RADAR

www.brookes.ac.uk/go/radar

OXFORD BROOKES UNIVERSITY

Directorate of **Learning Resources**

Research Article

A Selective Downlink Scheduling Algorithm to Enhance Quality of VOD Services for WAVE Networks

Shumao Ou,[1] Kun Yang,[1] Hsiao-Hwa Chen,[2] and Alex Galis[3]

[1] *Department of Computing and Electronic Systems, University of Essex, Colchester CO4 3SQ, UK*
[2] *Department of Engineering Science, National Cheng Kung University, Tainan City 70101, Taiwan*
[3] *Department of Electronic and Electrical Engineering, University College London (UCL), London WC1E 7JE, UK*

Correspondence should be addressed to Hsiao-Hwa Chen, hshwchen@ieee.org

Providing quality-of-service- (QoS-) guaranteed video on demand (VOD) services over wireless access in vehicular environments (WAVEs) is a challenge as WAVE adopts enhanced distributive channel access (EDCA), a contention-based channel access mechanism, for air interface access control. This paper proposes a selective downlink scheduling (SDS) algorithm to enhance the quality of VOD for WAVE networks. According to the importance of video decoding, video packets are categorized into high and low priorities. The categorized packets are put into different queues in roadside units (RSUs) to contend for transmission opportunities. Aiming to improve video playback quality and reduce video playback delay, the proposed SDS algorithm schedules video packets based on their importance, playback deadline, and their real-time parameters of receiving onboard units (OBUs), such as velocity and remaining dwelling time. The effectiveness of SDS algorithm is verified by simulations.

1. Introduction

Wireless access in vehicular environments (WAVEs) [1] is a next generation intelligent transportation technology that aims to improve transportation environments. The physical layer in WAVE is defined in IEEE 802.11 [2] and amended by IEEE 802.11p [3]. The spectrum of WAVE is allocated at the range of 5.86 ~ 5.92 GHz, and it is divided into seven channels each of 10 MHz bandwidth. Apart from improving transportation environment safety and intelligent management, the WAVE systems also provide a convenient means to provision data exchange services, such as video on demand (VOD), when people are traveling in vehicles. However, some specific features of WAVE networks, such as fast movement of vehicles and topology dynamic changes, render the provisioning of VOD services as a challenge. In this paper, we endeavor to tackle this problem by proposing a selective downlink scheduling (SDS) algorithm for VOD services in WAVE networks.

Communications in WAVE systems are conducted in two modes: vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. There are two types of devices in WAVE: roadside unit (RSU) and onboard unit (OBU). Both RSU and OBU support at least one control channel (CCH) and multiple service channels (SCHs). An RSU is installed in a fixed position along roadside to support communication with OBUs wirelessly in its cell coverage. An OBU is a mobile device, normally fitted in a vehicle, to support communications with RSUs and other OBUs. Figure 1 illustrates a WAVE network that is adopted in this paper, where RSUs have fixed links to the Internet, and OBUs moving in both directions communicate with RSUs in a V2I mode. A VOD server is connected to the Internet. The OBUs request VOD services through the RSUs and video streams data flows from the VOD server to the requested OBUs (e.g., OBUs 1, 3, 5, 4, and 7 as shown in Figure 1).

Compressed video streams are able to tolerate small amount of packet loss without degrading the quality of video playback very much, since some packets are less important for video decoding [4]. This feature can be used for schedule algorithms to deliver important packets and discard unimportant packets when traffic is heavy, or a wireless channel is unreliable. In this paper, the proposed SDS algorithm is designed to improve video playback quality
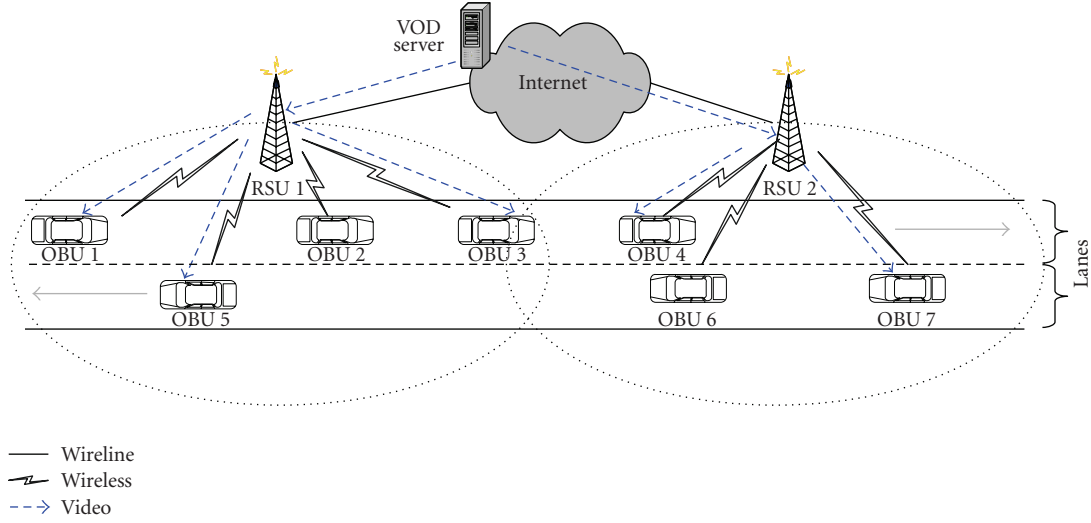
FIGURE 1: A WAVE network for VOD services.

and reduce video playback delay, especially when traffic load is heavy.

The existing works related to the issues concerned in this paper can be classified into two categories. The first category focuses on traffic scheduling of contention-based wireless networks, such as [5–9]. The second category concerns the issues on video streaming in wireless networks [10–13]. The WAVE standard does not specify particular scheduling algorithms for its media access control (MAC). The conventional first-come-first-service (FCFS) algorithm adopted by [6] is still recommended to be used in WAVE in coordination with IEEE 802.11e enhanced distributed channel access (EDCA) technology [1, 14]. The earliest deadline first (EDF) algorithm is widely used in traffic scheduling on wireless channels for multimedia [7, 9]. EDF can improve the QoS of multimedia services and overall system utilization when the deadline of each packet in the traffic is known. In [5], Chang et al. proposed a maximum freedom last (MFL) scheduling algorithm specifically for dedicated short-range communication (DSRC) networks. MFL is an enhancement of EDF which achieves a low-handoff rate by taking DSRC network parameters, such as cell size and vehicle dwelling time, into account.

Video streaming over IEEE 802.11 networks attracted much research interest. The authors of [11–13] proposed mechanisms to transport H.264 and MPEG-4 over IEEE 802.11e/a networks. However, their mechanisms are based on hybrid coordination function controlled channel access (HCCA) [14]. HCCA is a polling-based channel access mechanism which is different from contention-based EDCA used in WAVE. Furthermore, these mechanisms are only suitable for scalable video streams. Ksentini et al. [10] proposed a cross-layer architecture for H.264 over IEEE 802.11e. They utilized a specific feature of H.264 data partitioning (defined in the extended profile of H.264 standard). In their work, the bitstreaming of a frame is classified into three data partitions with different importance levels. The data partitions are delivered by different access categories (ACs)

in IEEE 802.11e MAC. However, they only dealt with the different importance of data partitions within a frame, and they did not distinguish the importance between different video frame-types.

