2

A QoS Guaranteed Energy-Efficient Scheduling for IEEE 802.16e

Wen-Hwa Liao and Wen-Ming Yen
Department of Information Management, Tatung University, Taipei
Taiwan

1. Introduction

Recently, the IEEE 802.16 standard (IEEE Std 802.16-2004, 2004), a solution to broadband wireless access commonly known as Worldwide Interoperability for Microwave Access (WiMAX), has been considered as a promising standard for next generation broadband wireless access networks. IEEE 802.16e (IEEE Std 802.16e-2005, 2005), also called Mobile WiMAX (Li et al., 2007), provides enhancements to IEEE 802.16 to support the mobility of Mobile Subscriber Stations (MSSs) at vehicular speed. Like other wireless systems, conserving energy is one of the critical issues for MSSs in IEEE 802.16e. Therefore, it is required for the protocol to offer a well-designed energy-efficient algorithm for an MSS.

IEEE 802.16e is expected to support Quality of Service (QoS) for real-time applications such as Voice over IP (VoIP), video streaming, and video conferencing with different QoS requirements (Wongthavarawat & Ganz, 2003; Zhu & Cao, 2004). Such applications are delay and delay variation susceptible. For example, when data packets incur vast delays and delay variations, the quality of the application seriously degrades. In order to avoid such situation, QoS provides the guarantee of transmission. IEEE 802.16e defines five types of service classed: Unsolicited Grant Service (UGS), Real-Time Variable Rate (RT-VR), Non-Real-Time Variable Rate (NRT-VR), Best Effort (BE), and Extended Real-Time Variable Rate (ERT-VR). Among them, the UGS is designed to support Constant Bit Rate (CBR) services, such as T1/E1 emulation, and VoIP without silence suppression. These kinds of services generate fixed-size data packets on a periodic basis. They usually require stringent QoS delay constraints, so determining the length of sleeping duration of an MSS in IEEE 802.16e is not only bounded by the total amount of traffic generated by the connections in the MSS, but is also restricted by the connections' QoS delay constraints. IEEE 802.16e was developed for the targets on mobile devices which are generally powered by energy-limited batteries. Thus, the energy-efficiency is an important issue to extend the lifetime of MSSs (Jang et al., 2006; Mukherjee et al., 2005; Tian et al., 2007). When a connection is established, an MSS may shift the operation status into sleep mode in order to save the power consumption if there are no packets to send or to receive in certain frame durations. Under sleep mode, there are two intervals: sleeping interval and listening interval. During the sleeping interval, an MSS can be powered down by putting its wireless network interface into sleep mode. Aside from this, the MSS would be unable to send or to receive packets during sleeping intervals. After a sleeping interval finishes, the MSS switches to listening interval. The MSS wakes up during the listening interval to check whether there are packets destined to it. Message packets are checked to determine whether the MSS should be woken up or not. IEEE 802.16e defined three types of Power-Saving Classes (PSCs) for connections with different characteristics, and each PSC is defined for a set of connections with common properties. A PSC is composed of interleaved listening windows and sleep windows. In PSC Type I, the sleep window is exponentially increased from a minimum value to a maximum value. This is typically done when the MSS is doing best-effort and non-real-time traffic. PSC Type II has a fixed-length sleep window and is used for UGS service. PSC Type III allows for a one-time sleep window and is typically used for multicast traffic or management traffic when the MSS knows when the next traffic is expected.

There are many previous researches that have devoted their efforts to adapting the sleeping duration of IEEE 802.11 and IEEE 802.15 (Liao & Wang, 2008; Liu & Liu, 2003; Tseng et al., 2002; Ye et al., 2004; Zheng et al., 2005). However, due to lack of QoS requirements, the results of those searches cannot be applied to IEEE 802.16e directly. Several studies have been proposed to analyze the IEEE 802.16e's power while an MSS operates in the power-saving mode (Han & Choi, 2006; Lei & Tsang, 2006; Seo et al., 2004). Several studies (Fang et al., 2006; Huang et al., 2007; Jang et al., 2006; Tsao & Chen, 2008) investigated the power consumption issues of IEEE 802.16e and suggested algorithms to determine the sleep interval in order to improve energy efficiency. In (Jang et al., 2006), the length of sleeping period is adapted according to the traffic type. This scenario is valid only under one MSS, and the QoS delay constraint is not considered. In (Tsao & Chen, 2008), although the QoS delay constraints are considered, the scenario cannot consider the energy costs of status transition. In (Fang et al., 2006), a scheduling algorithm for multiple MSSs with QoS delay constraints is proposed. To save power, the algorithm grants a primary MSS the right to use the bandwidth in burst mode. Secondary MSSs are only given the necessary bandwidth to meet the requirements of QoS delay constraints. However, its benefit only exhibits when the total traffic loading of all MSSs is low. In (Huang et al., 2007), although the constant bit rate traffic with QoS delay constraint is considered, the scenario cannot consider the jitter constraint.

In this chapter, we propose a QoS guaranteed energy-efficient scheduling for IEEE 802.16e. We consider that delay and jitter types of QoS should be scheduled at the same time and integrate sleep duration in one MSS. The packets would be scheduled successively to reduce the number of status transitions under QoS requirements for delay and jitter. The proposed approaches not only minimize the power consumption of the MSS but also guarantee both delay and jitter QoS of real-time connections.

2. The QoS guaranteed energy-efficient scheduling for IEEE 802.16e

In this section, we first describe the basic idea of our algorithm for QoS guaranteed energy-efficient scheduling. Second, we define the notations of our system model. Finally, we schedule packets in an MSS with our QoS guaranteed energy-efficient scheduling. Additionally, we consider the QoS requirements of jitter constraint to schedule the packets and achieve the guarantees of transmissions.

2.1 Basic idea

The idea behind our proposed algorithm, called successive scheduling scheme (SSS), is to schedule the packet transmission in successively fashion with the minimal interval of listen periods and a maximum interval of sleep periods without violating the QoS of all connections in an MSS. Additionally, the successive scheduling of time slots would reduce the number of status transitions between sleep periods and listen periods. This improvement greatly contributes to achieve the power-saving. The proposed approach can be adapted to the power-saving class of type III where the length of sleep and listen periods are variable.

2.2 System model

In this chapter, the centrally controlled IEEE 802.16e wireless network with a central BS and an MSS with multiple real-time connections is considered. The uplink and downlink channel is divided into fixed-size frames, and the frames are subdivided into fixed-size time slots. Both the energy consumption and the bandwidth are calculated in time slots. Different QoS parameters have been defined for various type of services in IEEE 802.16e, and all of them can be mapped into the minimum data rate requirements of the MSSs (Andrews et al., 2005). Therefore, we only apply the minimum data rate as the bandwidth requirement of QoS for each type of connection. Additionally, other QoS requirements such as the maximum latency and tolerated jitter would be considered in this chapter. The notations in this chapter are as follows: T_{aw} is the total number of time slots in which an MSS stays in the awake state; T_{st} denotes the total number of status transitions of an MSS from the sleep state to the awake state; Paw stands for the average energy consumption of each time slot by an MSS in the awake state; P_t represents the average energy consumption of each status transition from the sleep state to the awake state in an MSS; n denotes the index of time slot in an MSS; r_n stands for the data rate in which an MSS has been allocated by time slot n; $R_n^{\min}$ stands for the minimum data rate that an MSS should receive in order to guarantee its service quality in time slot n. We assume that there is no energy consumed during the sleep period of an MSS. Thus, the energy consumed of an MSS is determined by the number of the time slots it stays in the awake state and the number of status transitions it has from the sleep state to the awake state. The overall energy consumed by an MSS during period T, denoted as P, can be represented as follows:

$$P = T_{aw} \times P_{aw} + T_{st} \times P_t \tag{1}$$

The goal of the scheduling algorithm is to minimize the average energy consumed by an MSS during period T, while the QoS requirements such as minimum data rate, maximum delay constraint and tolerated jitter of an MSS must be guaranteed. Thus, we can minimize P by allocating the minimum time slots (T_{aw}) to satisfy the minimum data rates ($R_n^{\min}$) and successively schedule the packets to reduce the status transitions (T_{st}). In order to acquire the optimal result, the power-saving scheduling algorithm should consider the properties of the QoS requirements. We discuss the solutions of previous studies and present our QoS guaranteed energy-efficient scheduling to acquire the optimal result in the next section.

2.3 QoS guaranteed energy-efficient scheduling

First, we give the idea of our QoS guaranteed energy-efficient scheduling and perform the algorithm of our successive scheduling in an example. In the second part, we consider the jitter constraint of packet scheduling to provide more precise QoS guarantees.

2.3.1 The successive scheduling scheme (SSS)

To improve the power-saving performance, our algorithm will schedule packets into successive frames in order to reduce the number of status transitions in an MSS. The successive scheduling scheme is performed in two parts. The first part sorts all connections on the scheduling priorities of connections with tight delay requirements. The second part schedules the packets from the first priority connection into the successive frames. An MSS stays idle during sleep periods to save power, and only wakes up to transmit data packets during listen periods. Packets sent to the MSS during sleep periods are buffered at BS and are delivered to the MSS until the listen periods. In other words, the MSS only needs to receive and transmit data in listen periods and stay idle to conserve energy during sleep periods. The next paragraphs describe in detail the steps of our proposed successive scheduling scheme. Also, notations used in this chapter are summarized in Table 1.

Notation	Description
N	The number of connections
D_i	The delay constraint for connection i
I	The interval of packet arrival
$C_{i,j}$	The j^{th} packet for connection i
FIU	The frame-in-used; the frame which had already scheduled the packets without full-filled frame
FFU	The frame fully used; the frame which had already scheduled the packets without any available time slot
F_{next}	The unused frame that is next to the *FFU* and is more close to the next full-filled frame

Table 1. Notations and their descriptions.

To minimize the energy consumption of an MSS with multiple real-time connections, the successive scheduling scheme schedules the packets into their successive time slots under the radio resource and QoS requirements. Considering an MSS with N real-time connections, D_i is the delay constraint in milliseconds of any two consecutive packets for connection i, and I is the average inter-packet interval time in milliseconds for connection i. In this chapter, these connections could be either downlinked from a BS to an MSS or uplinked from an MSS to a BS. In the scheduling of downlink packets, our proposed scheme should be implemented on a BS. However, the proposed scheme must be realized on both a BS and an MSS if the proposed scheme is to be applied to the uplink packet scheduler. A BS can know the resource requirements of an MSS through negotiations in the requests from the MSS. Thus, a BS can determine the uplink packet schedule according to the proposed algorithm and provide transmission opportunities to an MSS. When a new connection to an MSS is initiated or any existing connection is released, the proposed scheme is activated to schedule or re-schedule resources in the following frames for the MSS. First, the successive scheduling scheme sorts all connections on an MSS according to their delay constraints and schedules these connections with tight delay requirements. The reason for this is that packets of these connections with tight delay requirements need to be sent or received within a small time window. The scheduler must consider these packets first in order not to violate their QoS requirements. Conversely, for packets that could tolerate more delays, the scheduler can postpone the packets to schedule them successively without violating their delay constraints. After the scheduler decides on the scheduling priorities of connections, the packets from the first priority connection, e.g. connection i, are scheduled. $C_{i,j}$ is represented the j^{th} packet of connection i and the proposed scheme schedules $C_{i,j}$ with following steps: (1) The frames that are within D_i for $C_{i,j}$ and have already scheduled the packets without full-filled frames, called *FIU*. For the various applications, the proposed scheme is based on either the shortest delay or the longest delay. For the shortest delay based, if there are two or more *FIU* for $C_{i,j}$, the *FIU* with shorter delay receives a higher priority for $C_{i,j}$. The shortest delay based is applied to the urgent applications that are very strict with delay requirements and is used to prevent packets loss. Additionally, it is done to reduce the intervals of listen periods and increase the interval of sleep periods. In other words, an MSS cannot sleep in the time slots where there are already schedule packets. Thus, *FIU* is assigned first if the time slots of *FIU* are still available to accommodate $C_{i,j}$. On the other hand, the *FIU* with longer delay receives a higher priority in being scheduled to $C_{i,j}$. The longest delay based applied to scenarios which have loose delay requirements. The BS can decide on which strategy to perform in specific applications. (2) If there is no *FIU* for $C_{i,j}$, the scheduler will then pick a set of frames that are within D_i for $C_{i,j}$ and which already have scheduled packets without any available time slot. These are called *FFU*. The frames in the set are sorted by the D_i for $C_{i,j}$ in ascending order. The *FFU* will be the first frame and last frame in the set with the shortest delay based and longest delay based individuals. In order to reduce the number of status transitions, the scheduler will schedule the packets in successive time slots. In the successive listen periods, the MSS will not enter the sleep periods, and the number of status transitions would be reduced. Additionally, the sleep periods will be longer after their successive listen periods. To schedule the packets successively, the scheduler will find an unused frame that is next to *FFU* and is closer to the next full-filled frame, called F_{next}. The reason for this is that F_{next} is closer to the next full-filled frame has more chances to schedule the listen periods successively. In other words, packets that are scheduled to F_{next} and that is next to *FFU* will become an *FIU*. Obviously, *FIU* gains more opportunities to serve other packets in the following connections. Therefore, *FIU* will become *FFU* after full-filled frame with packets, and this *FFU* will be successive. The listen periods will be continuously without the sleep periods and the number of status transitions would be reduced. (3) If there are no *FIU* and *FFU* within D_i for $C_{i,j}$, the scheduler will schedule the packet into the last unused frame within D_i for $C_{i,j}$ and the unused frame will then become *FIU*. The last unused frame is selected is because once a frame is scheduled to transmit or receive packets, the frame becomes an *FIU*. As we mentioned, an *FIU* has more opportunities to serve other packets in the following connections. After the above steps, the successive scheduling scheme performs packet scheduling and achieves the power-saving for an MSS.

