MuVi: A Multicast Video Delivery Scheme for 4G Cellular Networks

Jongwon Yoon*
University of Wisconsin
Madison, WI
yoonj@cs.wisc.edu

Honghai Zhang
NEC Labs America Inc.
Princeton, NJ
honghai@nec-labs.com

Suman Banerjee
University of Wisconsin
Madison, WI
suman@cs.wisc.edu

Sampath Rangarajan
NEC Labs America Inc.
Princeton, NJ
sampath@nec-labs.com

ABSTRACT

Although wireless broadband technologies have evolved significantly over the past decade, they are still insufficient to support the fast-growing mobile traffic, especially due to the increasing popularity of mobile video applications. Wireless multicast, aiming to exploit the wireless broadcast advantage, is a viable approach to bridge the gap between the limited wireless networking capacity and the ever-increasing mobile video traffic demand.

In this work, we propose MuVi, a Multicast Video delivery scheme in OFDMA-based 4G wireless networks, to optimize multicast video traffic. MuVi differentiates video frames based on their importance in reconstructing the video and incorporates an efficient radio resource allocation algorithm to optimize the overall video quality across all users in the multicast group. MuVi is a lightweight solution with most of the implementation in the gateway, slight modification in the base-station, and no modification at the clients. We implement MuVi on a WiMAX testbed and compare its performance to a Naive wireless multicast scheme that employs the most robust MCS (Modulation and Coding Scheme), and an Adaptive scheme that employs the highest MCS supportable by all clients. Experimental results show that MuVi improves the average video PSNR (Peak Signal-to-Noise Ratio) by up to 13 and 7 dB compared to the Naive and the Adaptive schemes, respectively. MuVi does not require modification to the video encoding scheme or the air interface. Thus it allows speedy deployment in existing systems.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communication

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

Video Multicast, Cellular Networks, OFDMA

1. INTRODUCTION

Mobile video streaming (e.g., Youtube [1], Hulu [2]) is one of the most popular applications in recent years and the amount of video traffic destined for mobile devices is increasing rapidly as the number of hand-held devices (e.g., smart phones, iPad) grows. According to the Cisco Visual Index, video accounted for about 40 percent of consumer Internet traffic in 2010 and will reach 62 percent by the end of 2015 [3]. Global mobile data traffic is expected to increase 26 times from 2010 to 2015. Moreover, high-definition (HD) video will surpass standard definition video by the end of 2012 and will become the dominant form of video traffic.

Current WiFi systems can not provide satisfactory quality of video streaming services due to the small coverage and relatively limited bandwidth as the number of mobile users increases. Even worse, WiFi networks are not robust enough to sustain user mobility. 3G mobile networks such as CDMA and UMTS can provide more robust wireless connections to mobile users. However, their bandwidth is not adequate to support applications with high bandwidth requirements (e.g., HD video). Limitations of current WiFi and 3G systems naturally turn our attentions to emerging 4G cellular networks.

4G cellular networks, such as WiMAX and LTE, have emerged as alternatives that can provide much higher bandwidth, spectrum efficiency, and extended coverage. 4G networks are more robust to user mobility compared to WiFi systems, so that they can provide seamless real-time video streaming services. More specifically, 4G technologies can provide peak data rate of 100 Mbps for high mobility applications and 1 Gbps for nomadic applications [12]. Despite the much higher bandwidth provided by 4G technologies, efficient resource utilization is still needed for meeting the high bandwidth and stringent delay requirements of video applications, because the wireless spectrum is shared by multiple users.

Multicast is an efficient way of leveraging the shared nature of wireless spectrum to deliver traffic to multiple clients simultaneously while minimizing the wireless resource usage. The opportunity for wireless multicast arises in many scenarios: **(i) Live sport programs.** In US, numerous sport programs are broadcast live on the Internet. In fact, ESPN provides applications to al-

---

*Jongwon Yoon interned in the Mobile Communications and Networking department at NEC Laboratories America Inc, Princeton during this work.

MobiCom'12, August 22–26, 2012, Istanbul, Turkey.

low smart phones and tablets to watch its live programs. These programs can be provided to mobile clients using *multicast services*. **(ii) Breaking news.** When breaking news happens such as the Indonesia Tsunami, Japan earthquake, or 911 terrorist attack, many people click on the same video clips released by major news agencies. These requests that are made around the same time can be batched, stream-merged, and fulfilled using multicast transmissions. **(iii) EMBS and MBMS Services.** LTE and WiMAX define EMBS (Evolved Multicast Broadcast Service) and MBMS (Multimedia Broadcast Multicast Service) as separate services independent of regular unicast data services while sharing the same devices and spectrum with the unicast traffic. Both the services can be deployed readily to provide mobile TV programs or other commercial video programs, which will require multicast transmission of video content over the wireless systems.

In this work, we develop a Multicast Video delivery scheme (MuVi) in 4G OFDMA-based mobile broadband wireless networks. MuVi is motivated by several recent theoretical works [13, 16, 18] on multicasting scalable videos in OFDMA-based wireless networks. These works make an ideal assumption that the video has *multiple layers* with fixed sizes and do not consider the practical issue of *packetization of video frames with variable sizes*. Although scalable videos are interesting to study, most Internet videos today are not scalable and are encoded using traditional technologies such as MPEG4 or H.264. Therefore, the schemes in [13, 16, 18] cannot apply directly to most popular Internet videos (such as Youtube). Fortunately, we identify similar dependency relationship between P frames in a non-scalable video to that between different layers in a scalable video, and leverage a dynamic-programming technique similar to the one used for scalable video multicasting in [18], for *non-scalable* video wireless multicasting. By doing so, we also address the practical issue of variable video frame sizes naturally.

MuVi Overview: MuVi is designed as a proxy in the Access Service Network (ASN) gateway as depicted in Figure 1. There are four key design elements in MuVi. **1)** MuVi collects the feedback of wireless channel quality information of all clients in a multicast group through the base-station and obtains the supportable MCS for each client. **2)** MuVi prioritizes video frames by setting a utility value for each frame based on the frame type. The utility may also depend on the user profile. Thus, MuVi supports user differentiation. **3)** MuVi performs efficient resource allocation to maximize the system utility considering both the available radio resources and the video packets to be transmitted. **4)** MuVi assigns Modulation and Coding Schemes (MCS) for the video packets and hands them over to the base-station, which sends the video packets over the air using the assigned MCSs.

Different from several recent works on video streaming [8, 14] which provide graceful video delivery, MuVi does NOT modify the video encoding nor the wireless transmission schemes. It performs resource allocation in operational wireless systems and video frame prioritization with existing video encoding technologies to optimize the resource utilization and video delivery. MuVi is also different from several WiFi-specific video optimizations which utilize the notion of differential value of data packets, e.g., Medusa [21], in that 1) MuVi is designed for OFDMA-based systems, and 2) MuVi does not require client side modifications. Since it does not modify the air interface or the clients, MuVi is standards compatible and allows immediate deployment in commercial wireless systems (such as WiMAX).

We implement MuVi on a PicoChip [11] WiMAX testbed including a video server, an ASN gateway, a PicoChip basestation, and several mobile clients. We evaluate the performance of MuVi and compare it with a Naive multicast scheme that uses the most robust MCS and the Adaptive scheme proposed in DirCast [9], which picks the highest MCS that is supportable by all clients. We have the following findings.

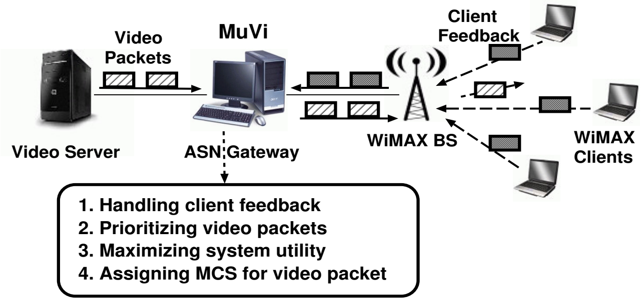


Figure 1: MuVi performs four operations while receiving video packets from the media server, and then it delivers video packets to the WiMAX base-station.

- MuVi improves the average video quality by up to 13 dB and 7 dB in terms of PSNR compared to the Naive scheme and the Adaptive scheme, respectively.
- MuVi reduces the inter-packet arrival delay by up to 80% and 55%, compared to the Naive and the Adaptive schemes, respectively. This indicates significantly fewer glitches and video stalls for MuVi.
- MuVi is a gateway solution and does not require any modification on the client side. It can be easily incorporated in cellular networks with only minimum modifications on the base-stations.
- Although it is implemented in WiMAX, MuVi is also applicable to other OFDMA-based systems, e.g., LTE and LTE-Advanced systems.

