Dynamic Multimedia Scheduling against Motion based DoS Attacks

Euijin Choo †1, Heejo Lee †2, Wan Yeon Lee ‡

†*Korea University*
1chksj@korea.ac.kr
2heejo@korea.ac.kr

‡*Hallym University*
wanlee@hallym.ac.kr

Abstract— **Many motions lead to drastic degradation of QoS in multimedia communications. However, previous scheduling schemes do not consider the effect of motions in video streaming. Here we highlight the importance of handling motions and introduce a new kind of attack manipulating the amount of motions, called *Motion based DoS* (MDoS) attacks. And, we propose a dynamic multimedia scheduling scheme against MDoS attacks, called *Scheduling Multimedia against MDoS* attacks (SMaM). SMaM dynamically schedules video streams adaptive to the current network conditions. Specifically, SMaM increases resilience against MDoS attacks coping with motion traffic. Experimental results show that SMaM maximizes video quality regardless of the change in the amount of motions. Especially, SMaM minimizes the amount of useless frames for dynamically changing video, e.g. 45 % improvement over conventional scheduling schemes, which we call preferential packet scheduling (PPS). Also, SMaM presents low and stable average delay, which shows 16 % improvement over PPS schemes.**

I. INTRODUCTION

Denial of Service (DoS) attacks are the most significant issue in modern network services. Under DoS attacks, legitimate packets will be dropped because of congestion control [1]. Due to the characteristics of real-time videos such as high data rate, time-constraint and components having different importance, even few amount of loss can pose a devastating effect on the video quality [2]. Accordingly, multimedia communications are more susceptible to DoS attacks than traditional networks. Especially, many motions give rise to high degradation of QoS or even a large discontinuity in video communications [3]. Therefore, deriving many motions can be regarded as DoS attacks in video communications. We term *Motion-based DoS* (MDoS) attacks which degrade video quality by injecting malicious motion traffic to video streams.

In this paper, we propose a dynamic multimedia scheduling scheme against MDoS attacks, called *Scheduling Multimedia Against MDoS*(SMaM). In order to prevent MDoS technically and maximize video quality, we must be prepared for many motions in a normal condition. This is due to the fact that network traffic induced by the MDoS attacks are not much different from those which are possible under a normal condition of dynamically changing videos. Hence, we will focus on MDoS attack prevention with the concept of congestion control, especially caused by large amount of motions, which can mitigate the impact of MDoS attacks. Since the routers typically do not actively provide congestion control, end-to-end congestion control is more recommended for networked multimedia transmission [1]. Along with this observation, we employed end-to-end buffer management policy.

The main contribution of SMaM lies in introducing a new type of DoS attacks manipulating the amount of motions and maximizing QoS adaptive to the amount of motions. Since data loss in a static video which has few changes between subsequent frames has a limited effect, the effect of data loss on the video communication depends on the amount of motions within the video [4]. Therefore, SMaM dynamically control multiple queues with consideration of the amount of motions. Experimental results show that SMaM can present reliable video services regardless of the change in network condition, especially change in the amount of motions.

The remaining of this paper is organized as follows. In Section 2, we will discuss the importance of handling motions in video applications and introduce a new type of DoS attacks using excessive motion traffic. Section 3 discusses scheduling requirements for video streaming and presents related works. In Section 4, we propose a dynamic multimedia scheduling scheme, SMaM. Section 5 analyzes SMaM with the experimental results. Finally, Section 6 concludes the paper.

II. MOTION BASED DOS ATTACKS

In this section, we show that many motions can pose significant inefficiencies in video communications and introduce a new type of DoS attacks using malicious motion traffic.

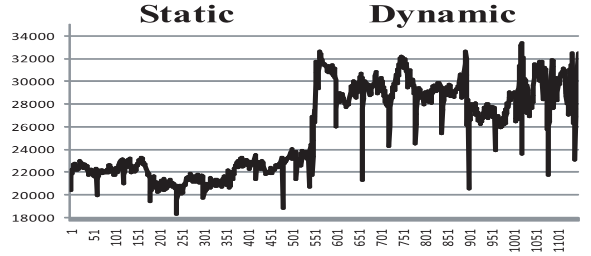


Fig. 1. The traffic trace of a video stream with the change in the amount of motions

