Bandwidth Sharing Schemes for Multimedia Traffic in the IEEE 802.11e Contention-Based WLANs

Yang Xiao, *Senior Member*, *IEEE*, Frank Haizhon Li, *Student Member*, *IEEE*, and Bo Li, *Senior Member*, *IEEE*

Abstract—Bandwidth allocation schemes have been well studied for mobile cellular networks. However, there is no study about this aspect reported for IEEE 802.11 contention-based distributed wireless LANs. In cellular networks, bandwidth is deterministic in terms of the number of channels by frequency division, time division, or code division. On the contrary, bandwidth allocation in contention-based distributed wireless LANs is extremely challenging due to its contention-based nature, packet-based network, and the most important aspect: only one channel is available, competed for by an unknown number of stations. As a consequence, guaranteeing bandwidth and allocating bandwidth are both challenging issues. In this paper, we address these difficult issues. We propose and study nine bandwidth allocation schemes, called sharing schemes, with guaranteed Quality of Service (QoS) for integrated voice/video/data traffic in IEEE 802.11e contention-based distributed wireless LANs. A guard period is proposed to prevent bandwidth allocation from overprovisioning and is for best-effort data traffic. Our study and analysis show that the guard period is a key concept for QoS guarantees in a contention-based channel. The proposed schemes are compared and evaluated via extensive simulations.

Index Terms—Admission control, IEEE 802.11e, quality of service, resource management, wireless LANs.

1 Introduction

In recent years, the market for IEEE 802.11 wireless local area networks (WLANs) has enjoyed tremendous growth, partially due to potential applications of WLANs such as convenient Internet/database access and high speed communications with reasonable costs. We have also witnessed phenomenal growth in cellular data services and emerging wireless multimedia applications. Bandwidth allocation schemes with Quality of Service (QoS) guarantees in wireless cellular networks have been well studied. In cellular networks, bandwidth is deterministic in terms of the number of channels by frequency division, time division, or code division. On the contrary, bandwidth allocation in distributed wireless LANs is extremely challenging due to its contention-based nature, packet-based network, and the most important aspect: only one channel is available, competed for by an unknown number of stations under different traffic patterns. Therefore, guaranteeing bandwidth and allocating bandwidth are both extremely difficult. These open up a new research avenue and call for novel ways to support QoS in contention-based distributed WLANs.

- *Y. Xiao is with the Department of Computer Science, University of Alabama, 101 Houser Hall, Box 870290, Tuscaloosa, AL 35487. E-mail: yangxiao@ieee.org.*
- *F.H. Li is with the Division of Mathematics and Computer Science, University of South Carolina Upstate, 800 University Way, Spartanburg, SC 29303. E-mail: fli@uscupstate.edu.*
- *B. Li is with the Computer Science Department, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China. E-mail: bli@cs.ust.hk.*

Manuscript received 12 Dec. 2004; revised 20 Dec. 2005; accepted 4 Aug. 2006; published online 11 May 2007.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0334-1204.
Digital Object Identifier no. 10.1109/TMC.2007.1054.

The medium access control (MAC) of the IEEE 802.11 WLAN employs a mandatory contention-based channel access function, called the Distributed Coordination Function (DCF), and an optional centrally controlled channel access function, called the Point Coordination Function (PCF) [1]. The DCF adopts a carrier sense multiple access with collision avoidance (CSMA/CA) with binary exponential backoff. The IEEE 802.11 DCF enables fast installation with minimal management and maintenance costs and is very robust protocol for the best-effort service. The popularity of the IEEE 802.11 WLAN is mainly due to the DCF, whereas the PCF is barely implemented in today's products due to its complexity and inefficiency for normal data transmissions, even though it has some limited QoS support. To support the MAC-level QoS, the IEEE 802.11 Working Group has recently developed IEEE 802.11e [2], providing QoS features and multimedia support to the existing 802.11a/b/g [3], [4] WLANs while maintaining a full backward compatibility with these legacy standards. The IEEE 802.11e MAC employs a channel access function, called the Hybrid Coordination Function (HCF), which includes both contention-based channel access and centrally-controlled channel access mechanisms. The contention-based channel access mechanism is also referred to as *Enhanced Distributed Channel Access* (EDCA). The EDCA provides a priority scheme by differentiating the interframe space as well as the initial and the maximum contention window sizes for backoff procedures.