The rest of the paper is organized as follows. Section 2 provides background on WiMAX and OFDMA systems. We describe basic operations of MuVi and functional building blocks in Section 3. Section 4 describes the prototype implementation and details of our WiMAX testbed. In Section 5, we evaluate the performance of MuVi using our testbed and present detailed experimental results. We discuss limitations and related work in Section 6 and 7, respectively. Section 8 concludes the paper.

2. BACKGROUND

WiMAX Preliminaries: While our work applies to Orthogonal Frequency Division Multiple Access (OFDMA)-based networks in general, we implement MuVi and conduct experiments on a WiMAX (802.16e [5]) testbed due to its availability[1]. Here, we give a brief overview of OFDMA and WiMAX networks (The detailed 802.16e specifications can be found in [5]). A WiMAX frame is a two-dimensional template across the time and the frequency domains. A slot, which is the minimum allocation unit in the frame, is made up of one sub-channel in the frequency domain and one, two, or three symbols in the time domain. Data to multiple clients are scheduled as rectangular bursts of slots in a frame. In TDD (time division duplexing) mode, a WiMAX frame is divided into two sub-frames: a downlink sub-frame and an uplink sub-frame, separated by Transmit-To-Receive and Receive-To-Transmit Gaps (TTG and TRG, respectively). An example of a WiMAX TDD frame is shown in Figure 2. Under the most commonly deployed

---

[1] LTE platform is not available yet.

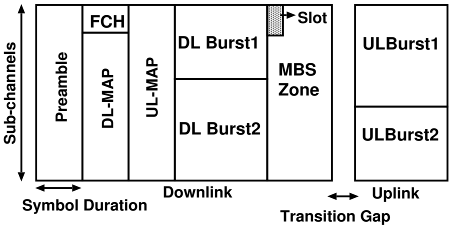


Figure 2: WiMAX frame structure.

Index	Modulation and Coding Rate
0	BPSK
1	QPSK(3/4)
2, 3	16 QAM(1/2, 3/4)
4, 5, 6, 7	64 QAM(1/2, 2/3, 3/4, 5/6)

Table 1: MCS indices available for data modulation and coding rates.

profile with 5 MHz and 10 MHz bandwidth, a WiMAX frame has a frame duration of 5 milliseconds.

A frame consists of a preamble, control data and payload data. The preamble is transmitted with power boost and allows mobile clients to lock on to the base-station. The control signaling consists of a Frame Control Header (FCH) and a MAP. The FCH contains the system control information and the information about the MAP. The DL-MAP indicates where each burst is placed in the frame, which client it is intended for, and what modulation and coding scheme (MCS) decodes it. Table 1 shows the different modulation levels employed in a WiMAX base-station. Similarly the UL-MAP indicates where the client should place its data on the uplink frame. The uplink sub-frame has dedicated sub-channels for HARQ, which are used by clients to explicitly acknowledge (ACK/NACK) the reception of each data burst. Clients also use the uplink sub-frame to report the instantaneous channel state information (CSI) to the base-station. Base-stations use a dedicated data burst zone, called MBS zone (in Figure 2), for multicast and broadcast services. While data bursts can be modulated using any MCS, the control parts (FCH and DL/UL MAPs) are always transmitted using QPSK and typically with multiple repetitions. The preamble is transmitted with a higher power compared to the other parts of the frame.

MPEG4: We describe the characteristics of MPEG4/H.264 video frames, since we use MPEG4 as example video encoding schemes. In MPEG4 or H.264, a video sequence is partitioned into Group of Pictures (GOP). Figure 3 shows an example of frame sequence in a GOP. Each GOP consists of a certain number of I, P, and B video frames, which are then further divided into packets typically with fixed length except for the last packet of each video frame. I frames are intra-coded pictures and it can be decoded independently. P frames are forward predicted pictures and require their preceding I or P frames to be decoded. B frames are bidirectionally predicted pictures and require both preceding and succeeding I or P frames for decoding. The arrows in Figure 3 represent the dependencies between frames. A typical GOP consists of 30 frames of repeated sequences (e.g., IBBPBBPBB...PBB).

3. DESIGN AND OPERATION OF MuVi

Our proposed system, MuVi, comprises a media server, media-aware gateway, base-station, and multiple clients in the multicast session as depicted in Figure 1. The basic operations of MuVi are as follows:

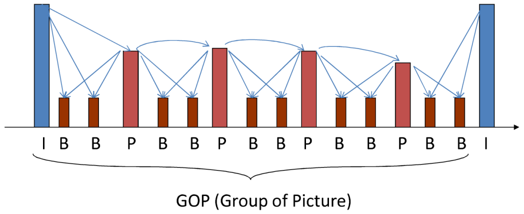


Figure 3: An example video frame sequence in a GOP. An arrow indicates that the frame at the head of the arrow depends on that at the tail of the arrow.

1) The WiMAX clients send the channel state information (CSI) to the base-station periodically. Providing CSI is a standard mechanism in 802.16e (and all mobile broadband systems). Upon receiving the channel feedback, the base-station computes appropriate moving average for each client, aggregates the averaged feedback, and forwards them to the ASN gateway. The ASN gateway then determines the supportable MCS for each client based on their average channel conditions, along with an MCS table (details in subsection 3.1).
2) The media server sends the video packets to the ASN gateway, which performs packet scheduling and possibly drops some video packets based on the available radio resources and the frame types. The packet scheduling includes packet re-ordering and determining the Modulation and Coding Scheme (MCS), i.e., the PHY rate, employed at the base-station for each video packet (details in subsection 3.3).
3) After receiving the video packets, the base-station applies the MCS selected by the ASN gateway and transmits them over the air using multicast (details in subsection 3.4).
4) When the clients receive the packets, they perform packet re-ordering and then pass the received packets to the video player (e.g., VLC [7]). Packet re-ordering is typically implemented in upper-layer protocols such as RTP (Realtime Transport Protocol).

To execute the above operations, we modify the ASN gateway and the base-station. Our major modifications lie at the ASN gateway, which consist of 1) collecting the supportable MCS for each client, 2) classifying packet types (i.e., video frame types), and 3) determining the PHY transmission rate for each video packet.

The ASN gateway can perform deep-packet-inspection to find the type of video frames, but this may cause extra overhead at the ASN gateway. Hence, we slightly modify the video player application (VLC player) running on the media server to ease the job at the ASN gateway. The customized VLC player inserts the video frame type information in the IP header (e.g., DiffServ field) before sending video packets to the ASN gateway. The ASN gateway simply looks up the IP header to categorize received packets.

The base-station needs to make small modifications to 1) forward the client channel feedback to the ASN gateway, and 2) transmit the packets using the MCS instructed by the ASN gateway. We believe these are within the scope of what BS vendors are willing to do. The clients are not required to change given that re-ordering can be performed by upper-layer protocols such as RTP, which is typically employed in real-time multicast streaming.

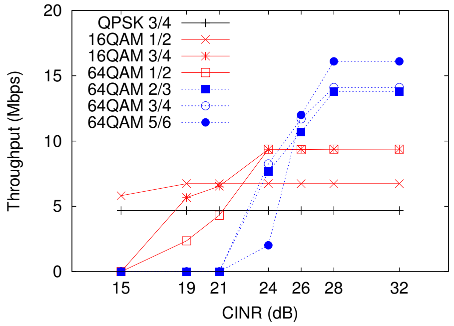


Figure 4: MuVi can find the supportable MCS for a client based on its CINR feedback.

In the following subsections, we describe the details of each design element.

3.1 Handling Client Feedback

The wireless channel suffers from packet losses and errors caused by interference and noise. It is hard to determine the target MCS (i.e., PHY transmission rate) in a wireless multicast system because of the different channel conditions amongst multiple users in the multicast group. Client channel condition is an important factor to determine the target PHY rate to multicast while avoiding resource under-utilization. To guarantee data delivery over wireless channels, traditional wireless multicast systems use the most robust MCS (i.e., the lowest PHY transmission rate). However, this approach leads to under-utilization of available radio resources when most multicast users can support higher MCS than the lowest one.

A carefully selected MCS can improve the throughput and the video quality of a client. The most appropriate MCS for each client depends on its channel conditions. Using a lower MCS leads to resource under utilization and using a higher MCS renders high packet loss rate. MuVi leverages the information of the supportable MCS for each client to make efficient use of radio resources when multicasting the video traffic.