Fig. 1 represents the traffic trace of a video stream with the change in the amount of motions. *Static* implies that when a video has few motions and *dynamic* means that when a video has many motions. As shown in Fig. 1, the average traffic volume of *dynamic* is higher, e.g. 40 % increment over *static*. Also, many peaks occur in *dynamic* while traffic rates of *static* maintain within a certain range. The range of traffic variations in *dynamic* are very wide e.g. 4 times wider than that in *static* as shown in Fig. 1. When there are more peaks in traffic, i.e., more packets in the burst, it takes a longer time to transmit all packets relating to a video frame [5]. This results in dropping packets due to playback deadline [1]. Packet loss leads to drastic degradation of QoS or even a large discontinuity when streaming videos. Hence, deriving excessive motions can be regarded as DoS attacks in video communications since it has similar effect to DoS attacks on video quality.

MDoS (Motion based DoS) attacks which we term continuously degrade QoS by exploiting vulnerabilities of video communications. Video streams can be attacked by flooding attacks as well as injecting motion traffic into streams. For example, attackers generate motion traffic with hijacked frames and inject the malicious motion traffic which seems normal traffic to video streams. The purpose of MDoS attacks is not to necessarily cripple a service, but rather to inflict significant degradation of service quality. This kind of attacks sharply contrasts to traditional DDoS attacks because MDoS attacks highly degrade QoS without overwhelming amount of packets. Moreover, attack traffic is similar to normal traffic of a dynamic video. Since users have little knowledge of original scenes in realtime systems such as surveillance systems, users may not even recognize whether they are currently suffered from attacks in which users received unintended scenes. This is because scenes for which MDoS attacks are generated are not greatly different from expected ones, while DDoS attacks wreck havoc on services resulting in black screen. As a result, attackers can significantly reduce video quality, while evading detection by consuming an unsuspicious network capacity. Since network traffic induced by the MDoS attacks are not much different from those which are possible under a normal condition of dynamically changing videos, we propose a dynamic scheduling scheme which mitigates the impact of MDoS attacks by handling motion traffic and adapting to network condition. In next section, we discuss scheduling requirements for providing reliable services in video communications and present related works.

III. SCHEDULING MULTIMEDIA STREAM

The representative technique for controlling congestion in video communications is to carry out buffer management [2]. A buffer management attempts to minimize the possibility of network congestion by dropping packets. Due to the characteristics of video streams, a buffer management for scheduling video streams should be specifically designed. The video scheduling requirements are as follows.

- R_1. **Minimize the amount of frame loss and loss impact**: Generally, much packet loss leads to degrade QoS. However, there is no linear relationship between QoS and the amount of frame loss in video communications since frames in a video have different priorities. In video streams, the I, P and B frames have priorities in descending order. Also, the earlier packet has a higher priority in the same priority level [1]. Hence, we must consider the loss impact of different prioritized streams. The *loss impact* implies the amount of dropped frames and useless frames due to priority.
- R_2. **Distribute frame loss uniformly and minimize GoP impact**: Much loss of consecutive frames causes skipping many motions, a large discontinuity or even entirely different scenes from the original [2]. Hence, frame loss should be uniformly distributed for each GoP [6]. Furthermore, frame loss to protect new frame must not affect another GoP, since damaging another scene instead of current scene is not desirable. Therefore, we should consider GoP impact when distributing frame loss for each GoP. *GoP Impact* implies whether a dropping target affects another GoP or not.
- R_3. **Maintain low and stable delay**: Real-time video requires low and stable delay so that packets are displayed in time [2]. Previous studies have shown that user can perceive the delay less than 100 ms [7]. For maintaining low delay, a queue must have the smallest available size and reflect the size of bursts [2].
- R_4. **Adapt to the amount of motions**: As the amount of motions increases, the slope of QoS degradation increases [3]. Moreover, the effect of data loss on video communications relies on the amount of motions within the video. Even though the amount of loss is equal in two video streams, loss in a dynamic video leads to more serious consequences on quality because there are many changes between even a few subsequent frames in dynamic videos [4]. Hence, video streams should be scheduled adaptively relying on the amount of motions.

We briefly classify previous buffer management schemes into three categories based on the packet dropping strategies under network congestion.