Inspired by EDF, MFL, and aforementioned video streaming schemes over wireless mechanisms, the SDS algorithm proposed in this paper aims to improve video playback quality and reduce video playback delay in WAVE networks. The SDS algorithm schedules video packets based on their importance, playback deadline, and their real-time parameters of receiving OBUs, such as velocity and remaining dwelling time. The SDS algorithm selectively drops unimportant video packets when it is not possible to transmit all packets due to limited dwelling time, heavy load, or undesired channel conditions. This selective packet dropping feature is of benefit to graceful quality degradation. In the SDS algorithm, video packets destined to different OBUs are coordinately scheduled by their playback deadline. This feature can be used to reduce the video playback delay of all OBUs. The proposed mechanism in this paper is not designed for only a specific video codec scheme (such as H.264), but it is also applicable to other popular video compression formats, such as H.263, MPEG-2.

The rest of the paper is outlined as follows. Section 2 presents some preliminaries, and Section 3 discusses the system model. In which the system architecture and system operations are discussed. The proposed scheduling algorithm is presented in Section 4. In Section 5, the scheduling algorithm is evaluated by simulations, and performance of the algorithms is analyzed. Finally, Section 6 concludes the paper.

2. Preliminaries

2.1. Video Frame and Packet. In video compression technologies, such as MEPG and H.264 [4], encoded pictures (or frames) are arranged in groups of pictures (GOPs). An encoded video stream consists of successive GOPs. A GOP
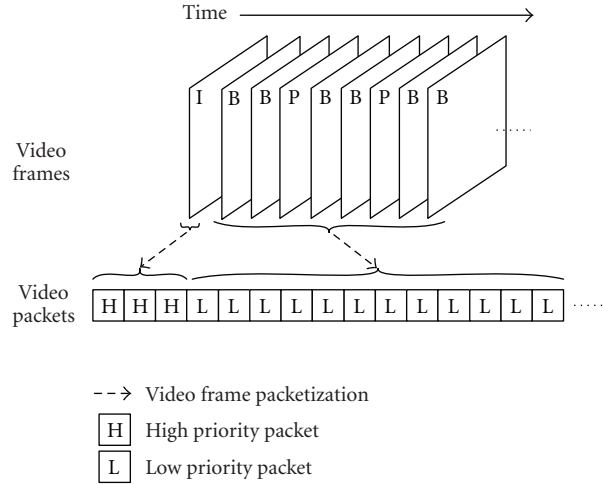
FIGURE 2: Video frame and packet.

can contain the following frame types: I-frame, P-frame, and B-frame. The order of intraframes and interframes is specified in a GOP. An I-frame is a reference picture which is intracoded corresponding to a fixed image and it is independent of other pictures. A P-frame is predictive-coded frame which contains motion-compensated difference information from the preceding I- or P-frame. A B-frame is bidirectionally predictive-coded frame which contains different information from the preceding and following I- or P-frame within a GOP. I-frame and P-frame are often referred to as anchor frames. A GOP always begins with an I-frame. Afterwards, several P-frames follow. The B-frames are inserted between two consecutive anchor frames.

Since I-frames contain all the information for image reconstruction without the need of referring to other frames, the more I-frames a video stream has, the more resilient it is for transmission over wireless links. However, having more I-frames increases the video stream size. In order to save bandwidth, it is common that video streams are encoded with only one I-frame per GOP. In WAVE, the errors of video stream decoding can occur in OBUs due to packet loss or transmission error. A successfully received I-frame can correct any errors caused by any preceding frames. Therefore, I-frames are much more important than P- and B-frames. In the cases of heavy traffic load or severe packet loss, it is wise to deliver I-frames to OBUs first to enhance the quality of video playback.

Figure 2 depicts an example of video encoding pattern ("IBBPBBPBB") and video stream packetization. Encoded video frames are packetized into video packets for transportation by lower layer protocols. For instance, a video stream encoded by H.264 can be transported over real-time transport protocol (RTP) [15]. In the procedures of video packetization, the importance of a video packet can be marked as either high or low priority, so that it can be scheduled and delivered differently in lower layers. As shown in Figure 2, I-frame packets are marked as high-priority

packets, whereas P- and B-frame packets are marked as low priority. The video packet priority marking will be discussed in detail in Section 3.1.

It should be noted that the sequence of video frames for transmission is different from the sequence of video playback. Taking the encoding pattern "IBBP..." as an example, the actual transmission sequence is "IPBB...," that is, the P-frame is transmitted before the B-frames. This is due to the fact that decoding B-frames requires both I- and P-frames received in an ONU's buffer. In this paper, the playback deadline of a frame or packet means the latest time that a frame (or packet) has to be available in an ONU's decoding buffer.

2.2. WAVE Channel Operations. Detailed description of multiple channel operations in WAVE networks can be found at [1]. In order to better understand the operations of the proposed SDS algorithm in Section 3, we summarize the typical channel operations as follows. First, upon power on, an OBU monitors the CCH until a WAVE service advertisement (WSA) sent by an RSU is received. A WSA carries the information of available SCHs and their access parameters, such as channel numbers. Based on the WSA information, the OBU then synchronizes with the RSU, and the OBU can exchange data with the RSU in SCHs.

Due to vehicles' fast movement, the topology of WAVE networks changes frequently. To maximum the time used for data exchange, WAVE devices do not use active scanning, association, and authenticating procedures as usually utilized in IEEE 802.11 [1]. WAVE defines a channel synchronization procedure which manages channel coordination. Figure 3 shows a WAVE sync interval which consists of a CCH interval, an SCH interval, and guard intervals between them. The CCH interval is used to send high-priority safety messages and WSAs, whereas the SCH interval is used for data services. The length of the CCH and SCH intervals, that is, T_{CCH} and T_{SCH}, can be adjusted adaptively. However, the total length of the sync interval, T_{SYNC}, is fixed to $T_{SYNC} =$
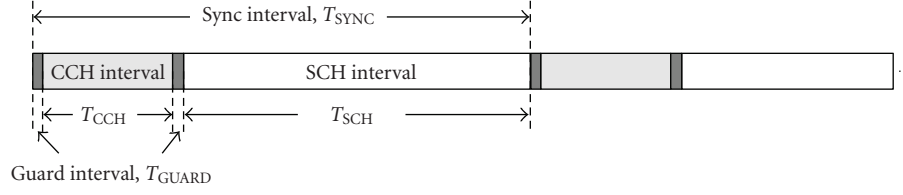
FIGURE 3: WAVE SYNC interval.

$T_{\text{CCH}} + T_{\text{SCH}} + 2 \times T_{\text{GUARD}}$. When joining to a WAVE basic service set (WBSS), OBUs are required to synchronize with the WBSS provider, that is, an RSU, in V2I model. The synchronization ensures the OBUs to monitor CCH during CCH intervals. Video packets are delivered in SCH intervals.

The discussion above is based on single-channel WAVE devices. With a single channel, an OBU can work on either CCH or SCH at a time. If two or more channels are facilitated in a WAVE device, the operations in CCH interval and SCH interval can be conducted simultaneously. In this paper, we consider single-channel WAVE devices only as this is common for OBUs.

3. System Model of VOD in WAVE Networks

3.1. System Architecture. The architecture of WAVE down-link scheduling for video streaming is shown in Figure 4(a). There are two types of transport protocols used in WAVE MAC: WAVE short message protocol (WSMP) and Internet protocol (IP). WSMP is used to transport WAVE system information, such as management, control, and safety information. WAVE short message (WSM) and WSA are transported by WSMP. IP is used to convey video packets and other application data. WSMP packets can be exchanged on both CCH and SCH. However, IP packets are only allowed on the SCH. These two protocol types are identified by the *EtherType* field of IEEE 802.2 header. The channel router routes packets to CCH or SCH according to this field.

