OEFMON: An Open Evaluation Framework for Multimedia over Networks

Chunho Lee, Myungchul Kim, Soon J. Hyun, and Sooyong Lee, Korean Advanced Institute of Science and Technology

Ben Lee, Oregon State University

Kyounghee Lee, Electronics and Telecommunications Research Institute

ABSTRACT

As various types of networks are proliferating with greater capacity, the demand for multimedia services, such as videoconferencing, VoIP, and IPTV, over underlying networks has been increasing. This demand has increased the need for an evaluation framework to analyze the quality of such multimedia services and network performances for the related research areas. This article proposes OEFMON, an open framework for evaluating the quality of multimedia transmissions over networks. OEFMON integrates a multimedia module and a network simulator. The major advantages of OEFMON compared to other existing work are support for any number of multimedia codecs, adaptive coding based on network feedback, on-the-fly observation of multimedia transmission, and easy extension for other tools. A couple of case studies are performed to demonstrate the applicability and usefulness of our framework.

INTRODUCTION

Multimedia devices such as smartphones, portable media players, and tablet/pad computers (e.g., iPad) have become widespread, and the demand for multimedia services for these devices has increased. In addition, as broadband networks and wireless LANs (WLANs) made it possible to widely deploy multimedia services, various applications such as voice over Internet Protocol (VoIP), IP multimedia communications (IPMC), video streaming, and videoconferencing have become prevalent. However, due to the characteristics of multimedia and networks, multimedia transmission over a network is very challenging for a number of reasons. First, multimedia transmission is time-sensitive: packets that arrive at a receiver after their playout time are considered lost. Second, frequent packet losses significantly degrade the perceptual quality of multimedia. Even a single lost packet can cause performance degradation on multiple elements since conventional multimedia coding algorithms reference contiguous elements to remove redundancies.

In order to mitigate the aforementioned problems, various research activities have been performed on reliable multimedia transmission over networks. Typically, network performance metrics, such as end-to-end delay, delay jitter, packet loss ratio, and bit error rate (BER), are used to evaluate multimedia transmissions. However, these metrics neglect user-perceived quality. Therefore, a framework that can simultaneously evaluate multimedia transmission over networks and resulting user-perceived quality is crucial since both are closely related.

In general, new ideas on multimedia transmission over networks have been proven either through simulations or on testbeds. Testbed experiments give more realistic results than simulations. However, huge cost and effort are required to configure and manage the hardware and network topology. In contrast, simulation studies have a number of advantages over testbeds in terms of cost, scalability, flexibility, and rapid development. In particular, simulations make it possible to easily and effectively evaluate multimedia transmission over network protocols, such as ad hoc networks, wireless mesh networks, and networks with cross-layer design, under various network conditions.

Despite the existence of multimedia frameworks, such as DirectShow [1] and Gstreamer [2], and network simulators, such as NS-2 [3] and QualNet [4], very little work exists on combining both capabilities into a single framework. The work closest to ours is EvalVid-RA [5]. However, EvalVid-RA has some disadvantages that stem from its predecessor, EvalVid [6], including limited codec support, frame-level-only video coding that does not support slice coding, trace-file-based simulation that only allows full-reference video quality measurement, and low extensibility and usability due to its trace-file-based structure and complex procedures. The

The two major components of OEFMON are the Multimedia Module and the Network Simulator. Multimedia source and configuration information are sent to the Multimedia Module, and network configuration and quality of service mapping information are passed to the Network Simulator.

most up-to-date version of EvalVid supports MPEG-4, H.263, and H.264 codecs, but the latest codecs, such as VC-1 and ON2, are not supported.

This article proposes the Open Evaluation Framework for Multimedia Over Networks (OEFMON) for assessing both measurable and perceptual qualities of multimedia transmission over networks. There are several novel features in OEFMON. First, a multimedia module and a network simulator are integrated to easily support various types of codecs and networks, respectively. Second, the network performance information is fed back to the multimedia module in real time, which allows the multimedia module to apply this information into its coding algorithms. Third, transmitted video, network performance, and video quality can all be observed at the same time during simulation. These features allow researchers and practitioners to better understand the intricate relationship between perceived video quality and network performance. Fourth, the proposed framework can be used to perform research on no-reference video quality measurement, which evaluates the quality of on-the-fly video without comparing it to the original raw video. Finally, OEFMON is easily extensible due to its modularized structure.

OEFMON

Figure 1 shows the structure of OEFMON, where blue and gray boxes represent newly implemented and third-party components, respectively. The two major components of OEFMON are the *Multimedia Module* and the *Network Simulator*. Multimedia source and configuration information are sent to the Multimedia Module, and network configuration and quality of service (QoS) mapping information are passed to the Network Simulator.

Before simulation, the Network Simulator references a scenario file, and builds a network topology and a network scenario. Then the Network Simulator executes the Multimedia Module. During simulation, the sender in the Network Simulator receives coded data from the encoder in the Multimedia Module and then generates and sends packets to the receiver in accordance with the QoS mapping information. The receiver passes data to the Multimedia Module where it is decoded. Based on this, the sent and received multimedia data can be displayed on the fly during simulation. After simulation, the network performance metrics can be gathered from log files and the video quality metrics, such as peak signal-to-noise ratio (PSNR), are measured using both the original and received raw videos.

OEFMON Components

The following provides a detailed explanation of each component.

Multimedia Module — The Multimedia Module was implemented using DirectShow, and consists of a video encoder filter and a video decoder filter. A *filter* in DirectShow represents one stage of multimedia data processing. Thus, a complex multimedia task (e.g., video encoder) can be represented by creating instances of the required filters and connecting them as a *filter graph*.