Fig. 1 shows the second step in the second part of the proposed algorithm. Based on the shortest delay, the scheduler chooses the first *FFU* to determine F_{next}. Because the 4th frame is an unused frame and is closer to the next *FFU*, which is the 5th frame, the scheduler determines the 4th frame as F_{next} and schedules the packet into the 4th frame. Once we determine the proper frame to be filled with packets, the time slots for transmission will be more successive for their following connections of scheduling. Thus, the 4th frame becomes *FIU* and has a greater chance to be filled with packets by the proposed algorithm. The status would not be switched from 3rd to 5th frame when the 4th frame is filled up with packets.

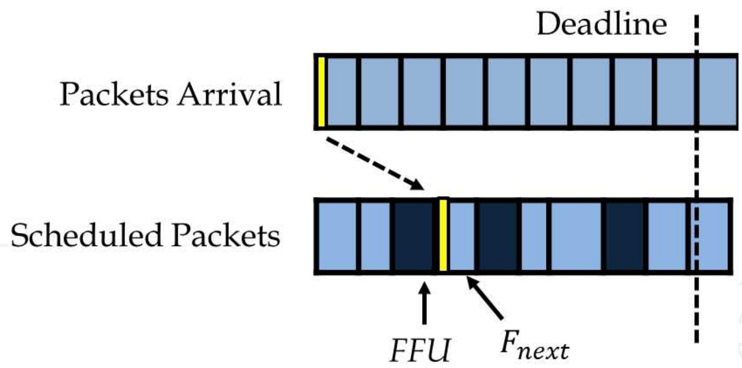


Fig. 1. The shortest delay based scheduling.

Therefore, we can reduce the number of the status transitions by scheduling packets successively and save energy consumption in the status transitions. The longest delay based scheduling is shown in Fig. 2.

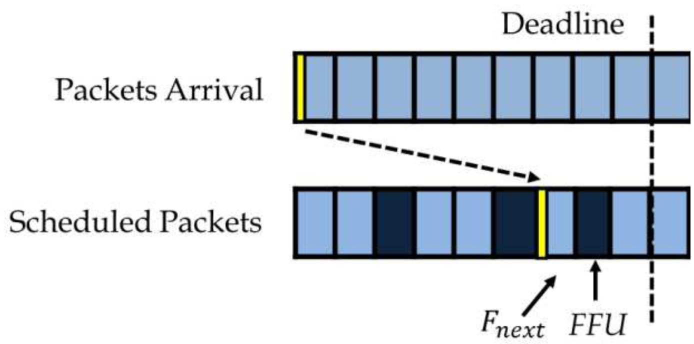


Fig. 2. The longest delay based scheduling.

In Fig. 3, we schedule the packets of connection 1 with the QoS requirement of UGS, and connection 2 with the QoS requirement of RT-VR in an MSS. With the shortest delay based, we schedule the first packet of connection 1 which is $C_{1,1}$. There is no *FIU* or *FFU* in the available frames under this delay constraint. In the third step of the second part in our proposed algorithm, we schedule $C_{1,1}$ into the 5th frame with the maximum delay without violating the constraints, and the 5th frame becomes *FIU*. After that, $C_{1,2}$ is scheduled into *FIU* which is the 5th frame according to the first step in the second part of our algorithm. $C_{1,3}$ and $C_{1,4}$ are also scheduled into *FIU*, which is the 5th frame by the first step in the second part of our algorithm. The 6th packet is scheduled into the 10th frame because there is no *FIU* or *FFU* within D_1 for $C_{1,6}$. The 10th frame becomes *FIU* after $C_{1,6}$ is scheduled inside. The rest packets of connection 1 are scheduled in the same way as are done in previous steps. When we schedule connection 2, the first packet will be scheduled into the 6th frame because there is no *FIU*, while the 5th frame is *FFU*. By the second step in the second part of the algorithm, the F_{next} is the 6th frame. Thus, we schedule $C_{2,1}$ into the 6th frame and $C_{2,2}$ is scheduled into *FIU*, which is the 6th frame. $C_{2,3}$ and $C_{2,4}$ are scheduled into the 9th and 14th frame, respectively. The longest delay based scheduling is shown in Fig. 4. In the result of our examples, our SSS algorithm will schedule the packets into the time slots successively and reduce the number of status transitions in an MSS and minimize the energy consumption of status transitions.

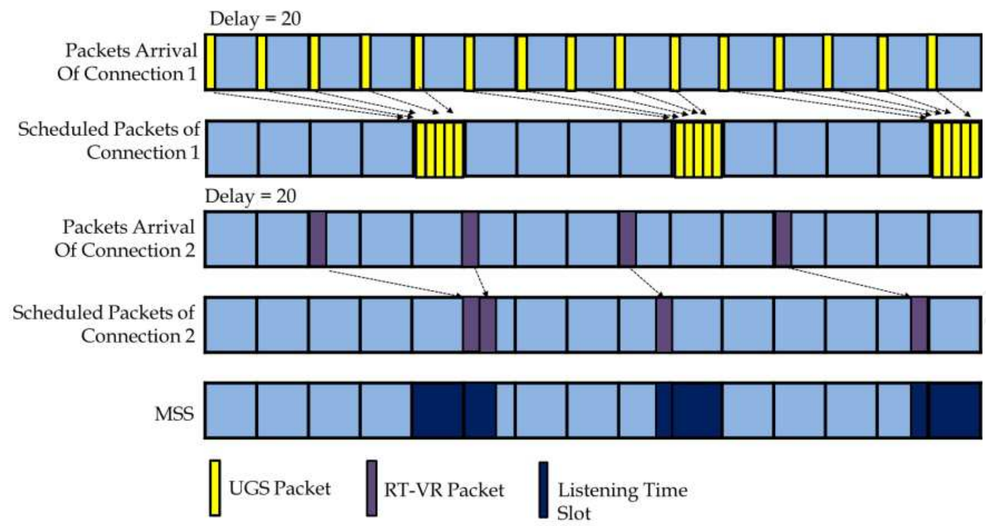


Fig. 3. Our SSS algorithm on the shortest delay based scheduling.

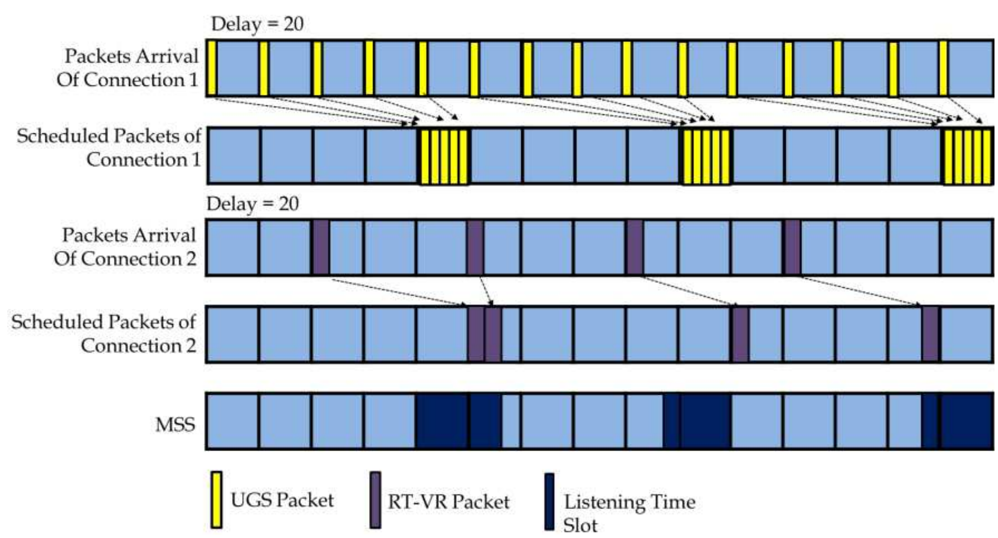


Fig. 4. Our SSS algorithm on the longest delay based scheduling.

2.3.2 The QoS requirements for jitter

In IEEE802.16e broadband wireless access networks, acrucial component of delay is the buffered packet delay between BS and MSS. Due to varying delays in transmission, the delays of scheduling from packet to packet may cause buffered packet delay. This phenomenon is called jitter (Wu & Chen, 2004).

As shown in Fig. 5, we denote $Packet_i$ as the i^{th} packet of certain connections, with the QoS requirement of delay having 7 time slots and 2 jitters. Assume $Packet_{i-1}$ was scheduled in the first time slot, and the delay of $Packet_{i-1}$ is 0. $Packet_i$ may schedule into the time slots of the 2nd time slot to the 8th time slot if we only consider the delay constraint of the QoS requirement. However, it is more realistic to consider the jitter constraint of the QoS requirement. Because the delay of $Packet_{i-1}$ and $Packet_i$ cause jitter, we need to consider the delay of $Packet_i$ to satisfy the jitter constraint. Assume we schedule $Packet_i$ in the 5th time slot, the delay of $Packet_i$ is 3 and the jitter will also 3, and this violates the jitter constraint. Thus, under the jitter

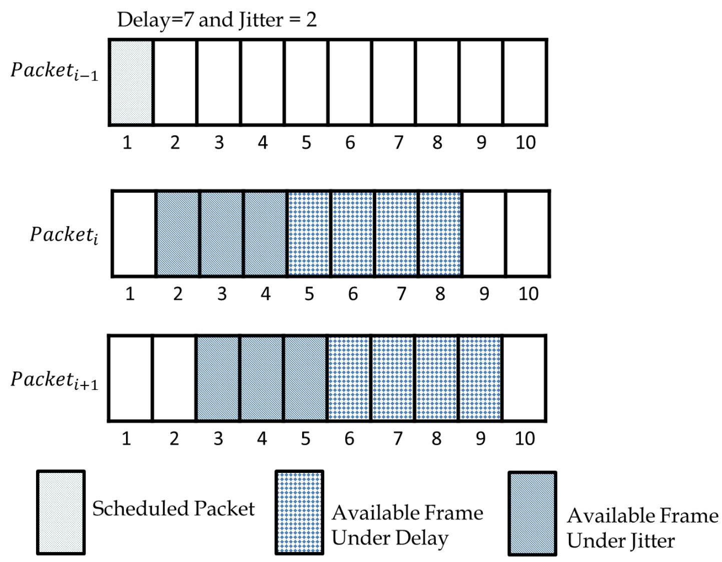


Fig. 5. An example of jitter.

constraint, $Packet_i$ may only schedule into the time slots of the 2nd time slot to the 4th time slot. Assume we schedule $Packet_i$ into the 2nd time slot, $Packet_{i+1}$ may only schedule into the time slots of the 5th time slot to the 7th time slot under the jitter constraint. Thus, the previous approaches to power-saving scheduling with QoS may cause the transmission failure when the jiter constraint is not considered.