Inside the CCH, there is a CCH classifier which categorizes packets into four queues identified by access category index (ACI). These four queues have different traffic specifications, such as access priorities, which are defined in IEEE 802.11e EDCA. Each access category has its own channel access function and uses specified EDCA parameters. These parameters are (1) arbitration interframe space (AIFS) which defines the minimum time interval between the wireless medium becoming idle and the start of another transmission; (2) contention windows (CW) which defines a random number generation window for ran-dom collision backoff mechanism; (3) transmit opportunity (TXOP) limit which defines the maximum time duration for which a WAVE device can transmit after obtaining a TXOP. The sum of TXOPs of these four queues is a CCH interval, that is, $T_{\text{CCH}} = \sum_{i=1}^{4} \text{TXOP}_{\text{CCH},i}$, where $\text{TXOP}_{\text{CCH},i}$ is the TXOP of the ACI $= i$ queue in the CCH channel. The SCH scheduler schedules these four access category queues based on their access parameters.

There is no specific scheduling algorithm defined in the WAVE standard, and the scheduling operations may differ in different implementations.

In the proposed scheduling architecture, SCH's packet queuing procedure is similar to that of CCH's. The SCH classifier is in charge of putting packets into these four access category queues. It is the same as CCH that $T_{\text{SCH}} = \sum_{i=1}^{4} \text{TXOP}_{\text{SCH},i}$. SCH differs from CCH in the following manners: (1) the ACI $= 2$ queue (Figure 4(c)) is dedicated for video streaming packets, and it is further separated into subqueues for each OBU to store high- and low-priority video packets; (2) the SCH scheduler implements the proposed SDS algorithm (as discussed in Section 4).

Before sending video packets to a channel router, the priority of each video packet is marked by the priority marking module. Some reserved bits in video bitstream header can be used to identify the importance of a packet. Taking H.264 over RTP [15] as an example, the *nal_unit_type* with values 30 and 31 are not specified in the standard [15]. As shown in Figure 4(b), we can mark high-priority video packets as type 30, and low priority video packets as type 31.

3.2. System Operations. Figure 5 illustrates the operations in an RSU. In a CCH interval, as shown in Figure 5(a), the RSU announces the video streaming services provisioned via WSA messages. The video streams are identified by provider service identifiers (PSIDs). Upon receiving the WSAs, an OBU which is interested to playback the video stream replies to the RSU to join in a WBSS. The OBU also sends its current location (coordinates from global positioning system, or GPS), direction, and speed (estimated by the OBU itself based on the GPS coordinates) to the RSU. Based on the OBU's information, the RSU estimates the time that the OBU leaves its radio coverage (or cell), that is, the out-of-cell time $T_{\text{out},i}$, where i is the index of the OBU. The RSU also monitors the signal strength of each OBU and estimates a transmission rate, denoted as R_i, for each OBU. The RSU maintains a WBSS user list which keeps each OBU's PSID, $T_{\text{out},i}$, and R_i (as shown in the middle of Figure 5).

Before delivering packets in an SCH interval, the RSU has to decide how to use the SCH interval for all OBUs. This is where scheduling algorithms come to play. In the proposed architecture, the SDS algorithm schedules downlink packets based on a *service list* to specify the sequence of the OBUs to be served and which packets in the queues are granted to deliver to OBUs. As shown in Figure 5, OBU 3 will be
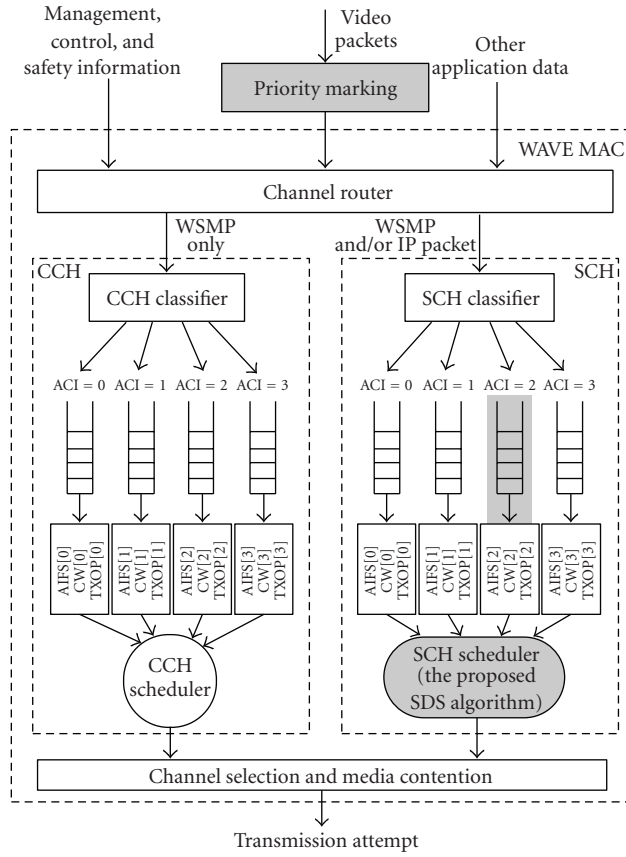
(a) The scheduling architecture of video streaming
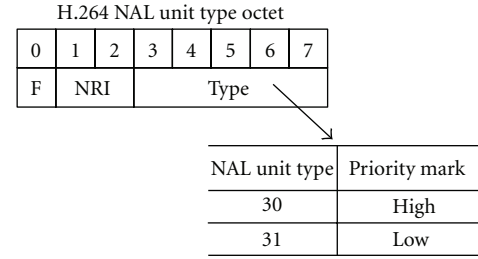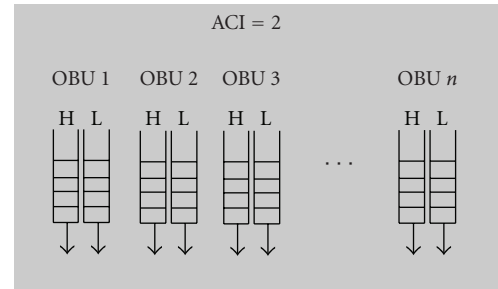
(c) Inside of the ACI = 2 queue

Figure 4: The proposed scheduling architecture of VOD in WAVE.

(a) RSU operation in CCH
at each CCH interval
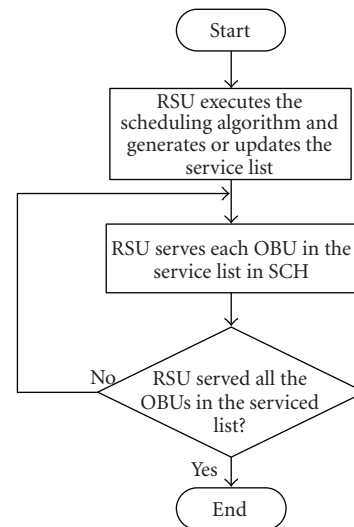
(b) RSU operation in SCH
at each SCH interval

Figure 5: RSU operations.
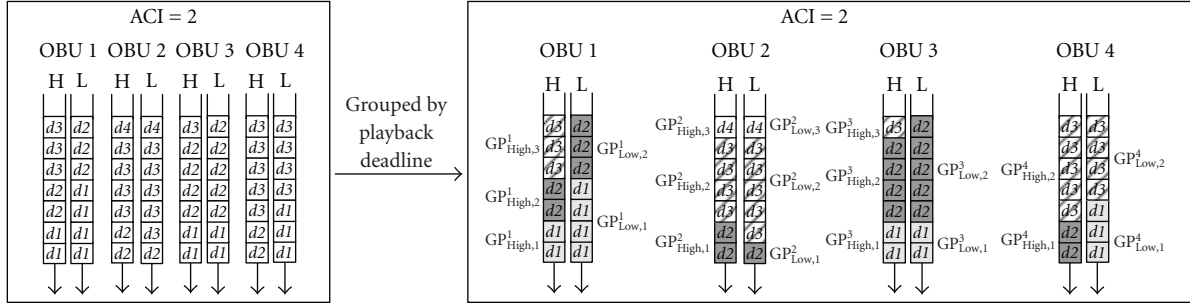
FIGURE 6: An example of video packet grouping.