The Multimedia Module also includes a video source filter and a video writer filter to read and write raw videos, respectively. The video source filter reads data from a raw video file and sends it to the video encoder filter. On the other hand, the raw video writer filter writes decoded frame data to a raw video file.

Network Simulator — The Network Simulator was implemented in QualNet, and consists of sender and receiver agents. The sender agent receives compressed videos from the video encoder filter via QualNet Connector, converts them into Real-Time Transport Protocol (RTP) packets, and transmits them to the receiver agent. If the sender agent receives feedback from the receiver agent over the network, this information is sent to the video encoder filter via DirectShow Connector to adjust the video coding.

The receiver agent depacketizes the received RTP packets into compressed video and sends them to the video decoder filter via QualNet Connector to decompress the video.

In order to integrate QualNet with other components of OEFMON, the sender agent, receiver agent, and user interface of the Network Simulator were implemented. Sender and receiver agents were implemented as applications in QualNet. QualNet uses scenario files that define network topology configurations and traffic scenarios. Since QualNet supports IEEE 802.11, 802.16, and other networks, scenarios can be configured for various network types, such as mobile ad hoc and wireless mesh networks.

Framework Inputs — OEFMON receives several input files. The input files for the Multimedia Module are raw video and DirectShow Graph generated using DirectShow GraphEdit. The input files for the Network Simulator are QualNet Scenario produced by QualNetUI and QoS Mapping Parameter provided by an XML editor.

GraphEdit is a utility program for visually building and simulating filter graphs in DirectShow. QualNetUI provides a configurable user interface for building network topologies and scenarios required by QualNet. QualNetUI was modified to allow users to configure scenario files for QualNet and DirectShow, such as playout rate, playout buffer size, and the file paths of DirectShow Connector and DirectShow Graph. This graphical user interface provides a more convenient way to set parameters than the command line user interface in EvalVid.

The QoS Mapping Parameter files specify the rules for mapping multimedia units into network priorities. The header information of the files indicates RTP with payload type (e.g., H.264). This is followed by data partitions delimited by payload tags, which represent the mapping method for the units. QualNet uses precedence value on each packet as a network priority. Thus, the precedence tag is a QoS parameter value in