To find the relationship between the client channel condition and the supportable MCS, we measure the client throughput from various locations where the channel conditions vary. We deploy the clients on several locations where clients experience different levels of channel conditions in terms of Carrier to Interference plus Noise Ratio (CINR). For each location, base-station sends UDP traffic using *iperf* to the client while trying out all possible MCSs and the client records the obtained UDP throughput with respect to each MCS used. We plot the measured throughput for all locations where each location is represented as a point in Figure 4.

Using this figure, we can obtain a MCS-CINR table, as shown in Table 2. By looking up the table, we can easily figure out the highest supportable MCS for a client based on its reported channel feedback. For example, a client can successfully decode packets transmitted with MCS 4 (64-QAM with 1/2 coding) or lower when its observed CINR is 24 dB. We comment that constructing the MCS-CINR table only needs to be done once.

For clients, providing channel feedback (CINR report) to base-station is part of the IEEE 802.16e (WiMAX) standard [5]. We modify the MAC code of the base-stations to compute the moving average of the CINR values for each client, aggregate the averaged CINR values of all clients, and forward them to the ASN gateway periodically. Once the ASN gateway receives the client feedback from the base-station, it can identify the highest MCS that can be decoded by each client from Table 2.

MCS index	CINR range (dB)
7	(28, ∞)
6	(26, 28]
5	(24, 26]
3, 4	(20, 24]
2	(15, 20]
1	(-∞, 15]

Table 2: CINR range and the corresponding MCS.

3.2 Packet Values

The video sequence consists of different frame types (i.e., I, P, and B frames) for MPEG4 or H.264 encoded video. Each video frame contains different video/audio information. For example, P and B frames carry only difference information so they depend on some other frames (i.e., I and P frames) to be decoded successfully.

MuVi prioritizes the packets based on their dependency and assign the priorities as a function of number of packets depending on them. The more packets depend on it, the higher priority is assigned to a packet. Typically, I frames have the highest priority and B frames have the lowest priority. P frames may have different priorities depending on their positions in a GOP. For example, a P frame in the earlier part of a GOP has more number of dependent frames compared to that in the later part of a GOP, hence the former has higher priority than the latter.

In addition, users may have different priorities. For example, a high-profile user may pay higher subscription fee and expect high video quality. So we allow a video frame (i.e., the packets) to have different priority values for different clients.

3.3 Utility Maximization

The main media optimization engine lies in the ASN gateway. It collects the channel feedback for each client in the multicast group from the base-station and determines the resource allocation for the video frames in order to maximize the total system utility.

The problem setup is as follows. There are Q available slots in a WiMAX frame for the video multicast session, where a slot is a two-dimensional minimum allocation unit in the time and frequency domains in an OFDMA frame. The OFDMA frame duration is τ ms ($\tau = 5$ in most WiMAX systems). There are M Modulation and Coding Schemes (MCS) representing different transmission rates. For MCS m, one OFDMA slot can deliver R_m bytes and the minimum required CINR for decoding MCS m is $\bar{\gamma}_m$, where $1 \leq m \leq M$. There are K clients in the multicast session and their CINR values are $\gamma_1, \gamma_2, \cdots, \gamma_K$.

We consider the transmission of a Group of Picture (GOP) with J video frames. A GOP includes an I frame, a sequence of P frames, and a group of B frames as depicted in Figure 3. The jth video frame has length L_j. Receiving frame j at client k obtains a utility u_j^k given that all frames it depends on are also received by client k. By including a superscript k in the utility function, we allow different priorities among different users. We assume that the inter-frame time is Δ second and thus the total number of slots available for transmitting the GOP is $T = \lfloor \Delta JQ/\tau \rfloor$. The objective is to maximize the total utility received by all clients subject to the total slot constraint.

The dependency relationship between I/P frames and B frames is different. All frames in the GOP need to refer I frame when decoding. P frames depend on their preceding P frames and are needed by both intermediate B frames and succeeding P frame. B frames are not referred by other frames. Considering different frame dependencies, we first schedule groups of I/P frames and B frames separately and then optimize the overall utility across the two groups.

3.3.1 I/P Frame Scheduling

We first consider the problem of scheduling only I and P frames in a GOP. For notational convenience, we assign a special sequence number P_0 to the only I frame in the GOP and hereafter, we treat I frame as a special P frame P_0. So we have a sequence of P frames P_j, where $j = 0, 1, \cdots, G$, such that frame P_{j+1} depends on P_j ($0 \leq j \leq G-1$), where G is the number of P frames. Let the total number of available slots for P frames be $t \leq T$ and the objective is to maximize the total utility subject to the total slot constraint t for all P frames.

To ensure minimum video quality at every client, we require that the first $j_0 \geq 0$ P frames be received by all clients, and set aside t_1 slots for transmitting these P frames using the highest MCS that can be decoded by all clients. Thus the available number of slots for the remaining P frames is $t' = t - t_1$. Without loss of generality, we assume $j_0 = 0$ (i.e., no P frame is required by all clients) in the following.

Let binary variable x_{jm} indicate that frame P_j is transmitted with MCS m and z_j^k indicate that frame j is valid for client k, which means that all frames on which frame j depends, including j, can be decoded by client k, so

$$z_j^k = \begin{cases} 1, & \text{if for all } 0 \leq l \leq j, \text{ there exists } m \\ & \text{such that } x_{l,m} = 1 \text{ and } \bar{\gamma}_m \leq \gamma_k, \\ 0 & \text{otherwise.} \end{cases}$$

Now the problem of utility maximization can be written as

$$\begin{aligned} \max \quad & \sum_{j=0}^{G} \sum_{k=1}^{K} z_j^k u_j^k \\ s.t. \quad & \sum_{m=1}^{M} x_{jm} \leq 1, \\ & \sum_{m} \left\lceil \sum_{j} \frac{x_{jm} L_j}{R_m} \right\rceil \leq t, \end{aligned} \tag{1}$$

where in the last equation, we assume that frames transmitted using the same MCS can be bundled together for the allocation of radio resources (i.e., slots). We slightly abuse the notations by using u_j^k to denote the utility of frame P_j for client k.

Interestingly, problem (1) is similar to the problem of maximizing sum of video quality when multicasting SVC-encoded videos by viewing the jth P frame as the jth layer in an SVC video. Indeed, the dependency relationship between P frames in non-scalable videos are equivalent to that between different layers in SVC videos. Therefore, we can leverage a dynamic programming algorithm similar to the one used in SVC video multicasting [18] to solve problem (1).

Define $U_P(j, m, t)$ as the optimal utility with the P frames $P_l, l = 0, \cdots, j$ with MCS up to m and at most t slots. Let $\tau_{j_1, j_2, m} = \lceil \sum_{l=j_1}^{j_2} L_l / R_m \rceil$ be the number of slots required to deliver frames $P_l, l = j_1, \cdots, j_2$ using MCS m. To compute the optimal utility $U_P(j, m, t)$, only the last few frames ending at P_j may choose MCS m. Assuming that frames $P_l, l = i+1, \cdots, j$, are transmitted with MCS m (note that if $i = j$, no frame is transmitted with MCS m), the utility $U_P(j, m, t)$ is then the summation of the optimal utility of the first i frames using MCS up to $m - 1$ with $t - \tau_{i+1,j,m}$ slots and the utility obtained by transmitting frames $i+1$ to j using MCS m. Maximizing over all possible i, we obtain the recursive equation for $U_P(j, m, t)$ as follows.

$$U_P(j, m, t) = \max_{0 \leq i \leq j} \left[U_P(i, m-1, t - \tau_{i+1,j,m}) + \sum_{l=i+1}^{j} \sum_{k \in S_m} u_l^k \right] \tag{2}$$

$$q(j, m, t) = \text{argmax}_i \left[U_P(i, m-1, t - \tau_{i+1,j,m}) + \sum_{l=i+1}^{j} \sum_{k \in S_m} u_l^k \right] \tag{3}$$

where S_m is the set of users who can decode MCS m. In Eq. (3), $q(j, m, t)$ keeps track of the best parameter i that maximizes Eq. (2) and it is used for finding the optimal resource allocation later.