TABLE 1: WAVE MAC EDCA parameters.

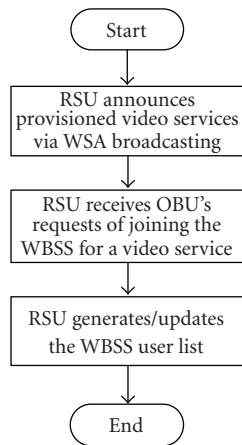| ACI	AC	CW_{min}	CW_{max}	AIFSN	TXOP
1	Background	$aCW_{min} \times 2$	$aCW_{max} \times 10$	9	0.5 ms
0	Best effort	aCW_{min}	aCW_{max}	6	1.5 ms
2	Video	$(aCW_{min}+1)/2-1$	aCW_{min}	2	5.0 ms
3	Voice	$(aCW_{min}+1)/2-1$	aCW_{min}	2	5.5 ms

TABLE 2: Parameters of video streams used in the simulations.

Parameter	Value
Encoder	H.264/AVC main profile
Resolution	CIF 352 × 288
Encoding pattern	IBBBPBBBPBBBPBBB (16 frames, with 3 B-frames per I- or P-frame)
Quantization parameters	I-P-B: 28-28-30
Frame rate	30 fps (frames per second)
Length	20 minutes

served first and then OBU 1. There are 21 packets $(0 \sim 20)$ in the high-priority queue and 31 packet $(0 \sim 30)$ in the low-priority queue granted to be delivered to OBU 3. For OBU 1, only high-priority packets $11 \sim 25$ are granted to be delivered in this SCH interval. The high-priority packets $0 \sim 10$ of OBU 1 are not granted, and they are probably expired and thus discarded. The SDS algorithm generates or updates the service list in the RSU just before each SCH interval. The service list is indexed by OBU IDs and it is ordered by a weight factor (as discussed in Section 4 below).

After the service list is generated or updated, as shown in Figure 5(b), the RSU serves the OBUs according to this list one by one by following the order. The granted video packets are delivered within the current SCH interval. The above operational procedure repeated in each WAVE sync interval.

4. Proposed SDS Algorithm

As aforementioned, an RSU delivers downlink packets to OBUs based on a service list. The first task for the SDS

algorithm to do is to generate or update the service list, so that the RSU can use it to deliver packets in SCH intervals.

To clarify the problem and the proposed algorithm, we give the following definitions. OBU i requests to playback a packetized video stream V. V consists of a sequence of packets $\{p_k \mid 1 \leq k \leq m\}$, where p_k is the kth packet, and m is the total number of video packets. Each packet p_k is defined by a tuple $\langle S_k, T_{\text{deadline},k}, \text{Pri}_k \rangle$, where S_k is the size of the packet, $T_{\text{deadline},k}$ is the playback deadline, and Pri_k is the priority of the packet, $\text{Pri}_k \in \{\text{High}, \text{Low}\}$. The sequences of packets are fed into the ACI = 2 queue, and they are stored in OBUs' high and low subqueues based on their destined OBUs and Pri_k (refer to Figure 4). In each subqueue, the packets with the same playback deadline are further grouped into a group of packets (GP) . Let $\text{GP}^i_{\text{High},j}$ and $\text{GP}^i_{\text{Low},j}$ represent the jth high-and low-priority GPs in OBU i. They are defined by $\langle S^i_{\text{High},j}, T_{\text{High,deadline},j} \rangle$, and $\langle S^i_{\text{Low},j}, T_{\text{Low,deadline},j} \rangle$, respectively. We have

$$
\begin{aligned}
\text{GP}^i_{\text{High},j} &= \bigcup_{T_{\text{deadline},k}=T_{\text{High,deadline},j}, \text{Pri}_k=\text{High}} p_k, \\
\text{GP}^i_{\text{Low},j} &= \bigcup_{T_{\text{deadline},k}=T_{\text{Low,deadline},j}, \text{Pri}_k=\text{Low}} p_k.
\end{aligned}
\tag{1}
$$

The sizes of $\text{GP}^i_{\text{High},j}$ and $\text{GP}^i_{\text{Low},j}$ are

$$
\begin{aligned}
S^i_{\text{High},j} &= \sum_{p_k \in \text{GP}_{\text{High},j}} S_k, \\
S^i_{\text{Low},j} &= \sum_{p_k \in \text{GP}_{\text{Low},j}} S_k.
\end{aligned}
\tag{2}
$$

Let T represents the current time on the absolute time axis. We use t_{UnAlloc} to denote the unallocated time in the current SCH interval for the ACI = 2 queue, $t_{\text{UnAlloc}} = \text{TXOP}_{\text{SCH},2}$, before running the algorithm.

As aforementioned, the RSU keeps each OBU's out-of-cell time, $T_{\text{out},i}$, and transmission rate R_i. The possible (maximum) transmission time can be allocated to OBU i is

$$
\min(\text{TXOP}_{\text{SCH},2}, (T_{\text{out},i} - T)),
\tag{3}
$$

that is, either the service interval is entirely allocated to OBU i or the RSU allocates the OBU's with its remaining
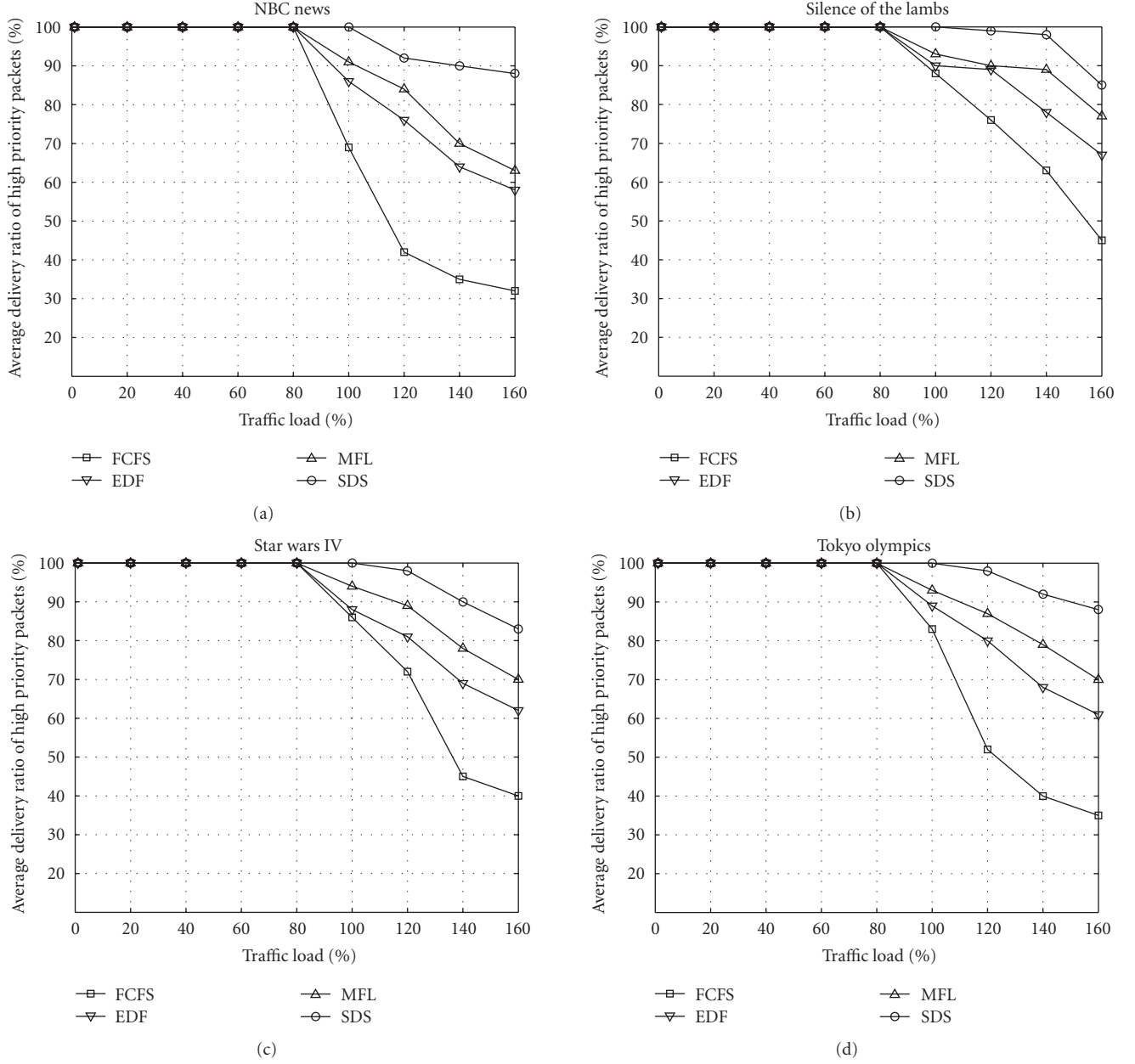
Figure 7: Average delivery ratio of high-priority packets.