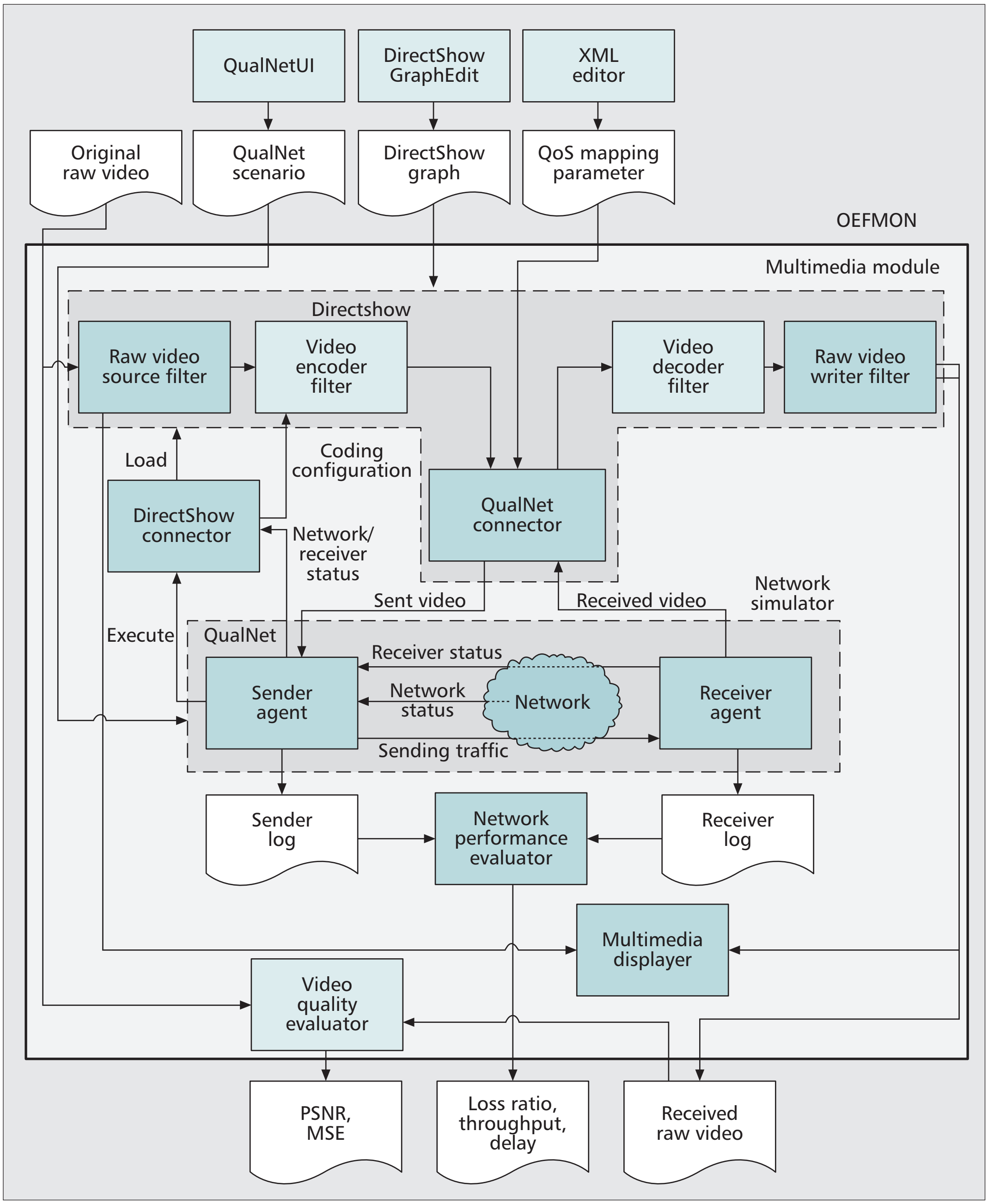

Figure 1. *The structure of OEFMON.*

QualNetUI was modified to allow users to configure scenario files for QualNet and DirectShow, such as playout rate, playout buffer size, and the file paths of DirectShow Connector and DirectShow Graph.

QualNet, and the value tag infers the type of each unit. For example, precedence values 7 and 3 mean the highest and second highest access categories (AC3 and AC2), respectively, in IEEE 802.11e.

QualNet Connector — The QualNet Connector was implemented as a DirectShow filter to integrate the Multimedia Module and Network Simulator. Since QualNet and DirectShow do not have functionalities for RTP and QoS mapping, the QualNet Connector is responsible for packetizing data from the video encoder filter using RTP and forwarding them to the sender agent based on the QoS mapping parameters. The QualNet Connector depacketizes video packets received from the receiver agent, and then sends them to the video decoder filter.

The QualNet Connector consists of three internal modules. The Packet Manager packetizes/depacketizes video data to/from an RTP stream. The Jitter Buffer collects packets from the receiver agent in QualNet, and then composes the packets into multimedia frames and sends them to the video decoder filter. The Inter-Process Communications (IPC) Manager is a set of communication mechanisms for exchanging data among different components. This module provides tight coupling between QualNet and DirectShow. The IPC Manager module uses Named PIPE as an IPC mechanism due to its efficiency and support for blocking I/O operation. This is done by creating Named PIPE severs in both the sender and receiver agents, and having the QualNet Connector establish connections to these servers. Then the QualNet Con-

Perceived video quality can be reduced due to not only packet losses but also network delay jitter, which causes video frames to be displayed either before or after the fixed playout time. Therefore, a playout buffer is used at the receiver side to compensate for packet delay variation.

nector can send and receive compressed videos to and from the sender and receiver agents, respectively, through the Named PIPEs.

DirectShow Connector — The responsibility of the DirectShow Connector is to load the DirectShow library, and then build filter graphs using DirectShow graph files. In addition, it feeds back the Network/Receiver status from the network into multimedia coding. This is done by utilizing the COM interfaces exposed by encoder and decoder filters, which allows these filters to be controlled externally. The DirectShow Connector adapts coding configuration, such as video rate and frame rate obtained from Network/Receiver Status, into video encoder and decoder filters by using their COM interfaces. Furthermore, the DirectShow Connector allows users to preview videos being sent and received on the Multimedia Displayer.

Video Quality Evaluator — To evaluate user-perceived video quality, both subjective and objective measurements are used. Subjective video quality measurements are considered unsuitable since they rely mainly on human vision. As an objective video quality measurement, PSNR has been commonly used. In general, PSNR values below 25 dB are considered unacceptable to the human visual system.

The MSU Video Quality Measurement Tool [7] is a third party program for video quality measurement, and supports various video quality metrics such as mean square error (MSE), PSNR, and video quality metric (VQM). This tool is used in OEFMON to compare received videos with original raw videos.

Network Performance Evaluator — The Network Performance Evaluator was implemented to analyze log files generated by the sender and receiver agents, and gives results such as end-to-end delay, packet loss rate, and frame loss rate.

Design and Implementation Issues

Handling Time Synchronization — Time synchronization problems can occur since simulation time is different from actual time, and the required execution time for multimedia coding varies depending on the system. To resolve this issue, the sender agent in QualNet is continuously suspended until the next coded data is received from the QualNet Connector by using a blocking I/O operation in Named PIPE.

Playout Buffer — Perceived video quality can be reduced due to not only packet losses but also network delay jitter, which causes video frames to be displayed either before or after the fixed playout time. Therefore, a playout buffer is used at the receiver side to compensate for packet delay variation. For simplicity, OEFMON supports fixed-size buffers that can be set by the user using QualNetUI. The buffer size is sent to QualNet Connector using the COM interface.

Handling Missing Frames — Some video frames may not be available for decoding due to packet loss and delay. However, full-reference video quality measurement compares the original and received video files in a frame-by-frame manner. Therefore, missing frames should be compensated with pre-decoded frames. For the purpose of compensating missing frames in OEFMON, the Video Writer Filter handles a missing frame by checking the timestamp of each media sample and determining whether or not a previous media sample is missing. If the timestamp gap between the current and previous media samples is larger than expected, the Video Writer Filter writes a pre-decoded frame for video quality measurement.

Case Studies

This section demonstrates the features of OEFMON using two case studies to better understand the interplay between multimedia data and networks, and its effect on network and multimedia performance. The first case study shows the effectiveness of *network-feedback-based video coding*, where quantization parameter (QP) values are dynamically changed based on network conditions. The second case study shows that OEFMON can easily be extended to support the *data partitioning* feature of H.264/AVC implemented as a simple DirectShow filter. Moreover, unlike other prior studies, both case studies provide on-the-fly observations of video transmissions. Although the case studies presented in this section consider only wireless networks and PSNR values, OEFMON is an extensible framework that can support any type of network and video quality measurement.