The initial conditions for $U_P(j, m, t)$ are

$$\begin{aligned} U_P(j, m, t) &= -\infty, \text{ if } t < 0 \\ U_P(j, 0, t) &= -\infty, \text{ if } j \geq 0, t \geq 0 \\ U_P(-1, m, t) &= 0, \text{ if } m \geq 0, t \geq 0 \end{aligned} \tag{4}$$

The first two equations state that $t < 0$, or $m = 0$ and $j \geq 0$ is not a valid choice for the utility function $U_P(j, m, t)$. We note that the valid MCS choices are 1 to M and $m = 0$ is a dummy MCS used for initialization. In the last equation above, $j = -1$ with MCS $m \geq 0$ indicates that all frames ($j \geq 0$) have been considered and the dummy frame $j = -1$ has zero utility.

From Eq. (2), we can see that $U_P(j, m+1, t) \geq U_P(j, m, t)$. Therefore, the optimal utility is always achieved at $m = M$. For each t, we can then compute the optimal utility U_P^* for each available time slot t:

$$\begin{aligned} U_P^*(t) &= \max_{j \geq 0} U_P(j, M, t) \\ j^*(t) &= \text{argmax}_{j \geq 0} U_P(j, M, t) \\ m^*(t) &= \min\{m : U_P(j^*(t), m, t) == U_P^*(t)\} \end{aligned} \tag{5}$$

where $j^*(t)$ achieves the optimal utility, indicating that P frames $j > j^*(t)$ are dropped. The dynamic-programming procedure for I/P frame scheduling is illustrated in Algorithm 1.

Algorithm 1 I/P frame scheduling

```
1: Use Eq. (4) to compute the utility $U_P(j, m, t)$ at the boundary.
2: for all $j, m, t$ do
3:     Compute $U_P(j, m, t)$ iteratively using Eq. (2).
4: end for
5: Find the optimal utility $U_P^*(t)$ for all $0 \leq t \leq T$ using Eq. (5).
```

3.3.2 B Frame Scheduling

Given that t out of total T slots are allocated for transmitting I and P frames, $T - t$ slots are left for B frames. Since some P frames may be dropped after the P frame scheduling, transmitting the B frames that depend on those dropped P frames does not produce any benefit. Hence, we only consider the set of B frames that are still useful. Let

$$\mathcal{B}(t) = \{b : \text{ B-Frame } b \text{ does not depend on P frame } j > j^*(t)\}.$$

As B frames are less important than P frames, we use higher (or equal) MCS for B frames than the last transmitted P frame $j^*(t)$ for efficient use of given resources (lower MCS requires more number of slots for transmission). Assume that B frames in $\mathcal{B}(t)$ are naturally ordered by their decoding time (or display time).

The problem of B frame scheduling can be formulated as, selecting a subset of B frames and an appropriate MCS $m \geq m^*(t)$ for each B frame, to maximize the total utility, such that the total number of slots does not exceed $T - t$. The problem can easily be seen as a multiple-choice knapsack problem and is NP-hard. Nevertheless, most B frames have relatively small sizes and small utility compared to I and P frames.

We use the following algorithm to obtain a sub-optimal solution by imposing the requirement that all B frames use the same MCS. To be specific, for each MCS $m \geq m^*(t)$, we find maximum number b_m of B frames that can be transmitted using MCS m under the slot constraint. Finally, we pick the MCS that maximizes the total utility. B frame scheduling algorithm is described in Algorithm 2.

Algorithm 2 B frame scheduling

```
for all $m^*(t) \leq m \leq M$ do
    Find the maximum $b_m$ such that the first $b_m$ B frames in $\mathcal{B}(t)$ can be transmitted with MCS $m$ in $T - t$ slots.
    The resulting utility is $U_B(m, T - t) = \sum_{b=1}^{b_m} \sum_{k \in S_m} u_b^k$.
end for
Find the optimal $m_0 = \text{argmax}_m U_B(m, T - t)$ and obtain the utility $U_B^*(T - t) = U_B(m_0, T - t)$.
```

3.3.3 *Joint I/P/B Frame Scheduling*

To perform the joint scheduling, we find the optimal resource allocation between I/P frames and B frames. Let

$$\begin{aligned} U^* &= \max_t U_P^*(t) + U_B^*(T - t) \\ t^* &= \text{argmax}_t U_P^*(t) + U_B^*(T - t) \end{aligned} \qquad (6)$$

be the optimal total utility and the optimal number of slots allocated to I/P frames, respectively. $T - t^*$ is the optimal number of slots allocated to B frames.

3.4 MCS assignment

Finally, MuVi determines the number of I/P frames and B frames transmitted and the MCS for each transmitted frame based on the result of utility optimization obtained in the previous subsection. The following steps are executed in order. Note that I frame is viewed as the special P frame with index 0.

1. t^* and $T - t^*$ (obtained from Eq. (6)) are the number of slots allocated to P frames and B frames, respectively.
2. The first $j^*(t^*)$ P frames (from Eq. (5)) are transmitted and the rest P frames are discarded.
3. $t = t^*, j = j^*(t^*), m = m^*(t^*), i = q(j, m, t)$.
4. P frames $P_l, l = i + 1, \cdots j$ are transmitted with MCS m (if $i == j$, no frames are transmitted with MCS m).
5. If $i < 0$, go to Step 6. Otherwise, $t = t - \tau_{i+1,j,m}$, $j = i, m = m - 1, i = q(j, m, t)$, go to step 4.
6. $m_0 = \text{argmax}_m U_B(m, T - t^*)$. The first b_{m_0} B frames are transmitted with MCS m_0 and the rest B frames are dropped.

Once the MCS for each video frame is determined, the packets belonging to each frame are marked with the assigned MCS index in the DiffServ field of the IP header.

4. WIMAX TESTBED AND PROTOTYPE IMPLEMENTATION

In this section, we describe our WiMAX network testbed and the prototype implementation on it.

Testbed: Our WiMAX testbed consists of four components: a WiMAX femto base-station, an Access Service Network (ASN) gateway, a video server, and several WiMAX clients as depicted in Figure 5. The base-station is a PicoChip [11] WiMAX platform based on IEEE 802.16e standard [5]. The PicoChip base-station is tuned to operate in a 10 MHz bandwidth with the center carrier frequency of 2.59 GHz, for which we have obtained an experimental license to transmit WiMAX signals over the air. Both the ASN gateway and the video server run on typical Linux machines with a 2 GHz processor and 1 GB memory. The WiMAX clients are Windows laptops with commercial USB dongle WiMAX cards [10] or Beceem PCMCIA [4] interfaces.

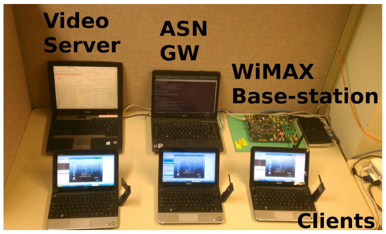


Figure 5: WiMAX testbed.

The clients can be associated with the WiMAX base-station through the ASN gateway. The ASN gateway controls and maintains both uplink and downlink connections between the WiMAX base-station and the clients through configuring service flows which are unidirectional flows of data traffic. All uplink and downlink traffic between the base-station and the video server are tunneled through the ASN gateway.

The base-station manages the scheduling for both downlink and uplink traffic. Since multiple users share the OFDMA wireless resources (i.e., the WiMAX frames), the base-station incorporates a scheduler for efficient resource management. The downlink and uplink scheduler assigns a certain number of slots to each flow and informs the clients about the resource allocations through DL/UL MAPs.

Prototype Implementation: Our prototype implementation includes modifications at the video server, the ASN gateway, and the WiMAX base-station. The main components of MuVi (handling client feedback, optimizing resource allocation, and frame re-ordering) are implemented on the ASN gateway using the Click Modular Router [19] as a user-level module to handle all video frames to the base-station. We significantly extend and modify the click module (about 2000 lines of C++ code) to realize MuVi's building blocks.

The modified version of VLC media player [7] runs on the video server to send media traffic to the group of multicast users. The VLC module first inserts the frame information in each video packet's IP header (i.e., DiffServ field) and then sends them to the ASN gateway. This marking process helps the ASN gateway classify the video packets easily without deep packet inspection. After classifying the packet types (i.e., video frame types), the ASN gateway puts them in different queues based on their types. Then, ASN gateway schedules video frames to the base-station after applying the packet re-ordering and rate selection algorithms as described in Section 3. In broadband cellular networks, gateways are typically sophisticated servers managing hundreds of base-stations. Since all downlink and uplink traffic go through the gateway, assigning a MCS for all packets at the gateway causes very little overhead, therefore, the overhead of MuVi at the gateway is minimal (evaluation in subsection 5.4). As a result of the scheduling algorithm, each packet is marked with the assigned MCS index in the DiffServ field in the IP header and is delivered to the base-station.