dwelling time $(T_{out,i} - T)$. Based on a GP's playback deadline, we can calculate a tolerable playback delay of this GP. For example, the tolerable playback delay of a high-priority GP j is $(T^i_{High,deadline,j} - T)$. The RSU also keeps the time that each GP already waited in the queue, that is, queuing delay. For example, the queuing delay of a high-priority GP j is $t^i_{High,queued,j}$. Based on the total size of a GP and the transmission rate to OBU i, the total time needed to transmit a GP to OBU i can be calculated, for example, for GP j it is $S^i_{High,j}/R_i$ or $S^i_{Low,j}/R_i$, for high- and low-priority GPs, respectively.

The objective of the scheduling algorithm is to generate or update a service list and then schedule packets based on this service list (see Algorithm 1). The service list indicates

the order of the OBUs to be served and the packets in each OBU queue to be delivered. Let n be the number OBUs connected to the RSU. We sort the OBUs by using a scheduling weight factor w_i which is defined as

$$w_i = \left[\frac{S^i_{High,1}}{R_i} - \left(T^i_{High,deadline,1} - T \right) \right] \times \frac{t^i_{High,queued,1}}{\left(T_{out,i} - T \right)},$$

(4)

where $S^i_{High,1}/R_i$ is the time needed to send the first high-priority GPs to OBU i, and $(T^i_{High,deadline,1} - T)$ is the tolerable playback deadline of the first high-priority GPs. The equation implies that the OBUs with small dwelling time, small tolerable playback delay, larger queuing delay,
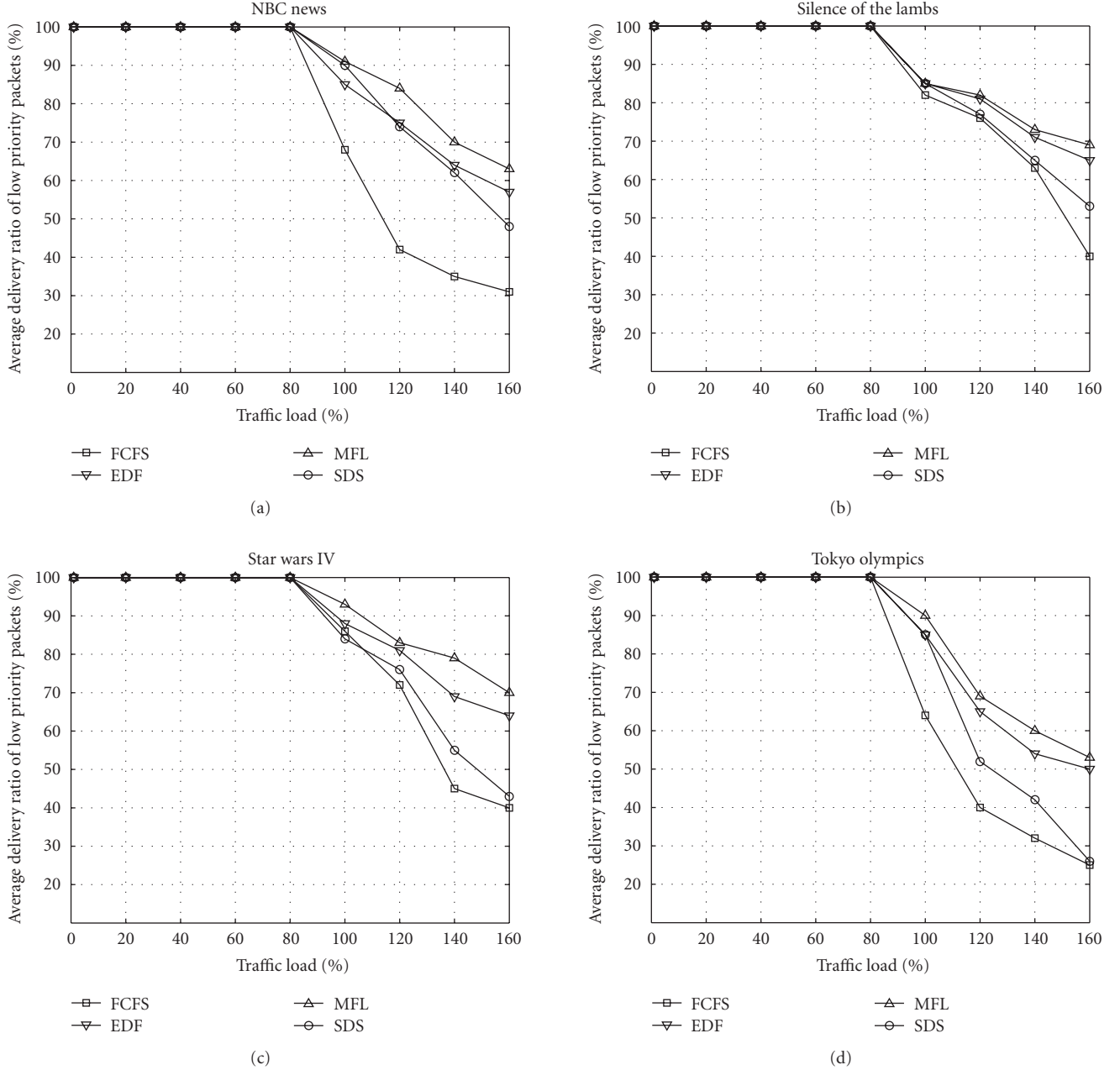
FIGURE 8: Average delivery ratio of low-priority packets.

and larger amount of high-priority packets will be scheduled first.