In the previous work in [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], the main focus was on studying the EDCA mechanisms and differentiated services. However, the schemes proposed in [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15] cannot provide guaranteed QoS. One of the key challenges in the design of bandwidth allocation policies is to guarantee the different QoS requirements while at the same time ensuring that the scarce bandwidth is utilized efficiently. Readers are recommended to refer to [13] for more work on differentiation/priority service. In [23], we proposed a global measurement-based data control scheme. In [24], we proposed a local measurement-based data control scheme as well as admission control without the presence of access points.

In this paper, we propose several bandwidth allocation mechanisms for the EDCA with guaranteed QoS. Guaranteed QoS is achieved by admission control mechanisms and best-effort data control mechanisms. In admission control mechanisms, video and voice flows are accepted or rejected based on the available budget(s), and a guard period is proposed to prevent bandwidth allocation from over-provisioning. In the data control mechanism, best-effort data parameters are dynamically controlled based on traffic conditions. The main focus of this paper is to study more efficient bandwidth allocation mechanisms with guaranteed QoS for voice and video. We propose nine bandwidth sharing schemes:

1. complete sharing,
2. forward-voice reserved sharing,
3. backward-voice reserved sharing,
4. forward-video reserved sharing,
5. backward-video reserved sharing,
6. forward reserved sharing,
7. backward reserved sharing,
8. backward-voice forward-video reserved sharing, and
9. forward-voice backward-video reserved sharing.

In the complete sharing scheme, voice flows and video flows share the bandwidth completely. Some bandwidth can be reserved for either voice flows or video flows or both if needed. The order of using bandwidth can also be classified into two approaches, i.e., forward and backward. All the schemes can provide guaranteed QoS for voice and video. The proposed schemes are evaluated with extensive simulations.

In this paper, a bandwidth guarantee means that the quality of voice and video flows is maintained by two mechanisms: admission control and data control. Admission control protects existing voice/video flows from new voice/video flows and data control protects existing voice/video flows from best-effort data traffic. The guarantee is a soft statistical guarantee, but not a hard guarantee.

The rest of paper is organized as follows: We briefly introduce the IEEE 802.11 DCF and the 802.11e EDCA in Section 2. Sharing schemes and a guard concept are proposed in Section 3. Section 4 presents detailed implementations of sharing schemes via admission control mechanisms and a data control algorithm. Performance studies are carried out in Section 5 with extensive simulation results. We conclude our paper in Section 6.

2 IEEE 802.11 DCF and IEEE 802.11e EDCA

We briefly introduce the IEEE 802.11 DCF and the IEEE 802.11e EDCA, an earlier 802.11e draft, i.e., IEEE 802.11e/D4.3, for differentiated services of EDCA in Section 2.1 and Section 2.1, respectively.

2.1 IEEE 802.11 DCF

The IEEE 802.11 MAC employs a mandatory DCF and an optional PCF. In a long run, time is divided into repetition intervals called *superframes*. Each superframe starts with a beacon frame, and the remaining time is further divided into a contention-free period (CFP) and a contention period (CP). The DCF works during the CP and the PCF works during the CFP. If the PCF is not active, superframes do not exist. However, the beacon frames are periodically transmitted irrespectively. The beacon frame is a management frame for synchronization, power management, and delivering network operation parameters. Beacon frames are generated in regular intervals called target beacon transmission times (TBTTs).