Now that the scheduling process is offloaded from the base-station to the ASN gateway, the base-station simply packs incoming video

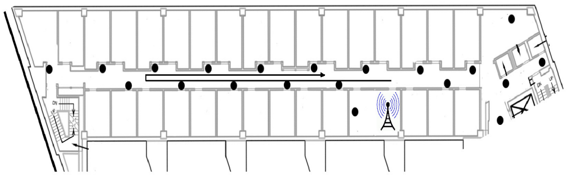


Figure 6: Dots represent the client locations for all experiments. The line represents the path of a mobile client under the mobility experiment.

frames into the WiMAX data bursts and transmits them to the clients in a multicast group. The reference design of PicoChip platform does not involve sophisticated scheduling routines and provides just a working link between the base-station and the clients. We modify the base-station in two aspects to incorporate MuVi. First, it reads the DiffServ Field of the IP header to extract the MCS information determined by the scheduling process in the ASN gateway and transmits the packets using the specified MCS. Second, we modify the base-station MAC code to provide the client feedback to the ASN gateway once every 100 milliseconds. Client channel conditions are periodically reported from the clients to the base-station as part of the standard operation, so the base-station just aggregates the feedback and forwards to the ASN gateway. The size of each channel feedback is relatively small (less than 40 Bytes), so this routine causes very little overhead at the base-station.

5. PROTOTYPE EVALUATION

We evaluate the efficacy of MuVi on our WiMAX testbed by comparing it with existing wireless multicast schemes. We first describe our experiment setup, two reference schemes, evaluation metrics, and test video we used for experiments.

Experiment Setup: Our WiMAX testbed (Figure 5) is deployed in a typical indoor environment as depicted in Figure 6. To generate various topologies, we deploy clients in multiple locations, where the channel quality varies in terms of Carrier to Interference plus Noise Ratio (CINR). Each client is exposed to a different level of channel condition. All clients have the same priority and the utility of each frame is set to the number of frames depending on it (including itself).

For the confidence results, we repeat each set of experiments more than 5 times and present the averaged results with a 95% confidence interval, except for the microscopic and mobility results, where we present the results of a single run with finer granularity. We maintain the same client topologies while running different video multicast schemes to ensure fair comparisons.

Reference Schemes: We compare the performance of MuVi with that of Adaptive and Naive approaches. The Naive scheme is a traditional WiFi/WiMAX multicast approach that uses the most robust (lowest) MCS for delivering data to the group of clients. This approach does not consider the client channel conditions and only guarantees successful deliveries without considering the real-time requirement.

For the Adaptive scheme, we only adapt transmission bit-rates for video frames like DirCast [9]. Based on the client channel conditions, Adaptive scheme uses the highest MCS which can be supported by all clients in the multicast group. The MCS used in Adaptive scheme to deliver the multicast data reflects instantaneous client channel conditions.

While Medusa [21] also determines the MCS based on video frame types, it requires the clients send the reception reports (i.e., ACK/NACKs) to the Medusa proxy and requires modifications at the clients. Thus it is not amenable to implementation in cellular networks, so we do not compare MuVi against it.

Video quality (MOS)	PSNR range
Excellent	> 37
Good	31-37
Fair	25-31
Poor	20-25
Bad	< 20

Table 3: The relationship between video quality (MOS) and the PSNR range.

Evaluation Metrics:

- *PSNR:* PSNR (Peak Signal-to-Noise Ratio) is a standard metric of video quality and is a function of the mean square error between the original and the received video frames. If a video frame is dropped or past the deadline, it is considered lost and is concealed by copying from the last received frame before it. The relationship between the user perception expressed in Mean Opinion Score (MOS) and the PSNR values [20] is summarized in Table 3.
- *Inter-packet delay:* We measure inter-packet delay of received packets to quantify the jitter of delivered video stream. To successfully display a streaming video on the client, all video packets belonging to the same video frame need to be received within a given deadline. High jitter values between packets cause bad visual quality (e.g., glitches and stopped video frames when displaying).
- *Ratio of packets past the deadline:* PSNR measures video quality based on the received video frames regardless of the deadline of video frames. However, the video frames past the deadline can not be used to display real-time streaming media. We measure the ratio of packets which miss the deadline to reflect the real-time streaming video quality.
- *MCS:* MCS selection for each video frame is important to satisfy the guaranteed delivery to all clients that are exposed under different channel conditions. MCS selection is even more important to prevent the under-utilization of given resources (i.e., number of slots in WiMAX frame). We measure the average MCS used for delivering video frames which reflects the efficacy of WiMAX frame resources usage.

Test Video: We use MPEG4-encoded [6] video streaming for system evaluations. In the experiments, the video is encoded at 2.5 Mbps and is multicasted to the clients. Although we consider non-scalable video sequences in the experiments, our scheme can also be applied to scalable videos by assigning packets with different utility based on the layers they belong to.

5.1 Performance under Various Resource Constraints

We evaluate the performance of MuVi and compare it to Adaptive and Naive schemes in the presence of cross-traffic (i.e., background traffic) while varying the number of available resources (i.e., slots) in a WiMAX frame for multicast transmissions. We use 2.5 Mbps video stream for all experiments.

The maximum data throughput depends on both the number of slots allocated in each WiMAX frame and the MCS level chosen in the WiMAX system. For example, allocating 60 slots per frame with the highest MCS (64-QAM with 5/6 coding) can yield 2.6 Mbps throughput, theoretically. In practical experiments, MCS

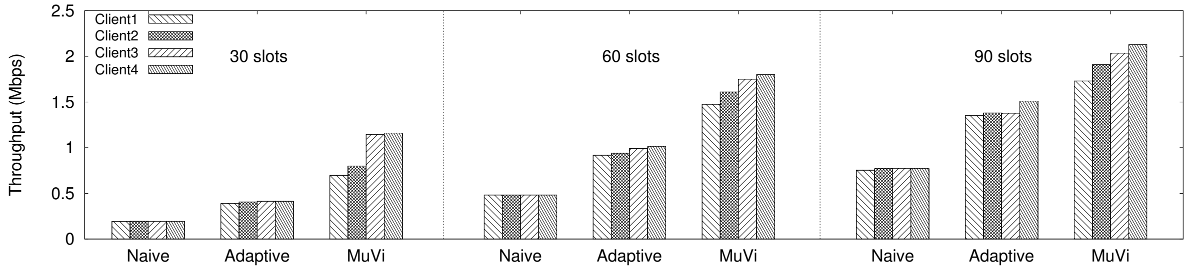
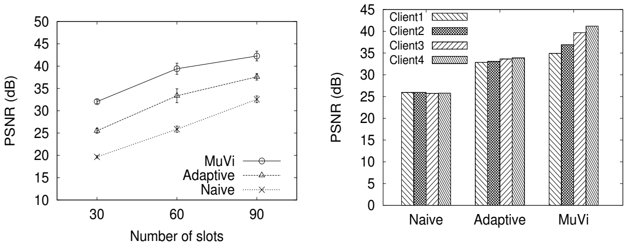


Figure 7: Measured throughput for all clients under different resource constraints. MuVi provides at most x4.9 and x2.35 throughput improvement comparing to Naive and Adaptive schemes, respectively.



(a) The average PSNR value and confidence interval

(b) PSNR value for each client under multicast to four clients.

Figure 8: (a) MuVi provides best video quality regardless of available resources. (b) MuVi provides differentiated service to the clients depends on their channel conditions.

used for multicast may be less than the highest one considering various channel conditions of the users in the multicast group. Hence, 60 slots may not be sufficient to successfully deliver the video encoded with 2.5 Mbps. MCS selection for transmitting multicast data depends on the clients' channel conditions, the available radio resources, and the optimization algorithm as described in Section 3. We vary the number of slots in a WiMAX frame to generate different resource constraints introduced by the background traffic. In the experiments, 30 and 60 slots per frame represent very tight resource constraints and 90 slots per frame represent just enough resources for delivering the video at 2.5 Mbps.

