Optimality of Frame Aggregation-Based Power-Saving Scheduling Algorithm for Broadband Wireless Networks

Wen-Jiunn Liu, *Student Member, IEEE*, Kai-Ten Feng, *Senior Member, IEEE*, and Po-Hsuan Tseng, *Member, IEEE*

***Abstract*—The limitation on battery lifetime has been a critical issue for the advancement of mobile computing. Different types of power-saving techniques have been proposed in various fields. In order to provide feasible energy-conserving mechanisms for the mobile subscriber stations (MSSs), three power-saving types have been proposed for the IEEE 802.16e broadband wireless networks. However, these power-saving types are primarily targeting for the cases with a single connection between the base station (BS) and the MSS. With the existence of multiple connections, the power efficiency obtained by adopting the conventional scheduling algorithm can be severely degraded. In this paper, with the consideration of multiple connections and their quality-of-service (QoS) constraints, a frame aggregation-based power-saving scheduling (FAPS) algorithm is proposed to enhance the power efficiency by aggregating multiple under-utilized frames into fully-utilized ones. The optimality on the minimum number of listen frames in the proposed FAPS algorithm is also provided, and is further validated via the correctness proofs. Performance evaluation of proposed FAPS scheme is conducted and compared via simulations. Simulation results show that the power efficiency of FAPS algorithm outperforms the other existing protocols with tolerable frame delay.**

***Index Terms*—Power-saving, scheduling, frame aggregation, optimality, broadband wireless networks.**

I. INTRODUCTION

THE IEEE 802.16 standard for the broadband wireless networks (BWNs) is designed to fulfill various demands for higher capacity, higher data rate, and advanced multimedia services [1]–[5]. In order to prolong the battery lifetime of the mobile subscriber stations (MSSs), the design of a feasible power-saving mechanism is considered a major issue in the IEEE 802.16e standard [6]–[9]. Three power-saving types are specified, i.e., types I, II, and III, in the IEEE 802.16e point-to-multipoint (PMP) mode such as to meet different demands of traffic between the base station (BS) and the MSSs. Two specific time intervals are defined as the sleep interval for saving energy and the listen interval for listening to the BS and conducting packet transmission. The power-saving class of type I defines the exponential-growing sleep intervals associated with fixed listen intervals. On the other hand, periodic occurrences of both the sleep and listen intervals are considered in type II. The power-saving class of type III consists of pre-determined longer sleep intervals without the existence of the listen period.

There are significant amount of research work [10]–[17] focusing on the energy-saving issues for battery-powered mobile devices. Different types of energy-efficient algorithms have been studied in [10] for generic central-controlled wireless data networks. Based on the IEEE 802.11 power-saving mechanism [11], several energy conservation schemes have been proposed in both centralized [12]–[16] and decentralized [17] manners. However, these existing techniques are not intended to satisfy the requirements as defined in the IEEE 802.16e standard. In recent research studies, performance analysis of the different IEEE 802.16e power-saving types are investigated. Most of the work concentrates on constructing the analytical models for the power-saving class of type I [18]–[22]; while the enhanced model as proposed in [23] switches the power-saving class between types I and II according to the network traffic. A longest virtual burst first (LVBF) scheduling algorithm has been proposed in [24], which considers both the energy conservation and resource allocation between the BS and multiple MSSs. In [25], the optimal traffic indication interval is proposed to replace the original negotiation messages. A power-saving mechanism with periodic traffic indications is proposed to reduce the delay for state transition in order to enhance the power-saving efficiency. Nevertheless, these analytical results and scheduling schemes only assume a single connection between the BS and each MSS, i.e., a single connection is assigned to each MSS.

In view of the multi-connection scenarios, connection-oriented methods have been investigated in different studies [26]–[31]. The maximum unavailability interval (MUI) scheme [26][27] is designed for the connections of power-saving class of type II in the IEEE 802.16e standard. Based on the Chinese remainder theorem, the proper start frame will be identified for each type II connection in order to reduce total

Manuscript received October 5, 2012; revised March 22 and September 21, 2013; accepted November 19, 2013. The associate editor coordinating the review of this paper and approving it for publication was G. Bianchi.

W.-J. Liu and K.-T. Feng (corresponding author) are with the Department of Electrical and Computer Engineering, National Chiao Tung University, Hsinchu, Taiwan (e-mail: jiunn.cm94g@nctu.edu.tw; ktfeng@mail.nctu.edu.tw).

P.-H. Tseng is with the Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan (e-mail: phtseng@ntut.edu.tw).

This work was in part funded by the Aiming for the Top University and Elite Research Center Development Plan, NSC 102-2221-E-009-018-MY3, the MediaTek research center at National Chiao Tung University, and the Telecommunication Laboratories at Chunghwa Telecom Co. Ltd, Taiwan. This material was also based on works by P.-H. Tseng supported in part by the NSC 102-2221-E-027-004.

Digital Object Identifier 10.1109/TW.2013.123013.121540

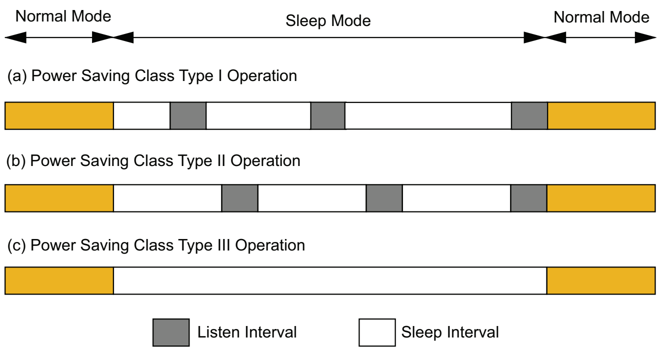


Fig. 1. Three power-saving classes defined in the IEEE 802.16e standard.

energy consumption. The extended maximizing unavailability interval (eMUI) [28] is further proposed to extend the MUI method for the mixture of power-saving classes of types I and II. By jointly considering traffic from multiple MSS, a Load-Based Power Saving (LBPS)[29] is proposed to design an adjustable sleep window size for each MSS based on measured traffic parameters. The periodic on-off scheme (PS) and the aperiodic on-off scheme (AS) are another two connection-oriented methods proposed in [30]. The objective of the PS scheme is to provide a scheduling algorithm with periodic sleep and listen intervals, which can elongate the total sleep intervals. However, since connections may have aperiodic traffic pattern, the PS scheme with periodic pattern is not ideal to accommodate those traffic in terms of power efficiency. For further enhancement, the AS scheme with aperiodic sleep and listen intervals is therefore suggested. According to the delay constraint of each connection, the connection-oriented AS scheme schedules each connection from the one with the tightest quality-of-service (QoS) requirement, i.e., the one with the smallest required delay time. With the QoS constraints, the AS scheme delays the packet as much as possible in order to acquire the opportunity of aggregating with other packets, which can improve the power efficiency.

In this paper, the packet-level frame aggregation-based power-saving scheduling (FAPS) algorithm is proposed with the considerations of multiple connections and their corresponding QoS constraints. Instead of adopting the connection-oriented methods, the proposed FAPS scheme considers power-saving scheduling at finer granularity from the packet-level perspective. The FAPS algorithm consists of two stages of processes, including the frame aggregation (FA) and the backward adjustment (BA) procedures. The proposed FA procedure is served as the default routine of maximizing the number of sleep frames by aggregating multiple under-utilized frames into fully-utilized ones; while the BA method is adopted when the FA process encounters procedure exceptions. The optimality on the minimum number of listen frames produced by the proposed FAPS algorithm can be obtained and verified via the proof of correctness under the consideration of stepwise grant space set. Performance evaluation of the FAPS scheme is subsequently conducted and compared via simulations under different situations. Simulation results show that better power efficiency with tolerable delay can be obtained by adopting the proposed FAPS algorithm compared to the baseline protocols.

The rest of the paper is organized as follows. The targeted problem and the corresponding system model are formulated in Section II. Section III explains the proposed FAPS algorithm. The optimality of the proposed FAPS protocol for the stepwise grant space set is further described and verified in Section IV; while performance evaluation of the FAPS scheme is conducted in Section V. Section VI draws the conclusions.

II. Problem Formulation

A. IEEE 802.16e Sleep Mode Operation

According to the IEEE 802.16e specification [2], the sleep mode is defined to reduce the power consumption of an MSS. As a connection is established between the BS and the MSS, the MSS can be switched into the sleep mode if there is no packet to be transmitted or received during a certain time period. The time duration within the sleep mode is divided into cycles, where each cycle can contain both the sleep and the listen intervals. In the listen interval, the MSS can either transmit/receive data or listen to the medium access control (MAC) messages acquired from the BS. During the sleep interval, on the other hand, the MSS may turn into its sleep power state or associate with other neighbor BSs for handover scanning purpose. It is noticed that the sleep mode can be initiated by either the MSS or the BS. For the MSS-initiated process, the MSS sends a MOB_SLP-REQ massage to the BS for requesting the permission of entering the sleep mode. The BS will reply with a MOB_SLP-RES massage, which also includes the parameters of the connection type, the size of the sleep and listen intervals, and the starting time for the sleep mode.

As mentioned in Section I, three power-saving types are specified for the connections between the BS and the MSS in order to facilitate different characteristics of services. As shown in Fig. 1, the sleep mode of the MSS with the power-saving class of type I consists of exponential-growing sleep intervals and fixed-length listen intervals. Within the listen intervals, the MSS will listen for the MOB_TRF-IND massage obtained from the BS in order to determine if it should return back to the normal mode. In the case that the MSS is determined not to switch back to the normal mode, the length of the next sleep interval will be doubled until the pre-defined maximum sleep window size is reached. Based on the QoS requirements as defined in [1], this type is suitable for non-realtime traffic variable-rate (NRT-VR) connections and the best-effort (BE) services between the BS and the MSSs. The power-saving class of type II defines the repetitive occurrences of the sleep and listen intervals, where the sizes of both intervals are pre-determined fixed parameters. The MSS is allowed to transmit/receive data periodically within the listen intervals. It is noticed that this power-saving type is especially suitable for QoS-guaranteed services, e.g., the unsolicited grant service (UGS) and the realtime traffic variable-rate (RT-VR) connections. Furthermore, without the assignment of listen interval, the power-saving class of type III pre-specifies a long sleep interval for the MSS before it returns back to the normal mode. This type is suggested to be utilized for multicast connections and management operations.

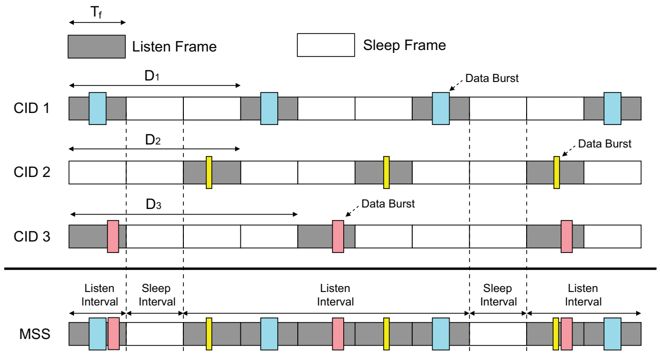


Fig. 2. Three connections with sleep mode operation between the BS and the MSS by adopting the conventional IEEE 802.16e power-saving algorithm.

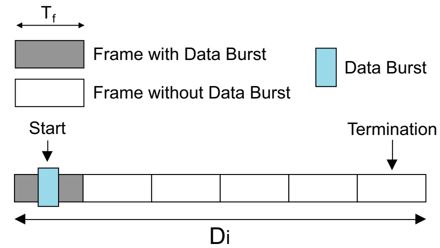


Fig. 3. The grant space $G_i(s_i, g_i, t_i)$ with the delay constraint D_i and its start and termination frames.

