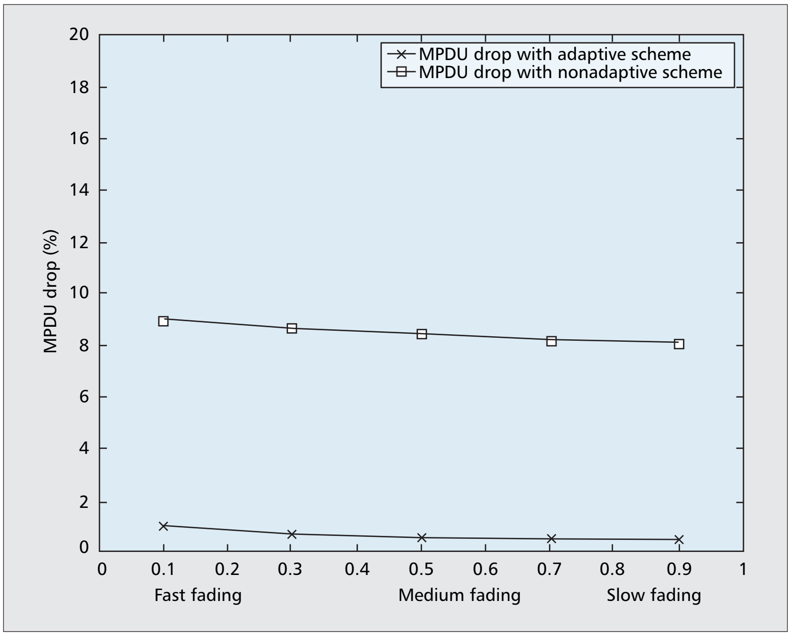
Simulation Parameters

First, it is necessary to obtain the increase/decrease in payload and code size if the goodput is to be optimized. Of course, the increase/decrease will also depend on the ratio of payload to code. In Fig. 4, we show one such example, where we use three different payload to code ratios. It can be observed that, with initial *payload: code* :: 500 : 50 the best goodput is obtained for byte payload increase. Similarly, for *payload: code* :: 500 : 100, increase of bytes gives the best goodput. As a result, it can be concluded that the change in payload and code depends on the payload code ratio.

We assumed an initial packet size of bytes, that is, payload bytes, code bytes, and header bytes. Upon receiving a feedback from the subscriber station, the base station fragments or packs the MSDUs adaptively per the proposed algorithm to form the MPDUs. For different types of feedback, the change in size of payload and code are shown in Table 2. Since, for feedback types 5 and 6, the MPDUs are either stalled or skipped, we do not show them in the table. It can be noted that these values apply to the initial payload to code ratio. Any other ratio will yield a different set of values.



■ **Figure 9.** *Goodput for different channels.*



■ **Figure 10.** *MPDU drop for different channels.*

Simulation Results

In Figs. 5 and 6, we show how the packet-restore probabilities vary over time for nonadaptive and adaptive MPDUs construction respectively. The channel was repeated for both adaptive and nonadaptive cases. Here, we assumed that the probability of having bad channel state was 15 percent, that is, on average the BER was for $> f$ of the time of the total duration of time. We show only a 10 msec (20 to 30 msec) window. It can be observed that the packet-restore probability is very low most of the time in the nonadaptive scheme, as compared to the adaptive scheme where it is maximum almost all the time. Though the packet restore probability drops (in four to five instances), they hardly reach zero signifying that the packets *may* still be recovered. It is evident from the figures that the proposed adaptive scheme outperforms the nonadaptive scheme.

We compare the goodput and MPDU drop percentage for both schemes which are shown in Figs. 7 and 8. For the nonadaptive scheme, payload and code were kept fixed. In case of the adaptive scheme, we first kept the code fixed, modifying only the payload field. Then, in the adaptive scheme, we changed both the payload and code field. From the figures, it is evident that, with the adaptation in both the payload and code field, the improvement is greater.

In Figs. 9 and 10, we show the effects of fast, medium, and slow fading on the goodput and MPDU drop probability for both schemes. Irrespective of the channel conditions, the adaptive scheme performs better.