The algorithm first initializes the service list by removing disconnected OBUs and adding newly connected OBUs (line 1). It then discards the packets with playback deadlines are reached (line 2). The algorithm calculates the scheduling weight factors (line 3) and sorts the OBUs by their factors (line 4). After that, for each OBU, all the transmission time required for transmitting high- and low-priority GPs with deadlines within current WAVE sync interval T_{SYNC} is calculated (line 5). This step is to find the packets which will be expired if they are not transmitted within current sync interval. Line 6 calculates the total time required to

transmit all high- and low-priority GPs. Line 7 initializes $t_{UnAlloc}$ to $TXOP_{SCH,2}$. If the unallocated time is enough for transmitting the high-priority GPs (line 8), all high-priority GPs will be granted (line 9). Line 10 updates the unallocated time $t_{UnAlloc}$. If there is still enough time for low-priority GPs (line 11), all packets in the low-priority GPs will be granted (line 12). If the unallocated time $t_{UnAlloc}$ is not enough to transmit all low-priority GPs (line 15), $t_{UnAlloc}$ will be used for allocating the GPs according to the order yielded in line 4, until the TXOP is fully allocated, that is, $t_{UnAlloc} = 0$. It is the same if $TXOP_{SCH,2}$ is not enough for all high-priority GPs (line 19), and only part of high-priority GPs can be granted according to the
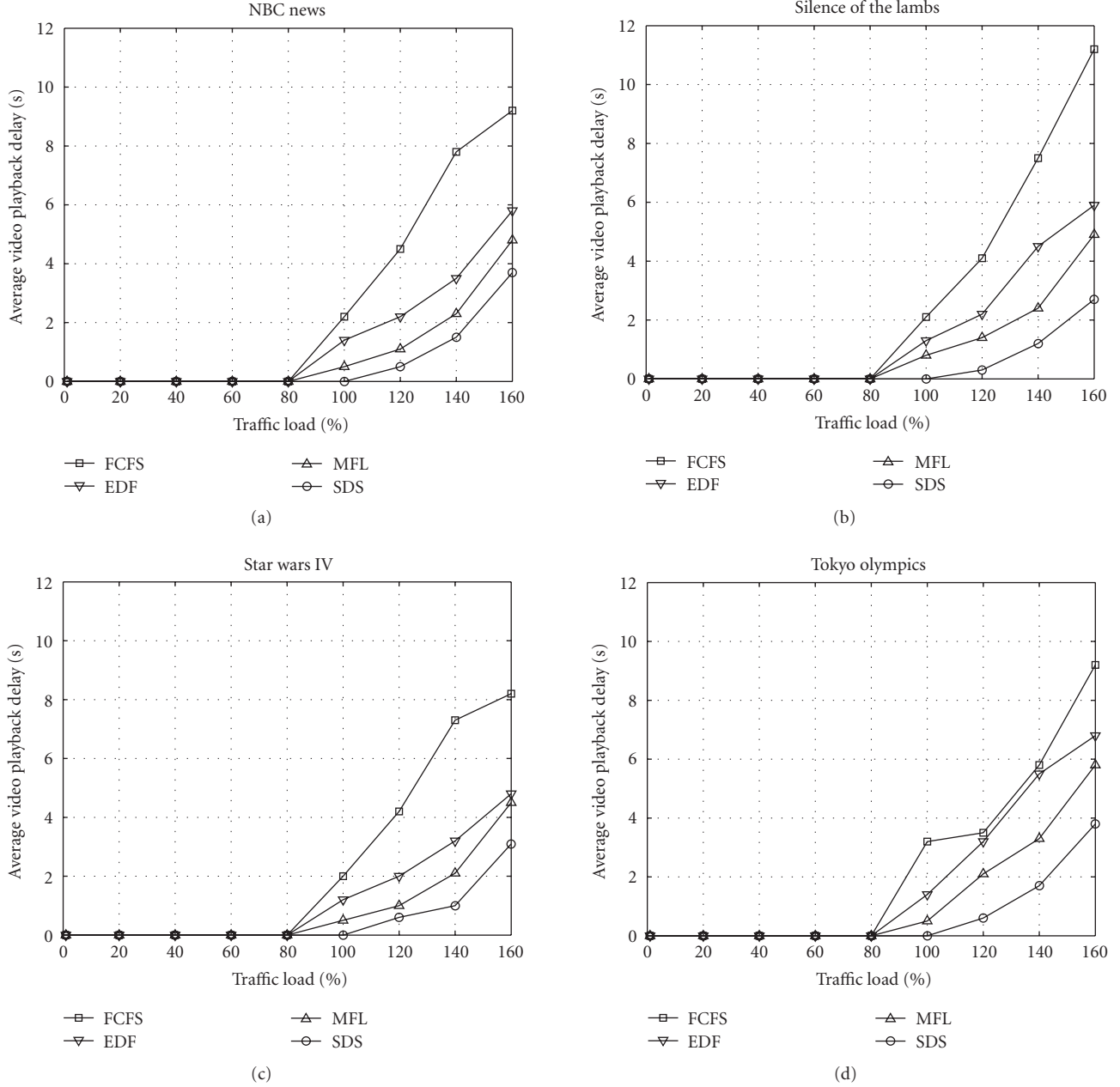
FIGURE 9: Average video playback delay.

order yielded in line 4. If there is still some time left after all the high- and low-priority GPs with the playback deadlines within current T_{SYNC} are granted (line 22), the remaining time will be used to grant the packets in all high-priority subqueues and then all low-priority subqueues in a round-robin manner in the sorted OBU order. The reason to use round-robin here is for computational simplicity, since these packets' playback deadlines are not reached until next SCH interval and they can be transmitted in next sync interval if they are not scheduled in current T_{SYNC}.

Figure 6 shows an example of video packet grouping. There are four OBUs currently connected to an RSU, and

there are four playback deadlines $d1$, $d2$, $d3$, and $d4$ as marked in the figure, where $d1 < d2 < d3 < d4$. Assume that we have $d2 < T_{\text{SYNC}} < d3$, that is, the packets with playback deadlines as $d1$ and $d2$ have to be transmitted in this TXOP. The video packets with the same playback deadlines are grouped together to form GPs. Assume that the order of the OBUs in the service list is OBU 3, OBU 1, OBU 4, and OBU 2, which is decided by their scheduling weight factors (4). According to the algorithm, the granting sequence of the GPs will be $GP^3_{\text{High},1}$, $GP^3_{\text{High},2}$, $GP^1_{\text{High},1}$, $GP^1_{\text{High},2}$, $GP^4_{\text{High},1}$, and $GP^2_{\text{High},1}$ for the high-priority GPs. If there is still some time left, the low-priority GPs will be granted in the sequence of $GP^3_{\text{Low},1}$, $GP^3_{\text{Low},2}$, $GP^1_{\text{Low},1}$, $GP^1_{\text{Low},2}$,

> (1) Initialize the service list: remove an OBU from the list if it is no longer connected to the RSU;
>   add an OBU into the list if it is newly connected to the RSU.
> (2) In each sub-queue, discard the packets if their playback deadlines are reached.
> (3) Calculate w_i, $(i = 1, \ldots, n)$ for each OBU by using (4).
> (4) Sort the service list (indexed by OBUs ID) by weight factor w_i, the weightiest the first.
> (5) for all i, $i = 1, \ldots, n$, calculate $T_{\mathrm{High},i} = \sum_j S_{\mathrm{High},j}^i / R_i$ and $T_{\mathrm{Low},i} = \sum_j S_{\mathrm{Low},j}^i / R_i$ for all GP j that
> $(T_{\mathrm{High,deadline},j}^i - T) \le T_{\mathrm{SYNC}}$ and $(T_{\mathrm{Low,deadline},j}^i - T) \le T_{\mathrm{SYNC}}$.
> (6) Calculate $T_{\mathrm{High}} = \sum_{i=1}^n T_{\mathrm{High},i}$ and $T_{\mathrm{Low}} = \sum_{i=1}^n T_{\mathrm{Low},i}$.
> (7) $t_{\mathrm{UnAlloc}} = \mathrm{TXOP}_{\mathrm{SCH},2}$.
> (8) **if** $T_{\mathrm{High},i} \le t_{\mathrm{UnAlloc}}$ **then**
> (9)   Grant all the high priority GPs with deadlines in current T_{SYNC}.
> (10)   Update $t_{\mathrm{UnAlloc}} = t_{\mathrm{UnAlloc}} - T_{\mathrm{High},i}$.
> (11)   **if** $T_{\mathrm{Low},i} \le t_{\mathrm{UnAlloc}}$ **then**
> (12)   Grant all the low priority packets with deadlines in current T_{SYNC}.
> (13)   Update $t_{\mathrm{UnAlloc}} = t_{\mathrm{UnAlloc}} - T_{\mathrm{Low},i}$.
> (14)   **else** only part of low priority GPs can be sent
> (15)   Grant low priority GPs in the sorted OBU order and update t_{UnAlloc} once a GP is granted, until $t_{\mathrm{UnAlloc}} = 0$.
> (16)   **algorithm ends**
> (17)   **endif**
> (18) **else**
> (19)   Grant high priority GPs in the sorted OBU order and update t_{UnAlloc} once a GP is granted, until $t_{\mathrm{UnAlloc}} = 0$.
> (20)   **algorithm ends**
> (21) **endif**
> (22) **if** $t_{\mathrm{UnAlloc}} > 0$ **then** still some time left
> (23) Grant the packets in all high priority sub-queues and then all low priority sub-queues in a round-robin
>   manner and in the sorted OBU order, update t_{UnAlloc} once a packets is granted, until $t_{\mathrm{UnAlloc}} = 0$.
> (24) **endif**
> (25) **algorithm ends**