QoS requirements include the delay and jitter constraints in scheduling packets. However, previous studies focused on delay constraint without considering the effect of the jitter. Therefore, we take the jitter constraint into account in the scheduling algorithm. In Fig. 6, the first packet was scheduled into the 4th frame which is *FIU* (Tsao & Chen, 2008). Thus, the first packet's delay is 17 and satisfies the delay constraint. The second packet is scheduled into the 4th frame, which is *FIU*. The delay of the second packet is 8 and the jitter between the first and second packet is 9, which satisfies the jitter constraint. The third packet is scheduled into the 9th frame, according to the priorities of the frames. If there is no *FIU*, the first priority will be the frame which has the maximum delay. Therefore, the delay of the third packet would be 20 and the jitter between the second and third packet would be 18, which violates the jitter constraint. Once the scheduling violates the jitter constraint, the QoS is no longer guaranteed.

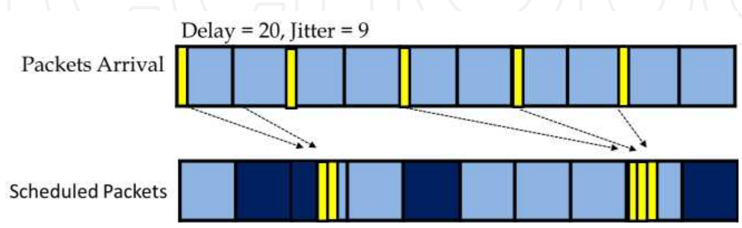


Fig. 6. Example of the scheduling approach (Tsao & Chen, 2008) without considering jitter.

The algorithm of our proposed successive scheduling, which considers jitter constraints, is described in the following two parts. In the first part of our algorithm, the scheduler sorts all connections on an MSS by their delay constraints, and schedules these connections with tight delay requirements. The reason for scheduling connections with tight delay requirements first is to not violate their QoS requirements, as we mentioned previously. The second part of our algorithm is composed of three steps we described in the Section 2.3.1. In addition to these, the scheduler examines the difference in the delay between the present packet and the previous packet when scheduling each packet in each step. The difference in the delay between the present packet and the previous packet can be viewed as jitter. The scheduler schedules the packets to be earlier or later and into the proper time slots in order to satisfy the jitter constraints.

An example of our algorithm is represented in Fig. 7. The first packet is scheduled into the 4th frame, which is *FIU* within D_1 for $C_{1,1}$. Thus, the delay of $C_{1,1}$ is 17. $C_{1,2}$ is scheduled into the 4th frame, which is *FIU* and with a delay of $C_{1,2}$ being 8. Thus, the jitter between $C_{1,1}$ and $C_{1,2}$ is 9, which satisfies the jitter constraint. $C_{1,3}$ is scheduled into the 5th frame according to our algorithm of successive scheduling scheme and with a delay 4. The jitter between $C_{1,2}$ and $C_{1,3}$ is 4, which is smaller than a jitter constraint of 9. $C_{1,4}$ is scheduled into the 7th frame, with a delay of zero time slots and satisfies the jitter constraint of 9 between $C_{1,3}$ and $C_{1,4}$. $C_{1,5}$ is scheduled into the 9th frame with a delay of 4 and the jitter between $C_{1,4}$ and $C_{1,5}$ being 4. Therefore, in order to provide the QoS guarantees of packets scheduling, we need to satisfy the delay and the jitter constraints.

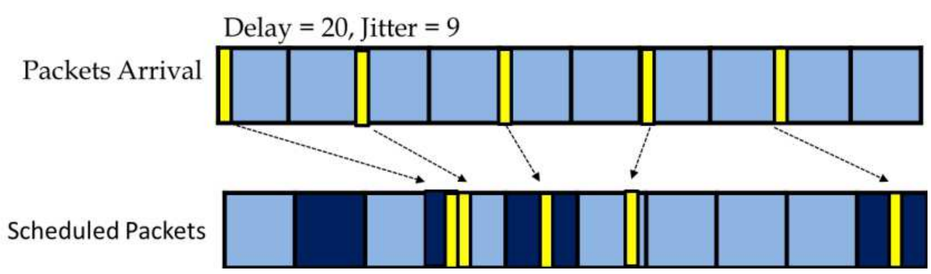


Fig. 7. Example of our SSS algorithm with jitter constraint.

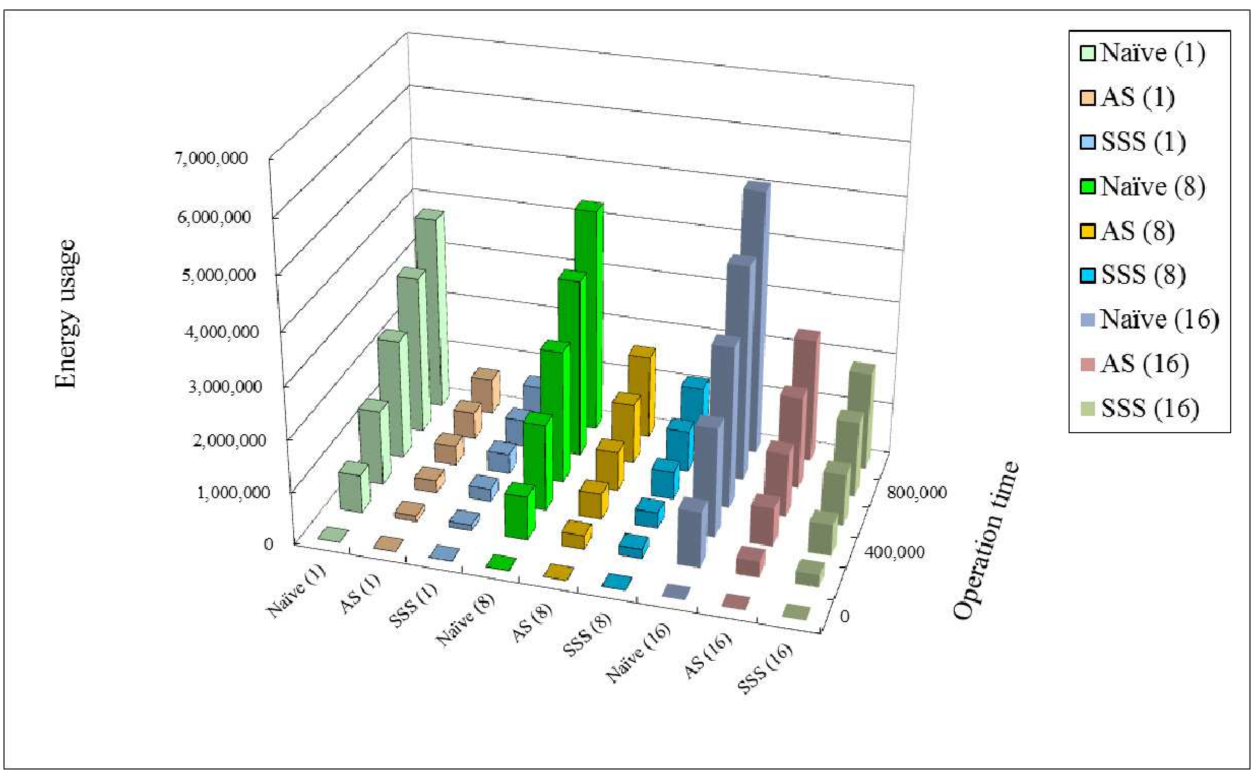
3. Simulation results

This section evaluated the power consumption of an MSS in terms of the number of listen time slots and status transitions. The QoS requirements of *A*, *B*, *C*, and *D* are listed in Table 2. Both connection types *A* and *B* are VoIP connections. Both connection types *C* and *D* are video streaming connections. The first four connection types on the top half of the list are real-time connections that do not consider the tolerated jitter, and the last four connection types are the same as the first four connection types, but with constrained tolerated jitter. The total energy of an MSS is 1,000,000 units. We compare our proposed SSS algorithm with the Naïve approach without optimizations and the AS approach (Tsao & Chen, 2008). The Naïve approach implies that each connection associates with its preferred type of power-saving class and parameters, and minimizs that packet delay and power consumption for that single connection.

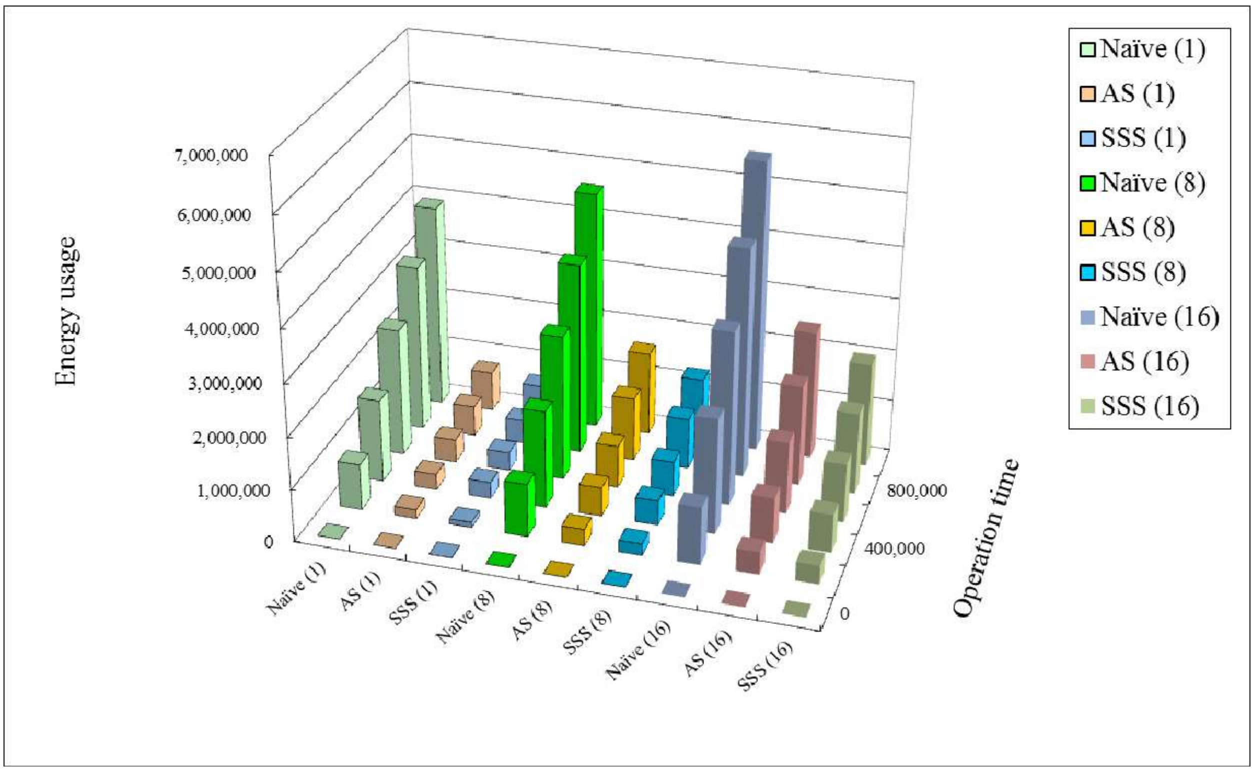
Fig. 8 shows the operation time and energy usage of an MSS by applying three different scheduling schemes in the different connection types with a varied number of connections without the jitter constraints. In the Naïve approach, the energy usage increases faster than the other two approaches. Because the Naïve approach does not consider the optimization of packet scheduling, it results in the enormous energy consumption in status transitions. The energy usage in the AS approach performs the same as our SSS approach when there is only one connection in an MSS. This is because the two approaches maximize the delay of packets scheduling and schedule the packets into minimal listen periods. However, since we consider status transitions in scheduling the packets, our SSS approach chooses successive frames in scheduling the packets and reducing the number of status transitions. When more connections take into account the scheduling, our SSS approach reduces the number of status transitions by successive scheduling. In other words, while successive time slots are scheduled with packets, they do not place the status transitions in the time slots. Thus, our SSS algorithm saves energy and prolongs the operation time in an MSS.