5.1.1 PSNR and Throughput

The average PSNR values and 95% confidence intervals for each multicast scheme obtained from 10 experiments are presented in Figure 8(a). For each multicast scheme, we aggregate the PSNR values from all clients in the multicast group and present the average of them with confidence interval from multiple runs. Figure 8(a) shows MuVi significantly outperforms the other two schemes under all resource constraints we evaluated. Specifically, the average PSNR value is excellent for MuVi, good for Adaptive, and only fair for Naive when 90 slots per frame for multicasting is used. Even when the radio resources are severely insufficient (i.e., 30 slots per frame), MuVi still provides good video quality while the other two schemes suffer. In average, MuVi improves the video quality by up to 13 dB and 7 dB, compared to the Naive scheme and the Adaptive scheme, respectively.

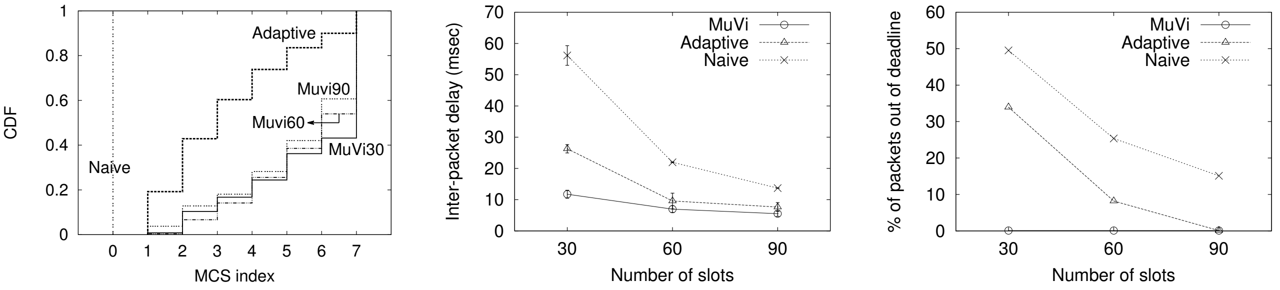
Similar subjective results are observed when we watch streaming video on the client side. We notice more frequent glitches and stalls with Adaptive and Naive schemes, while we see much smoother streaming with MuVi. The reason that MuVi outperforms the other two schemes is mainly due to the packet scheduling (e.g., resource optimization) and MCS assignment discussed in Section 3.3. Using the most robust MCS can guarantee successful deliveries, but it leads to very low efficacy of resource utilization and eventually lots of frames cannot be delivered before their display or decoding deadlines. Instead, MuVi selectively drops some less important video frames, and judiciously assigns the MCS of packets based on their priorities in order to achieve the maximum overall system utility while guaranteeing that the deadlines of all packets are met.

In the experiments, we deploy the clients in different locations to create distinct channel conditions. Specifically, clients 1, 2, 3, and 4 are placed in the CINR range of 19-21 dB, 22-24 dB, 25-27 dB, and 28-32 dB, respectively. Figure 8(b) shows the PSNR values of each client. The Naive scheme uses the most robust MCS (i.e., MCS index 0) for delivering all video frames, hence, the PSNR values for different clients are almost equal. Similarly, the Adaptive scheme uses a single MCS for all video frames limited by the client who has the poorest channel conditions. Therefore, all clients also experience similar video quality. On the other hand, MuVi uses different MCS levels for different video frames, and hence the client's PSNR values highly depend on their channel conditions. Higher video quality can be expected with better channel condition due to different MCS employed on different video frames. The client 4, which has the best channel condition, experiences the highest video quality amongst all clients.

During the same experiments, we measure the throughput for each client and present them in Figure 7 with respect to the different resource constraints. The throughput pattern is very similar to that of PSNR (Figure 8(b)), since throughput is highly related to the video packet receptions. The more packets a client receives, the higher throughput and PSNR value are observed. The throughput results for both Naive and Adaptive schemes are linearly correlated to the resource constraints. By contrast, MuVi provides differentiated service to the clients and guaranties relatively high throughput even under very limited resource conditions (i.e., 30 slots).

5.1.2 MCS Selection

To understand the MCS usage, we present the CDF of selected MCSs for all video frames obtained from multiple runs of various topologies in Figure 9(a), where the curve MuVi30 (60, 90) represents the experiment using MuVi with 30 (60, 90) slots per WiMAX frame. MuVi uses higher MCSs than the other schemes and provides higher PSNR values as we see in Figure 8(a). The Adaptive scheme transmits 50% of frames using MCS 3 or higher while MuVi transmits 50% of frames using MCS 6 or 7. Recall that, both



(a) CDF of selected MCS for all frames (b) Average inter-packet delay (c) Ratio of packets missing a deadline

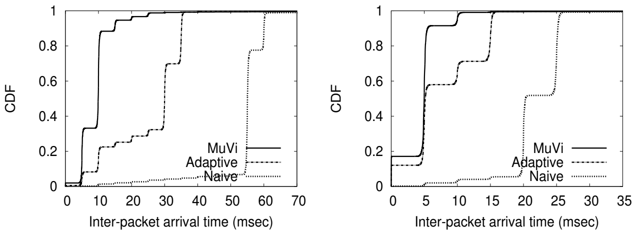
Figure 9: (a) MuVi uses higher MCS compared to the Adaptive and Naive schemes. (b) and (c) MuVi outperforms the Adaptive and the Naive schemes in terms of inter-packet delay and percentage of packets missing the deadline under various resource constraints. Especially, MuVi guarantees on-time delivery of all transmitted video packets.

MuVi and Adaptive schemes tune the MCS for frames instantaneously based on the clients' feedback of their channel conditions. The Adaptive scheme keeps the same MCS for all video frames, however MuVi assigns different MCS to each frame reflecting its system utility discussed in Section 3.3. The frames with higher utilities (e.g., I and P frames) are transmitted with lower, more reliable MCSs which can be supported by most of clients. Higher MCSs are used for the frames with lower utilities (e.g., B frames) which only target the clients in relatively good channel conditions. Some clients cannot receive all video frames due to their bad channel conditions and the higher MCSs employed with some frames, but it helps to improve the video quality for other clients with good channel conditions. This is especially beneficial when the resource constraint is very tight (e.g., 30 slots per frame). This explains why MuVi selects higher MCSs than the Adaptive and the Naive schemes on average and provides differentiated service to the clients.

The other interesting fact we notice is that MuVi aggressively uses higher MCS when there are insufficient resources for delivering all video frames (e.g., MuVi30 and MuVi60 curves in Figure 9(a)). Specifically, MuVi selects 40% of frames with MCS 7 when 90 slots are available per frame. However, it selects 60% of frames with MCS 7 when 30 slots are available per frame. The reason behind this is that when the resources are severely insufficient, MuVi tends to sacrifice some users with weak channel conditions by selecting higher MCSs and reducing the transmission time of selected video packets.

5.1.3 Inter-packet Delay

The average inter-packet delay for the three schemes with respect to various resource constraints is presented in Figure 9(b). We see that MuVi keeps the inter-packet delay low compared to the other two schemes regardless of the number of available slots. The inter-packet delay depends on the MCSs used for transmitting packets. A lower MCS leads to longer transmission time, which in turn results in larger inter-packet delay. The gap between MuVi and other schemes is larger under tight resource constraints (e.g., 30 slots), and it becomes smaller as the number of slots increases. Increasing radio resources significantly increases the number of packets packed in a frame and reduces the delay for the Naive and the Adaptive schemes. However, the improvement (reduced inter-packet delay) due to the increased resources is marginal for MuVi because MCS selection for each video frame and packet scheduling are already optimized.

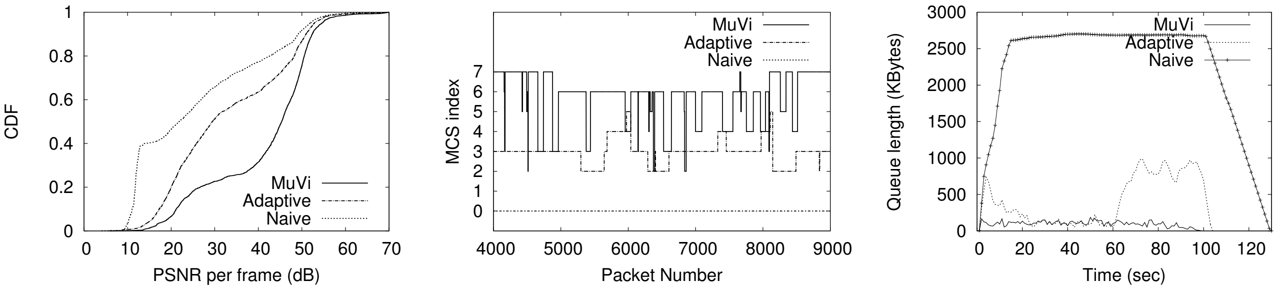


(a) 30 slots (per frame) (b) 60 slots (per frame)

Figure 10: CDF of inter-packet arrival time under different resource constraints. MuVi keeps inter-packet time close to 5 milliseconds (same as WiMAX frame interval) regardless of the available resources.