Case Study 1: Rate Adaptive H.264 Video Transmission over IEEE 802.11a

The quality of video transmission will degrade when the required bit rate is higher than the available bandwidth. In order to mitigate network congestion and achieve better quality, a video encoder can adjust its bit rate depending on the current network situation. Joint work on video rate control and available bandwidth estimation has been discussed in [8]. However, to the best of our knowledge, research on developing a framework that can fully support video rate control has not been sufficiently studied. For example, EvalVid-RA only supports rate adaptive video transmission on a per Group of Pictures (GOP) basis with a limited choice of codecs. In contrast, OEFMON provides this support on a per-frame basis and for a variety for codecs. Therefore, this case study demonstrates the effectiveness of OEFMON for using network performance feedback information for video transmission.

Simulation Environment — OEFMON adjusts bit rate based on available bandwidth estimated by the Wireless Bandwidth (WBest) estimation tool [9]. The simulated network topology represents an infrastructure mode WLAN consisting of one access point, three stations (i.e., clients), a background traffic server, and a video server.

The IEEE 802.11 distributed coordination function (DCF) is used as the medium access control (MAC) layer protocol, and the channel capacity is 6 Mb/s. The simulation is based on

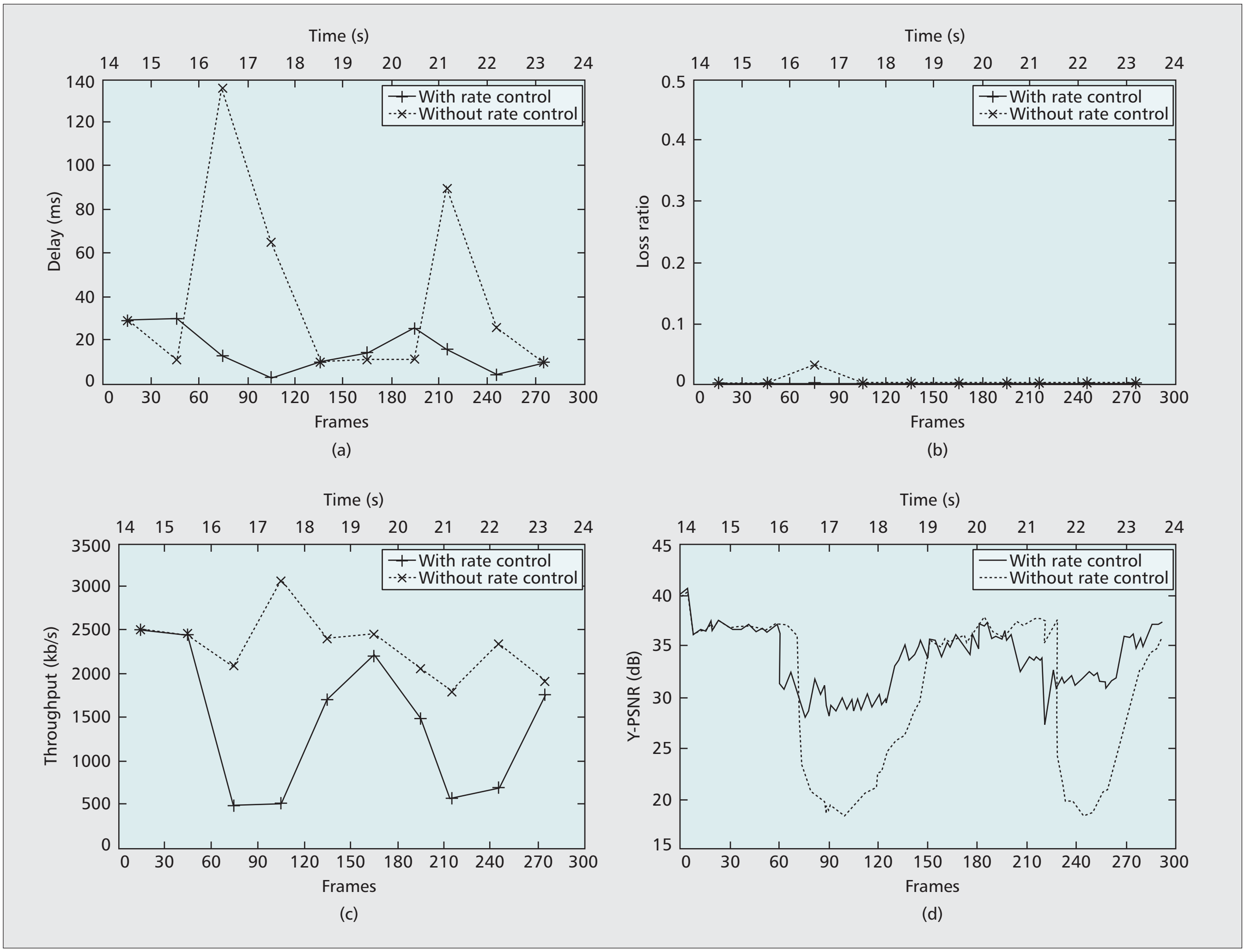

Figure 2. *Case study 1 performance: a) end-to-end delay; b) loss ratio; c) throughput; d) PSNR.*

the MONOGRAM x264 encoder, which is based on H.264/AVC, and a video clip *Crew* that consists of 300 frames in 4CIF resolution.