Conclusions

WiMax is a promising broadband wireless technology that is designed to provide last-mile DSL or T1-level connectivity. In this article we have studied the problem of real-time streaming media over WiMax and have exploited the flexible features present in the MAC layer of 802.16a. We have proposed that the size of MAC packet data units be made adaptive to the instantaneous wireless channel state condition. Based on the type of feedback received, variable-size MPDUs were constructed either by packing or fragmenting the MAC service data units. We then conducted simulation experiments to verify the validity of our proposed scheme.

References

[1] WiMax Forum, 2006, http://www.wimaxforum.org

[2] S. J. Vaughan-Nichols, "Achieving Wireless Broadband with WiMax," *IEEE Comp.*, vol. 37, issue 6, June 2004, pp. 10–13.

[3] I. Joe, "An Adaptive Hybrid ARQ Scheme with Concatenated FEC Codes for Wireless ATM," *Proc. ACM/IEEE MobiCom '97*, pp 131–38.
[4] B. Vucetic, "An adaptive Coding Scheme for Time-Varying Channels," *IEEE Trans. Commun.*, vol. 39, issue 5, May 1991, pp. 653–63.
[5] M. van der Schaar *et al.*, "Adaptive Cross-Layer Protection Strategies for Robust Scalable Video Transmission over 802.11 WLANs," *IEEE JSAC*, vol. 21, issue 10, Dec. 2003, pp. 1752–63.
[6] Y. Li *et al.*, "Joint Power/Playout Control Schemes for Media Streaming over Wireless Links," *Proc. IEEE Packet Video Wksp. 2004*, Dec. 2004.
[7] D. G. Sachs *et al.*, "Hybrid ARQ for Robust Video Streaming over Wireless LANs," *Int'l. Conf. Info. Tech.: Coding and Computing*, 2001, pp. 317–21.
[8] L. Hang and M El Zarki, "Performance of H.263 Video Transmission over Wireless Channels Using Hybrid ARQ," *IEEE JSAC*, vol. 15, no. 9, Dec. 1997, pp. 1775–86.
[9] A. Majumdar *et al.*, "Multicast and Unicast Real-Time Video Streaming over Wireless LANs," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 12, issue 6, June 2002, pp. 524–34.
[10] B. Sklar, *Digital Comnunications*, 2nd ed., Prentice Hall.

BIOGRAPHIES

MAINAK CHATTERJEE (mainak@cpe.ucf.edu) received a Ph.D. from the Department of Computer Science and Engineering at the University of Texas at Arlington in 2002. Prior to that he completed a B.Sc. in physics (Hons) from the University of Calcutta in 1994 and an M.E. in electrical communications engineering from the Indian Institute of Science, Bangalore, in 1998. He is currently an assistant professor in the School of Electrical Engineering and Computer Science at the University of Central Florida. His research interests include economic issues in wireless networks, applied game theory, resource management and quality-of-service provisioning, ad hoc and sensor networks, CDMA data networking, and link layer protocols. He serves on the executive and technical program committee of several international conferences.

SHAMIK SENGUPTA [S] (shamik@cpe.ucf.edu) is a Ph.D. candidate in the School of Electrical Engineering and Computer Science at the University of Central Florida. Before joining the Ph.D. program he worked for Tata Consultancy Services, India, as an assistant systems engineer. He received a B.E. degree with first class honors in computer science and engineering from Jadavpur University, Calcutta, in 2002. His research interests include resource management in wireless networks, auction and game theories, pricing, and WMAN technologies.

SAMRAT GANGULY (samrat@nec-labs.com) received a B.Sc. degree in physics from Indian Institute of Technology, Kharagpur, in 1994, an M.E. in computer science from the Indian Institute of Science, Bangalore, in 1998, and a Ph.D. degree in computer science from Rutgers University, New Jersey, in 2003. Since 2001 he has been a research staff member at NEC Laboratories America, Princeton, New Jersey. His research interests include distributed algorithm design and performance optimization in wireless, overlay, and content delivery networks.