Figure 10 shows the CDF of inter-packet arrival time from the same set of experiments as shown in Figure 9(b) with respect to the different resource constraints. As we can see from Figure 10(b), for MuVi with 60 slots per frame, more than 94% of packets have inter-packet arrival time less than or equal to 5 milliseconds, which is the WiMAX frame duration. This shows that MuVi delivers most of the packets in one WiMAX frame. Even when the radio resources are insufficient (i.e., 30 slots), MuVi delivers 90% of the frames within 10 milliseconds (Figure 10(a)). On the other hand, the inter-packet delay for the Adaptive and the Naive schemes highly depends on the number of slots available. Hence, the performance of the Adaptive and the Naive schemes is limited by the amount of available resources.

Figure 9(c) shows the percentage of received packets that miss their deadlines. This pattern is very similar to the average inter-packet delay presented in Figure 9(b) because higher inter-packet delay leads to later arrival time. MuVi guarantees delivery of video frames within their deadlines, and hence all packets meet their deadlines (no packets out of deadline regardless of resource constraints).



(a) CDF of per frame PSNR (b) MCS for each packet (c) Queue length at the WiMAX BS

Figure 11: (a) MuVi's per frame PSNR is higher than the other two schemes. (b) MuVi uses higher MCS for transmitting frames. (c) MuVi keeps the queue size significantly smaller than the other two schemes.

5.2 Micro-benchmark

We have confirmed that MuVi provides the best video quality amongst the three schemes. For further understanding, we present the microscopic view of the performance of the three schemes. All results are obtained from the experiments with 60 slots for multicast traffic (we observe similar behavior for 30 and 90 slots and omit them for the sake of brevity).

5.2.1 Per frame PSNR

In Figure 11(a), we plot the CDF of per frame PSNR from a single experiment. The video frame rate is 30 frames per second and each video frame consists of multiple packets. Per frame PSNR value is determined by the reception of video packets. We pick a client in the multicast group whose CINR values are in the range of 25-27 dB and present the CDF of per frame PSNR of the selected client. And we observe similar patterns from all other runs and for clients with different channel conditions.

We can see that the per frame PSNR of MuVi is significantly higher than that of the other two schemes. The median of per frame PSNR for MuVi, Adaptive and Naive schemes is 45, 29 and 21 dB, respectively. Moreover, the PSNR value of 75% of video frames in MuVi is greater than 33 dB (considered good in terms of MOS). This leads to the higher average PSNR values as we have seen in the previous subsection.

5.2.2 Per packet MCS

In Figure 11(b), we present the MCS used for each packet from the same experiment as above. Here again, we observe similar patterns from all other experiments. MuVi aggressively selects higher MCS than the Adaptive and the Naive schemes. We also notice that the variation of chosen MCS for MuVi is higher than that of the other two schemes because MuVi adaptively changes the MCS for each video frame based on the system utility and the client's feedback. Typically, higher MCSs (64-QAM with 2/3 coding or higher) are used for transmitting B frames and lower MCSs are used for transmitting I and P frames in MuVi. The Adaptive scheme uses MCS 3 for most of the packets while adapting it for some packets based on the clients' channel conditions.

MCS used for MuVi is higher than the other schemes, but MuVi provides better video quality as presented in Figure 11(a). Although the Adaptive and the Naive schemes use lower MCSs for robust delivery regardless of client's channel conditions, this requires longer time to deliver the video packets. As a result, many packets miss their deadlines and are discarded at the client, which results in lower PSNR values. In other words, Adaptive and Naive schemes waste the given resources while focusing on robust transmission rather than resource optimization.

5.2.3 Queue Length

Figure 11(c) shows the instantaneous queue size (in KBytes) at the base-station for the three schemes. The queue length of MuVi is constantly below 100 KBytes and is significantly smaller than the other two schemes. The queue length highly depends on the MCS (i.e., the transmission rate) used for transmission. When the base-station uses higher transmission data rate, it can quickly send data and empty the queue[2]. As we see in Figure 11(b), MCS used for MuVi is higher, and hence it leads to smaller queue length than Adaptive and Naive schemes. The duration of video we used for the experiments is 100 seconds. With given resources (60 slots), MuVi can deliver all incoming video traffic in time, while Adaptive scheme completely empties its queue at around 104 seconds. Even worse, Naive scheme finishes at 125 seconds. The frames which are received late is shown as the number of frames missing their deadlines in Figure 9(c).

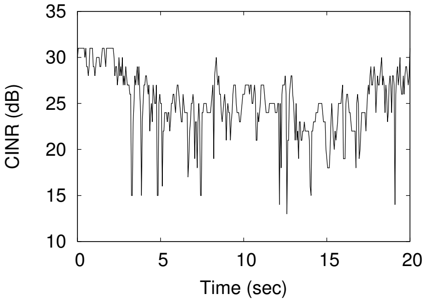
5.3 Client Mobility

We have seen the performance improvement of MuVi under static conditions where all clients remain stationary. In this subsection, we evaluate the performance of MuVi under client mobility. In this set of experiments, the multicast group contains two clients, with one stationary and the other moving at walking speed. The mobile client starts from a location close to the base-station, moves away from it, and then moves back, according to the route depicted in Figure 6.

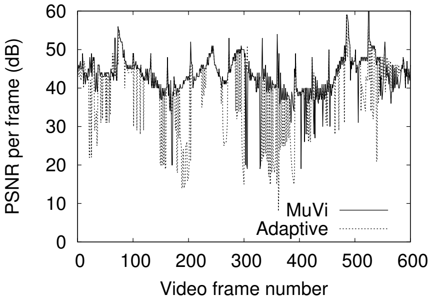
Figure 12(a) shows the observed CINR of the mobile client. We see that the CINR decreases till 15 seconds and then increases. The CINR increase between 8 and 13 seconds and multiple downside spikes are probably caused by random channel fading. Figure 12(b) plots the PSNR value of each frame for both MuVi and the Adaptive scheme under the same mobile trajectory. We see from Figure 12 that the PSNR values under both MuVi and the Adaptive scheme are highly correlated to the CINR of the mobile client, but the Adaptive scheme obtains much lower average PSNR values and

---

[2] In this work, we do not consider re-transmissions, which are typically not employed for multicast traffic to avoid feedback explosion.



(a) Instantaneous CINR of the mobile client



(b) PSNR values of first 600 frames

Figure 12: Mobility experiments. The video frame rate is 30 frames per second, so the total number of video frames generated in 20 seconds is about 600.

much higher variation than MuVi. Although MuVi obtains low PSNR values for some frames, most of them are due to dropped B frames, which do not affect other frames. On the contrary, under the Adaptive scheme, some P frames may get lost, which will affect several other dependent frames. This indicates that instantly adapting MCS based on the weakest client's feedback is not robust enough to sustain user mobility. Simultaneously optimizing resource allocations and differentiating packet types by using different MCSs can provide more robust performance under client mobility.

WiMAX base-station receives the CINR report (which is part of the WiMAX standard) from all clients periodically (e.g., every 5 msec) through the uplink channel. MuVi leverages this information to determine the highest MCS for each client. Since MuVi incorporates the client's channel conditions in real-time, it provides the best performance even for mobile clients.

5.4 Overhead

MuVi's core engine lies at the ASN gateway like Opal [15]. Typically, a single ASN gateway manages multiple base-stations in a cellular network. The overhead at the ASN gateway could be problematic considering the number of base-stations associated with it. Hence, any implementations or additional work load on the ASN gateway will require appropriate provisioning.

To investigate the computational overhead of MuVi's operations (optimizing resource allocation, determining MCS for each packets and processing packet re-ordering), we measure the execution time of MuVi while delivering video frames to the multicast group. The total execution time for MuVi's algorithm is 83.63 milliseconds while the total experimental duration (the same as the video length) is 100 seconds. It amounts to less than 0.1% of the total CPU time. Therefore, the overhead of MuVi is almost negligible for handling a single base-station and would be small even if several hundred base-stations are managed by a single gateway.