The MONOGRAM x264 encoder is extended to prevent frame drops during frame-by-frame rate control and generates logs on QP variation. The rate control mechanism used in the current MONOGRAM x264 encoder causes frame losses because frames that are already encoded are thrown out when QP is changed. Therefore, the mechanism to apply a new video bit rate was modified to prevent the encoder from throwing out encoded frames.

After the simulation starts, the medium is idle for 14 s, and then the video traffic begins. At 15 and 16 s, two 1.7 Mb/s streams are generated as background traffic. From 19 to 21 s, 1.2 Mb/s background traffic is added every second. The WBest server estimates the available bandwidth every 1 s, and this result is taken into account during video encoding. The quality of the rate adaptive video transmission is then compared with the normal video transmission bit rate of 2.3 Mb/s.

Results — Figure 2a shows that end-to-end delay significantly increases without rate control due to background traffic from 16 to 18 s and from 21 to 22 s. In contrast, end-to-end delay remains relatively constant when rate control is applied. Figure 2b shows that there are no lost packets with rate control but some packets are lost without rate control. Figure 2c indicates that throughput decreases and video bit rate decreases with rate control due to severe network congestion caused by background traffic. However, without rate control, throughput is almost constant. Figure 2d shows that user-perceived video quality significantly degrades without rate control when background traffic is applied. In contrast, applying rate control results in only minor video quality degradation. Note that adjusting QP with respect to network congestion in general improves perceived video quality, but still results in minor PSNR degradation.

Figure 3 shows that the quality of frames with rate control is much better than without it and is consistent with the PSNR results shown in Fig 2d.

CASE STUDY 2: DATA PARTITIONED H.264/AVC VIDEO TRANSMISSION OVER IEEE 802.11E

This case study demonstrates simulation of a cross-layer design for H.264/AVC data partitioned video transmission in IEEE 802.11e. Our

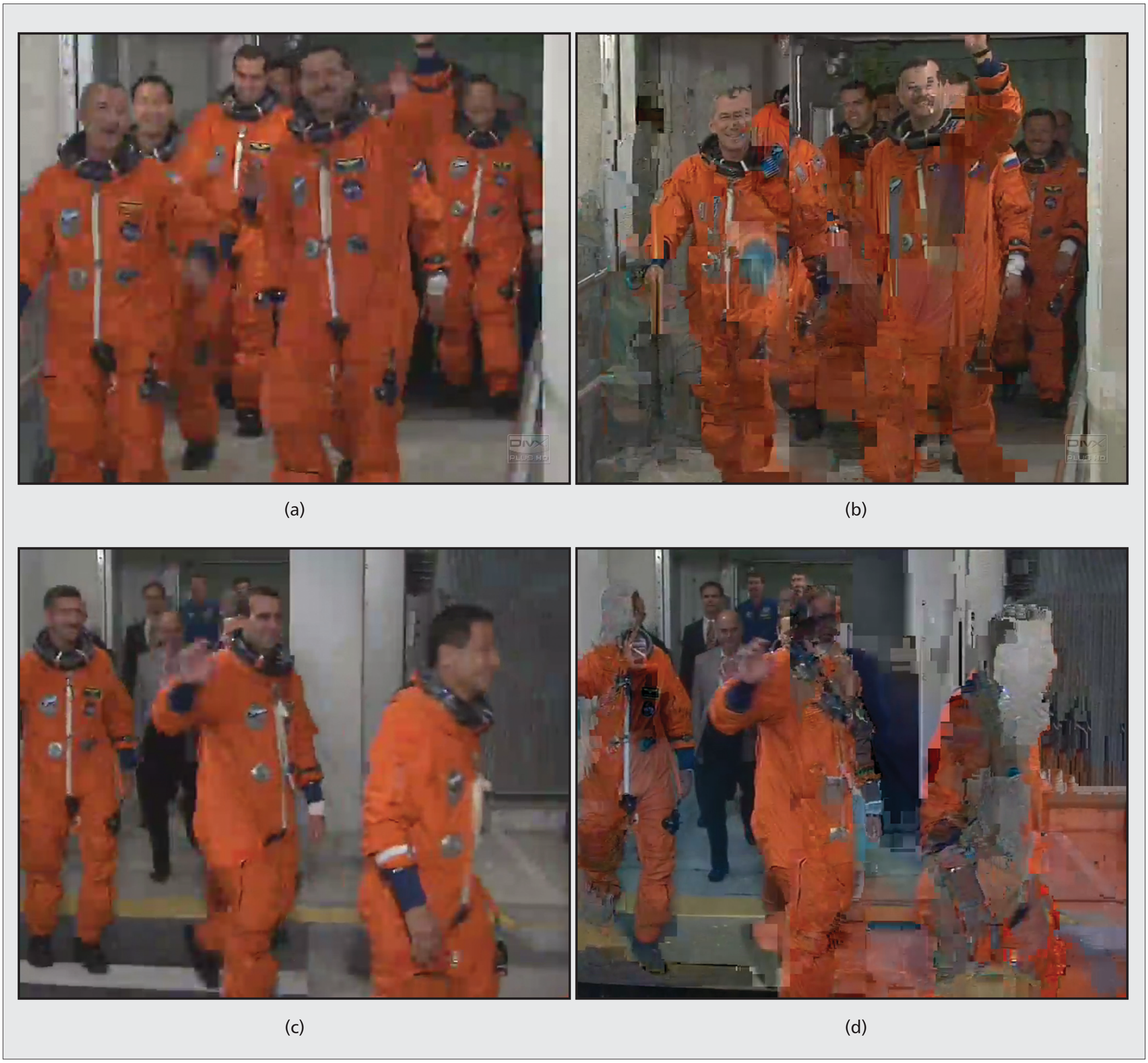

Figure 3. *100th and 250th frames with and without rate control: a) 100th frame with rate control; b) 100th frame without rate control; (c) 250th frame with rate control; d) 250th frame without rate control.*

experiment with a QoS architecture, where some units in a video sequence are given higher priority, results in better user-perceived video quality than existing methods even when quantitative network performance measures are very similar.