ALGORITHM 1: Service list generation and updating algorithm.

$GP_{\mathrm{Low},1}^4$, and $GP_{\mathrm{Low},1}^2$. After that, if there is still some time left, the rest GPs can be scheduled in the order of $GP_{\mathrm{High},3}^3$, $GP_{\mathrm{High},3}^1$, $GP_{\mathrm{High},2}^4$, $GP_{\mathrm{High},2}^2$, $GP_{\mathrm{High},3}^2$, $GP_{\mathrm{Low},2}^4$, $GP_{\mathrm{Low},2}^2$, and $GP_{\mathrm{Low},3}^2$.

5. Simulation and Result Analysis

5.1. Simulation Setup. We have developed a WAVE network simulator to simulate WAVE networks. It simulates a two-lane straight highway of 30 km with each lane for each direction of traffic flow. In each lane, there are more than one sublane all carrying traffic to same direction. These sublanes enable vehicles overtaking each other. There are 50 RSUs uniformly distributed in the highway, each has 300 m transmission coverage radius. This makes a full coverage of the simulated highway. Simulated vehicles with mounted single-channel OBUs randomly enter the highway with exponentially distributed interarrival time at a rate of 100 vehicles per minute. Vehicle speed follows a Gaussian distribution with a mean of 100 km/h and a standard deviation of 10 km/h. The capacity of the RSU is 54 Mbps. Channel data rate varies between 1 to 54 Mbps and is a function of distance in our simulation for simplicity. The EDCA parameters used in the WAVE MAC are listed in Table 1. The contention window size is $a\mathrm{CW}_{\min} = 7$ milliseconds and $a\mathrm{CW}_{\max} = 15$ milliseconds.

Since the focus of the proposed algorithm is to schedule the packets in the RSU queues, handover is not considered directly in this paper. In the simulations, we assumed that the video packets associated to an OBU are always available in its currently connected RSU. If the OBU is not associated with the RSU due to mobility, the proposed algorithm will discard all the packets related to the OBU in the RSU queues.

We tested our algorithm with several other benchmark video streams [16]. The encoding parameters of the benchmark videos and the parameters used in our experiments are listed in Table 2. The statistics of the four video streams used are presented in Tables 3 and 4. The maximum packet size of the video streaming is set to 1000 bytes, so that large frames will be packetized into two or more packets. The ratios between high- and low-priority packets are fixed for the chosen video clip which is listed in Table 5.

In the simulations, we assume that all OBUs request VOD services after they enter the simulated highway. The requested video streams are randomly selected from the aforementioned four benchmark video streams. Except for the video data (ACI = 2), other types of downlink traffics are generated by following rules: (1) voice traffic is generated randomly in the range of (32 kbps, 128 kbps) (2) self-similar BE traffic is generated randomly in the range of (0 kbps, 128 kbps) (3) background traffic is generated randomly in the range of (0 kbps, 256 kbps).

TABLE 3: Statistics of video streams used in the simulations (high priority frame and packet).

Video stream	Video size	No. of Frm	Maximum Frm size	Average Frm size	No. of Pkts	Maximum Pkt size	Average Pkt size
NBC news	57.642 MB	2,251	22,815 B	9,468.720 B	22,452	1,000 B	949.318 B
Silence of the lambs	22.986 MB	2,251	23,198 B	4,412.618 B	11,125	1,000 B	892.636 B
Star wars IV	23.144 MB	2,251	10,500 B	3,660.987 B	9,373	1,000 B	879.215 B
Tokyo Olympics	56.362 MB	2,251	27,850 B	7,415.579 B	17,810	1,000 B	937.253 B

TABLE 4: Statistics of video streams used in the simulations (low priority frame and packet).

Video stream	Video size	No. of Frm	Maximum Frm size	Average Frm Size	No. of Pkts	Maximum Pkt size	Average Pkt Size
NBC news	57.642 MB	33,750	21,176 B	1,159.340 B	58,799	1,000 B	665.449 B
Silence of the lambs	22.986 MB	33,750	15,174 B	419.837 B	40,677	1,000 B	348.342 B
Star wars IV	23.144 MB	33,750	8,995 B	474.877 B	40,886	1,000 B	391.995 B
Tokyo Olympics	56.362 MB	33,750	25,762 B	1,256.521 B	64,848	1,000 B	653.954 B

TABLE 5: The ratio between high- and low-priority packets.

Video stream	Ratio between high- and low-priority packets
NBC news	1 : 2.62
Silence of lambs	1 : 3.65
Star wars IV	1 : 4.36
Tokyo Olympics	1 : 3.64

In the simulations, the proposed SDS algorithm is compared with the conventional FCFS [6], EDF [7], and MFL [5] algorithms in terms of video packet delivery ratio and video playback delay. We run the simulation 50 rounds and collect the results for analysis.

5.2. Delivery Ratio of Video Packets. More high-priority packets delivered means better video playback quality at OBUs. We measure the number of high- and low-priority packets delivered in all simulation runs with different traffic loads (from 1 to 160%). The average delivery ratio of high- and low-priority packets is shown in Figures 7 and 8, respectively.

Figure 7 shows the average delivery ratio of high-priority packets. When the total traffic load is under 80%, there is no high-priority packet loss. After the traffic load reaches a saturation point (100% of traffic load), the delivery ratio drops. The FCFS algorithm drops the fastest. This is because it does not distinguish the difference in packets. By distinguishing playback deadlines, the EDF and MFL perform better than FCFS in all video streams. Amongst EDF and MFL, MFL performs better due to the fact that it considered some specific features of vehicular networks, such as remaining dwelling time and tolerable maximum transmission delay. As EDF and MFL do not take the importance of different types of video packet into account, they treat high- and low-priority packets in the same way. The SDS algorithm outperforms the others due to the fact that the high-priority packets are preferentially treated. In all four video streams, the SDS algorithm always achieves the highest delivery ratio. Even though the overall traffic load is 160%, the SDS algorithm's average delivery ratio is still

higher than 80% in all video streams. This reflects the specific feature of the SDS algorithm, that is, to deliver high-priority packets first.

Apart from the delivery ratio of high-priority packets, we are also interested to analyze the delivery ratio of low-priority packets. As shown in Figure 8, when the traffic load becomes higher than the saturation point, the delivery ratios of all algorithms drop quickly for all video streams. As can be observed, the delivery ratio of the SDS algorithm is lower than EDF and MFL in most of the cases. It is even lower than FCFS in video stream *Tokyo Olympics* after the traffic load is larger than 140%. This reflects the fact that the high-delivery ratio of high-priority packets is at the cost of low-delivery ratio of low-priority packets when the traffic load is very heavy.