The DCF defines a basic access mechanism and an optional request-to-send/clear-to-send (RTS/CTS) mechanism. Under the DCF, a station with a frame to transmit monitors the channel activities until an idle period equal to a distributed interframe space (DIFS) is detected. After sensing an idle DIFS, the station waits for a random backoff interval before transmitting. The backoff time counter is decremented in terms of slot time as long as the channel is sensed idle. The counter is suspended when a transmission is detected on the channel and resumed with the old remaining backoff interval when the channel is sensed idle again for a DIFS interval. The station transmits its frame when the backoff timer reaches zero. For each new transmission attempt, the backoff interval is uniformly chosen from the range [0, $CW - 1$] in terms of timeslots, where CW is the current backoff window size. At the very first transmission attempt, CW equals the initial backoff window size CW_{min}. After each unsuccessful transmission, CW is doubled until a maximum backoff window size value CW_{max} is reached. After the destination station successfully receives the frame, it transmits an acknowledgment frame (ACK) following a short interframe space (SIFS) time. If the transmitter station does not receive an ACK within a specified ACK timeout, it reschedules the frame transmission according to the backoff rules discussed above.

2.2 IEEE 802.11e EDCA

IEEE 802.11e provides a channel access function, called the Hybrid Coordination Function (HCF), to support applications with QoS requirements. The HCF includes both contention-based channel access and centrally controlled channel access schemes. The contention-based channel access of the HCF is also referred to as the Enhanced Distributed Coordination Function (EDCA).

The EDCA works with four Access Categories (ACs), which are virtual DCFs, where each AC achieves a differentiated channel access. This differentiation is achieved through varying the amount of time a station would sense the channel to be idle and the length of the contention window for a backoff. The EDCA supports eight different priorities which are further mapped into four ACs, where AC_VO (or AC 3), AC_VI (or AC 2), AC_BE (or AC 1), and AC_BK (or AC 0) correspond to voice, video, best-effort, and background traffic, respectively. We, in this work, assume that data traffic is served via AC 0. Differentiated ACs are achieved by differentiating the arbitration interframe space (AIFS), the initial window size, and the maximum window

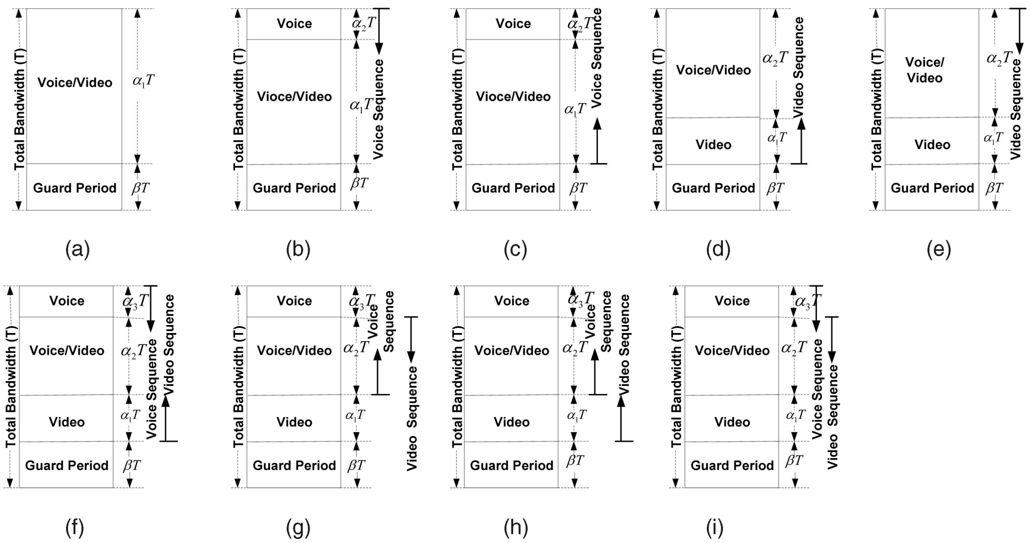

Fig. 1. Sharing schemes. For (a) to (i), we have $(\beta + sum(\alpha_i) = 1)$: (a) complete sharing, (b) forward-voice reserved sharing, (c) backward-voice reserved sharing, (d) forward-video reserved sharing, (e) backward-video reserved sharing, (f) forward reserved sharing, (g) backward reserved sharing, (h) backward-voice forward-video reserved sharing, and (i) forward-voice backward-video reserved sharing.