	Service type of QoS	Packets size (bytes)	Interval of packets arrival (ms)	Delay constraint (ms)	Tolerated jitter (ms)
A	UGS	32	10	50	∞
B	UGS	128	10	50	∞
C	RT-VR	512	30	100	∞
D	RT-VR	1024	30	100	∞
A′	UGS	32	10	50	10
B′	UGS	128	10	50	10
C′	RT-VR	512	30	100	20
D′	RT-VR	1024	30	100	20

Table 2. QoS parameters of four real-time connections.

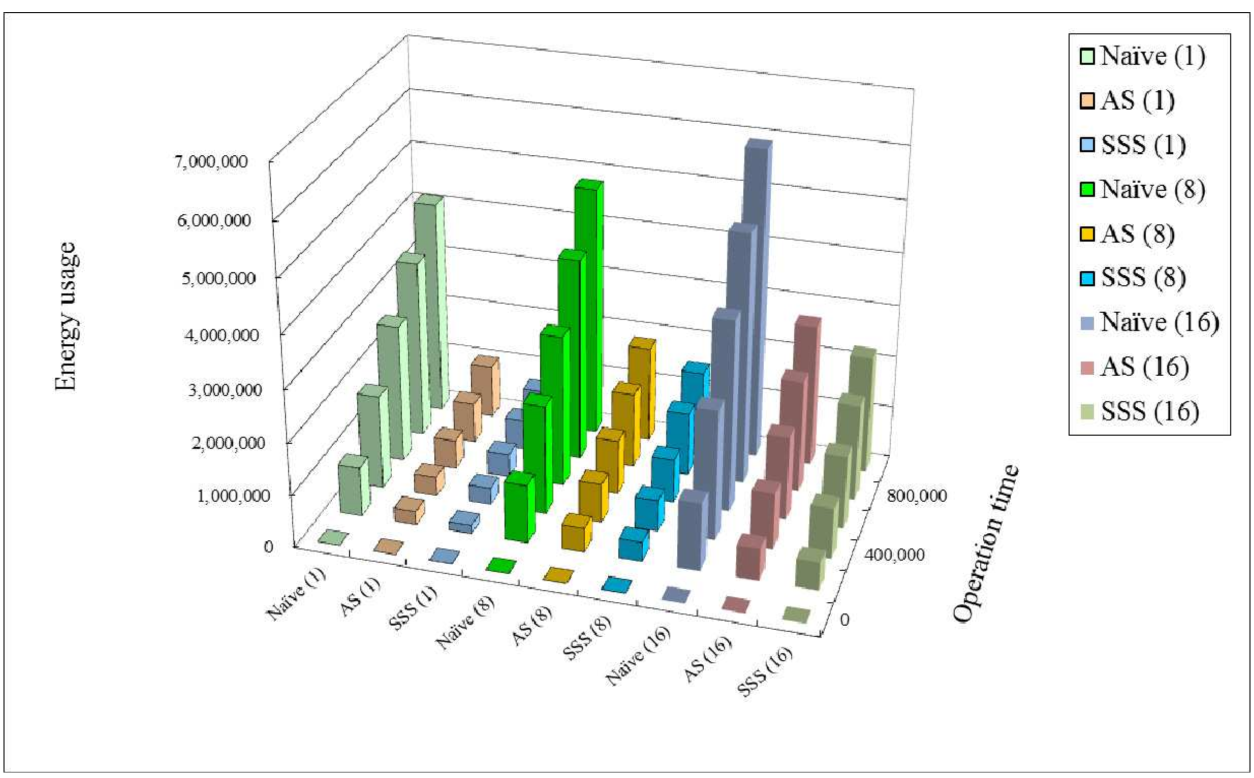


(a)



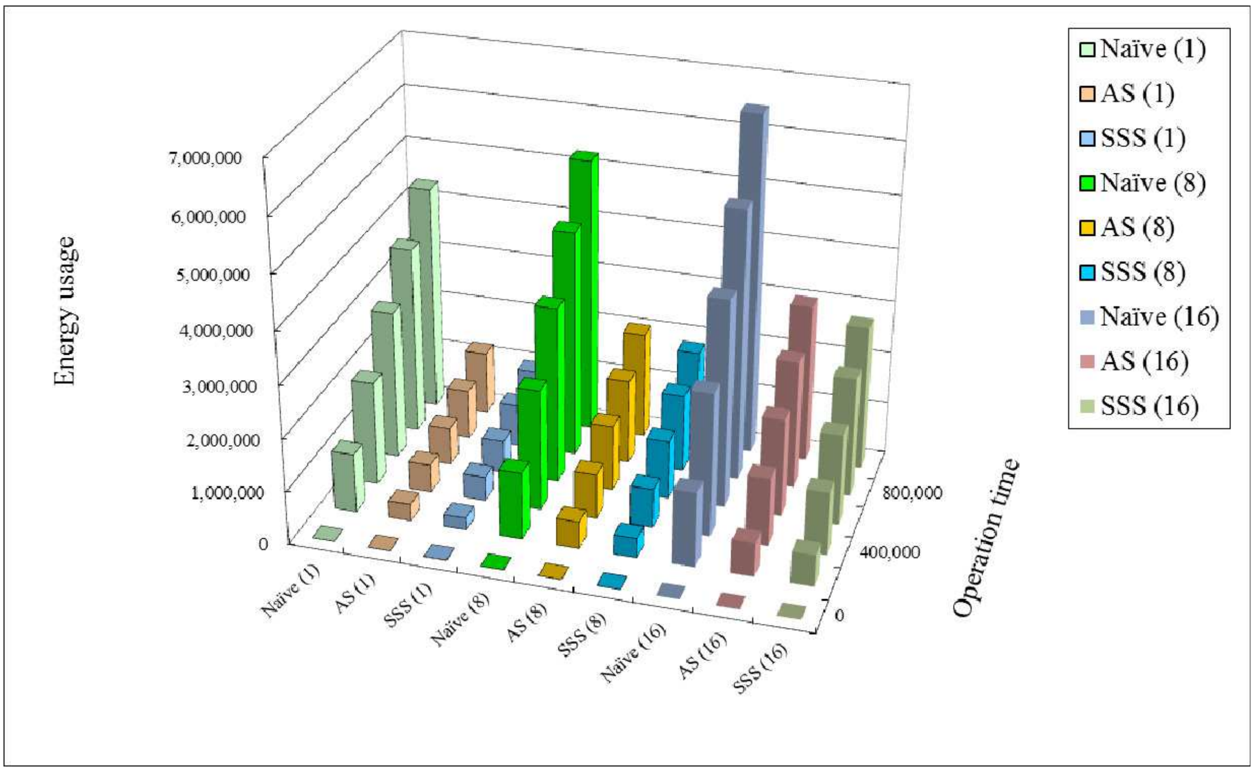
Naïve (1)
AS (1)
SSS (1)
Naïve (8)
AS (8)
SSS (8)
Naïve (16)
AS (16)
SSS (16)
Energy usage
7,000,000
6,000,000
5,000,000
4,000,000
3,000,000
2,000,000
1,000,000
0
Operation time
800,000
400,000
0

(b)



Naïve (1)
AS (1)
SSS (1)
Naïve (8)
AS (8)
SSS (8)
Naïve (16)
AS (16)
SSS (16)
Energy usage
7,000,000
6,000,000
5,000,000
4,000,000
3,000,000
2,000,000
1,000,000
0
Operation time
800,000
400,000
0

(c)



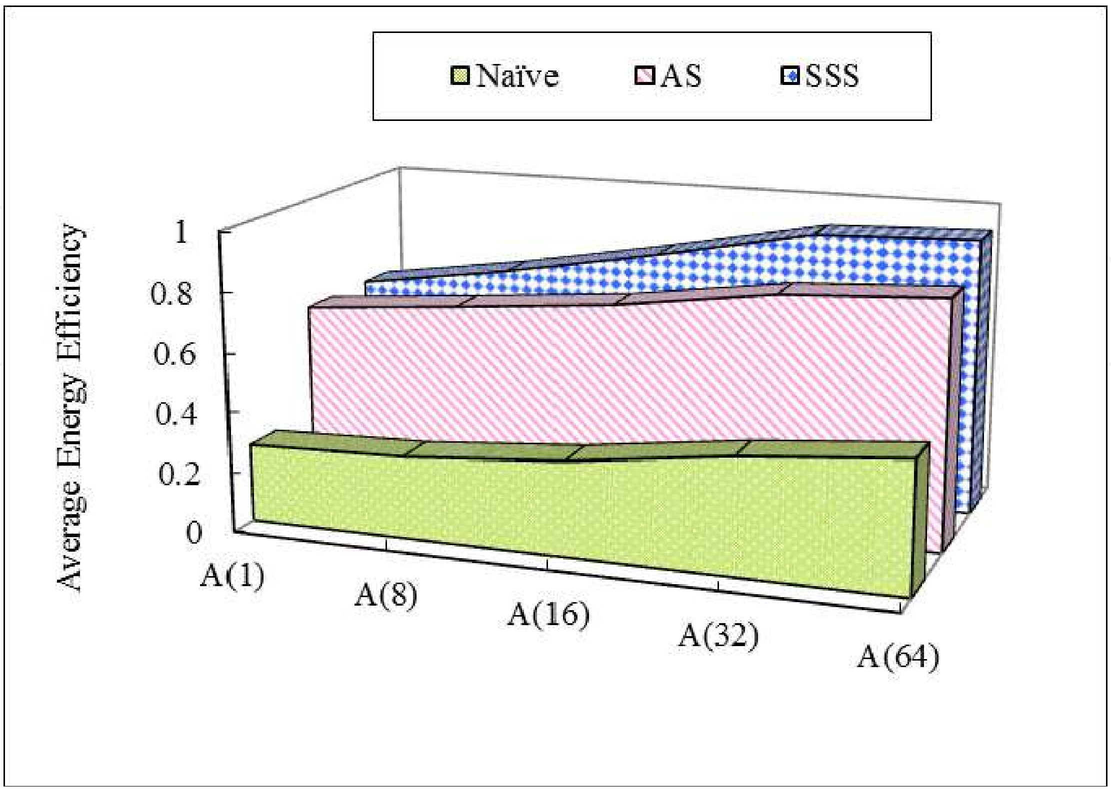
(d)

Fig. 8. The operation time and energy usage of an MSS for three schemes with four connection types with a varied number of connections: (a) connection type *A*, (b) connection type *A*+*B*, (c) connection type *A*+*B*+*C*, and (d) connection type *A*+*B*+*C*+*D*.

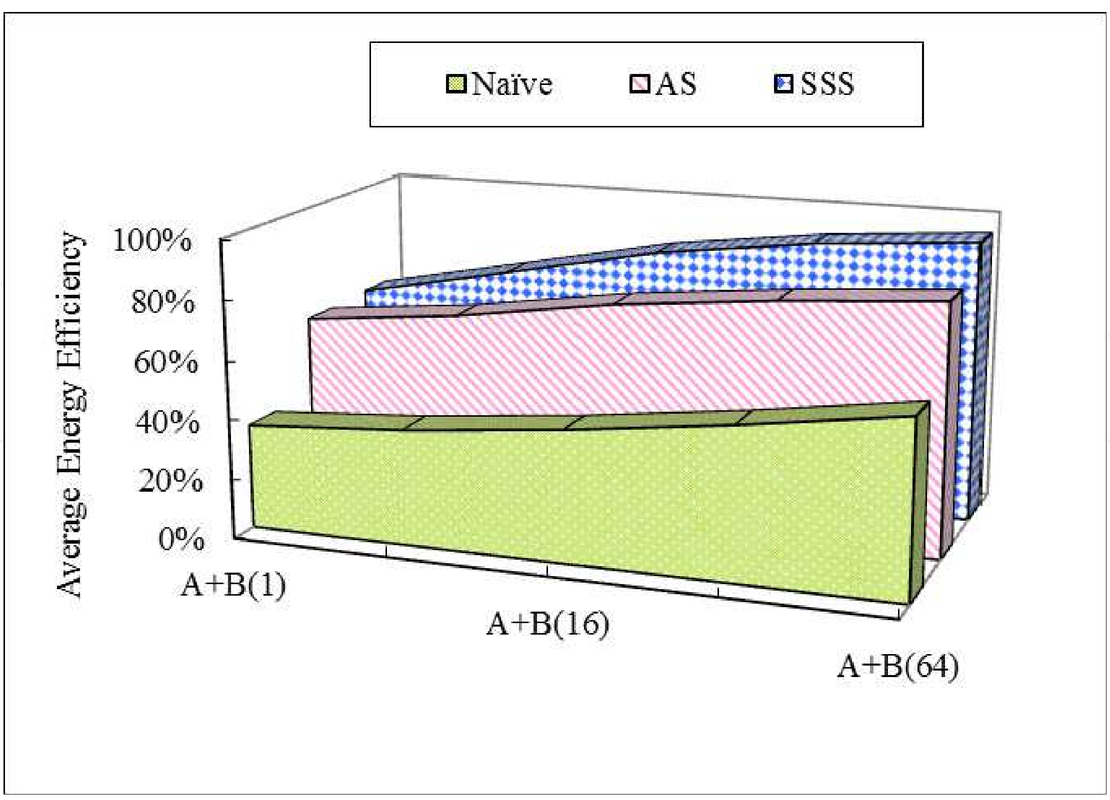
Fig. 9 shows the average energy efficiency of an MSS by applying three different scheduling schemes for different connection types with a varied number of connections without the jitter constraints. We defined E_{trans} as the energy usage for the packet transmission of an MSS during a time period *T*; E_{total} represents the total energy usage in an MSS during *T*. The average energy efficiency (*AEE*) for an MSS during *T* can be represented as follows:

$$AEE = E_{trans} / E_{total} \quad (2)$$

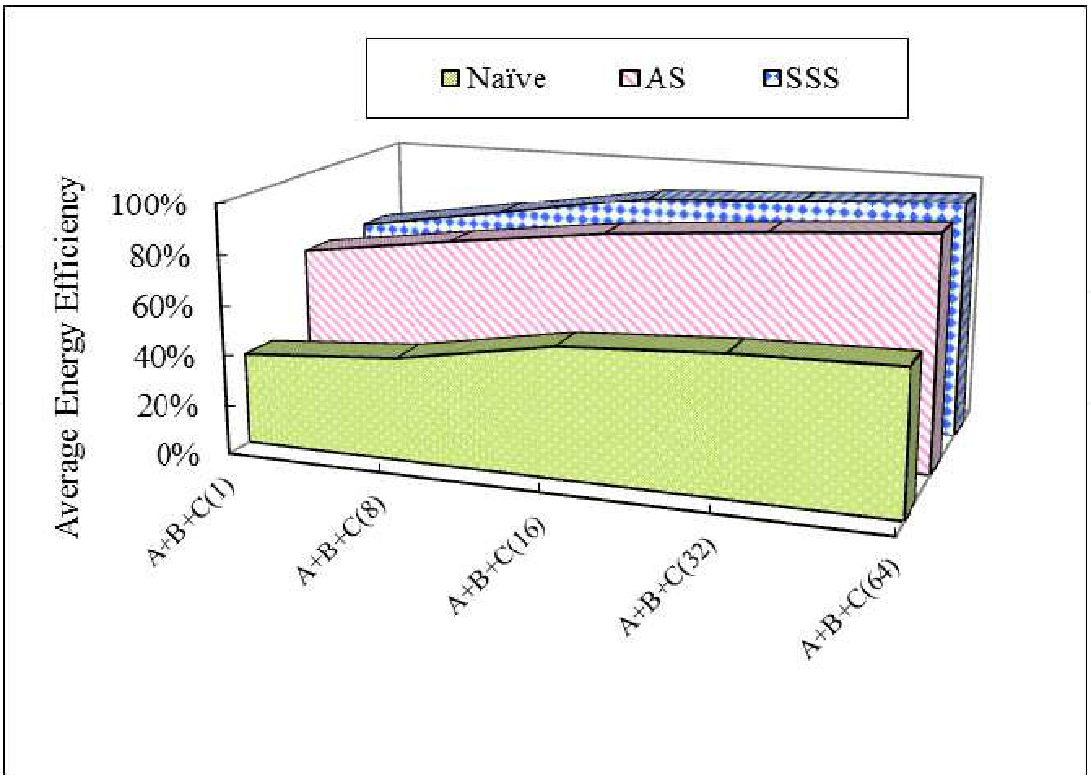
In the Naïve approach, the average energy efficiency is lower than the other two approaches. This is because the Naïve approach processes packets immediately when they arrive, so number of status transitions increase enormously. The energy for status transitions reduce the energy usage for packet transmission from the total energy usage in an MSS. In our SSS algorithm, the average energy efficiency performed the same as the AS approach, where there is only one connection in an MSS. The reason for this is the same as the previous simulation matrix. When there is only one connection in an MSS, the two approaches maximize the delay in packet scheduling and schedules the packets into their minimal listen periods without violating the delay constraints. Thus, the number of status transitions is the same. However, the average energy efficiency in our SSS approach grows up when the number of connections increases. This is because the packets are scheduled more successively when the packets are small, and the number of connections grows large under our proposed algorithm. Fig. 9(c) and (d) reveal that, when the transmission loading encounters a bottleneck, the average energy efficiency stops increasing.


Naïve
AS
SSS
Average Energy Efficiency
1
0.8
0.6
0.4
0.2
0
A(1)
A(8)
A(16)
A(32)
A(64)

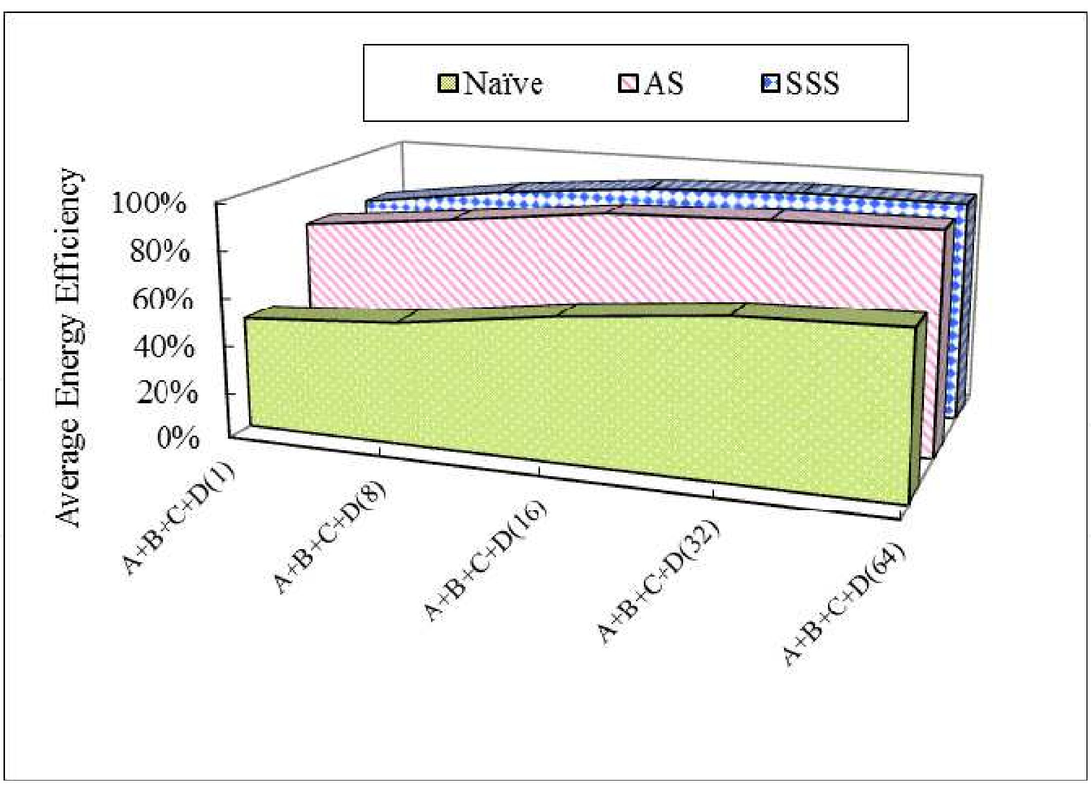
(a)


Naïve
AS
SSS
Average Energy Efficiency
100%
80%
60%
40%
20%
0%
A+B(1)
A+B(16)
A+B(64)

(b)



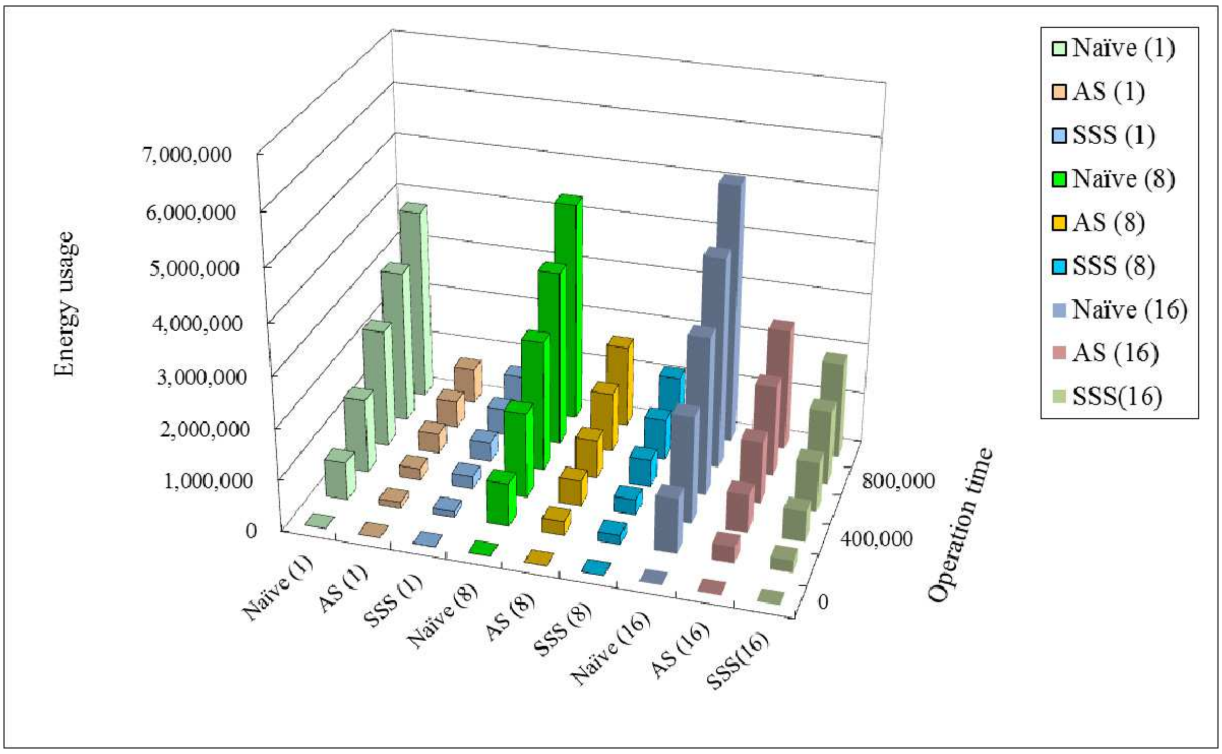
(c)