5.3. Video Playback Delay. Apart from higher delivery ratio of high-priority packets, another main objective of the proposed SDS algorithm is to reduce video playback delay. In the situations of heavy traffic load and severe channel condition, video playback delay is unavoidable. In this simulation, video playback delay is defined as the accumulated time of the video discontinuity during the overall video stream playback. Video discontinuity is caused because the required high-priority packets (which constitute I-frames) are undelivered within their playback deadlines.

Figure 9 depicts the average video playback delay in all simulation runs. As can be observed, FCFS has the longest delay in all simulations. EDF and MFL perform better than FCFS, and MFL is slightly better than EDF. It can be also observed that video streams with a smaller packet size, such as the video stream *silence of the lambs*, suffer less playback delay. Since high-priority packets are always delivered first, our SDS algorithm achieves the smallest delay in all video streams. The playback delay of the SDS algorithm is nearly half of the MFL for all cases.

6. Conclusion

In this paper, we have proposed a selective downlink scheduling algorithm, SDS, for WAVE networks in order to improve

the quality of VOD services. Video packets are marked into high and low priorities before they are fed into WAVE MAC. Marked video packets are put into different access category queues. Our proposed SDS algorithm schedules the video packets according to their playback deadline and received OBU's network parameters, such as the time remained to communicate with current RSU and so on. The SDS algorithm is evaluated through simulations, showing its effectiveness and efficiency for video streaming over WAVE networks. In this work, we only focus on increasing the delivery ratio of high-priority packets and reducing video playback delay. We believe that these two parameters are fundamental for the quality of video streaming over WAVE. Other video quality evaluations, such as peak signal-to-noise ratio (PSNR) and mean opinion score (MOS), will be studied in our future work.

Acknowledgments

References

[1] IEEE Std 1609.4-2006, "IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE)—Multi-Channel Operation," 2004.

[2] IEEE Std 802.11, IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 2001.

[3] IEEE P802.11p, "Draft Amendment to STANDARD FOR Information technology—Telecommunications and information exchange between systems—LAN/MAN Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Wireless Access in Vehicular Environments (WAVE)".

[4] ISO/IEC 14496-10 and ITU-T Recommendation H.264, Advanced Video Coding, 2003.

[5] C.-J. Chang, R.-G. Cheng, H.-T. Shih, and Y.-S. Chen, "Maximum freedom last scheduling algorithm for downlinks of DSRC networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 223–232, 2007.

[6] IEEE 1455-1999, "IEEE standard for message sets for vehicle/roadside communications," 1999.

[7] T. M. Lim, B.-S. Lee, and C. K. Yeo, "Quantum-based earliest deadline first scheduling for multiservices," *IEEE Transactions on Multimedia*, vol. 9, no. 1, pp. 157–168, 2007.

[8] Z. Ouan and J.-M. Chung, "A statistical framework for EDF scheduling," *IEEE Communications Letters*, vol. 7, no. 10, pp. 493–495, 2003.

[9] K. Kang, Y. Cho, J. Cho, and H. Shin, "Scheduling scalable multimedia streams for 3G cellular broadcast and multicast services," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 5, pp. 2655–2672, 2007.

[10] A. Ksentini, M. Naimi, and A. Guéroui, "Toward an improvement of H.264 video transmission over IEEE 802.11e through a cross-layer architecture," *IEEE Communications Magazine*, vol. 44, no. 1, pp. 107–114, 2006.

[11] M. van der Schaar, Y. Andreopoulos, and Z. Hu, "Optimized scalable video streaming over IEEE 802.11 a/e HCCA wireless networks under delay constraints," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 755–768, 2006.

[12] N. S. Shankar and M. van der Schaar, "Performance analysis of video transmission over IEEE 802.11a/e WLANs," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 4, part 2, pp. 2346–2362, 2007.

[13] Q. Li, Y. Andreopoulos, and M. van der Schaar, "Streaming-viability analysis and packet scheduling for video over in-vehicle wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, part 1, pp. 3533–3549, 2007.

[14] IEEE 802.11e/D13.0, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), draft supplement, January 2005.

[15] S. Wenger, M. M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Singer, "RTP Payload Format for H.264 Video," IETF RFC 3984, February 2005.

[16] Video Stream Traces, 2008, http://trace.eas.asu.edu/.

EUSIPCO 2011

19th European Signal Processing Conference
August 29- September 2, 2011, Barcelona (Spain)

Photograph © Turisme de Barcelona / J. Trullàs

Preliminary call for papers

The 2011 European Signal Processing Conference (EUSIPCO-2011) is the nineteenth in a series of conferences promoted by the European Association for Signal Processing (EURASIP, www.eurasip.org). This year edition will take place in Barcelona, capital city of Catalonia (Spain), and will be jointly organized by the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC) and the Universitat Politècnica de Catalunya (UPC).

EUSIPCO-2011 will focus on key aspects of signal processing theory and applications as listed below. Acceptance of submissions will be based on quality, relevance and originality. Accepted papers will be published in the EUSIPCO proceedings and presented during the conference. Paper submissions, proposals for tutorials and proposals for special sessions are invited in, but not limited to, the following areas of interest.

Areas of Interest

- Audio and electro-acoustics.
- Design, implementation, and applications of signal processing systems.
- Multimedia signal processing and coding.
- Image and multidimensional signal processing.
- Signal detection and estimation.
- Sensor array and multi-channel signal processing.
- Sensor fusion in networked systems.
- Signal processing for communications.
- Medical imaging and image analysis.
- Non-stationary, non-linear and non-Gaussian signal processing.

Submissions

Procedures to submit a paper and proposals for special sessions and tutorials will be detailed at www.eusipco2011.org. Submitted papers must be camera-ready, no more than 5 pages long, and conforming to the standard specified on the EUSIPCO 2011 web site. First authors who are registered students can participate in the best student paper competition.

Important Deadlines:

Proposals for special sessions	15 Dec 2010
Proposals for tutorials	18 Feb 2011
Electronic submission of full papers	**21 Feb 2011**
Notification of acceptance	23 May 2011
Submission of camera-ready papers	6 Jun 2011

Webpage: www.eusipco2011.org

Organizing Committee

Honorary Chair
Miguel A. Lagunas (CTTC)

General Chair
Ana I. Pérez-Neira (UPC)

General Vice-Chair
Carles Antón-Haro (CTTC)

Technical Program Chair
Xavier Mestre (CTTC)

Technical Program Co-Chairs
Javier Hernando (UPC)
Montserrat Pardàs (UPC)

Plenary Talks
Ferran Marqués (UPC)
Yonina Eldar (Technion)

Special Sessions
Ignacio Santamaría (Unversidad de Cantabria)
Mats Bengtsson (KTH)

Finances
Montserrat Nájar (UPC)

Tutorials
Daniel P. Palomar
(Hong Kong UST)
Beatrice Pesquet-Popescu (ENST)

Publicity
Stephan Pfletschinger (CTTC)
Mònica Navarro (CTTC)

Publications
Antonio Pascual (UPC)
Carles Fernández (CTTC)

Industrial Liaison & Exhibits
Angeliki Alexiou
(University of Piraeus)
Albert Sitjà (CTTC)

International Liaison
Ju Liu (Shandong University-China)
Jinhong Yuan (UNSW-Australia)
Tamas Sziranyi (SZTAKI -Hungary)
Rich Stern (CMU-USA)
Ricardo L. de Queiroz (UNB-Brazil)