size. That is, for AC i $(i = 0, 1, 2, 3)$, the initial backoff window size is $CW_{min}[i]$, the maximum backoff window size is $CW_{max}[i]$, and the arbitration interframe space is $AIFS[i]$. For $0 \leq i < j \leq 3$, we have $CW_{min}[i] \geq CW_{min}[j]$, $CW_{max}[i] \geq CW_{max}[j]$, and $AIFS[i] \geq AIFS[j]$. In other words, the EDCA employs $AIFS[i]$, $CW_{min}[i]$, and $CW_{max}[i]$ (all for $i = 0, 1, 2, 3$) instead of DIFS, CW_{min}, and CW_{max}, respectively. If one AC has a smaller $AIFS$, CW_{min}, or CW_{max}, the AC's traffic has a better chance to access the wireless medium earlier.

Four transmission queues are implemented in a station, and each queue supports one AC behaving roughly as a single DCF entity in the original IEEE 802.11 MAC. It is assumed that a payload from a higher layer is labeled with a priority value and it is enqueued into the corresponding queue according to the mapping. Each queue acts as an independent MAC entity and performs the channel access with a different interframe space $(AIFS[i])$, a different initial window size $(CW_{min}[i])$, and a different maximum window size $(CW_{max}[i])$. Each queue has its own backoff counter $(BO[i])$, which acts independently in the same way as the original DCF backoff counter. If there is more than one queue finishing the backoff at the same time, the highest AC frame is chosen to transmit by the virtual collision handler. Other lower AC frames whose backoff counters also reach zero will increase their backoff counters with $CW_{min}[i](i = 0, \ldots, 3)$, accordingly.

The values of $AIFS[i]$, $CW_{min}[i]$, and $CW_{max}[i]$ (all for $i = 0, \ldots, 3$) are referred to as the EDCA parameters, which will be announced by the QoS Access Point (QAP) via periodically transmitted beacon frames. The QAP can also adaptively adjust these EDCA parameters based on the network traffic conditions.

3 Sharing Schemes and Guard Period

In this section, we propose sharing schemes and a guard period concept in Section 3.1 and Section 3.2, respectively. In Section 3.3, we provide an approximation and analysis on choice of the guard period.

3.1 Sharing Schemes

If a portion of bandwidth is shared, it is shared without differentiations among the involving ACs. On the other hand, if a portion of bandwidth is reserved, one particular AC occupies the whole portion.

To define bandwidth in a contention-based channel is not as simple as in a cellular network. In this paper, measurements are conducted during each regular time interval, which can be a beacon interval or several beacon intervals. We define bandwidth as the time interval between two measurements, and it is a constant value. We propose in total nine sharing schemes shown in Fig. 1: Fig. 1a shows complete sharing, Fig. 1b shows forward-voice reserved sharing, Fig. 1c shows backward-voice reserved sharing, Fig. 1d shows forward-video reserved sharing, Fig. 1e shows backward-video reserved sharing, Fig. 1f shows forward reserved sharing, Fig. 1g shows backward reserved sharing, Fig. 1h shows backward-voice forward-video reserved sharing, and Fig. 1i shows forward-voice backward-video reserved sharing.