The process of handling client's CINR report for getting supportable MCS for each client will be done by referring the MCS-CINR table we summarized in Table 2. The base-station has a MCS-CINR table and hence whenever base-station receives the CINR report from the clients it can easily read the supportable MCS value accordingly. The complexity of this process is O(1). The uplink channels are used to send the client's CINR report to the base-station, it would not affect the downlink resources for video traffic to the multicast clients.

6. DISCUSSION

In this work, we consider non-scalable (e.g., MPEG4) video and differentiate video frames based on their types. MuVi's optimization algorithm is also applicable to SVC-encoded videos in a similar manner. Scalable video consists of one base layer which provides minimum video quality and multiple enhancement layers for refined video quality. To implement MuVi with scalable videos, we can prioritize frames based on their layers and apply similar optimization algorithm for efficient resource allocation. Video packets that belong to lower layers will be assigned relatively higher priority comparing to the other frames in enhancement layers. Implementing a multicast system for SVC-encoded videos in 4G networks (including both the media server and the player for clients) is an interesting topic for our future work.

MuVi employs scheduling, resource allocation and PHY rate selection for optimizing multicast video delivery. Other MAC and PHY techniques, such as packet re-transmission, power control, and beam-forming, are also possible to improve video delivery performance, which are part of our future work.

7. RELATED WORK

Video multicast in wireless (WiFi): There are many prior works on wireless multicast that aim to improve the performance through PHY and MAC layer design. DirCast [9] applies association control to minimize the total multicast delay, and within each AP (Access Point), DirCast chooses the transmission rate based on the channel condition of the "worst" client in the multicast group.

A recent work Medusa [21] employs a proxy-based solution to improve media streaming performance in wireless LAN. Medusa focuses on a content-dependent PHY rate selection and packet value awareness for WiFi multicast. However, Medusa is designed for asynchronous, WiFi systems where the wireless media is shared via contention-based random channel access. As a result, Medusa does not perform radio resource allocations and only employs a heuristic-based rate adaptation algorithm. Moreover, Medusa requires that clients periodically send reception reports about the packets transmitted previously and performs retransmissions using network coding, and thus it requires modifications at the client side, which are challenging in cellular networks (because it involves modifications of the wireless standards).

By contrast, MuVi is designed for OFDMA-based, synchronous broadband mobile wireless systems (e.g., WiMAX, LTE) where the radio resources are allocated by the base-stations. As a result, MuVi employs a near-optimal resource allocation algorithm to maximize the total system utility by intelligent resource allocation (e.g., frame dropping) and PHY rate adaptation of each video packet. MuVi does not require packet reception reports or perform re-transmissions. Instead, MuVi incorporates CINR reports that are available in current cellular networks to adjust the OFDMA resource allocation and PHY rate adaptation. Thus it does not require modification at the client-side or the air interface, and has minimum control overhead.

Scalable video multicast in wireless systems: In [13], Deb *et al.* studied the problem of multicasting scalable video (SVC) in WiMAX cellular networks with the goal of maximizing the system utility. They developed a greedy algorithm to allocate radio resources and adaptively assign the MCS for each transmitted video layer. In [16, 17], the authors use dynamic programming approach to find the optimal system utility and to assign modulation and coding schemes for scalable video traffic in mobile cellular networks. In [18], Li *et al.* considered joint-layer resource allocation to further improve multicast performance and developed approximation algorithms to trade off algorithm complexity with performance. All these works were only evaluated through theoretical analysis and simulations based on fixed layering sizes. The dynamics of video traffic across different video frames were not considered and no system implementation has been conducted.

Channel-unaware wireless video transmission: A couple of recent works proposed wireless video encoding and transmission schemes that need not be aware of the wireless channel conditions. SoftCast [14] proposed a joint channel encoding and video source coding scheme for mobile video transmission. This is essentially an analog approach for delivering video over the wireless. FlexCast [8] modified the MPEG4 video codec and incorporated rateless coding for efficient video streaming in wireless systems. Neither SoftCast nor FlexCast requires any feedback about the wireless channel conditions. The received video quality automatically adjusts depending on the channel quality at the clients. As a result, these two schemes provide natural support for wireless video multicast although they are not specifically designed for it. Nevertheless, both SoftCast and FlexCast require heavy modifications to video source coding, the air interface, and the mobile clients. By contrast, MuVi does not require any changes in these elements. MuVi requires long-term channel feedback and optimizes the video multicast transmission via efficient radio resource allocation and frame prioritization under the existing video and wireless standards. Thus it allows speedy deployment.

8. CONCLUSION

This paper presents MuVi, a wireless video multicast scheme in OFDMA-based cellular networks. We design and implement MuVi using a WiMAX testbed to show its efficacy in real systems. MuVi incorporates the clients' channel feedback and different video frame priorities to adapt the MCS for each video packet. MuVi's efficient resource allocation scheme helps to optimize the overall video quality in the multicast group. MuVi also allows providing differentiated services among users in the multicast group by assigning different utility to different users for the same packet while optimizing the overall video quality in the multicast group. MuVi does not modify the video encoding nor the wireless transmission schemes. Instead, it only involves resource allocations along with video frame prioritizations. MuVi is also applicable to other OFDMA-based wireless technologies (such as LTE and LTE-Advanced) and it does not require any modifications at the mobile clients, so it is readily deployable with commercial off-the-shelf devices.

Acknowledgments

We thank our shepherd, Dr. Thyaga Nandagopal, for his guidance through the camera-ready submission process. We thank Rajesh Mahindra from NEC Labs for invaluable support and suggestions during the preparation of this work. Jongwon Yoon and Suman Banerjee have been supported in part by the following grants of the US National Science Foundation: CNS-1040648, CNS-0916955, CNS-0855201, CNS-0747177, CNS-1064944, and CNS-1059306.

9. REFERENCES

[1] YouTube, *http://www.youtube.com/*.

[2] Hulu, *http://www.hulu.com/*.

[3] Cisco Visual Networking Index: Forecast and Methodology, 2010-2015, *http://www.cisco.com*.

[4] Beceem wimax chips for mobile applications, *http://www.beceem.com*.

[5] IEEE 802.16e-2005 Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, *IEEE 802.16e standard*.

[6] MPEG4 standard group, *http://mpeg.chiariglione.org/standards/mpeg-4/mpeg-4.htm*.

[7] VLC media player, *http://www.videolan.org/vlc*.

[8] S. T. Aditya and S. Katti, "FlexCast: Graceful Wireless Video Streaming", *ACM MobiCom*, 2011.

[9] R. Chandra, S. Kandula, T. Moscibroda, V. Navda, J. Padhye, R. Ramjee and L. Ravindrananth, "DirCast: A Practical and Efficient Wi-Fi Multicast System", *IEEE ICNP*, 2009.

[10] Accton, http://www.accton.com/.

[11] PicoChip femtocell solutions, http://www.picochip.com/.

[12] J. M. Costa, "More Frequencies Needed for Mobiles - Terrestrial Spectrum Sought for IMT", *ITU News, No3*, 2007.

[13] S. Deb, S. Jaiswal and K. Nagaraj, "Real-time Video Multicast in WiMAX Networks", *IEEE Infocom*, 2008.

[14] S. Jakubczak and D. Katabi, "A Cross-Layer Design for Scalable Mobile Video", *ACM MoiCom*, 2011.

[15] R. Kokku, R. Mahindra, S. Rangarajan and H. Zhang, "Opportunistic Alignment of Advertisement Delivery with Cellular Basestation Overloads", *ACM MobiSys*, 2011.

[16] P. Li, H. Zhang, B. Zhao and S. Rangarajan, "Scalable Video Multicast in Multi-carrier Wireless Data Systems", *IEEE ICNP*, Oct. 2009.

[17] P. Li, H. Zhang, B. Zhao and S. Rangarajan, "Fast Optimal Resource Allocation for Scalable Video Multicast in Broadband Wireless Networks", *IEEE ICC*, 2010.

[18] P. Li, H. Zhang, B. Zhao and S. Rangarajan, "Scalable Video Multicast with Joint Layer Resource Allocation in Broadband Wireless Networks", *IEEE ICNP*, 2010.

[19] R. Morris and E. Kohler and J. Jannotti and M. F. Kaashoek, "The click module router", *SIGOPS Operating System Rev.*, volume 33(5), pages 217-231, 1999.

[20] Z. Orlov, "Network-driven Adaptive Video Streaming in Wireless Environments", *PIMRC*, 2008.

[21] S. Sen, N. K. Madabhushi and S. Banerjee, "Scalable WiFi Media Delivery Through Adaptive Broadcasts", *USENIX NSDI*, 2010.