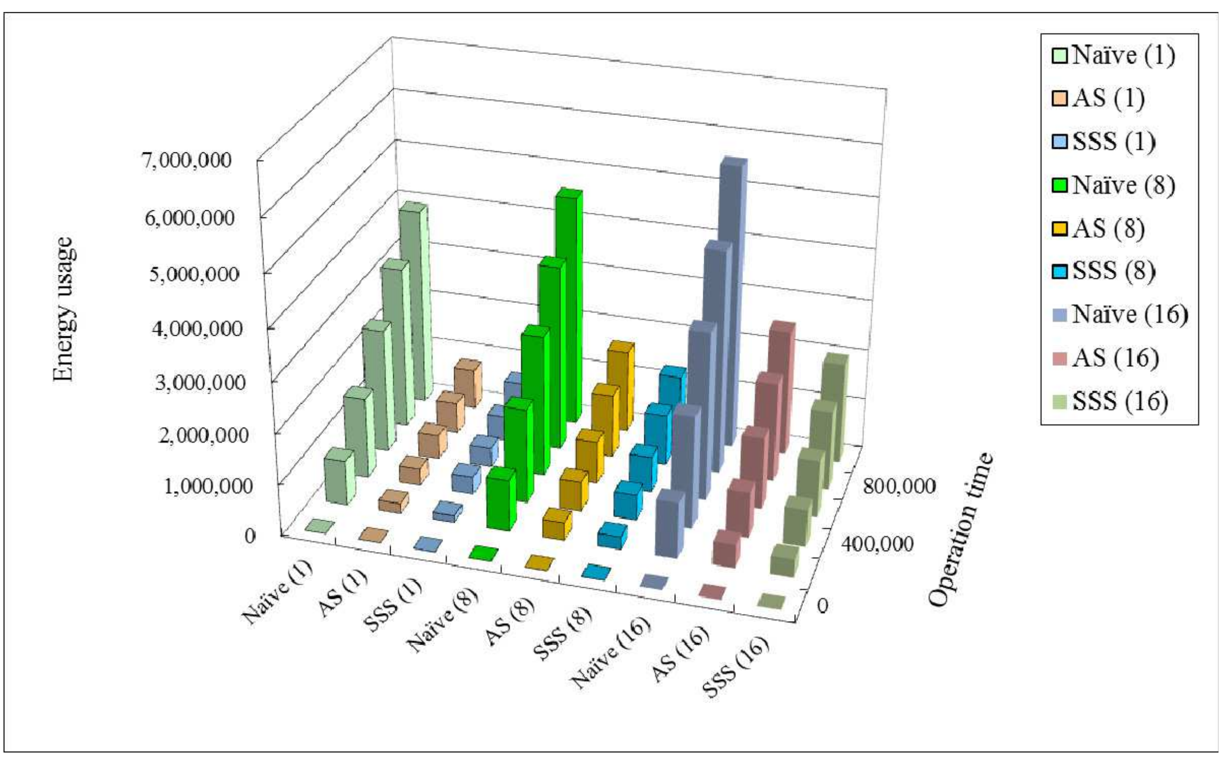
(d)

Fig. 9. The average energy efficiency of an MSS with three schemes and four connection types with a varied number of connections: (a) connection type *A*, (b) connection type *A*+*B*, (c) connection type *A*+*B*+*C*, and (d) connection type *A*+*B*+*C*+*D*.

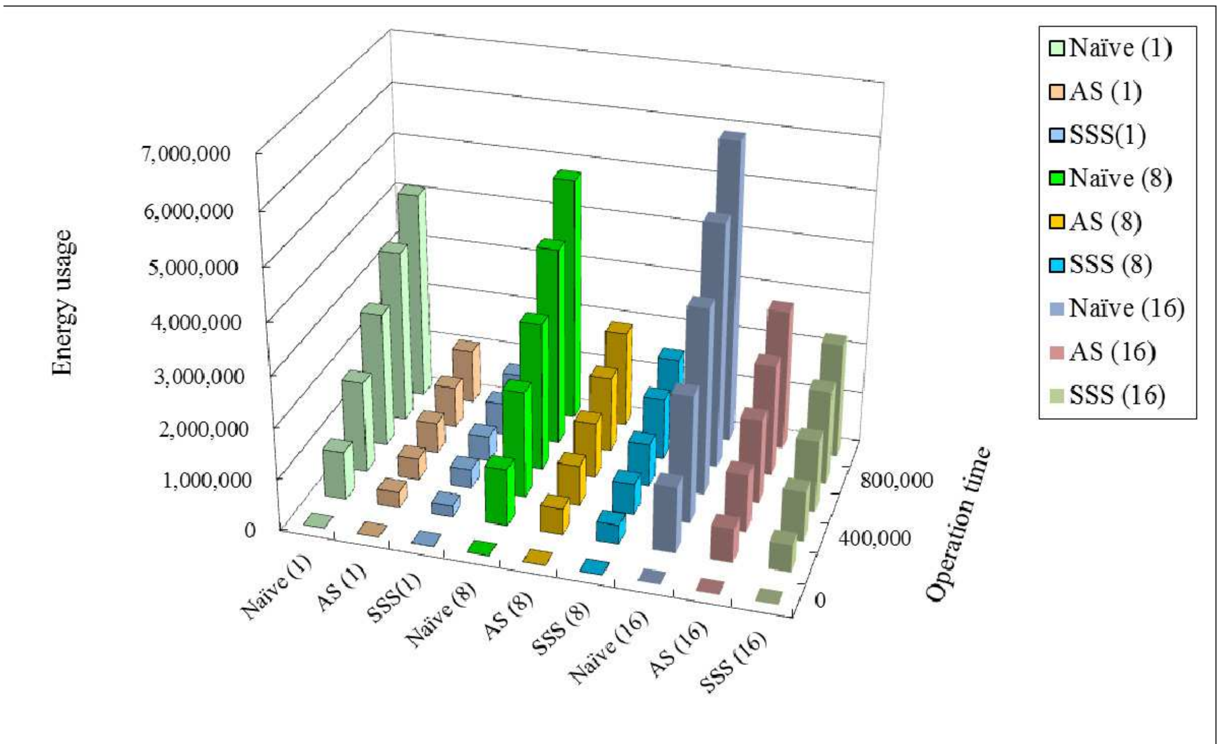
Fig. 10 shows the operation time and energy usage of an MSS by applying three different scheduling schemes for different connection types with a varied numbers of connections with the jitter constraints. The energy usage of different three approaches is higher than the one that does not consider the jitter constraints. The reason for this is that the process is limited



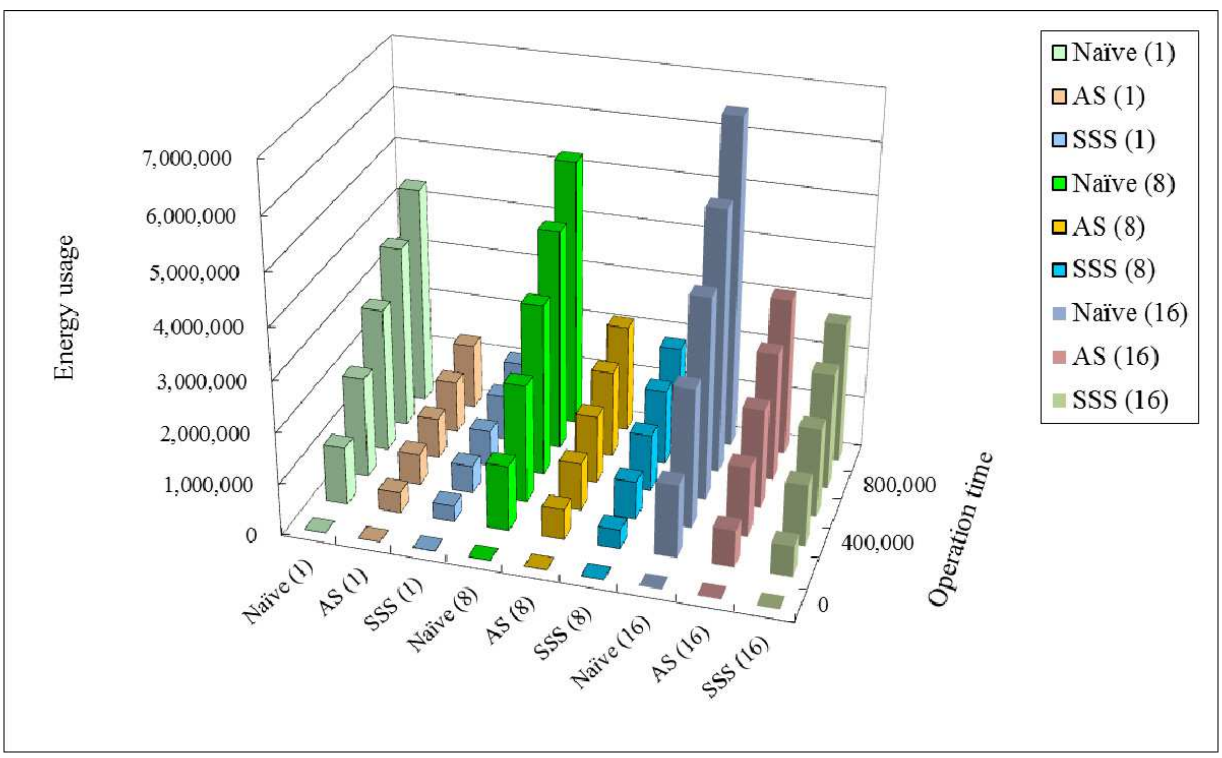
(a)



(b)



(c)

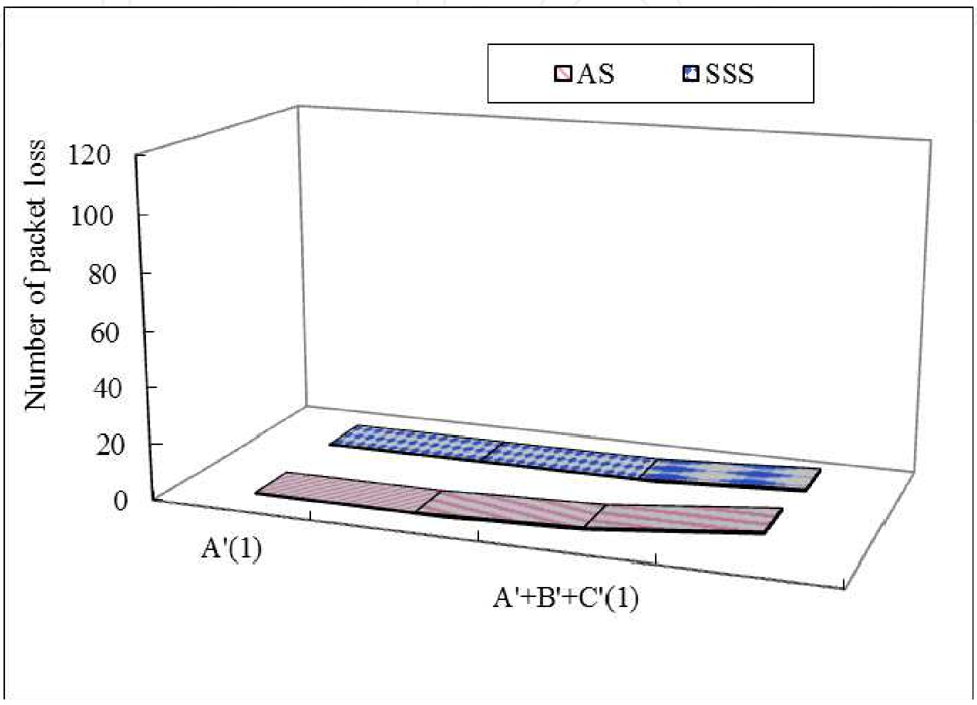


(d)

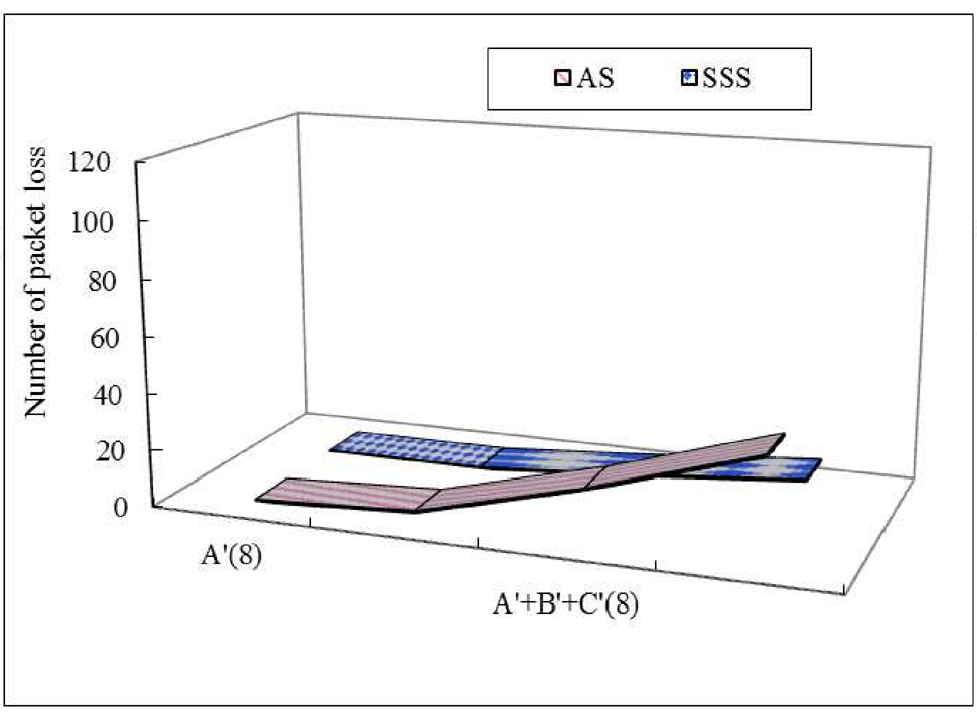
Fig. 10. The operation time of an MSS with three schemes and four connection types with a varied number of connections with jitter constraints: (a) connection type *A*′, (b) connection type *A*′+*B*′, (c) connection type *A*′+*B*′+*C*′, (d) connection type *A*′+*B*′+*C*′+*D*′.

by the jitter constraints, and the limited scheduling increases the number of status transitions. In our SSS approach, the energy usage is lower than the other two approaches under the same connection types. That is because the more connections gain the more chances to be scheduled successively, so the energy consumption of status transitions is reduced.