B. Inefficiency of IEEE 802.16e Sleep Mode Operation

Since the power-saving types are defined based on a single connection between the BS and the MSS, the inefficiency for energy conservation from the allowable multiple connections has not been considered in the IEEE 802.16e specification. For example, three realtime UGS connections are considered in Fig. 2. The parameter D_i is denoted as the delay constraint for the ith connection, which indicates that the data burst in this connection should be transmitted in the defined D_i interval. The time interval T_f is defined as the duration of a frame as shown in Fig. 2. Noted that the power-saving class of type II is considered for all the three connections with connection ID (CID) 1, 2, and 3, which are characterized as follows: (a) CIDs 1 and 2 are with packet arrival period $= 3 \cdot T_f$, sleep interval $= 2 \cdot T_f$, listen interval $= T_f$, and $D_1 = 3 \cdot T_f$; and CID 3 is with packet arrival period $= 4 \cdot T_f$, sleep interval $= 3 \cdot T_f$, listen interval $= T_f$, and $D_3 = 4 \cdot T_f$.

It can be observed that only one sleep frame per four frames will be obtained by directly combining the sleep intervals from these three connections, i.e., with the adoption of conventional IEEE 802.16e scheme as shown in Fig. 2. It can easily be extended that the sleep interval may become zero frame in certain multi-connection scenarios, which can severely degrade the efficiency for power conservation. Therefore, it is necessary to provide a feasible scheduling algorithm in order to reschedule the sleep intervals based on the combined effects from multiple connections.

C. Problem Formulation of Packet-based Power-saving Scheduling (PPS) Algorithm

Considering the aforementioned inefficiency problem of the conventional IEEE 802.16e sleep mode operation, a feasible scheduling algorithm should be proposed to enhance the efficiency of power scheduling under the scenario of multiple connections. Before diving into the design of scheduling algorithm, both the system model and the targeted problem will be described first. In order to model the combined effects of both the multiple connections and the QoS delay constraints, a packet-based modeling technique is suggested since all the connections can be individually partitioned into a series of data packets. The proposed grant space (GS) is utilized as the QoS data packet model to represent each QoS data burst in the multiple connections, which is defined as follows.

Definition 1 (grant space). *Given a frame s_i with a pre-scheduled grant for a data burst, a grant space $G_i(s_i, g_i, t_i)$ is defined as the adjacent frames ranging from s_i to $t_i = s_i + D_i - 1$, where D_i is the maximum QoS delay constraint for this data burst. The frames s_i and t_i are respectively called the start and the termination for this grant space, and the grant frame g_i is the frame that contains the data burst.*

For practical applicability in the WiMAX system, the realtime connection with unsolicited grant service (UGS) can be directly modeled by the grant spaces since the data bursts can be exactly predicted and the delay constraint is also given in the parameter of maximum latency. Moreover, for the connections with realtime Polling Service (rtPS) and extended realtime Polling Service (ertPS), the mandatory parameters of maximum latency (D_{ps}) and minimum reserved traffic rate (TR_{min}) [1][2] can also be used to remodel them into pseudo UGS connections as follows: Separate each polling service into time frames with each frame of time length equal to $D_{ps}/2$. The expected data bursts in each time frame will be $D_{ps}/2$ multiplied by TR_{min}. These bursts in a single time frame will at least have the delay constraint of $D_{ps}/2$. Two pseudo UGS connections can therefore be formed by the odd-numbered time frame group and the even-numbered time frame group. Each pseudo UGS connection has the delay constraint of $D_{ps}/2$ with predictable payload of $(D_{ps}/2) \times TR_{min}$. With the help of pseudo UGS connections, the grant space model of both rtPS and ertPS connections can therefore be obtained, which validates model applicability for the WiMAX systems.

Fig. 3 illustrates the grant space $G_i(s_i, g_i, t_i)$ with the delay constraint D_i. The start and the termination of $G_i(s_i, g_i, t_i)$ are also indicated at the two terminal frames respectively. The data burst should be scheduled within the grant frame g_i, where $s_i \leq g_i \leq t_i$, i.e., between the start and the termination frames. If the data burst is successfully scheduled within $G_i(s_i, g_i, t_i)$, it is considered that the QoS requirement for this data burst can be satisfied. With the adoption of grant spaces, the entire system can therefore be modeled by the composition of grant spaces that are acquired from multiple connections. As shown in Fig. 4, there are nine connections with CIDs from 1 to 9 and each connection consists of multiple data bursts with their QoS delay constraints. Based on Definition 1, each data burst with its delay constraint can be modeled as a grant space. For example, the first data burst u of CID 1 with delay constraint $5 \cdot T_f$ is modeled as a grant space with $D_u = 5 \cdot T_f$ and its start and termination are at frame $s_u = 3$ and frame $t_u = 7$ respectively. The second data burst d of CID

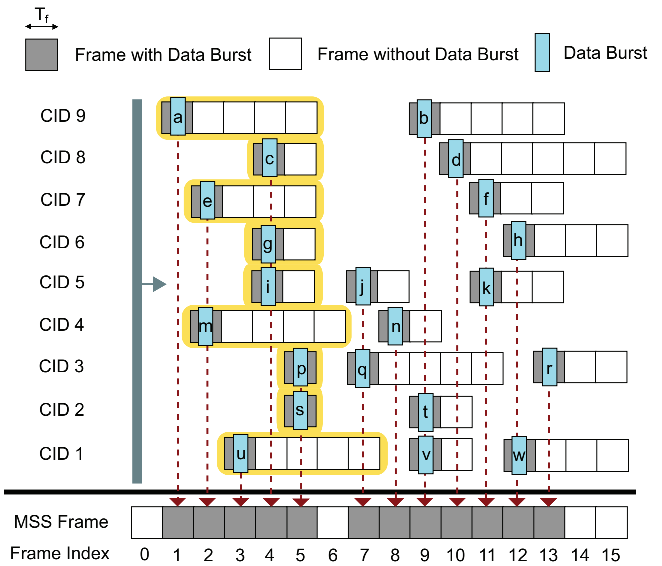


Fig. 4. The entire system modeled by multiple grant spaces that are acquired from multiple connections.

8 with delay constraint $6 \cdot T_f$ is represented as a grant space with $D_d = 6 \cdot T_f$ and its start and termination are at frame $s_d = 10$ and frame $t_d = 15$ respectively. After completing the representation of data bursts via the grant spaces, the listen and sleep frames of the MSS can therefore be determined based on the arrangement of each data burst.

For example, frame 1 of the MSS should be a listen frame since the first data packet a of CID 9 is scheduled within this frame; while frame 6 is a sleep frame since no packet is scheduled in this frame. As shown in Fig. 4, all the data packets from different connections of the MSS are categorized and result in the listen frames in grey color; while the sleep frames are identified with white color. Moreover, it is considered that the maximum allowable number of data bursts in each MSS frame will be limited, which is specified by the parameter of frame capacity F_{max}. After the description of system model, the targeted packet-based power-saving scheduling (PPS) problem can be formulated as follows.

Problem 1 (packet-based power-saving scheduling). *Given a set* $\mathbf{G} = G_i(s_i,\, g_i, t_i)$ $\forall i$ *of multiple grant spaces and the frame capacity* F_{max}*, how to arrange the data bursts within the grant spaces in order to maximize the power efficiency of the MSS, i.e., to maximize the total number of sleep frames?*

III. Proposed Frame Aggregation-based Power-saving Scheduling (FAPS) Algorithm

With the consideration of multiple connections and purely packet-level scheduling method, a frame aggregation-based power-saving scheduling (FAPS) algorithm is proposed to resolve the PPS problem as was formulated in Problem 1. The proposed FAPS algorithm can enhance the power efficiency of the MSS with proper arrangement of data bursts under the constraint of pre-specified QoS delay requirement. Two procedures are utilized in the FAPS scheme, including the frame aggregation (FA) and the backward adjustment (BA) mechanisms. The FA procedure is served as the default routine to maximize the number of sleep frames in the proposed FAPS scheme; while the BA process is utilized if the FA procedure encounters exceptions. These two procedures will be described in the following subsections. The practicability of FAPS scheme will be addressed in the last subsection.

A. Frame Aggregation (FA) Procedure

The major design concept of proposed FA procedure is to aggregate those under-utilized listen frames into fully-utilized ones, which can consequently provide more sleep frames. For example, it is assumed that the frame capacity, i.e., the maximum allowable number of data bursts in an MSS frame F_{max}, is set to be 3. As shown in Fig. 4, the two under-utilized listen frames, i.e., frame 7 and frame 8, can be aggregated into frame 8 together since the three corresponding data bursts, i.e., bursts j and q in frame 7 and burst n in frame 8, can be scheduled at frame 8 without breaking the QoS delay constraints specified in their grant spaces. Additional sleep frame at frame 7 can therefore be acquired, which improves the power efficiency of the system. The proposed FA procedure is described as follows.

1) Forward Collection Mechanism: In order to acquire more sleep frames, a data burst λ should be delayed as much as possible to obtain more chance to aggregate with other data bursts. However, according to the QoS delay constraint, the data burst λ must be scheduled before the termination t_λ specified within its grant space $G_\lambda(s_\lambda, g_\lambda, t_\lambda)$. Therefore, the first step of the proposed FA procedure is to delay all data bursts from left to right until some of the data bursts reach their delay constraints. For example, as shown in Fig. 4, all the data bursts within the set $\mathbf{D} = \{a,\, c,\, e,\, g,\, i,\, m,\, p,\, s,\, u\}$ defined by their grant spaces with yellow background color can be delayed to the maximum at frame 5. Note that data bursts m and u can further be delayed to frames 6 and 7 respectively based on the requirements from their grant spaces. This procedure can be analogues to the action of a windshield wiper as represented by the solid bar at frame 0 in Fig. 4. The windshield wiper will be stuck at the so-called stuck frame $x_{\mathbf{G}} = 5$ since both the data bursts p and s in this group can not be further proceeded based on their QoS constraints. Therefore, the unscheduled grant spaces in the set $\mathbf{D}$ will be stuck at frame $x_{\mathbf{G}}$, which can be formally defined as the stuck group as follows.

Definition 2 (stuck group). *Given a set* $\mathbf{G} = G_i(s_i,\, g_i, t_i)$ $\forall i$ *of unscheduled grant spaces, the stuck group of* $\mathbf{G}$ *is defined as the set*

$$\mathbf{S_G} = \{G_\zeta \in \mathbf{G} \,|\, s_\zeta \leq x_{\mathbf{G}}\}, \tag{1}$$

where s_ζ *is the start of grant space* G_ζ *and* $x_{\mathbf{G}} = \min(t_i)$ $\forall i$ *is called the stuck frame of* $\mathbf{G}$ *with* t_i *denoting the termination of grant space* G_i*.*

As was mentioned in the previous paragraph, even in the stuck group $\mathbf{S_G}$, there may still exist data bursts that can be delayed beyond the stuck frame $x_{\mathbf{G}}$ based on their grant spaces, e.g., the data bursts m and u as shown in Fig. 4 can be delayed to frames 6 and 7 respectively. In order to identify this type

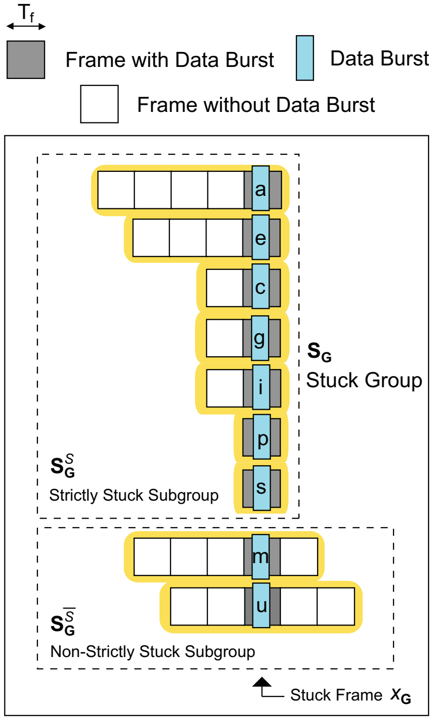


Fig. 5. The strictly stuck subgroup $\mathbf{S}_\mathbf{G}^S$ and non-strictly stuck subgroup $\mathbf{S}_\mathbf{G}^{\overline{S}}$.

of data bursts and their corresponding grant spaces, the stuck group $\mathbf{S_G}$ can further be divided into two sorted subgroups as follows.