- **General Scheduling (GS)**: The traditional approach is the bufferfull scheduling which drops packets when queue becomes full such as “taildrop” [2]. In order to solve “Full Queue” problem in bufferfull scheduling [2], active queue management(AQM) such as RED and its many variants drops packets before queue becomes full [2]. While AQM achieves better performance in terms of packet loss, AQM can not always minimize the loss of video quality, since scheduling requirements are not considered [2].
- **Frame Priority Based Scheduling(FPBS)**: In order to consider interdependency of each frame, FPBS uses static priority labels for I, P and B frames and randomly drops frames with low priority [1]. However, FPBS did not consider loss impact, reducing the video quality [1].
- **Preferential Packet Scheduling(PPS)**: PPS drops less important packets of video streams, such as frame in the

later position of a GoP [1] and enhancement layer packet in scalable-coded video [2].

These previous approaches did not guarantee all scheduling requirements - minimization of frame loss and loss impact (R_1), uniformity of loss distribution (R_2), low and stable delay (R_3) and adaptiveness to the amount of motions (R_4) which are discussed above. To be specific, GS only addressed minimization of packet loss. Even though FPBS discussed R_1 and R_3 and PPS discussed R_1 R_3, they were not able to achieve balance between requirements. Moreover, GS, FPBS and PPS did not take R_4 into account. This paper presents a dynamic scheduling scheme which serves all requirements in balanced manner. Especially, the proposed scheme alleviates the impact of MDoS attacks with consideration of R_4.

IV. SMaM: Dynamic Scheduling Scheme of Multimedia Stream against MDoS attacks

This section describes the proposed scheduling scheme called *Scheduling Multimedia Against MDoS*(SMaM). Since attack traffic is not much different from normal traffic of a dynamic video, SMaM manages MDoS attacks with the concept of congestion control, especially caused by large amount of motions. For presenting reliable video services continuously, SMaM periodically analyzes current traffic and schedules video streams deliberating scheduling requirements. By analyzing current traffic, SMaM checks current traffic volume and its variations which could represent changes in amount of motions. Depending on the analyzed traffic, SMaM dynamically changes queue spaces that makes the best use of empty spaces. Under congestion, SMaM selects drop targets in the range of all GoPs with loss impact, uniformity of loss distribution and GoP impact. This is available in limiting the amount of frame loss within each GoP and overall frame loss and minimizing the effect of loss.

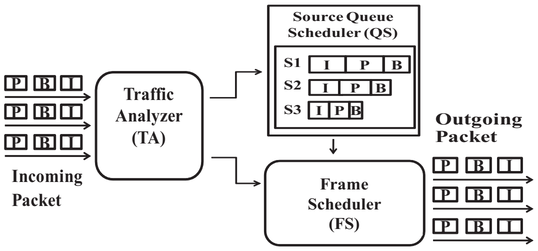


Fig. 2. Functional components of SMaM

Fig. 2 represents three functional components of SMaM, i.e., *traffic analyzer*(TA), *source queue scheduler*(QS), and *frame scheduler*(FS). First, TA filters out malformed video traffic or bypasses non-video traffic. And, TA periodically observes traffic volumes and their variations. SMaM decides whether network is congested or not with traffic rates. With analysis of range fluctuations between traffic volumes, SMaM watches the changes in the amount of motions by periods. The large range of variations between traffic volumes during periods implies many motions and the small range of variation means few motions.