Let T denote the total bandwidth, which is the time interval between two measurements. In the complete sharing scheme shown in Fig. 1a, total bandwidth is divided into two portions: a complete sharing region $(\alpha_1 T)$ for both voice and video and a guard period (βT) to prevent bandwidth allocation from over provisioning and for best-effort data traffic, where $\beta + \alpha_1 = 1$. The guard period will be explained in the next section. Some bandwidth can be reserved for either voice flows or video flows, or both if needed. The order of using bandwidth can be also classified into two approaches, i.e., forward and backward. In a forward scheme, bandwidth is used first in the reserved region if having any, and then in the shared region. In a backward scheme, bandwidth is used first in the shared region, and then in the reserved region if having any. For example, in the forward-voice reserved sharing scheme shown in Fig. 1b, an $\alpha_2 T$ portion of the total bandwidth is reserved for voice flows only, an $\alpha_1 T$ portion of the total bandwidth is for complete sharing among voice and video flows, and a βT portion of the total bandwidth is for the guard period, where $\beta + \alpha_1 + \alpha_2 = 1$; since Fig. 1b is a forward scheme, voice flows first use bandwidth in the reserved region, and if the reserved region has no bandwidth/budget left, the shared region can be used for voice flows too. The budget concept will be introduced in the next section. Parameters such as β and α are system-tunable parameters, which are chosen based on a system's requirements, such as what percentage of the bandwidth is reserved for video and voice.

Intuitively, the complete sharing scheme is the best in terms of utilization among the nine sharing schemes; however, with a reserved region for an AC, some minimum traffic for this AC can be guaranteed and more traffic for the reserved AC may be accepted in the system depending on the need for this AC. A forward scheme is more efficient than a backward scheme since the backward scheme is a selfish and greedy scheme for the reserved AC; however, in a backward scheme, more traffic for the reserved AC may be accepted in the system. These intuitions will be verified with simulations in later sections.

3.2 Guard Period and Data Traffic

In each sharing scheme, a guard period, shown in Fig. 1, is needed for two reasons: 1) as a guard bandwidth reserved partially for collision and idle time to prevent bandwidth allocation from over provisioning and 2) as a period reserved partially for best-effort data traffic.

The reason that the guard period has the function as a guard bandwidth reserved partially for collision and idle time to prevent bandwidth allocation from overprovisioning is stated as follows: The channel is a contention channel, in which there are idle time, successful transmissions (including associated overhead such as ACK frames and interframe spaces such as DIFS and SIFS time), and collisions (including associated overhead such as ACK frames and interframe spaces such as DIFS and SIFS time). No matter how smartly the MAC protocol is designed, there is no way to eliminate both idle period and failure transmission at the same time as long as the channel is contention-based. The following behaviors have been observed in many contention-based protocols such as Aloha, Slot-Aloha, CSMA (carrier sense multiple access), CSMA/CD (collision detection), and CSMA/CA (collision avoidance) [16], [17], [18]. In a contention-based protocol, as shown in Fig. 2, when the traffic load and the number of competing stations are very small, the portion of idle time is very large, and the portion of collisions is very small; as the traffic load and the number of competing stations increase, the portion of idle time decreases and the portion of collisions increase; finally, when the traffic load and the number of competing stations are extreme large (infinite) so that every frame is collided, the portion of idle time is almost zero, and the portion of collisions reaches almost one. We have the following assumptions for the above analysis: 1) measurements are conducted during a fixed and relative large time interval, and 2) the frame sizes are the same and fixed. In Fig. 2, we illustrate that the portion of

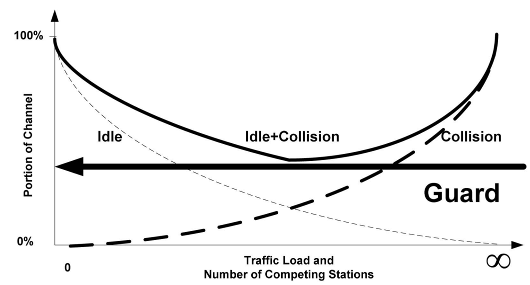

Fig. 2. Idle and collision periods.

idle and collision time is larger than a fixed positive number all the time, which is referred as to a guard period. Therefore, the total bandwidth, i.e., the length of measurement interval, cannot be totally used for bandwidth allocation, and a guard period is adopted in this paper to protect bandwidth allocation from overprovisioning. Therefore, we claim that a guard period is needed and reserved partially for idle and collision time to prevent bandwidth allocation from overprovisioning.