Definition 3 (strictly and non-strictly stuck subgroups). *Given the grant spaces $G_\zeta(s_\zeta,\ g_\zeta, t_\zeta)$ in a stuck group $\mathbf{S_G}$ that are sorted in ascending order according to the termination of each grant space. If two grant spaces possess the same termination, the start of the grant spaces is utilized as the parameter to perform sorting. The strictly and non-strictly stuck subgroups of $\mathbf{S_G}$ are respectively defined as the set*

$$\mathbf{S}_\mathbf{G}^S = \{G_\zeta \in \mathbf{S_G} \,|\, t_\zeta = x_\mathbf{G}\}, \tag{2}$$

and the set

$$\mathbf{S}_\mathbf{G}^{\overline{S}} = \mathbf{S_G} - \mathbf{S}_\mathbf{G}^S, \tag{3}$$

where $x_\mathbf{G}$ is the stuck frame defined for this stuck group $\mathbf{S_G}$.

2) Backward Push Mechanism: As shown in Fig. 5, the stuck group $\mathbf{S_G}$ obtained from the data burst set $\mathbf{D} = \{a,\ c,\ e,\ g,\ i,\ m,\ p,\ s,\ u\}$ in Fig. 4 is sorted and separated into the strictly stuck subgroup $\mathbf{S}_\mathbf{G}^S$ and the non-strictly stuck subgroup $\mathbf{S}_\mathbf{G}^{\overline{S}}$. It can be observed that all data bursts in $\mathbf{D}$ will be stuck at the stuck frame $x_\mathbf{G}$ by adopting the forward collection mechanism. However, it is impossible to schedule all the data bursts into frame $x_\mathbf{G}$ if the maximum allowable number of data bursts F_{max} of an MSS frame is less than the size of $\mathbf{S_G}$. Therefore, the constraint originated from the parameter F_{max} will be considered in the second step of the proposed FA procedure. Moreover, the scheduling of data bursts specified in the grant spaces of $\mathbf{S}_\mathbf{G}^S$ should be performed prior to that of $\mathbf{S}_\mathbf{G}^{\overline{S}}$ since all the data bursts in $\mathbf{S}_\mathbf{G}^S$ must be scheduled before the stuck frame $x_\mathbf{G}$.

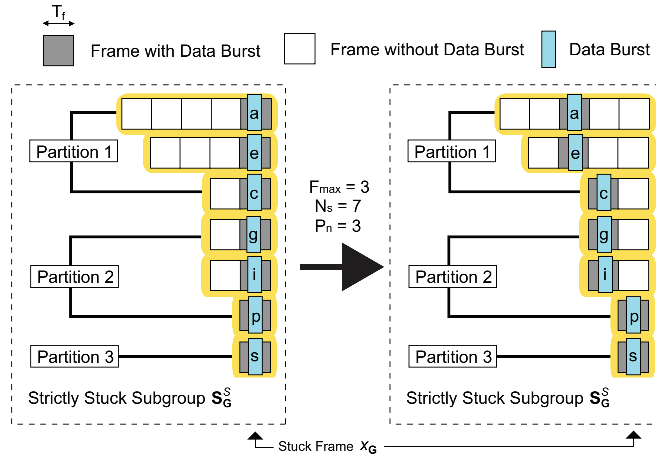


Fig. 6. The arrangement of data bursts in the strictly stuck subgroup $\mathbf{S}_\mathbf{G}^S$.

The arrangement of data bursts in the strictly stuck subgroup $\mathbf{S}_\mathbf{G}^S$ is explained as follows. Let N_s be the number of grant spaces in $\mathbf{S}_\mathbf{G}^S$ and $P_n = \lceil N_s/F_{max} \rceil$ be the number of partitions utilized to separate $\mathbf{S}_\mathbf{G}^S$ counting from the sorted first grant space. Note that each partition in $\mathbf{S}_\mathbf{G}^S$ will have at most F_{max} data bursts in order to meet the requirement of frame capacity. Finally, referenced from the stuck frame $x_\mathbf{G}$, the data bursts in each partition i will be moved back $P_n - i$ frames gradually if it is allowed based on their corresponding grant space constraints. If certain data burst can not be moved all the way to the designated frame, it will stay at the frame unable to proceed back. If no exception occurs, the arrangement of data bursts specified in $\mathbf{S}_\mathbf{G}^S$ is completed.

For example, as shown in Fig. 6, the frame capacity is $F_{max} = 3$, the number of grant spaces in $\mathbf{S}_\mathbf{G}^S$ is $N_s = 7$, and the number of partitions is $P_n = \lceil N_s/F_{max} \rceil = 3$. The data bursts in partition 1 should be moved back $P_n - 1 = 2$ frames. However, the data burst c will only be moved back one frame since it is limited by the start frame s_c of its grant space $G_c(s_c, g_c, t_c)$. In partition 2, same situation can be observed at the data burst p which is constrained by its grant space $G_p(s_p, g_p, t_p)$. Finally, the data bursts a and e will be scheduled in the same frame. The bursts c, g, and i can also be aggregated; while the stuck frame can accommodate the remaining data bursts p and s. It can be observed that all data bursts in $\mathbf{S}_\mathbf{G}^S$ is properly scheduled without breaking their QoS delay constraints and the maximum allowable number of data bursts of an MSS frame.

3) Packet-Padding for Backward Push Mechanism: On the other hand, the scheduling of non-strictly stuck subgroup $\mathbf{S}_\mathbf{G}^{\overline{S}}$ will be considered according to the packet-padding for backward push mechanism. In order to improve the power efficiency, the main concept is to pad the data bursts of $\mathbf{S}_\mathbf{G}^{\overline{S}}$ into those under-utilized listen frame after the arrangement of $\mathbf{S}_\mathbf{G}^S$. Therefore, the data bursts specified in $\mathbf{S}_\mathbf{G}^{\overline{S}}$ can be scheduled within some of the listen frames that are under-utilized in $\mathbf{S}_\mathbf{G}^S$, which is explained as follows. Acquiring a under-utilized listen frame δ starting from the left-most frame, the design procedure is to find a proper number of grant spaces in $\mathbf{S}_\mathbf{G}^{\overline{S}}$ whose data bursts can be scheduled in the frame δ. Note that the acquisition order of the grant spaces is the same as the order defined in Definition 3.

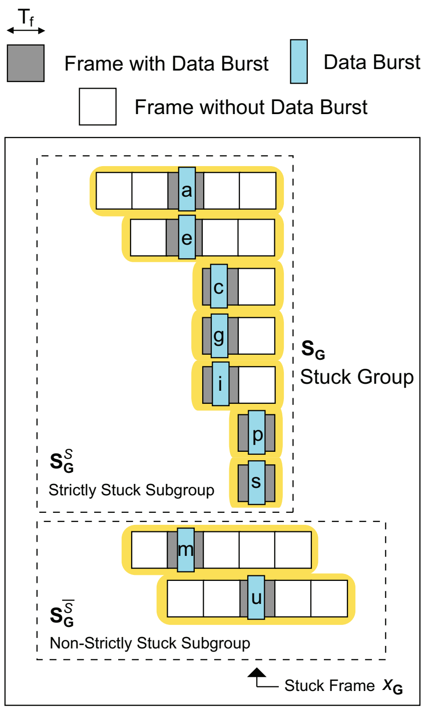


Fig. 7. The complete arrangement of the strictly stuck subgroup $\mathbf{S}_{\mathbf{G}}^{S}$ and the non-strictly stuck subgroup $\mathbf{S}_{\mathbf{G}}^{\overline{S}}$.

As shown in Fig. 7, the first under-utilized listen frame is the frame with data bursts a and e, and the second one is the frame with data bursts p and s. Based on the aforementioned technique, the data burst m will be arranged together with the data bursts a and e; while the burst u will be accommodated with bursts p and s, leading to the fully-utilized listen frames. Finally, if there still remain data bursts specified in the grant spaces of $\mathbf{S}_{\mathbf{G}}^{\overline{S}}$, these data bursts should be scheduled with the other unscheduled grant spaces as the new input grant spaces for the next round of the FA procedure. It can be observed that the set of data burst $\mathbf{D} = \{a, c, e, g, i, m, p, s, u\}$ in Fig. 4 can be successfully scheduled within 3 listen frames, which is smaller than the original 5 listen frames. Furthermore, Fig. 8 shows the scheduling result of the complete power-saving system exemplified in Fig. 4 after executing five rounds of the FA procedure. It can be seen that the number of required listen frames is decreased from the original 12 to 7 listen frames by adopting the FA algorithm, which effectively enhances the power efficiency of the system.

4) Complexity Discussion for FA Procedure: Let n_{gs} be the number of all unscheduled GSs. In the forward collection mechanism, the stuck group can be obtained by (a) finding the smallest termination, (b) grouping the suitable GSs, and then (c) sorting these GSs. The time complexity required for steps (a), (b), and (c) are respectively $O(n_{gs})$, $O(n_{gs})$, and $O(n_{gs} \cdot \log(n_{gs}))$. Let n_{gs}^{S} be the GS number of obtained strictly stuck subgroup and D_{max} the maximum delay for all connections. In the backward push mechanism, the time complexity will be $O(n_{gs}^{S} \cdot D_{max})$ since all GSs in $\mathbf{S}_{\mathbf{G}}^{S}$ should do schedulability test backwardly in its delay constraint. The same reason can be applied to the packet padding, i.e., the last step of FA procedure. The time complexity will be $O(n_{gs}^{\overline{S}} \cdot D_{max})$, where $n_{gs}^{\overline{S}}$ is the GS number of obtained non-strictly stuck subgroup.

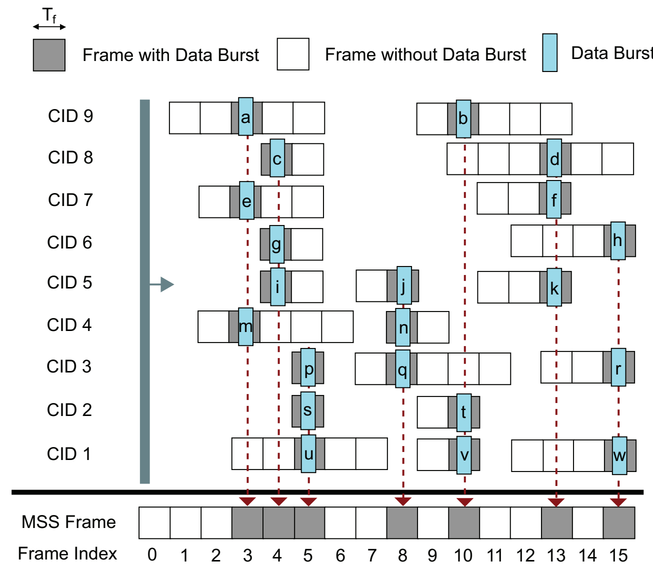


Fig. 8. The scheduling result of the proposed FA procedure for the example system in Fig. 4 with multiple connections.

B. Backward Adjustment (BA) Procedure

As shown in Fig. 9, there are eight data bursts from a to h with their own grant spaces defined from their individual connections. The maximum allowable number of data bursts F_{max} of an MSS frame is assumed to be 2. In the first round of the FA procedure, data bursts a and b can be intuitively and completely scheduled. Note that each round of the FA procedure is conducted according to the stuck group as defined in Definition 2. Data bursts c and d are also successfully aggregated in the second round without exceptions. In the third round of execution, there are four unscheduled grant spaces for data bursts e, f, g, and h, which are separated into the strictly and non-strictly stuck subgroups $\mathbf{S}_{\mathbf{G}}^{S}$ and $\mathbf{S}_{\mathbf{G}}^{\overline{S}}$. Based on the FA procedure, the grant spaces of $\mathbf{S}_{\mathbf{G}}^{S}$ will be divided into two partitions, including partition 1 consisting of data bursts e and f and partition 2 containing g as shown in Fig. 9.

In partition 1, each data burst should be moved backward $P_n - 1 = \lceil N_s/F_{max} \rceil - 1 = 1$ frame. However, it is unable to move these bursts backward since frame 5 is already full of data bursts, i.e., data bursts c and d. Data bursts e and f will still be placed at frame 6. On the other hand, data burst g specified in partition 2 will also be scheduled at frame 6. Therefore, the exception of proposed FA procedure occurs since frame 6 with frame capacity $F_{max} = 2$ is utilized to allocate data bursts e and f and frame 6 is the only choice for partition 2. It will violate the frame capacity constraint if no proper scheduling adjustment is performed. The proposed BA procedure is therefore utilized to solve the scheduling exception problems that are encountered by the aforementioned FA process. The BA procedure is described in the following two steps.