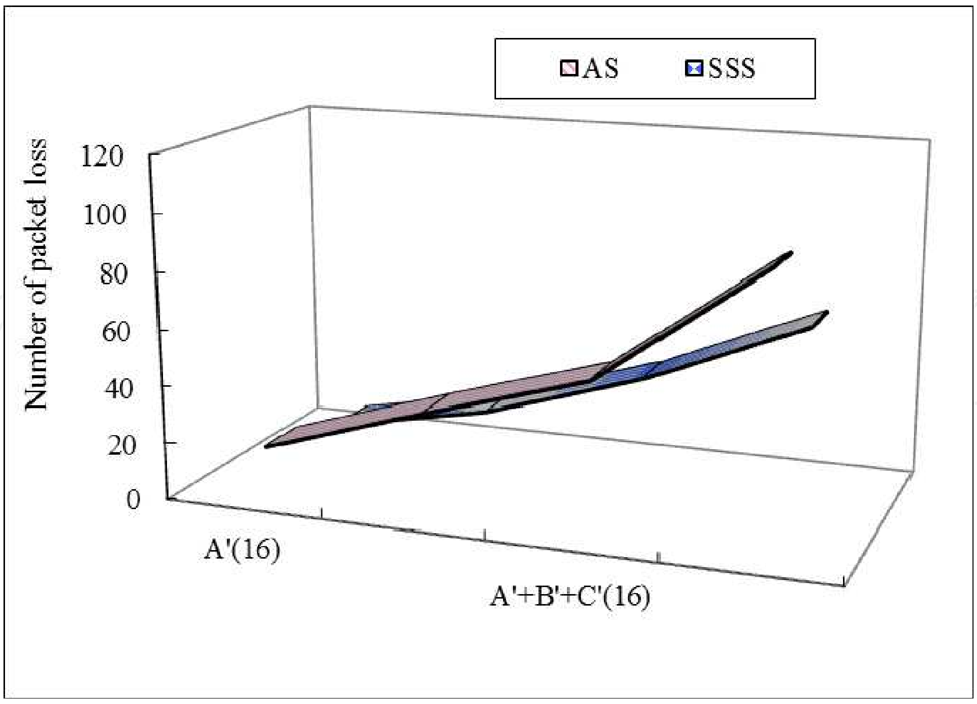
Fig. 11 shows the amount of packet loss of an MSS which applies two different scheduling schemes for different connection types with a varied number of connections with the jitter



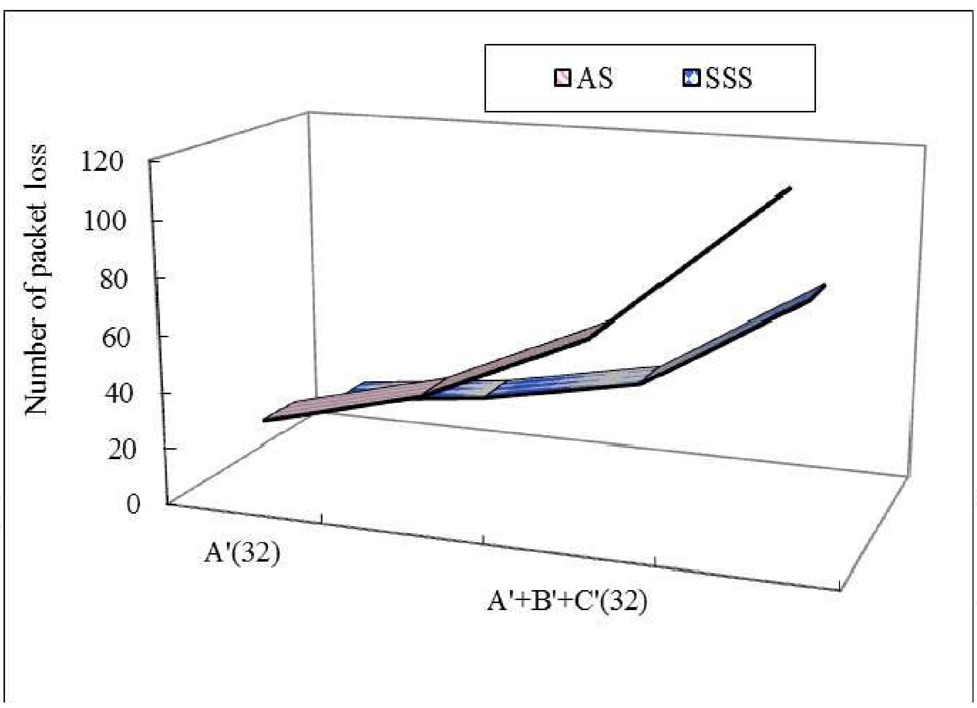
(a)



(b)



(c)

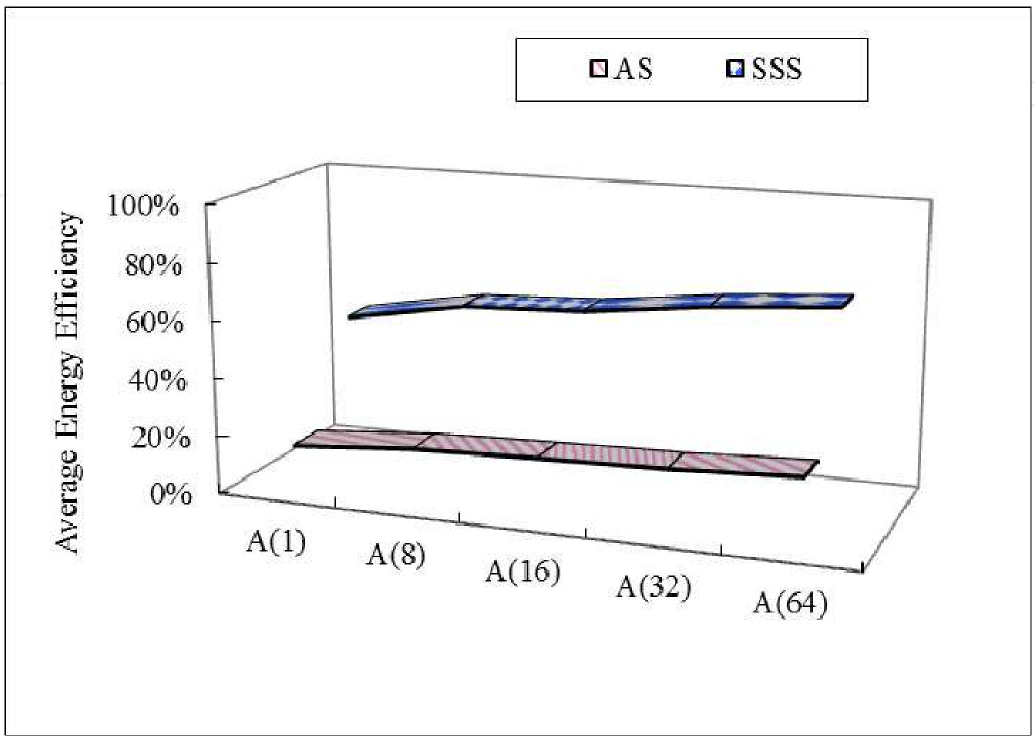


(d)

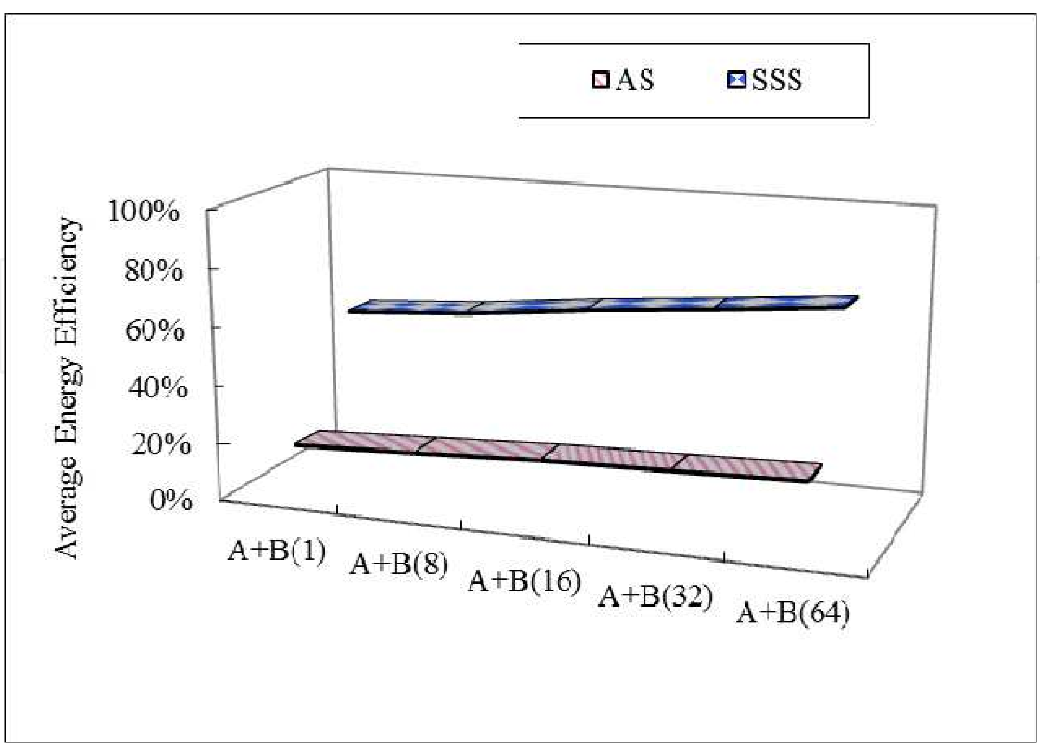
Fig. 11. The amount of packet loss of an MSS with two schemes and four connection types with a varied number of connections: (a) with 1 connection, (b) with 8 connections, (c) with 16 connections, and (d) with 32 connections.

constraints. We only compare the SSS and the AS approaches, which delay the packets, when processing the scheduling. The amount of packet loss is increased when the packet load is raised. In our SSS approach, the number of packet loss is minimized by the algorithm that considers jitter constraints. The scheduler chooses the proper time slots to schedule the packets in order so as not violate the jitter constraints between each packet.