The second function of the guard period is partially for best-effort data traffic, referred as to data traffic hereafter. Data traffic is not shown in Fig. 1 since data traffic is best-effort so that no allocation is for it. In other words, data traffic uses the leftover bandwidth of voice and video traffic. Furthermore, data traffic can also consume some bandwidth in the guard period if we intentionally choose a larger guard period, in which case, the guard period overprotects the system, but the overportion of the guard period can be potentially consumed by data traffic. Therefore, the guard period is partially for data traffic. Data traffic consumes not only bandwidth in the guard period, but also leftover bandwidth of voice and video traffic. When voice and video traffic is very large and consumes almost all the bandwidth for voice and video, data traffic almost cannot obtain any leftover bandwidth from voice and video and, in this situation, the larger guard period is especially useful for data traffic. In other words, a larger guard period can prevent starvation of data traffic if necessary.

3.3 Choice of Guard Period

In this section, we study how to choose the guard period. We can use Bianchi's analytical model [21] or an improved model [18] to approximately estimate the guard period. If we assume that each station has at least a frame ready to send all the time, from [18], [21], a saturation (normalized) throughput $S(N)$ is given as

$$S(N) = \frac{p_s T_{E(L)}}{(1 - p_b)\delta + p_s T_s + [p_b - p_s] T_c}, \tag{1}$$

where p_s denotes the probability that a successful transmission occurs in a slot time, $T_{E(L)}$ denotes the time to transmit the average payload, δ denotes the duration of an empty slot time, p_b denotes the probability that the channel is busy, T_s denotes the average time that the channel is sensed busy because of a successful transmission, T_c denotes the average time that the channel has a collision, and N is the number of competing stations. Please refer to [18], [21] for how to

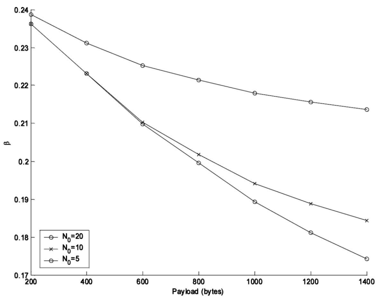

Fig. 3. Approximated β versus payload size.

calculate all these parameters. A guard β portion can be chosen as follows:

$$\beta \approx \min_{N \geq N_0} \left(\frac{(1-p_b)\delta + [p_b - p_s]T_c}{(1-p_b)\delta + p_s T_s + [p_b - p_s]T_c} \right), \qquad (2)$$

where N_0 is the minimum number of competing stations in the system.

Based on (2), we can obtain Fig. 3 for approximated β values over the average payload size (bytes) under the following parameters: $CW_{min} = 32$, $CW_{max} = 1{,}024$, the retry limit is 7, the data rate is 54 Mbps, and only data traffic is considered. Note that the condition that only data traffic is considered is enough for our purpose here to approximate the β value. Fig. 3 shows that β depends on both the average frame size and the minimum number of active stations, N_0, in (2). We observe that, by our approximation in Fig. 3, when $N_0 = 20$, β is around $22\% \sim 24\%$, and when $N_0 = 10$ and $N_0 = 5$, β is around $18\% \sim 23\%$. In our later extensive simulations, we choose $\beta = 20\%$, under which the performance is very good.

However, as we discussed before, a guard period has another function, i.e., as a period reserved partially for best-effort data traffic. Therefore, a larger guard value does not hurt the data traffic and QoS, but fewer voice and video flows can be accepted in the system. But, too small of a guard value is not recommended, as shown in later simulation results. Therefore, how to choose a guard period depends on the system requirements.