1) Recursive Backward Movement: The recursive backward movement of the proposed BA procedure can be found in Algorithm 1 and is described as follows. Given a data burst λ and the targeted frame θ, a function $makespace(\lambda, \theta)$ is utilized to allocate a space for λ at frame θ by recursively moving the data burst λ backward. If λ can be scheduled in

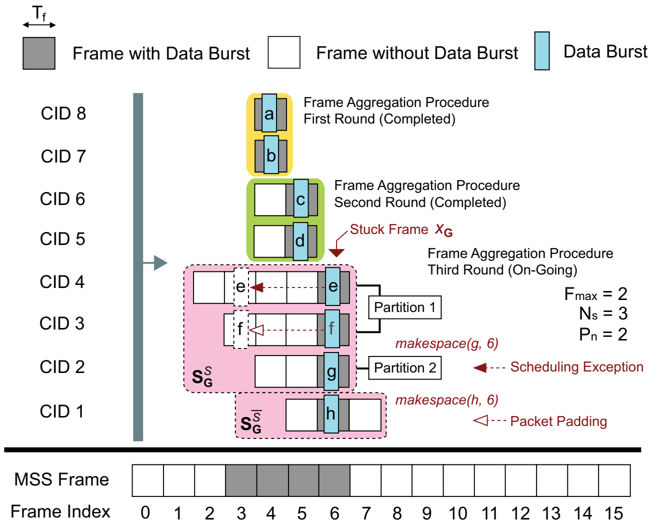


Fig. 9. The proposed backward adjustment procedure for the scheduling exceptions of the FA procedure.

θ, the function $makespace$ will return a $true$ value which consequently solves the exception problem resulting from the FA procedure. If λ can not be scheduled into frame θ, another data burst ξ in θ whose grant space $G_\xi(s_\xi, g_\xi, t_\xi)$ has the smallest start frame will be selected to perform the recursive backward movement. Subsequently, based on the same function $makespace$, the process will be continued by backward searching the frame π ranging from frame $(\theta - 1)$ to the start frame s_ξ of data burst ξ. If a frame π is found to be non-saturated, data burst ξ will be scheduled into this corresponding frame π. Finally, the unscheduled data burst λ can be allocated into frame θ, which completes the recursive backward movement of the proposed BA procedure for scheduling the data burst λ.

As shown in Fig. 9, data burst g in partition 2 will encounter the exception problem while adopting the third round of the FA procedure. Therefore, based on the recursive backward movement of the proposed BA procedure, the function $makespace(\lambda, \theta)$ will be invoked, where $\lambda = g$ and $\theta = 6$. For those data bursts e and f scheduled in frame 6, the grant space $G_e(s_e, g_e, t_e)$ of data burst e has the smallest start frame, i.e., $s_e = 2$. Consequently, data burst e will be selected as the burst that should be moved backward. Subsequently, the function $makespace(\lambda, \pi)$ will be re-conducted for the new inputs $\lambda = e$ and $\pi = \theta - 1, \ldots, s_e$, i.e., $\pi = 5$, 4, 3, and 2 for each time. As the first function $makespace(e, 5)$ for scheduling data burst e is invoked, the next selected data burst which may be moved back is burst c in frame 5 since it has the smallest start frame, i.e., frame 4. Recursively, when $makespace(c, 4)$ for arranging data burst c is initiated, burst a is also chosen as the candidate to move backward. However, burst a can not be moved back since it is at the start frame 4 of the grant space $G_a(s_a, g_a, t_a)$. Since burst a can not be moved, data burst c can only be scheduled at its current frame 5. Moreover, since it is not possible to move burst c backward, the data burst e should not be scheduled at the frame 5 which indicates that $makespace(e, 5)$ can not allocate a space for e at the frame 5.

The second function $makespace(e, 4)$ for scheduling data burst e will continually be invoked to verify if burst e can be scheduled at frame 4. For the same reason as explained in the previous paragraph, burst e also can not be arranged at frame 4. Finally, data burst e will be scheduled at frame 3 since the third function $makespace(e, 3)$ for scheduling burst e will return a $true$ value, representing the available vacancy of frame 3. As data burst e is placed into frame 3, there exists a vacancy in frame 6 for data burst g to be scheduled which completes the recursive backward movement.

Algorithm 1: *Recursive Backward Movement: makespace(λ, θ)*

Input: λ: data burst, θ: targeted frame
Output: $true$ or $false$: can λ be scheduled in θ ?

```
begin
    if λ can be placed into frame θ then
        return true;
    else
        let ξ be the data burst in frame θ whose grant space
        G_ξ(s_ξ, g_ξ, t_ξ) has the smallest start frame;
        for π = θ − 1 to s_ξ do
            if makespace(ξ, π) then
                put data burst ξ into frame π;
                return true;
            end
        end
        return false;
    end
end
```

2) Packet-Padding for Recursive Backward Movement: After the execution of recursive backward movement, there may still exist under-utilized frames that are capable to allocate additional data bursts. For example, in the scheduling of data burst g as shown in Fig. 9, data burst e will be pushed backward to frame 3 based on the $makespace$ algorithm, which makes the frame 3 as an under-utilized frame. In order to enhance the power efficiency, some data bursts should be aggregated into the frame 3. It can also be observed that the conduct of $makespace(h, 6)$ can achieve this goal. The data burst f can be moved to the frame 3 due to the $makespace$ algorithm. Based on this observation, the packet padding mechanism for the recursive backward movement can be designed to adopt the $makespace$ with function inputs as the data burst in the non-strictly stuck subgroup $\mathbf{S}_\mathbf{G}^{\bar{S}}$ and the stuck frame $x_\mathbf{G}$. This packet padding process is continuously conducted until either a $false$ return value is obtained or before an additional listen frame is introduced. In other words, this packet padding mechanism will not lead to the creation of new listen frames. As shown in Fig. 9, based on the packet padding mechanism, data burst h in $\mathbf{S}_\mathbf{G}^{\bar{S}}$ can be scheduled at the stuck frame $x_\mathbf{G} = 6$ by conducting the function $makespace(h, 6)$ which effectively makes frame 6 to be a fully-utilized frame. The original under-utilized frame 3 also reaches its maximum capacity. With the recursive backward movement and the packet padding scheme of the proposed BA procedure, the third round of proposed FA procedure can therefore be completed with enhanced power efficiency.

3) Complexity Discussion for BA Procedure: The recursive backward movement is based on the function $makespace(\lambda, \theta)$ in Algorithm 1. It can be observed that

θ will lie in the range of grant space $G_\xi(s_\xi, g_\xi, t_\xi)$, i.e., $s_\xi \leq \theta \leq t_\xi$. The iteration number of the **for** loop with parameter π will be accordingly less than D_{max} times. In the worst case, the $makespace(\lambda, \theta)$ function will be executed $O((D_{max})^{n_{gs}^S})$ times since there are at most n_{gs}^S grant spaces in the strictly stuck subgroup which need to run the BA procedure. Finally, in the packet padding for recursive backward movement, there are at most $n_{gs}^{\overline{S}}$ grant spaces in the non-strictly stuck subgroup which have the chance to be padded into under-utilized frames. Similar time complexity $O((D_{max})^{n_{gs}^{\overline{S}}})$ can also be acquired.

C. Practicability of FAPS Scheme

The benefits and usage scenarios of proposed FAPS scheme is summarized as follows. In this paper, a complicated scheduling problem has been decomposed into several parts in accordance with the procedures in IEEE 802.16 networks. The original scheduling mechanism includes three different stages as follows:

(S1) Admission control to map each application to its connection ID;

(S2) Request bandwidth and design sleep pattern from packet scheduling for each connection; and

(S3) System scheduling at BS side.

With proper admission control to decide the grant space for each connection, the proposed FAPS scheme mainly focuses on the second item *(S2)* for packet scheduling. With this problem decomposition, the proposed FAPS scheme is proven to be optimal in the sub-problem *(S2)*. The MSS can apply the FAPS algorithm for concurrent CIDs to achieve power conservation. A practical situation for higher number of concurrent CIDs will be the inter-networking scenario for WiFi and WiMAX networks. The WiFi-AP conserves energy by aggregating its traffic flows from WiFi users to request an optimal sleep pattern to WiMAX network.

Considering the DL scenario, the proposed FAPS scheme can also be implemented at BS side providing that the entire scheduling method is decomposed as the same procedure above. The FAPS scheme can aggregate multiple connections from multiple MSSs for achieving an optimal power-saving pattern to conserve the BS's power, which fulfills the requirements for green communications. Furthermore, without additional modifications, the FAPS algorithm can also be utilized for *(S1)* to aggregate multiple applications into one connection ID. It is intuitive that we can implement this stage as follows: (a) virtually map one service flow from the application to one CID for parameter calculation only, i.e., to form a grant space for each service flow without assigning real CID; and (b) aggregate different grant spaces into one CID using the FAPS scheme. This is considered a useful scenario since there are more and more applications with social networking (e.g., Facebook) or real-time information (e.g., Weather) which would periodically update its latest information. In summary, a feasible power-saving scheme (e.g., the proposed FAPS algorithm) is required to deal with the three cases as stated above, where the total number of multiple connections even up to 20 might still be reasonable. Consequently, the FAPS scheme can provide the optimal sleep pattern to these usage cases, which can be practically implemented.

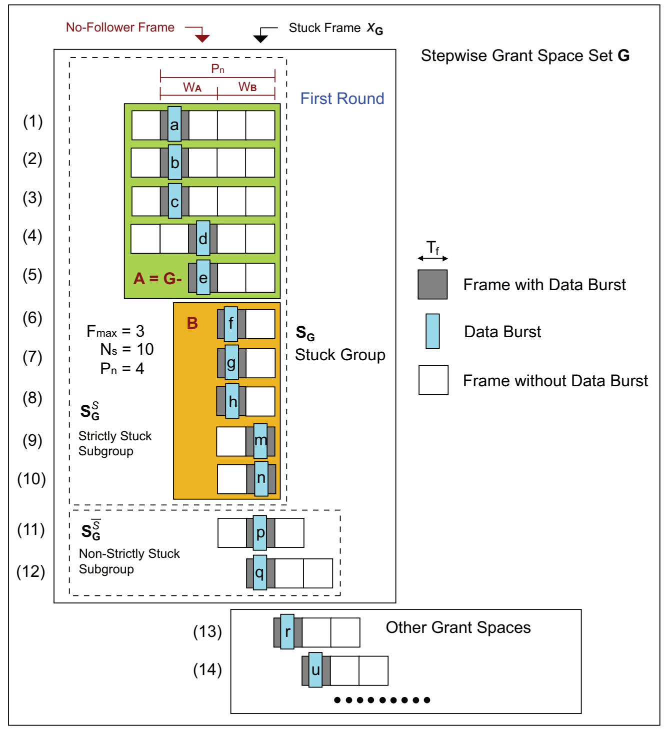


Fig. 10. The exemplified packet arrangement in the first round of the proposed backward push mechanism.

IV. Optimality of Proposed FAPS Algorithm

In this section, the optimality for the proposed FAPS algorithm to result in the minimum number of listen frames will be proven under the input scenario with stepwise grant space set. The stepwise grant space set is described as follows.

Definition 4 (stepwise grant space set). *A grant space set* $\mathbf{G} = G_i(s_i, g_i, t_i)$ *is called stepwise if the following two properties can be satisfied:*

1) *The order for each grant space in* $\mathbf{G}$ *is identified in the ascending manner according to the value of termination, and is further sorted by the start if same termination is encountered.*
2) *For each pair of consecutive grant spaces* $G_i(s_i, g_i, t_i)$ *and* $G_j(s_j, g_j, t_j)$ *in* $\mathbf{G}$*, the start and the termination of the former are respectively less than or equal to those of the latter, i.e.,* $s_i \leq s_j$ *and* $t_i \leq t_j$.