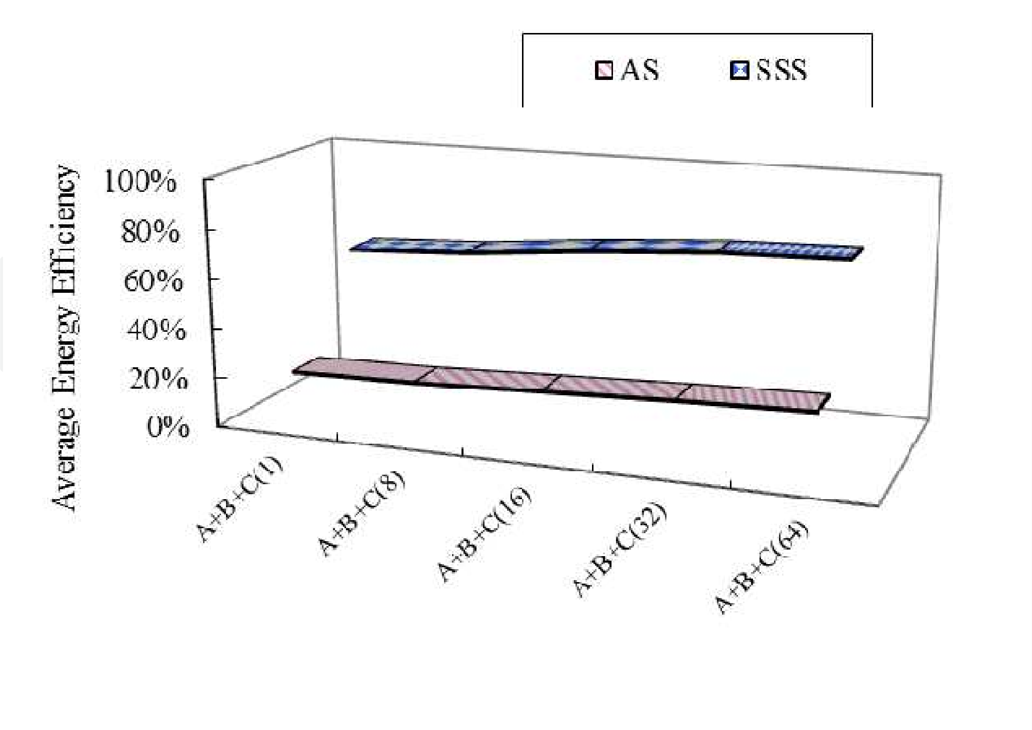
Fig. 12 shows the average energy efficiency of an MSS by applying two different scheduling schemes for different connection types with a varied number of connections with jitter



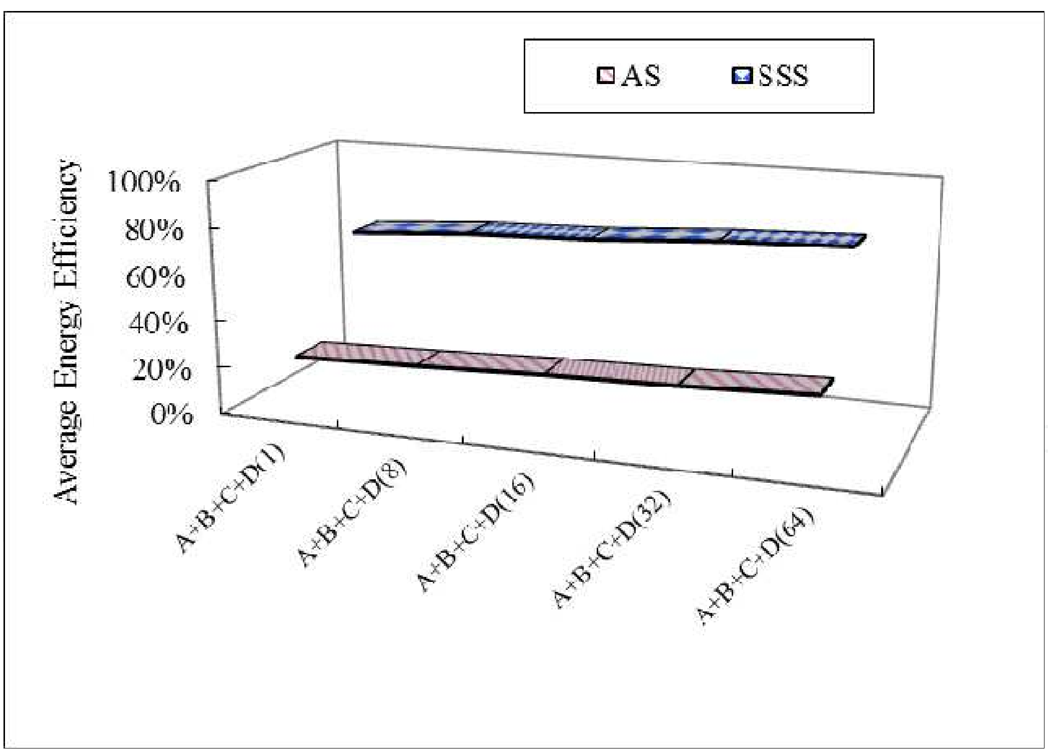
(a)



(b)



(c)



(d)

Fig. 12. The average energy efficiency of an MSS with two schemes and four connection types with a varied number of connections under jitter constraints: (a) connection type *A′*, (b) connection type *A′*+*B′*, (c) connection type *A′*+*B′*+*C′*, and (d) connection type *A′*+*B′*+*C′*+*D′*.

constraints. In this simulation, we only compare the SSS and AS approaches, which delay the packets when processing the scheduling. In our SSS algorithm, the average energy efficient is better than the performance of the AS approach. Due to QoS constraints, the available time slots for scheduling was limited by the delay and jitter constraints. Aside from the energy usage of status transitions, the packets will not be delivered if the scheduling violates the delay and jitter constraints. Meanwhile, the AS does not take the jitter constraints into account when they scheduling the packets. Thus, our SSS approach transmits more packets than the AS, and the average energy efficient in our SSS approach is better than the AS.

4. Conclusion

An energy-efficient scheduling scheme to improve the energy efficiency and guarantee Quality of Service in IEEE 802.16e was proposed. The previous literature only considers the delay constraint of QoS requirement in one MSS. We first consider both the jitter and delay constraints of QoS requirement to schedulethe real-time connections in one MSS. Our proposed algorithmis to schedule the packet transmission in successively fashion with the minimal interval of listen periods and maximal interval of sleep periods without violating the QoS of all connections in an MSS. Additionally, the successive scheduling of time slots would reduce the number of status transitions between the sleep periods and listen periods. The proposed approach can be adapted to the power-saving class of type III where the length of sleep and listen periods arevariable. Simulation results show that, incomparison with the AS and Naïve schemes, the proposed SSS scheduling algorithm can result in a significant overall energy saving and can guarantee the delay and jitter QoS.

5. References

Andrews, M.; Qian, L. & Stolyar, A. (2005). Optimal Utility Based Multi-user Throughput Allocation Subject to Throughput Constraints, *IEEE INFOCOM*, pp. 2415- 2424, 2005.

Fang, G.; Dutkiewicz, E.; Sun, Y.; Zhou, J.; Shi, J. & Li, Z. (2006). Improving Mobile Station Energy Efficiency in IEEE 802.16e WMAN by Burst Scheduling, *IEEE International Conference on Global Telecommunications Conference (GLOBECOM'06)*, pp. 1-5, 2006.

Han, K. & Choi, S. (2006). Performance Analysis of Sleep Mode Operation in IEEE 802.16e Mobile Broadband Wireless Access Systems, *IEEE 63rd Vehicular Technology Conference (VTC 2006-Spring)*, pp. 1141-1145, 2006.

Huang, S.-C.; Jan, R.-H. & Chen, C. (2007). Energy Efficient Scheduling with QoS Guarantee for IEEE 802.16e Broadband Wireless Access Networks, *International Conference on Wireless Communications and Mobile Computing (IWCMC'07)*, pp. 547-552, 2007.

IEEE Std 802.16-2004 (2004). IEEE Standard for Local and Metropolitan Area Networks. Part 16: Air Interface for Fixed Broadband Wireless Access Systems, 2004.

IEEE Std 802.16e-2005 (2005). IEEE Standard for Local and Metropolitan Area Networks. Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, 2005.

Jang, J.; Han, K. & Choi, S. (2006). Adaptive Power Saving Strategies of IEEE 802.16e Mobile Broadband Wireless Access, *Asia-Pacific Conference on Communications (APCC'06)*, pp. 1-5, 2006.

Lei, K. & Tsang, D.H.K. (2006). Performance Study of Power Saving Classes of Type I and II in IEEE 802.16e, *IEEE Conference on Local Computer Networks*, pp. 20-27, 2006.

Li, B.; Qin, Y.; Low, C.P. & Gwee, C.L. (2007). A Survey on Mobile WiMAX, *IEEE Communications Magazine*, Vol. 45, No. 12, pp. 70-75, Dec. 2007.

Liao, W.-H. & Wang, H.-H. (2008). An Asynchronous MAC Protocol for Wireless Sensor Networks, *Journal of Network and Computer Applications*, Vol. 31, No. 4, pp. 807-820, 2008.

Liu, M. & Liu, M. T. (2003). A Power-Saving Scheduling for IEEE 802.11 Mobile Ad Hoc Network, *IEEE International Conference on Computer Networks and Mobile Computing(ICCNMC 2003)*, pp.238 – 245, 2003.

Mukherjee, S.; Leung, K.K. & Rittenhouse, G. E. (2005). Protocol and Control Mechanisms to Save Terminal Energy in IEEE 802.16 Networks, *IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing (PACRIM)*, pp. 5-8, 2005.

Seo, J.B.; Lee, S.Q.; Park, N.H.; Lee, H.W. & Cho, C.H. (2004). Performance Analysis of Sleep Mode Operation in IEEE 802.16e, *IEEE 60th Vehicular Technology Conference, 2004 (VTC2004-Fall)*, pp. 1169-1173, 2004.

Tian, L.; Yang, Y.; Shi, J.; Dutkiewicz, E. & Fang, G. (2007). Energy Efficient Integrated Scheduling of Unicast and Multicast Traffic in 802.16e WMANs, *IEEE International Conference on Global Telecommunications Conference (GLOBECOM'07)*, pp. 3478-3482, 2007.

Tsao, S.-L. & Chen, Y.-L. (2008). Energy-Efficient Packet Scheduling Algorithms for Real-Time Communications in a Mobile WiMAX System, *Computer Communications*, Vol. 31, No. 10, pp. 2350-2359, 2008.

Tseng, Y.-C.; Hsu, C.-S. & Hsieh, T.-Y. (2002). Power-Saving Protocols for IEEE 802.11-Based Multi-Hop Ad Hoc Networks, *IEEE INFOCOM*, pp. 200-209, 2002.

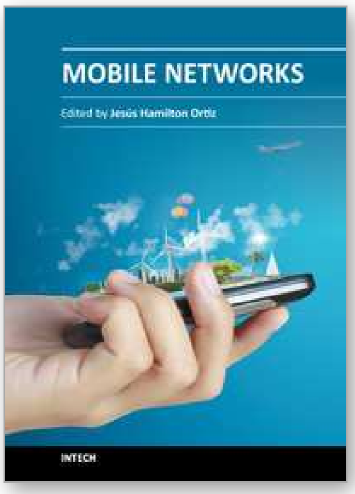
Wu, E. H.-K. & Chen, M.-Z. (2004). JTCP: Jitter-Based TCP for Heterogeneous Wireless Networks, *IEEE Journal on Selected Areas in Communications*, Vol. 22, No. 4, pp. 757-766, May 2004.

Wongthavarawat, K. & Ganz, A. (2003). Packet Scheduling for QoS Support in IEEE 802.16 Broadband Wireless Access Systems, *International Journal of Communication Systems*, Vol. 16, No. 1, pp. 81-96, Feb. 2003.

Ye, W.; Heidemann, J. & Estrin, D. (2004). Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks, *IEEE/ACM Transactions on Networking*, Vol. 12, No. 3, pp. 493-506, Jun. 2004.

Zheng, T.; Radhakrishnan, S. & Sarangan, V. (2005). PMAC: An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks, *IEEE International Parallel and Distributed Processing Symposium (IDPDS'05)*, pp. 65-72, 2005.

Zhu, H. & Cao, G. (2004). A Power-Aware and QoS-Aware Service Model on Wireless Networks, *IEEE INFOCOM*, pp. 1393-1403, 2004.



Mobile Networks
Edited by Dr. Jesús Ortiz

ISBN 978-953-51-0593-0
Hard cover, 192 pages
Publisher InTech
Published online 09, May, 2012
Published in print edition May, 2012

The growth in the use of mobile networks has come mainly with the third generation systems and voice traffic. With the current third generation and the arrival of the 4G, the number of mobile users in the world will exceed the number of landlines users. Audio and video streaming have had a significant increase, parallel to the requirements of bandwidth and quality of service demanded by those applications. Mobile networks require that the applications and protocols that have worked successfully in fixed networks can be used with the same level of quality in mobile scenarios. Until the third generation of mobile networks, the need to ensure reliable handovers was still an important issue. On the eve of a new generation of access networks (4G) and increased connectivity between networks of different characteristics commonly called hybrid (satellite, ad-hoc, sensors, wired, WIMAX, LAN, etc.), it is necessary to transfer mechanisms of mobility to future generations of networks. In order to achieve this, it is essential to carry out a comprehensive evaluation of the performance of current protocols and the diverse topologies to suit the new mobility conditions.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821