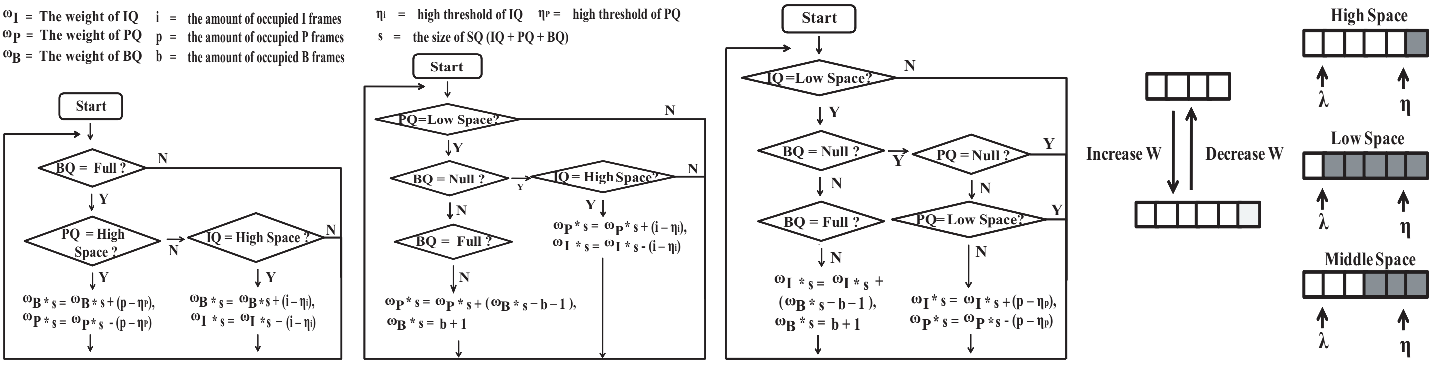
SMaM has multiple queues for different sources and different priority frames which are controlled by QS and FS, respectively. Depending on periodically analyzed traffic, QS decides the weight w_S for each multiple source and dynamically allocates a queue with w_S. A large w_S is assigned to the source with continuously many motions or burst traffic. After the source queue(SQ)'s are assigned to each different source, QS decides the weight of I, P, and B, i.e., w_I, w_P and w_B ($w_I + w_P + w_B = 1$). w_I, w_P and w_B are decided with the frame priority and defined amount of frames in a GoP. Since B frame has no effects on the other frame type, w_B is the smallest. Even though I frame has the highest priority, the amount of P frames is larger than the amount of I frames. Therefore, initially we set w_I and w_P are same. After setting w_I, w_P and w_B, SQ is partitioned into I frame queue(IQ), P frame queue(PQ) and B frame queue(BQ) with w_I, w_P and w_B.

FS drops the low priority frame or forwards the high priority frame while dynamically changing w_I, w_P and w_B. Changing w_I, w_P and w_B is available in controlling queue spaces effectively and minimizing frame loss. Increment of weight implies extension of a queue space and decrement of weight means reduction of a space. According to the empty queue space, we divide the state of each queue as shown in Fig. 3(d). BQ has three states: $Null$ (w_B = 0), $Full$ and $Notfull$. PQ has four states: $Null$ (w_P = 0), $Lowspace$ (PQ empty space $< \lambda_P$), $Highspace$ (PQ empty space $> \eta_P$), and $Middlespace$ ($\lambda_P <$ PQ empty space $< \eta_P$). And, IQ has three states: $Lowspace$ (IQ empty space $< \lambda_I$), $Highspace$ (IQ empty space $> \eta_I$), and $Middlespace$ ($\lambda_I <$ IQ empty space $< \eta_I$). Along with the change of each queue state as shown in Fig. 3(a), Fig. 3(b), and Fig. 3(c), FS schedules an incoming frame.

A queue extends when other queues have enough spaces. Since B frames have the lowest priority, w_B increases only when IQ or PQ is $Highspace$ as shown in Fig. 3(a). As shown in Fig. 3(b), w_P increases when IQ is $Highspace$ or BQ is $Notfull$. Similarly, w_I increases when BQ is $Notfull$ or PQ is not $Lowspace$ as shown in Fig. 3(c).

FS schedules incoming frames depends on each queue state. If BQ is $Notfull$, the incoming B frame B_{new} is inserted to BQ. FS drops B_{new} when BQ is $Full$, PQ is not $Highspace$ and IQ is not $Highspace$. The incoming P frame P_{new} is inserted to PQ, if PQ is not $Lowspace$. Otherwise, P_{new} is scheduled with frame priority, loss impact and GoP impact. Since P frames are more important than B frames, FS drops the last B frame first, decreases w_B, increases w_P and enqueue P_{new}. When BQ is $Null$ and IQ is not $Highspace$, FS drops P frames selected by following rules.

- If P_{new} has an allowable loss impact ($< \delta_P$) in a GoP or P_{new} has the smallest loss impact among all P frames in PQ, drop P_{new}.
- If there is P frame whose loss impact is less than δ_P in



(a) B queue state (b) P queue state (c) I queue state (d) Conceptual queue state

Fig. 3. The change of queue state

a GoP, drop the P frame.

- Otherwise, drop P_{new}.