As shown in Fig. 10, all grant spaces in the stepwise grant space set $\mathbf{G}$ are listed based on the criterions in Definition 4, and the sequential numbers are also specified at the left. If there exists the stepwise grant space set $\mathbf{G}$, minimum number of listen frames can be obtained by adopting the proposed FAPS algorithm that provides the optimal packet arrangement under the frame capacity constraint F_{max}. The flow of correctness proof for the FAPS algorithm is depicted in Fig. 11, and is explained as follows.

In the first round, based on the given stepwise grant space set $\mathbf{G}$, the proposed FAPS algorithm will perform the forward collection mechanism to construct the stuck group and further

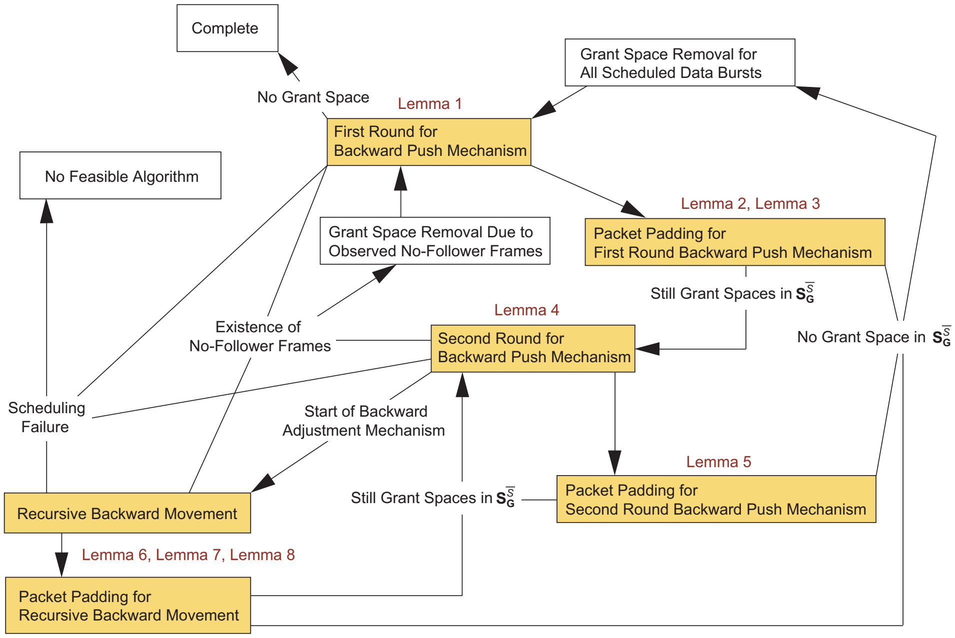


Fig. 11. The flow of the correctness proof for the proposed FAPS algorithm.

determine both the strictly and non-strictly stuck subgroups. Subsequently, the backward push mechanism will be executed to properly rearrange the data bursts according to the frame capacity $F_{max} = 3$ data bursts. As illustrated in Fig. 10, the stepwise grant space set $\mathbf{G}$ consists of the grant spaces of data bursts $\{a, b, \ldots r, u, \ldots\}$. Based on the backward push mechanism, the first partition $\{a, b, c\}$ should be scheduled at the frame $x_{\mathbf{G}} - 3$; while the second partition $\{d, e, f\}$ will be placed at frame $x_{\mathbf{G}} - 2$. However, due to the grant space constraint on the start frame s_f, data burst f will be scheduled at frame $x_{\mathbf{G}} - 1$. The remaining data bursts in the strictly stuck subgroup $\mathbf{S}_{\mathbf{G}}^{S}$ are arranged as shown in Fig. 10. In order to facilitate the correctness proof, it is required to define the no-follower frame as follows.

Definition 5 (no-follower frame). *With the adoption of the proposed FAPS algorithm on a stepwise grant space set $\mathbf{G}$, some grant spaces can be scheduled together in a frame θ_n and the last grant space to be scheduled in θ_n is specified as $G_\xi(s_\xi, g_\xi, t_\xi)$. The frame θ_n is defined as a no-follower frame if there does not exist any other data burst after the grant space $G_\xi(s_\xi, g_\xi, t_\xi)$ that has the chance to be scheduled in this frame.*

As shown in Fig. 10, due to the properties of stepwise grant space set in Definition 4, the frame $(x_{\mathbf{G}} - 2)$ is a no-follower frame since there is no other data burst after the grant space $G_e(s_e, g_e, t_e)$ which can be scheduled in this frame.

Fact 1. *Given the frame capacity F_{max} and a set $\mathbf{G}$ with $M_{\mathbf{G}}$ grant spaces, all the start and the termination frames of the grant spaces in $\mathbf{G}$ are bounded in the frame range $[\theta_s, \theta_t]$ with width $W = \theta_t - \theta_s + 1$. If the number $M_{\mathbf{G}}$ can fulfill the equality of $\lceil M_{\mathbf{G}}/F_{max} \rceil = W$, all the frames ranging from θ_s to θ_t must be necessary listen frames. On the other hand, if the number $M_{\mathbf{G}}$ results in the inequality as $\lceil M_{\mathbf{G}}/F_{max} \rceil > W$, there does not exist any scheduling algorithm that can arrange these data bursts based on the QoS constraints in the grant space set $\mathbf{G}$.*

Fact 2. *Assuming that at least U_L necessary listen frames exist before the end of an arbitrary frame θ_n in $[\theta_s, \theta_t]$, the entire grant space set $\mathbf{G}$ is partitioned into two disjoint sets, including the removable set $\mathbf{G}_-$ and the remaining set $\mathbf{G}_r$. The set $\mathbf{G}_-$ can be scheduled within the minimum number of listen frames and can be removed without interfering the scheduling of $\mathbf{G}_r$ if the following two conditions are satisfied: (a) the set $\mathbf{G}_-$ can be feasibly scheduled within U_L frames before the end of frame θ_n; and (b) it is not possible to place the data bursts in $\mathbf{G}_r$ into those U_L necessary listen frames.*

Lemma 1. *In the first round of the backward push mechanism, given the first observed no-follower frame θ_n, the set $\mathbf{G}_-$ consisting of all grant spaces in $\mathbf{G}$ whose data bursts are scheduled before the end of frame θ_n can be removed without interfering the scheduling of the remaining grant spaces $\mathbf{G}_r$. The set $\mathbf{G}_-$ is considered completely scheduled within the minimum number of listen frames.*

Proof: The strictly stuck subgroup is denoted as $\mathbf{S}_{\mathbf{G}}^{S}$ with N_s grant spaces after the first round of the backward push mechanism. Based on the backward push mechanism, the number of partitions is obtained as

$$P_n = \lceil N_s/F_{max} \rceil. \quad (4)$$

For example, as shown in Fig. 10, the values can be obtained as $N_s = 10$, $F_{max} = 3$, and $P_n = 4$. The listen frames for accommodating all the data bursts specified in the grant spaces of $\mathbf{S}_{\mathbf{G}}^{S}$ will be within the range $[x_{\mathbf{G}} - P_n + 1, x_{\mathbf{G}}]$ since the maximum backward movement in the proposed backward push mechanism is $P_n - 1$ frames. Moreover, since the first no-follower frame θ_n can be observed during the first round of the backward push mechanism, the frame θ_n must be within the same listen frame range, i.e., $\theta_n \in [x_{\mathbf{G}} - P_n + 1, x_{\mathbf{G}}]$.

Based on the last grant space $G_\xi(s_\xi, g_\xi, t_\xi)$ scheduled in the no-follower frame θ_n, $\mathbf{S}_\mathbf{G}^S$ can further be divided into the following two parts: (a) the set $\mathbf{A}$ consisting of all the grant spaces before and including $G_\xi(s_\xi, g_\xi, t_\xi)$; and (b) the other set $\mathbf{B}$ containing the remaining grant spaces after $G_\xi(s_\xi, g_\xi, t_\xi)$. As shown in Fig. 10, for example, the last grant space in the no-follower frame θ_n is $G_e(s_e, g_e, t_e)$. The data bursts of the set $\mathbf{A}$ will be $\{a, b, c, d, e\}$; while those of the set $\mathbf{B}$ will be $\{f, g, h, m, n\}$. Given the size M_A of the set $\mathbf{A}$ and based on the backward push mechanism, set $\mathbf{A}$ must be accommodated in the range $[x_\mathbf{G} - P_n + 1, \theta_n]$ with width $W_\mathbf{A}$, which can be determined as

$$W_\mathbf{A} = \lceil M_\mathbf{A}/F_{max} \rceil. \tag{5}$$

The grant spaces in the set $\mathbf{B}$ are constrained in the range $[\theta_n + 1, x_\mathbf{G}]$ due to the stuck frame and the no-follower frame definitions in Definitions 2 and 5, respectively. The width of the range $[\theta_n + 1, x_\mathbf{G}]$ can be denoted and determined as

$$W_\mathbf{B} = P_n - W_\mathbf{A}. \tag{6}$$

Furthermore, the size $M_\mathbf{B}$ of the set $\mathbf{B}$ is obtained as

$$M_\mathbf{B} = N_s - M_\mathbf{A}. \tag{7}$$

Therefore, based on (4) to (7), the relationship between $M_\mathbf{B}$ and $W_\mathbf{B}$ can be derived as

$$\begin{aligned} &\lceil M_\mathbf{B}/F_{max} \rceil = \lceil (N_s - M_\mathbf{A})/F_{max} \rceil \\ &\quad \geq \lceil N_s/F_{max} \rceil - \lceil M_\mathbf{A}/F_{max} \rceil = P_n - W_\mathbf{A} = W_\mathbf{B}. \end{aligned} \tag{8}$$

Based on (8) and Fact 1, since set $\mathbf{B}$ is bounded in the range $[\theta_n + 1, x_\mathbf{G}]$ with width $W_\mathbf{B}$, the frames in the range $[\theta_n + 1, x_\mathbf{G}]$ must be necessary listen frames if a feasible scheduling method exists, i.e., $W_\mathbf{B} = \lceil M_\mathbf{B}/F_{max} \rceil$. Furthermore, there are at least $P_n = \lceil N_s/F_{max} \rceil$ listen frames required by all N_s data bursts in $\mathbf{S}_\mathbf{G}^S$ before the end of the stuck frame $x_\mathbf{G}$. Therefore, there must be at least $U_L = P_n - W_\mathbf{B} = W_\mathbf{A}$ listen frames before the end of the no-follower frame θ_n since those $W_\mathbf{B}$ frames in the range $[\theta_n + 1, x_\mathbf{G}]$ have been proven to be necessary listen frames.

The set $\mathbf{A}$ can fulfill with the requirement of set $\mathbf{G}_-$ since all data bursts of $\mathbf{A}$ are scheduled before the end of θ_n and no more grant space can be arranged within that range due to Definition 5. Furthermore, all data bursts in $\mathbf{A} = \mathbf{G}_-$ can be scheduled before the end of θ_n with $W_A = U_L$ frames. Therefore, based on Fact 2, the set $\mathbf{G}_-$ can be removed without interfering the scheduling of the remaining grant spaces. The scheduling of set $\mathbf{G}_-$ is completed with the minimum number of listen frames. It completes the proof. □

Based on each observed no-follower frame, the corresponding grant space set $\mathbf{G}_-$ can be repeatedly removed from the stepwise grant space set $\mathbf{G}$ according to Lemma 1. All data bursts in the removed grant spaces are properly scheduled within the minimum number of listen frames.

Lemma 2. *After completing the potential grant space removal in $\mathbf{S}_\mathbf{G}^S$ based on Lemma 1, the remaining frames with data bursts are all fully-utilized except for the stuck frame which can be an under-utilized frame.*

Proof: Based on the backward push mechanism, all grant spaces will be partitioned by the frame capacity F_{max}. All partitions are full of F_{max} grant spaces except for the last partition that can have the number of grant spaces less than F_{max}. Furthermore, all data bursts in the same partition will have the same pre-determined target frame, i.e., the ith partition will be moved back $P_n - i$ frames from the stuck frame $x_\mathbf{G}$. If all data bursts in the same partition can reach the target frame, this target frame must be fully-utilized. On the other hand, if the data burst ξ can not be scheduled in its target frame θ_ξ^t, two possible situations can occur as follows. The first situation is that there does not exist additional vacancy in frame θ_ξ^t which denotes full utilization of frame θ_ξ^t. The other situation happens when the start frame index s_ξ of data burst ξ in the grant space $G_\xi(s_\xi, g_\xi, t_\xi)$ is larger than frame θ_ξ^t, i.e., $s_\xi > \theta_\xi^t$. For example, as shown in Fig. 10, the data bursts in the first partition $\{a, b, c\}$ will be scheduled in frame $(x_\mathbf{G} - 3)$ and occupy the entire frame. On the other hand, the second partition $\{d, e, f\}$ should all be scheduled in the frame $x_\mathbf{G} - 2$ as planned. However, due to the start frame constraint of the grant space, data burst f can not be scheduled to reach frame $x_\mathbf{G} - 2$.