An H.264/AVC video sequence consists of coded pictures, where each coded picture represents a single frame or a field, and a parameter set that contains header information for the sequence. The beginning of an H.264/AVC video sequence is an Instantaneous Decoding Refresh (IDR) picture, which contains an intrapicture that does not reference other pictures but is referenced by other pictures. In addition, coded pictures cannot be decoded without the coding information contained in the parameter set. Therefore, IDR and parameter set information are considered more important than others. Moreover, some parts of a coded picture are more valuable than other parts. For example, data partitioning, which is an error resilience feature in H.264/AVC, allows a coded picture to be separated into up to three data partitions (A, B, and C) according to their importance. Thus, data partition A contains slice header information that includes QP and motion vectors. Data partition B contains intra-slice macroblocks. Data partition C includes intercoded macroblocks. Data partition A is considered the most important because data partitions B and C cannot be decoded without it. Data partition B is more important than data partition C since it prevents error propagation.

In IEEE 802.11 DCF, all stations compete equally for the medium, and thus there is no QoS consideration for multimedia services. IEEE 802.11e was introduced as an amendment to IEEE 802.11 for QoS support with a new channel access method, called Enhanced Distributed Channel Access (EDCA). EDCA achieves QoS provisioning by providing access categories (ACs) for service differentiation. EDCA defines four ACs: AC3, AC2, AC1, and AC0, which are designed for VoIP, video, best-effort, and background traffic services, respectively. Each AC has its own queue, backoff entity, and AC-specific parameters called *EDCA Parameter Set*, which consists of *Arbitration Inter-Frame Space* (AIFS), and *Contention Window (*CW) *minimum* and *maximum*. Channel access delay of each AC is determined based on the EDCA parameter set and backoff entity. There have been a number of efforts to develop QoS mapping schemes for IEEE 802.11e and H.264/AVC error resilience tools [10, 11]. A cross-layer architecture based on a QoS mapping scheme for data partitioning and EDCA was proposed in [10], where the parameter set, data partition A, and IDR are considered the most important information and have higher priority than data partitions B and C. However, to the best of our knowledge, existing frameworks for evaluating video transmission do not support the H.264 error resilience features. In contrast, OEFMON supports error-resilient tools for H.264/AVC as well as QoS cross-layer mapping schemes.

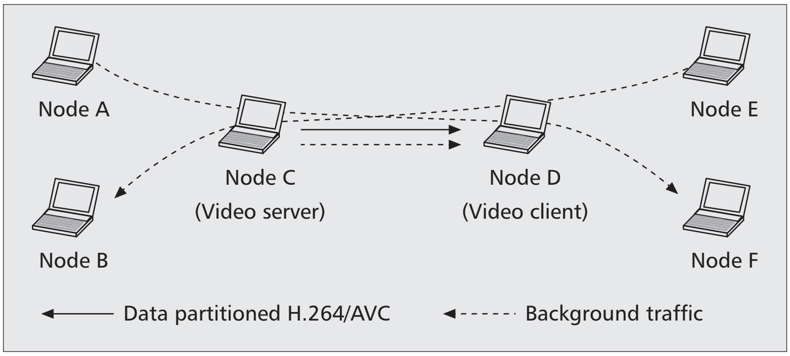

Figure 4. *Network topology and traffic for case study 2.*

	IDR	PS	DP A	DP B	DP C
Original file	7	4	293	293	286
Scheme 1	7	4	283	277	108
Scheme 2	2	4	283	278	221
Scheme 3	1	4	254	275	41

Table 1. *Number of received IDR slices, parameter set (PS), and data partitions (DPs).*

Simulation Environment — The simulated network topology is shown in Fig 4, which is similar to the environment presented in [10]. The ad hoc network consists of six nodes equipped with IEEE 802.11a radios with bit rate of 6 Mb/s. The distance between nodes is 300 m, and each node has a transmission range of 350 m and a carrier sense range of 700 m. Node C is a server that transmits video to node D at 4 s. Also, node C generates 600 kb/s constant bit-rate (CBR) background traffic at 4.5 s. Nodes A and E generate 600 kb/s CBR background traffic at 5 and 5.5 s, respectively. Transmitted video contains 300 frames of the CIF Foreman clip encoded as data partitioned H.264/AVC video using the JM encoder [12].

Three QoS mapping schemes were compared. Scheme 1 mimics the QoS architecture proposed in [10], where parameter sets (PSs), such as sequence parameter set and picture parameter set, are mapped to AC3, IDR and data partition (DP) A are mapped to AC2, data partitions B and C are mapped to AC1, and background traffic is mapped to AC0. In scheme 2, all the H.264/AVC data and background traffic are mapped to AC2 and AC0, respectively. Finally, scheme 3 represents the default DCF where all the traffic is mapped to AC0.

Results — Table 1 shows the number of received IDR slices, parameter set, and data partitions for the three schemes.