The incoming I frame I_{new} is inserted to IQ, if IQ is not $Lowspace$. If both PQ and BQ are $Null$, FS drops I_{new}. Otherwise, FS schedules I_{new} with frame priority, loss impact and GoP impact similar to the P frame scheduling. Since B frame has the smallest priority, FS drops the last B frame first, decreases w_B, increases w_I and enqueue I_{new}. If BQ is $Null$, FS drops P frames selected by following rules, decreases w_P, increases w_I and enqueue I_{new}.

- If the loss impact of the latest P frame P_l is less than δ_P, drop P_l.
- Find the latest P frame whose loss impact is less than δ_P in a GoP, drop the P frame.
- Otherwise, drop the P frame who has the smallest loss impact.

V. EXPERIMENTAL RESULTS AND ANALYSIS

In order to evaluate the performance of SMaM, we implemented a network model with Pentium4 processor PCs that operate on Windows XP. For the purpose of comparing the effect of motions in each scheduling scheme and minimizing the effect of different network environment, we used a client-server model. We tested two video streams, “Lab” which is a recorded video in a laboratory and “The war of flowers” which is a part of movie for comparing each scheme with the amount of motions in video. “Lab” has few motions, i.e., a static video and “The war of flowers” has many motions, i.e., a dynamic video. The video sequences are represented in 640 × 480 pixels with a frame rate of 30 fps. We tested each video during streaming 15 GoPs which consist of 450 frames. In the following subsections, we analyze SMaM in terms of four scheduling requirements R_1, R_2, R_3 and R_4 comparing with the existing schemes classified in Section 3.

5.1. Frame loss and loss impact

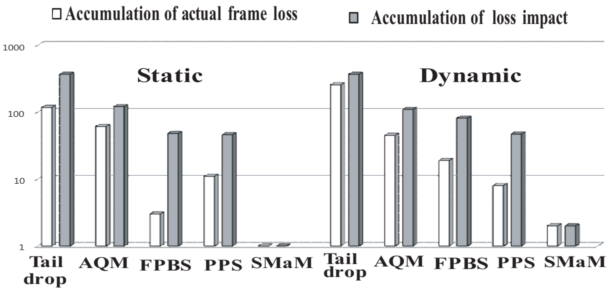


Fig. 4. Accumulation of frame loss and loss impact

Fig. 4 represents frame loss and loss impact while streaming two videos including different amount of motions. As shown in Fig. 4, SMaM outperforms other scheduling schemes in terms of R_1 and R_4. First, SMaM minimizes frame loss under congestion by controlling empty queue spaces effectively. Second, SMaM minimizes the amount of useless frames. In static video, there are few frame loss and useless frame. Also, SMaM minimizes loss impact, e.g. 45 % improvement over PPS in dynamic video. Since FPBS and PPS did not consider a loss distribution, they often drop the high priority packet under congestion, resulting in the increase of useless frames and drastically degraded QoS. Even under congestion, SMaM limits the amount of frame loss within each GoP and minimizes overall frame loss because SMaM takes into account the uniformity of loss distribution. Finally, the amount of motions does not affect frame loss in SMaM. In other schemes, the difference between actual frame loss and frame loss with useless frames increases, as the amount of motions increases. Since SMaM operates with motion traffic analysis, change in the amount of motions does not affect the scheduling. By motion-adaptive scheduling, SMaM present reliable video services even under streaming dynamic videos or MDoS attacks.

5.2. Deviation

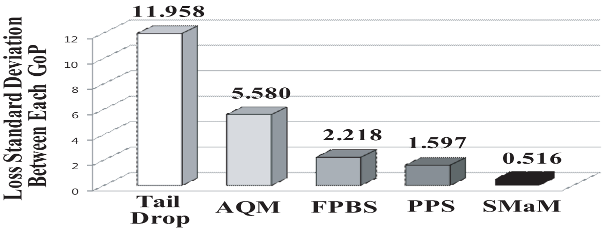
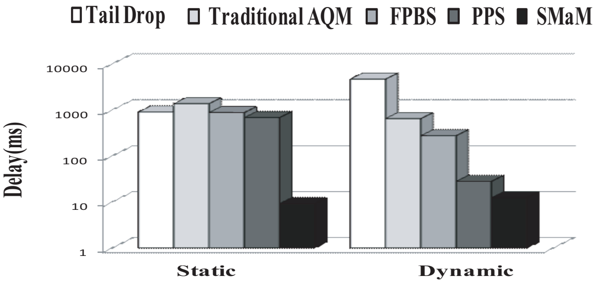