As the reason stated for the second situation, the target frame θ_ξ^t can become an under-utilized frame. Nevertheless, this frame must be a no-follower frame according to Definition 5 since there is no other data burst after grant space $G_\xi(s_\xi, g_\xi, t_\xi)$ that has the chance to be scheduled in frame θ_ξ^t. For example, as illustrated in Fig. 10, target frame θ_ξ^t of data burst $\xi = f$ corresponds to frame $x_\mathbf{G} - 2$, and there is no data burst after burst f that can be placed at this frame $x_\mathbf{G} - 2$. Based on Lemma 1, all the grant spaces with the grant frame $g_i \leq \theta_\xi^t$ can be removed. Therefore, after completing all possible grant space removals, the remaining frames with data bursts are fully-utilized except for the potential under-utilized stuck frame. It completes the proof. □

After the execution of the backward push mechanism, the packet padding in the proposed FA mechanism will be conducted. The data bursts specified in the non-strictly stuck subgroup $\mathbf{S}_\mathbf{G}^{\overline{S}}$ will be placed in the vacant locations within the under-utilized frame. As shown in Fig. 10, data burst p will be scheduled at the same frame along with the data bursts m and n.

Lemma 3. *If all data bursts specified in the non-strictly stuck subgroup $\mathbf{S}_\mathbf{G}^{\overline{S}}$ are scheduled within the under-utilized frames in the first round of backward push mechanism, the stuck group $\mathbf{S}_\mathbf{G}$ can be removed without interfering the scheduling of the remaining grant spaces. All the removed grant spaces are considered completely scheduled within the minimum number of listen frames.*

Proof: Based on Lemma 1, all the grant spaces in $\mathbf{G}$ whose data bursts are scheduled before the end of the no-follower frame can be removed without interfering the scheduling of the remaining grant spaces. This leads to the size reduction for the strictly stuck subgroup $\mathbf{S}_\mathbf{G}^S$. Based on Lemma 2, the remaining frames with data bursts will all be fully-utilized except for the stuck frame $x_\mathbf{G}$. In other words, the stuck frame $x_\mathbf{G}$ is the only frame which can accommodate the data bursts specified in the non-strictly stuck subgroup

$\mathbf{S}_{\mathbf{G}}^{\overline{S}}$. If all data bursts specified in $\mathbf{S}_{\mathbf{G}}^{\overline{S}}$ are scheduled within the stuck frame $x_{\mathbf{G}}$, the size M_n of the grant spaces in the stuck group $\mathbf{S}_{\mathbf{G}}$ and the total number of listen frames L_n will possess the relationship as $L_n = \lceil M_n / F_{max} \rceil$. Note that L_n is also denoted as the minimum number of required frames before the end of the stuck frame $x_{\mathbf{G}}$. Moreover, due to the stuck group definition in Definition 2, there is no other grant space in $(\mathbf{G} - \mathbf{S}_{\mathbf{G}})$ which can be scheduled at any frame $\theta_\xi \leq x_{\mathbf{G}}$. The listen frames for accommodating the stuck group $\mathbf{S}_{\mathbf{G}}$ will not be further updated or influenced by the scheduling of $(\mathbf{G} - \mathbf{S}_{\mathbf{G}})$. Therefore, based on Fact 2, the stuck group $\mathbf{S}_{\mathbf{G}} = \mathbf{G}_{-}$ can be removed without interfering the scheduling of the remaining grant spaces. All the removed grant spaces are considered completely scheduled with the minimum number of listen frames. It completes the proof. □

If the stuck group $\mathbf{S}_{\mathbf{G}}$ can be removed based on Lemma 3, the proposed FA mechanism will be re-conducted as the first round with the new stepwise grant space set $(\mathbf{G} - \mathbf{S}_{\mathbf{G}})$. On the other hand, after the execution of the packet padding mechanism, if there still exist some grant spaces in $\mathbf{S}_{\mathbf{G}}^{\overline{S}}$, the second round of backward push mechanism will be performed. The new stuck frame $x_{\mathbf{G}}$ will therefore be updated.

Lemma 4. *In the second round of the backward push mechanism, given the first observed no-follower frame θ_n, the set $\mathbf{G}_{-}$ consisting of all the grant spaces in $\mathbf{G}$ whose data bursts are scheduled before the end of frame θ_n can be removed without interfering the scheduling of the remaining grant spaces. The set $\mathbf{G}_{-}$ is considered completely scheduled within the minimum number of listen frames.*

Proof: In the second round of the backward push mechanism, based on the last grant space $G_\xi(s_\xi, g_\xi, t_\xi)$ that is scheduled in θ_n, the current strictly stuck subgroup can be divided into the grant space sets $\mathbf{A}$ and $\mathbf{B}$. The set $\mathbf{A}$ contains the grant space $G_\xi(s_\xi, g_\xi, t_\xi)$ and those before $G_\xi(s_\xi, g_\xi, t_\xi)$; while the remaining grant spaces are in set $\mathbf{B}$. Moreover, the grant space set $\mathbf{C}$ is also introduced to include the previously scheduled and not-yet removed grant spaces. Note that set $\mathbf{C}$ is perfectly scheduled within $L_{\mathbf{C}}$ fully-utilized frames, i.e., the following relationship will hold as $M_{\mathbf{C}} = L_{\mathbf{C}} \cdot F_{max}$ where $M_{\mathbf{C}}$ is the number of grant spaces in set $\mathbf{C}$. The listen frames that contain set $\mathbf{C}$, i.e., the previous scheduling results, may or may not influence the second round of the backward push mechanism. Therefore, the integer variable δ is introduced to represent the number of frames blocked by set $\mathbf{C}$. The relationship in (6) will therefore be updated as

$$P_n = \delta + W_{\mathbf{A}} + W_{\mathbf{B}}. \tag{9}$$

With the new P_n in (9), the inequality specified in (8) will still be satisfied since $W_{\mathbf{B}} + \delta \geq W_{\mathbf{B}}$. Based on (8) and Fact 1, since set $\mathbf{B}$ is bounded in the range $[\theta_n + 1, x_{\mathbf{G}}]$ with width $W_{\mathbf{B}}$, the frames in the range $[\theta_n + 1, x_{\mathbf{G}}]$ must be necessary listen frames if there exists a feasible scheduling method, i.e., $W_{\mathbf{B}} = \lceil M_{\mathbf{B}} / F_{max} \rceil$. In other words, the variable δ must be zero in order to have a feasible solution.

Note that there are at least $V = \lceil (M_{\mathbf{C}} + M_{\mathbf{A}} + M_{\mathbf{B}}) / F_{max} \rceil$ listen frames before the end of the stuck frame $x_{\mathbf{G}}$ in order to contain sets $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$. Moreover, since the $W_{\mathbf{B}}$ frames in the range $[\theta_n + 1, x_{\mathbf{G}}]$ have been proven as necessary listen frames, there must be at least $U_L = V - W_{\mathbf{B}} = L_{\mathbf{C}} + P_n - W_{\mathbf{B}} = L_{\mathbf{C}} + W_{\mathbf{A}}$ listen frames before the end of the no-follower frame θ_n. It can be observed that all data bursts in sets $\mathbf{A}$ and $\mathbf{C}$ can be scheduled before the end of the no-follower frame θ_n with $U_L = L_{\mathbf{C}} + W_{\mathbf{A}}$ listen frames. Furthermore, based on Definition 5, there will be no other grant space after the last grant space $G_\xi(s_\xi, g_\xi, t_\xi)$ scheduled in θ_n which can affect the scheduling of sets $\mathbf{A}$ and $\mathbf{C}$. Therefore, based on Fact 2, the set $\mathbf{G}_{-} = \mathbf{A} + \mathbf{C}$ can be removed without interfering the scheduling of the remaining grant spaces. The scheduling of $\mathbf{G}_{-}$ is completed with the minimum number of listen frames. It completes the proof. □

If there exists a no-follower frame in the second round of the backward push mechanism, the corresponding grant space set $\mathbf{G}_{-}$ can be removed from the stepwise grant space set $\mathbf{G}$ based on Lemma 4. All data bursts in the removed grant spaces are properly scheduled in the minimum number of listen frames. The remaining grant spaces will result in a new stepwise grant space set which initiates a first round execution of the backward push mechanism. On the other hand, if no-follower frames are not encountered, either the packet padding for the proposed FA mechanism (Lemma 5) or the proposed BA mechanism (Lemma 6) will be executed. The first case that conducts packet padding is considered as follows.

Lemma 5. *If all data bursts specified in the non-strictly stuck subgroup $\mathbf{S}_{\mathbf{G}}^{\overline{S}}$ are scheduled within the under-utilized frames of the second round backward push mechanism, the stuck group $\mathbf{S}_{\mathbf{G}}$ and all previously scheduled grant spaces can be removed without interfering the scheduling of the remaining grant spaces. All the removed grant spaces are considered completely scheduled within the minimum number of listen frames.*

Proof: It can be observed that all listen frames occupied by the data bursts of $\mathbf{S}_{\mathbf{G}}$ and all previously scheduled grant spaces are fully-utilized except for the possible under-utilized stuck frame. Since all data bursts of $\mathbf{S}_{\mathbf{G}}^{\overline{S}}$ can be placed within the stuck frame, the stuck group $\mathbf{S}_{\mathbf{G}}$ and all previously scheduled grant spaces are properly scheduled within the minimum number of listen frames. Moreover, based on Definition 2, the remaining grant spaces do not have the chance to be placed in either the stuck frame or the frames before. Finally, based on Fact 2, the stuck group $\mathbf{S}_{\mathbf{G}}$ and all previously scheduled grant spaces can be removed without interfering the scheduling of the remaining grant spaces. All those removed grant spaces are completely scheduled with the minimum number of listen frames. It completes the proof. □

If the grant space removal can be done based on Lemma 5, the remaining unscheduled grant spaces can form a new stepwise grant space set as the input for the first round of the backward push mechanism. The second case that executes the proposed BA mechanism is addressed as follows.

Lemma 6. *If the recursive backward movement of the proposed BA mechanism can not properly schedule the data bursts, there does not exist a feasible scheduling algorithm.*

Proof: When the recursive backward movement of the proposed BA mechanism is employed, the stuck frame $x_{\mathbf{G}}$ must be full of data bursts and there is still at least one data burst that should be scheduled in either the stuck frame or those frames before the stuck frame. Based on the recursive backward movement, the grant space scheduled in $x_{\mathbf{G}}$ which has the smallest start frame will be moved back one frame in order to verify whether there exists any vacancy in frame $x_{\mathbf{G}}-1$ next to the current frame $\theta_c^1 = x_{\mathbf{G}}$. If frame $x_{\mathbf{G}}-1$ is also fully-utilized, same process will be repeatedly conducted to verify the availability of frame $x_{\mathbf{G}}-2$ next to the current frame $\theta_c^2 = x_{\mathbf{G}}-1$. The case for the recursive backward movement not being able to complete the scheduling is under the condition that the current frame θ_c^i is the same as all the start frames of the grant spaces arranged in θ_c^i, and the current frame θ_c^i is unfortunately full of data bursts. It can be noted that there will be at least $(x_{\mathbf{G}} - \theta_c^i + 1) \cdot F_{max} + 1$ grant spaces bounded in the range $[\theta_c^i, x_{\mathbf{G}}]$. Therefore, based on Fact 1, there does not exist a feasible scheduling algorithm which completes the proof. □

Lemma 7. *If there is a no-follower frame θ_n observed during the execution of the proposed BA mechanism, the set $\mathbf{G}_-$ consisting of all grant spaces whose data bursts scheduled in the frame θ_n and those frames before θ_n can be removed without interfering the scheduling of the remaining grant spaces. All the removed grant spaces are considered completely scheduled with the minimum number of listen frames.*

Proof: It is assumed that the no-follower frame θ_n is obtained during the scheduling the grant space $G_\xi(s_\xi, g_\xi, t_\xi)$. Based on Lemma 6, the data burst ξ in $G_\xi(s_\xi, g_\xi, t_\xi)$ will be properly scheduled if a feasible scheduling algorithm exists. In order to provide a vacancy for ξ, one data burst will be moved to the first observed under-utilized frame θ_u in the reverse direction. All previously scheduled and not-yet removed grant spaces can be divided into three sets. Set $\mathbf{A}$ contains all grant spaces whose data bursts are arranged in the range $[\theta_n+1, x_{\mathbf{G}}]$. The set $\mathbf{B}$ includes those grant spaces in the range $[\theta_u, \theta_n]$; while the remaining ones are in set $\mathbf{C}$. M_ξ and L_ξ are respectively denoted as the number of data bursts and listen frames in the set, i.e., $\xi \in \{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$. It can be observed that the listen frames for set $\mathbf{A}$ and $\mathbf{C}$ are fully-utilized, i.e., $M_{\mathbf{A}} = L_{\mathbf{A}} \cdot F_{max}$ and $M_{\mathbf{C}} = L_{\mathbf{C}} \cdot F_{max}$. Furthermore, the listen frames for set $\mathbf{B}$ possess the relationship as $L_{\mathbf{B}} = \lceil M_{\mathbf{B}}/F_{max} \rceil$.