The total number of received IDR slices and data partitions for schemes 1 and 2 are higher than for scheme 3. This is to be expected since video packets in schemes 1 and 2 are mapped to AC2 or AC3 and thus transmitted with higher priority than video packets in scheme 3, which are transmitted with equal priority as background traffic. Therefore, the probabilities that packets are lost in schemes 1 and 2 are lower than in scheme 3. Moreover, the number of received IDR slices for scheme 1 is higher, but the number of received data partitions B and C is lower compared to scheme 2. This is because all the video packets in scheme 2 have the same priority as IDR slices and data partition A in scheme 1, which are mapped to AC2 and thus have more chances for transmission. However, it is difficult to determine user-perceived video quality based only on this information.

Figure 5 shows the network performance for the three schemes. Based on the results shown in Figs. 5a–5c, scheme 1 provides the best overall network performance. However, both schemes 1 and 2 have similar performance between frames 160 and 250.

In contrast to network performance results, scheme 1 gives the best user-perceived quality during the entire simulation, as shown in Fig. 5d. This is interesting since there is no correlation between the user-perceived quality shown in Fig. 5d and the network performance metrics shown in Figs. 5a–5c. After around the 170th frame, there is hardly any difference between schemes 1 and 2 in terms of network performance metrics. However, Fig. 5d shows that the PSNR values of scheme 2 are much lower than those of scheme 1 after around 170th frame. This is because most of the lost packets in scheme 2 are IDR slices, which significantly affect user-perceived video quality. In comparison, Table 1 shows that mainly data partitions C are lost in scheme 1, which are less important in terms of user-perceived quality. This can be verified from the on-the-fly observations of the 100th and 200th decoded frames for the three schemes. The visual quality of the 100th frame is similar for both schemes 1 and 2, while scheme 3 results in very poor quali-

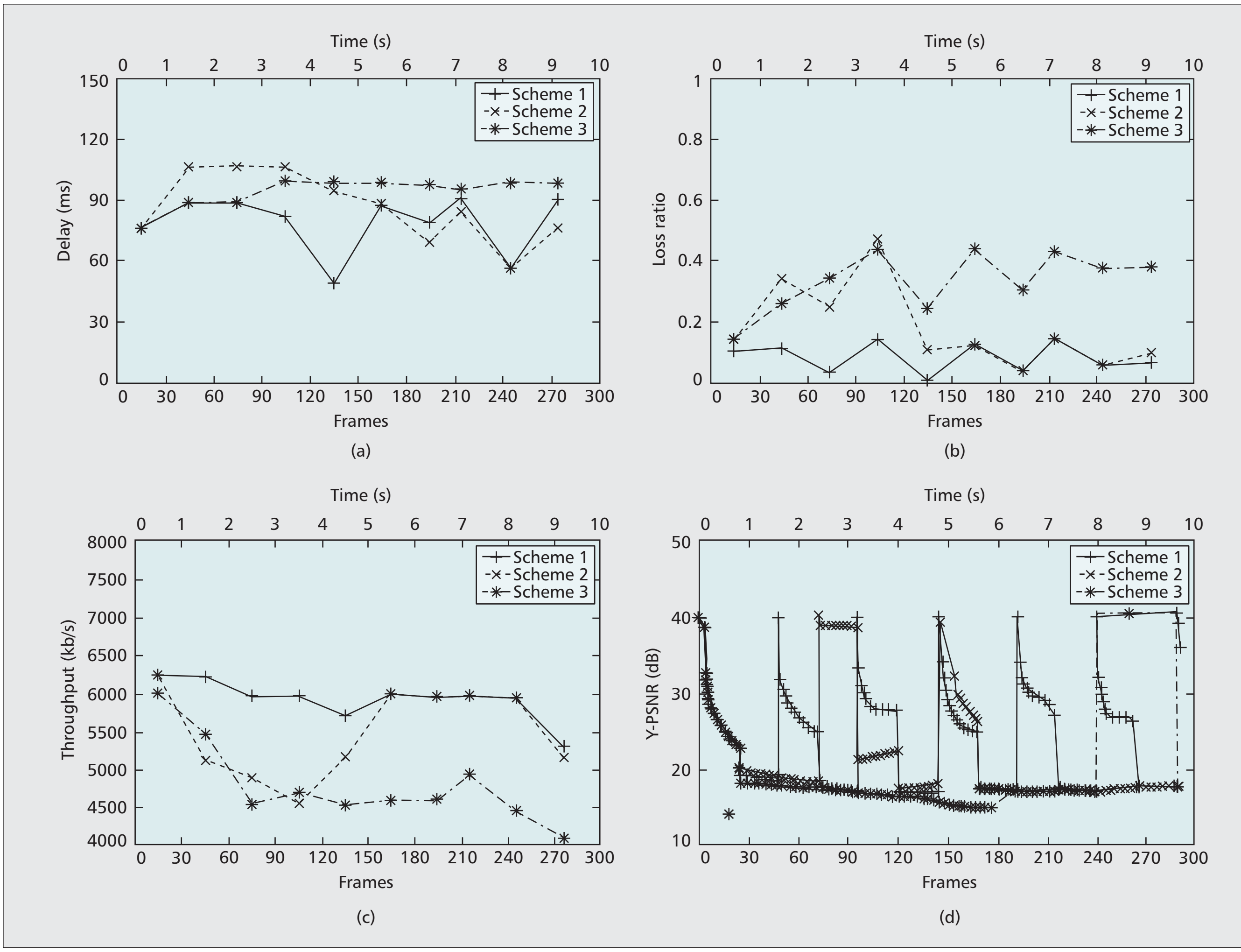

Figure 5. *Case study 2 performance: a) end-to-end delay; b) loss ratio; c) throughput; d) PNSR.*

ty. However, the visual quality of the 200th frame for both schemes 2 and 3 is significantly lower than for scheme 1. These results show that giving higher priority to IDR slices and data partition A leads to better video quality even though network performance may not reflect this.