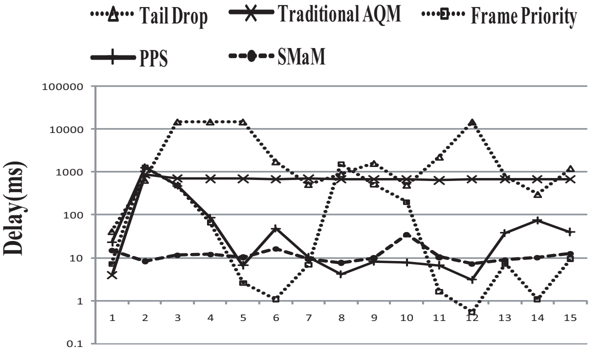
Fig. 5. Standard deviation of frame loss between each GoP

In order to evaluate the uniformity of loss distribution with considering GoP impact in each scheme, we compare the standard deviation of frame loss between each GoP which is shown in Fig. 5. As shown in Fig. 5, SMaM outperforms other scheduling schemes in terms of R_2. The standard deviation of SMaM is close to zero. It implies that frame loss is uniformly distributed for each GoP in SMaM.

5.3. Delay



(a) Average delay of static and dynamic video



(b) Delay of dynamic video during 15 GoPs

Fig. 6. Delay between sending streams and displaying

Fig. 6 represents the delay between server and client, i.e., the time from sending streams to displaying. By dropping many packets, FPBS and PPS minimizes delay under congestion. However, dropping many packets leads to considerably low QoS. As shown in Fig. 6, SMaM outperforms other scheduling schemes in terms of R_3 and R_4. SMaM presents a reliable video streaming with low and stable delay regardless of the amount of motions. Especially in dynamic video, SMaM shows 16 % improvement over PPS after dropping frames regardless of the amount of motions. This is because SMaM minimizes each queue size while a queue absorbs the size of bursts by a dynamic queue management.

VI. CONCLUSION

This paper presents a dynamic multimedia scheduling scheme, SMaM. Since many motions in a video stream lead to drastic degradation of QoS, we emphasized the importance of handling motions in video communications and introduced a new type of DoS attacks manipulating the amount of motions, called MDoS (Motion based DoS) attacks. In order to prevent MDoS technically and maximize video quality, we must be prepared for a normal condition of dynamic videos. Hence, we have focused on mitigating the impact of MDoS attacks in video applications with multiple queues controlled by two queue schedulers coping with motion traffic. Experimental results show that SMaM can present reliable video services regardless of changes in the amount of motions.

ACKNOWLEDGMENT

This work was supported in part by the ITRC program of the Korea Ministry of Information & Communications, and the Basic Research Program of the Korea Science & Engineering Foundation

REFERENCES

[1] H. Luo and M. Shyu, “Differentiated service protection of multimedia transmission via detection of traffic anomalies,” in *IEEE Int’l Conf. on Multimedia and Expo*, 2007.

[2] Y. Bai and M. R. Ito, “Application-aware buffer management: new metrics and techniques,” *IEEE Trans. on Broadcasting*, vol. 51, pp. 114–121, Mar. 2005.

[3] H. Fitzek, B. Can, R. Prasad, and M. Katz, “Traffic analysis and video quality evaluation of multiple description coded video services for fourth generation wireless ip networks,” *Wireless Personal Communications*, vol. 35, Oct. 2005.

[4] Y. Lin, A. Kim, E. Gurses, and A. Perkis, “Rate-distortion optimized i-slice selection for low delay video transmission,” in *IEEE 9th Wkshp. on Multimedia Signal Processing*. IEEE, Oct. 2007.

[5] N. Cranley and M. Davis, “An experimental investigation of ieee 802.11e TXOP facility for real-time video streaming,” in *IEEE Global Telecommunications Conf.* IEEE, Nov. 2007.

[6] W. Wen, H. Hsiao, and J. Yu, “Dynamic FEC-distortion optimization for H.264 scalable video streaming,” in *IEEE 9th Wkshp. on Multimedia Signal Processing*, Oct. 2007, pp. 147–150.

[7] J. Oh, J. Hwang, and Y. Han, “A packet-by-packet scheduling algorithm for wireless multimedia systems,” in *IEEE 66th Vehicular Technology Conf.* IEEE, Oct 2007.