Based on Definitions 2 and 5, the grant spaces in $\mathbf{A}$ are bounded in the range $[\theta_n+1, x_{\mathbf{G}}]$. According to Fact 1, all $L_{\mathbf{A}}$ are necessary listen frames which result in at least $U_L = \lceil (M_{\mathbf{A}} + M_{\mathbf{B}} + M_{\mathbf{C}})/F_{max} \rceil - L_{\mathbf{A}} = L_{\mathbf{B}} + L_{\mathbf{C}}$ listen frames before the end of frame θ_n. Based on Fact 2, the set $\mathbf{G}_- = \mathbf{B} + \mathbf{C}$ can be scheduled in these U_L listen frames and no other grant space can affect the scheduling of $\mathbf{G}_-$. Therefore, set $\mathbf{G}_-$ can be removed without interfering the scheduling of the remaining grant spaces. All the removed grant spaces are considered completely scheduled within the minimum number of listen frames. It completes the proof. □

Lemma 8. *After executing the packet padding of the proposed BA mechanism, if no more grant space in the non-strictly stuck subgroup can be used, all scheduled grant spaces can be removed without interfering the scheduling of the remaining grant spaces. All the removed grant spaces are considered completely scheduled within the minimum number of listen frames.*

Proof: The packet padding mechanism will occur if there exists an under-utilized frame θ_u after the execution of the recursive backward movement. This frame θ_u is also the only under-utilized frame in the current scheduled frames since the recursive backward movement will skip all the fully-utilized frames from the stuck frame and find the first under-utilized frame to schedule the data burst. The packet padding mechanism will at most make the frame θ_u to become a fully-utilized frame. Therefore, if all grant spaces in the non-strictly stuck subgroup are utilized in the packet padding process, the grant spaces scheduled before the end of stuck frame can be arranged in the minimum required number of listen frames. Moreover, since no more grant space in the non-strictly stuck subgroup can be used, there is no other grant space which can be scheduled in the stuck frame and those frames before the stuck frame. Based on Fact 2, all scheduled grant spaces can be removed without interfering the scheduling of the remaining grant spaces. Consequently, all the removed grant spaces are completely scheduled within the minimum number of listen frames. It completes the proof. □

If the removal process is conducted, a new stepwise grant space set consisting of the remaining grant spaces can be constructed and the first round of the backward push mechanism will be restarted on this new set. On the other hand, if there are still grant spaces in the non-strictly stuck subgroup, the second round of the backward push mechanism will be repeatedly performed in order to make all the frames fully-utilized. At the end, based on the above proofs of correctness for the flow of the proposed FAPS algorithm in Fig. 11, if there exists a feasible packet arrangement for a given stepwise grant space set $\mathbf{G}$ under the frame capacity F_{max}, the proposed FAPS algorithm can result in the minimum number of listen frames. The optimality of the proposed FAPS algorithm can therefore be obtained.

V. Performance Evaluation

As shown in Fig. 1, type III power-saving class can be directly adopted for the proposed FAPS scheme since all kind of combinations for a listen interval followed by a sleep interval is suitable for type III power saving class. Moreover, it is also possible to transform the power-saving type II class into type III class such that the proposed FAPS scheme can still be applied. The work in [31] implemented a fold-and-demultiplex (FD) method in order to provide a demultiplexing mechanism. The FD scheme first folds all the type II traffic together into a series of 1s interleaved by 0s to calculate the bandwidth requirement. The wake-up period (i.e., sleep period plus listen period) of each connection is adopted as half of the delay constraint, and then all the wake-up periods are adjusted with integer multiples of the smallest one. This work further proves that the QoS requirement can be guaranteed if the delay constraint is two times larger than the smallest wake-up period. The sleep pattern that matches type III power-saving class can be further demultiplexed into multiple type II patterns. Therefore, the FAPS scheme can either be implemented based

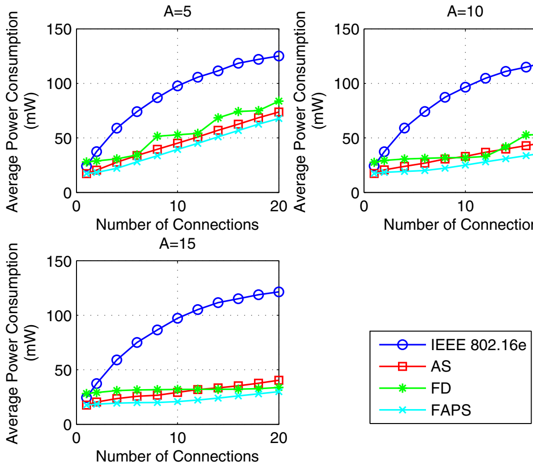


Fig. 12. Performance comparison of average power consumption versus number of connections under different frame capacities $A = F_{max} = 5$, 10, and 15.

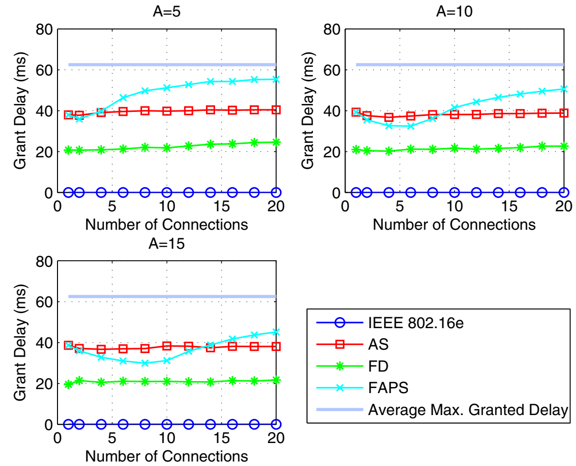


Fig. 13. Performance comparison of grant delay versus number of connections under different frame capacities $A = F_{max} = 5$, 10, and 15.

on type III class or adopt similar demultiplexing method as FD scheme for the implementation of type II class.

In this section, simulations are conducted to evaluate the performance of the proposed FAPS scheduling algorithms in comparison with the AS scheme [30], the FD scheme [31], and the original power-saving mechanism in the IEEE 802.16e specification. A single BS/MSS pair with multiple connections are considered as the simulation scenario. 100 frames with each frame duration of 5 ms are utilized in the simulations. The default settings of each connection are listed as follows. The maximum allowable data bursts in a frame, i.e., the frame capacity F_{max}, is equal to 5. The connection period for each CID is uniformly distributed from $[6, 10]$, which denotes the number of frames between each two consecutive packet arrivals. The delay constraint is assumed to range from 50 ms to 75 ms for each data burst. As suggested in the [25], the power consumption of the listen interval is 150 mW and that of the sleep interval is 10 mW. Two performance metrics are adopted for performance comparison as follows: (a) the average power consumption [25]: the energy consumption divided by the total simulated duration; and (b) the grant delay: the average value of (g_i - s_i) for each grant space $G_i(s_i, g_i, t_i)$.

As shown in Fig. 12, performance comparison of the average power consumption versus the number of connections is conducted under different frame capacities $F_{max} = 5$, 10, and 15. The curves for the IEEE 802.16e standard are almost the same under different values of F_{max} since it does not rearrange the data packets and only schedules the packets at the original start frames specified in the packets' corresponding grant spaces. On the other hand, the performance of the average power consumption for the proposed FAPS protocol outperforms the other connection-oriented schemes under specific F_{max} since the FAPS protocol can effectively rearrange the data packets in the packet level. As the number of connections is increased, the average power consumption is increased in all schemes since more frames should be in the awake state to serve the augmented number of packets. Note that when the frame capacity becomes larger, the average power consumption is decreased since less number of frames will be required to contain the data packets.

Fig. 13 shows the performance curves of the grant delay under different frame capacities F_{max}. The grant delay performance of the IEEE 802.16e protocol should be zero since it does not delay its packets. As the frame capacity becomes smaller, it can be seen from the proposed FAPS, the AS and the FD schemes that additional grant delay will be generated since more packets are delayed in order to seek for proper positions to accommodate themselves. Due to the connection-oriented manner, the AS scheme maintains at the same level of delay performance under different numbers of connections since it conducts scheduling mechanism according to the connection with the smallest delay constraint, which dominates the performance curves. On the other hand, the FD scheme recalculates the listen and sleep interval in order not to violate half of the delay constraint. Intuitively, the grant delay of the proposed FAPS scheme should be larger than that of the FD and the AS algorithms since the proposed FAPS scheme will result in the best performance in the average power consumption. However, owing to its packet-level design concept for packet aggregation, the proposed FAPS scheme will result in smaller delay in certain cases compared to the AS method. Moreover, it can also be observed that all the performance curves are below the average maximum grant delay, which fulfills the QoS delay requirements.

The performance curves of the average power consumption versus the number of connections under different connection periods are shown in Fig. 14. Note that larger connection period denotes that packets arrive less frequently compared to that with smaller connection period. Since shorter connection period indicates higher number of packet arrivals, all the four protocols result in increased average power consumption with smaller connection period. Moreover, all these three schemes show the increasing trend with regard to the number of connections since more frames are required to accommodate the packets provided by the connections. It can be observed that the proposed FAPS protocol will possess lower average power consumption compared to the other two connection-

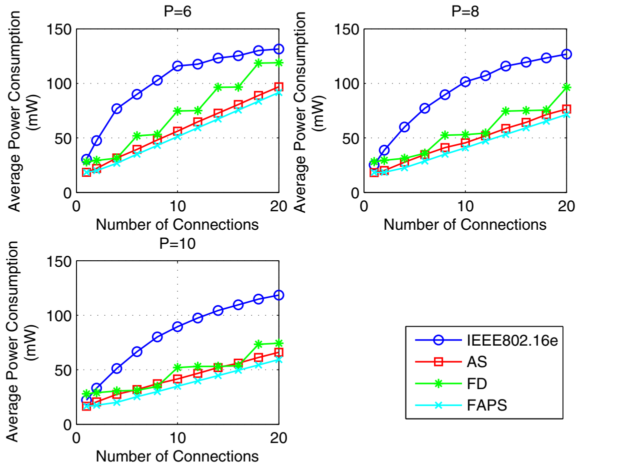


Fig. 14. Performance comparison of average power consumption versus number of connections under different connection periods $P = 6$, 8, and 10.