Conclusion

This article proposes OEFMON, which is an evaluation framework for multimedia over networks. OEFMON integrates a multimedia module, a network simulator, and evaluation tools in order to simultaneously evaluate multimedia transmission over networks and resulting user-perceived quality. OEFMON overcomes some limitations of existing work by providing several novel features. These include support for various codecs, ability to adapt network performance information into multimedia coding in real time, preview of transmitted videos, no-reference video quality measurements, and easy extensibility to add new features.

As a future work, OEFMON will be extended to support multiview and scalable video coding, and to investigate their transmission behavior and QoS with various networks. In order to encourage research on multimedia over networks using OEFMON, we have made it freely available at http://winslab.kaist.ac.kr/oefmon.

References

[1] M. D. Pesce, *Programming Microsoft DirectShow for Digital Video and Television*, Microsoft Press, 2003.
[2] Gstreamer, http://www.gstreamer.net/.
[3] NS2, http://www.isi.edu/nsnam/ns/.
[4] QualNet, http://www.qualnet.com/.
[5] A. Lie and J. Klaue, "Evalvid-RA: Trace Driven Simulation of Rate Adaptive MPEG-4 VBR Video," *Multimedia Systems*, vol. 14, no. 1, 2008, pp. 33–50.
[6] J. Klaue, B. Rathke, and A. Wolisz, "EvalVid — A Framework for Video Transmission and Quality Evaluation," *Proc. 13th Int'l. Conf. Modeling Techniques and Tools for Comp. Performance Eval.*, 2003, pp. 255–72.
[7] D. Vatolin *et al.*, "MSU Video Quality Measurement Tool," 2009.
[8] M. Chen and A. Zakhor, "Rate Control for Streaming Video Over Wireless," *IEEE Wireless Commun.*, vol. 12, no. 4, 2005, pp. 32–41.
[9] M. Li, M. Claypool, and R. E. Kinicki, "WBest: A Bandwidth Estimation Tool for IEEE 802.11 Wireless Networks," *Proc. 33rd Annual IEEE Conf. Local Computer Networks*, 2008, pp. 374–81.
[10] A. Ksentini, M. Naimi, and A. Gueroui, "Toward an Improvement of H.264 Video Transmission over IEEE 802.11e Through A Cross-Layer Architecture," *IEEE Commun. Mag.*, vol. 44, no. 1, 2006, pp. 107–14.
[11] R. MacKenzie, D. Hands, and T. O'Farrell, "Effectiveness of H.264 Error Resilience Techniques in 802.11e WLANs," *Proc. IEEE WCNC*, 2009, pp. 2774–79.
[12] JM Reference Software, http://iphome.hhi.de/suehring/tml/

BIOGRAPHIES

CHUNHO LEE (handera@kaist.ac.kr) received his B.A. in computer engineering from Chungbuk National University in 2009, and his M.S. in computer science from the Korea Advanced Institute of Science and Technology (KAIST) in 2011. He is currently an assistant engineer in the DMC R&D Center at Samsung Electronics. His research interests include multimedia streaming in wireless and mobile Internet.

MYUNGCHUL KIM (mck@kaist.ac.kr) received his B.A. in electronics engineering from Ajou University, Korea, in 1982, his M.S. in computer science from KAIST in 1984, and his Ph.D. in Computer Science from the University of British Columbia, Canada, in 1993. Currently, he is with the faculty of KAIST as a professor. His research interests include Internet, mobile Internet, protocol engineering, mobile computing, and wireless mesh networks.

BEN LEE (benl@eecs.orst.edu) received his B.E. degree in 1984 from the State University of New York at Stony Brook, and his Ph.D. degree in computer engineering in 1991 from Pennsylvania State University. He is currently with the School of Electrical Engineering and Computer Science at Oregon State University. He has published over 80 conference proceedings, book chapters, and journal articles in the areas of embedded systems, computer architecture, multithreading and thread-level speculation, and wireless networks.

SOON J. HYUN (sjhyun@kaist.ac.kr) received his M.E. degree in electrical engineering from the Katholike Universitate Leuven (KUL), Belgium, and his Ph.D. degree in computer science and engineering from the University of Florida in 1987 and 1995, respectively. Presently, he is a professor in the Computer Science Department, KAIST. His current research interests include temporal data management, context data management for context-aware ubiquitous computing, and sensor network databases.

KYOUNGHEE LEE (leekhe@etri.re.kr) received his M.S. and Ph.D. degrees in computer science from KAIST in 2000 and 2006, respectively. In 2006 he joined Electronics and Telecommunications Research Institute (ETRI) as a senior researcher. His research interests include IP mobility, wireless network, Internet QoS, and multimedia service. He has made some contributions in ITU-T standardization as an editor of several Recommendations in the next-generation network area.

SOOYONG LEE (torshong@kaist.ac.kr) received his B.S. degree in information and computer engineering from Ajou University in 2005, and his M.S. in computer science from KAIST in 2007. Currently, he is a Ph.D. candidate at KAIST. His research interests include QoS signaling protocol in mobile Internet, multimedia streaming in wireless and mobile Internet, and routing protocols in wireless mesh networks.

As future work, OEFMON will be extended to support Multiview Video Coding and Scalable Video Coding and to investigate their transmission behavior and QoS with various networks.