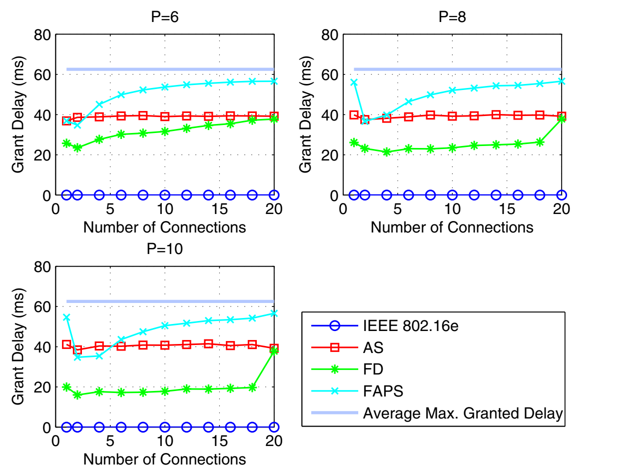


Fig. 15. Performance comparison of grant delay versus number of connections under different connection periods $P = 6$, 8, and 10.

oriented schemes under each specific connection period. The reason is owing to the design of FAPS algorithm that considers the packet as the unit of scheduling instead of the connection, which effectively results in larger number of sleep intervals.

Fig. 15 shows the performance comparison of the grant delay versus the number of connections under different connection periods. Without conducting packet arrangement, zero value of grant delay can be observed for the IEEE 802.16e standard. In the proposed FAPS scheme and the FD scheme, as the connection period becomes smaller, more grant delay will be generated in order to properly accommodate those additional packet arrivals. On the contrary, the AS scheme will possess higher grant delay as the connection period becomes larger. This phenomenon comes from the design of the AS scheme which arranges packets starting from the termination of each connection. Therefore, in the case of a larger connection period, a great amount of packets will be placed around the termination of connection leading to the higher grant delay performance. The grant delay of the proposed FAPS scheme is larger than that of the FD and the AS algorithms since the FAPS protocol has the best performance in the average power consumption. Note that the QoS delay requirements can be achieved by all the four schemes since all curves are below the average maximum grant delay. It can be observed that the merits of the proposed FAPS scheme are fully supported via simulations with tolerable delay.

VI. Conclusion

In this paper, with the consideration of multiple connections and their QoS constraints, a packet-level frame aggregation-based power-saving scheduling (FAPS) algorithm is proposed to maximize the number of total sleep frames by frame aggregation techniques. Based on the stepwise grant space set, the optimality on the minimum number of listen frames by adopting the proposed FAPS algorithm is provided and is further verified via the proof of correctness. Simulation studies show that the power efficiency of the proposed FAPS algorithm outperforms the other baseline protocols with tolerable delay.

Acknowledgements

The authors would like to thank Li-Ping Tung, Wun-Ci Su, and Jui-Hung Chu for their valuable suggestions of the manuscript. They would also like to thank the anonymous reviewers for their constructive comments.

References

[1] IEEE 802.16 WG, IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interference for Fixed Broadband Wireless Access Systems, IEEE Standards Association Std., 2004.

[2] IEEE 802.16 WG, IEEE Standard 802.16e-2005: Part 16: Air Interference for Fixed Broadband Wireless Access Systems, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1, IEEE Standards Association Std., 2006.

[3] I. Koffman and V. Roman, "Broadband wireless access solutions based on OFDM access in IEEE 802.16," *IEEE Commun. Mag.*, vol. 40, pp. 96–103, Apr. 2004.

[4] C. Cicconetti, L. Lenzini, E. Mingozzi, and C. Eklund, "Quality-of-service support in IEEE 802.16 networks," *IEEE Netw.*, vol. 20, pp. 50–55, Mar. 2006.

[5] A. Ghosh, D. R. Wolter, J. G. Andrews, and R. Chen, "Broadband wireless access with WiMAX/802.16: current performance benchmarks and future potential," *IEEE Commun. Mag.*, vol. 43, pp. 129–136, Feb. 2005.

[6] Y. Zhang, Y. Xiao, and V. C. M. Leung, "Energy management analysis and enhancement in IEEE 802.16e wirelessMAN," *IEEE Trans. Veh. Technol.*, vol. 58, pp. 3738–3752, Sep. 2009.

[7] C. Cicconetti, L. Lenzini, E. Mingozzi, and C. Vallati, "Reducing power consumption with QoS constraints in IEEE 802.16e wireless networks," *IEEE Trans. Mobile Comput.*, vol. 9, pp. 1008–1021, July 2010.

[8] L. Kong, G. K. W. Wong, and D. H. K. Tsang, "Performance study and system optimization on sleep mode operation in IEEE 802.16e," *IEEE Trans. Wireless Commun.*, vol. 8, pp. 4518–4528, Sep. 2009.

[9] M. G. Kim, J. Choi, and M. Kang, "Enhanced power-saving mechanism to maximize operational efficiency in IEEE 802.16e systems," *IEEE Trans. Wireless Commun.*, vol. 8, pp. 4710–4719, Sep. 2009.

[10] J. A. Stine and G. Veciana, "Improving energy efficiency of centrally controlled wireless data networks," *ACM/Baltzer Wireless Netw.*, vol. 8, pp. 681–700, 2002.

[11] IEEE 802.11 WG, IEEE 802.11n-2009: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment 5: Enhancements for Higher Throughput, IEEE Standards Association Std., Oct. 2009.

[12] R. Krashinsky and H. Balakrishnan, "Minimizing energy for wireless web access with bounded slow-down," in *Proc. 2002 ACM/IEEE Int. Conf. Mobile Comput. Netw.*, pp. 119–130.

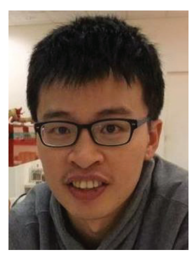
[13] D. Qiao, S. Choi, A. Soomro, and K. G. Shin, "Energy-efficient PCF operation of IEEE 802.11a wireless LAN," in *Proc. 2002 IEEE Int. Conf. Comput. Commun.*, pp. 580–589.
[14] G. Anastasi, M. Conti, E. Gregori, and A. Passarella, "A performance study of power-saving polices for Wi-Fi hotspots," *Comput. Netw.*, vol. 45, pp. 295–318, June 2004.
[15] F. Zhang, T. D. Todd, D. Zhao, and V. Kezys, "Power saving access points for IEEE 802.11 wireless network infrastructure," *IEEE Trans. Mobile Comput.*, vol. 5, pp. 144–156, Feb. 2006.
[16] J. C. Chen and K. W. Cheng, "EDCA/CA: enhancement of IEEE 802.11e EDCA by contention adaption for energy efficiency," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 2866–2870, Aug. 2008.
[17] K. T. Feng and K. H. Chou, "Intelligent router-assisted power saving medium access control for mobile ad hoc networks," in *Proc. 2006 IEEE Veh. Technol. Conf. – Spring*, pp. 304–308.
[18] Y. Xiao, "Energy saving mechanism in the IEEE 802.16e wireless MAN," *IEEE Commun. Lett.*, vol. 9, pp. 595–597, July 2005.
[19] K. Han and S. Choi, "Performance analysis of sleep mode operation in IEEE 802.16e mobile broadband wireless access systems," in *Proc. 2006 IEEE Veh. Technol. Conf. – Spring*, pp. 1141–1145.
[20] Y. Park and G. U. Hwang, "Performance modelling and analysis of the sleep-mode in IEEE802.16e WMAN," in *Proc. 2007 IEEE Veh. Technol. Conf. – Spring*, pp. 2801–2806.
[21] Y. Zhang, "Performance modeling of energy management mechanism in IEEE 802.16e mobile WiMAX," in *Proc. 2007 IEEE Wireless Commun. Netw. Conf.*, pp. 3205–3209.
[22] Y. Xiao, "Performance analysis of an energy saving mechanism in the IEEE 802.16e wireless MAN," in *Proc. 2006 Consum. Commun. Netw. Conf.*, pp. 406–410.
[23] L. Kong and D. H. K. Tsang, "Optimal selection of power saving classes in IEEE 802.16e," in *Proc. 2007 IEEE Wireless Commun. Netw. Conf.*, pp. 1836–1841.
[24] J. Shi, G. Fang, Y. Sun, J. Zhou, Z. Li, and E. Dutkiewicz, "Improving mobile station energy efficiency in IEEE 802.16e WMAN by burst scheduling," in *Proc. 2006 IEEE Global Commun. Conf.*, pp. 1–5.
[25] E. Hwang, K. J. Kim, J. J. Son, and B. D. Choi, "The power-saving mechanism with periodic traffic indications in the IEEE 802.16e/m," *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 319–334, Jan. 2010.
[26] T. C. Chen, Y. Y. Chen, and J. C. Chen, "An efficient energy saving mechanism for IEEE 802.16e wireless MANs," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 3708–3712, Oct. 2008.
[27] T. C. Chen, J. C. Chen, and Y. Y. Chen, "Maximizing unavailability interval for energy saving in IEEE 802.16e wireless MANs," *IEEE Trans. Mobile Comput.*, vol. 8, pp. 475–487, Apr. 2009.
[28] T. C. Chen and J. C. Chen, "Extended maximizing unavailability interval (eMUI): maximizing energy saving in IEEE 802.16e for mixing type I and type II PSCs," *IEEE Commun. Lett.*, vol. 13, pp. 151–153, Feb. 2009.
[29] C. C. Yang, Y. T. Mai, J. Y. Chen, Y. S. Shen, and Y. C. Kuo, "LBPS: load-based power saving in the IEEE 802.16e network," *Elsevier Comput. Electrical Eng.*, vol. 38, no. 4, pp. 891–905, 2012.
[30] Y. L. Chen and S. L. Tsao, "Energy-efficient sleep-mode operations for broadband wireless access systems," in *Proc. 2006 IEEE Veh. Technol. Conf. – Fall*, pp. 1–5.
[31] Y. C. Tseng, J. J. Chen, and Y. C. Yang, "Managing power saving classes in IEEE 802.16 wireless MANs: a fold-and-demultiplex method," *IEEE Trans. Mobile Comput.*, vol. 10, no. 9, pp. 1237–1247, 2011.

Wen-Jiunn Liu (S'07) received his B.S. and Ph.D. degrees in communications engineering from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in June 2005 and June 2010, respectively. His current research interests include MAC and network protocol design for wireless sensor networks and broadband wireless networks. Since Oct. 2010, he has been with Realtek Semiconductor Corp., Hsinchu, Taiwan, R.O.C., working on Wi-Fi protocol stack feature development.

Kai-Ten Feng (M'03-SM'13) received the B.S. degree from National Taiwan University, Taipei, Taiwan, in 1992, the M.S. degree from the University of Michigan, Ann Arbor, in 1996, and the Ph.D. degree from the University of California, Berkeley, in 2000.

Since August 2011, he has been a full Professor with the Department of Electrical and Computer Engineering, National Chiao Tung University (NCTU), Hsinchu, Taiwan, where he was an Associate Professor and Assistant Professor from August 2007 to July 2011 and from February 2003 to July 2007, respectively. He also serves as the Director of the Digital Content Production Center at NCTU (since October 2011). From July 2009 to March 2010, he was a Visiting Research Fellow with the Department of Electrical and Computer Engineering, University of California at Davis. Between 2000 and 2003, he was an In-Vehicle Development Manager/Senior Technologist with OnStar Corporation, a subsidiary of General Motors Corporation, where he worked on the design of future telematics platforms and in-vehicle networks. His current research interests include broadband wireless networks, cooperative and cognitive networks, smart phone and embedded system designs, wireless location technologies, and intelligent transportation systems.

Dr. Feng received the Best Paper Award from the Spring 2006 IEEE Vehicular Technology Conference, which ranked his paper first among the 615 accepted papers. He also received the Outstanding Youth Electrical Engineer Award in 2007 from the Chinese Institute of Electrical Engineering, and the Distinguished Researcher Award from NCTU in 2008, 2010, and 2011. He has also served on the technical program committee of various international conferences.



Po-Hsuan Tseng (M'12) received the B.S. and Ph.D. degrees in communication engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2005 and 2011, respectively. Since August 2012, he has been an assistant professor with the Department of Electronic Engineering, National Taipei University of Technology (NTUT), Taipei, Taiwan. From January 2010 to October 2010, he was a visiting researcher with the University of California at Davis. His research interests are in the areas of signal processing for networking and communications, including location estimation and tracking, cooperative localization, and mobile broadband wireless access system design. He is an honorary member of the Phi Tau Phi Scholastic Honor Society of R.O.C. He is a member of the IEEE.