4 Bandwidth Sharing Algorithms

The previous section provides basic ideas of sharing schemes. In this section, we propose algorithms to implement these ideas via different admission control algorithms. Any sharing scheme is implemented with two parts: 1) admission control for voice and video flows and 2) data control for data traffic. Admission control algorithms are proposed for different sharing schemes in Sections 4.1, 4.2, 4.3, and 4.4. Data control is the same for all sharing schemes and is proposed in Section 4.5. Finally, in Section 4.6, we introduce a new concept for the guard period: outside guard versus inside guard. For the rest of this paper, we assume that a measurement interval is a beacon interval. However, it can be easily applied to any measurement interval such as several beacon intervals.

4.1 Complete Sharing Scheme

The complete sharing scheme is shown in Fig. 1a. We only discuss the admission control part here, and data control is discussed in a later section. In this section, an AC stands for either voice AC or video AC.

The distributed admission control is developed to protect active QoS flows, i.e., voice and video flows. The QAP announces the transmission budget via beacon frames, and the budget is shared by both voice and video. The budget indicates the allowable transmission time in addition to how much is being utilized. QoS Stations (QSTAs) determine an internal transmission limit per AC for each beacon interval, based on the transmission count during the previous beacon period and the transmission budget announced from the QAP. The local voice/video transmission time per beacon interval shall not exceed the internal transmission limit per AC. When the transmission budget is depleted, new flows will not be able to gain transmission time, while existing flows will not be able to increase the transmission time per beacon interval, which they are already using. This mechanism protects existing flows. Readers are also recommended to read our previously published related work in [20].

4.1.1 *Procedure at QAP*

The QoS Parameter Set Element (QPSE) provides information needed by QSTAs for a proper operation of the QoS facility during a contention period. The QPSE includes $CW_{min}[i]$, $CW_{max}[i]$, and $AIFS[i]$ for $(i = 0, \ldots, 3)$ and $TXOPBudget$ and $SurplusFactor[i]$ for $(i = 1, 2, 3)$. These are global variables in the sense that they are maintained by QAP and transmitted to QSTAs via beacon frames. The first three variables/parameters were already discussed in the previous sections. $TXOPBudget$ specifies the additional amount of time available during the next beacon interval, and $SurplusFactor[i] (> 1)$ represents the ratio of over-the-air bandwidth reserved for AC i to bandwidth of the transported frames required for successful transmission. Note that bandwidth more than the minimum required is typically reserved to compensate for potential transmission failures, e.g., due to collisions. The QPSE is calculated by the QAP for each beacon interval and embedded into the next beacon frame.

The QAP shall measure the amount of time occupied by transmissions from each AC during the beacon period, including associated SIFS and ACK times if applicable. The QAP shall maintain a set of counters $TxTime[i]$, which shall be set to zero immediately following the transmission of a beacon. For each data frame transmission (either uplink or downlink), the QAP shall add the time, equal to the frame transmission time and all overhead involved such as SIFS and ACK, to the $TxTime$ counter corresponding to the AC of that frame. The QAP determines $TXOPBudget$ by

$$TXOPBudget = \max(\alpha_1 T - \sum_{i=1}^{3} TxTime[i] \times SurplusFactor[i], 0), \qquad (3)$$

where $\alpha_1 T$ is defined in Fig. 1a and associated explanations. Note that $TXOPBudget$ in (3) does not have "$[i]$" since it shared with both the video AC and voice AC. How to choose $SurplusFactor$ is well studied in [20] and its journal version in [22].

4.1.2 Procedure at Each QSTA

When the transmission budget is depleted, new QSTAs cannot gain transmission time, while existing QSTAs cannot increase the transmission time per beacon interval, which they are already utilizing. Accordingly, this mechanism protects existing flows.

Each QSTA has to maintain the following local variables for each AC: $TxUsed[i]$, $TxSuccess[i]$, $TxLimit[i]$, $TxRemainder[i]$, and $TxMemory[i]$. These are local variables in the sense that each station locally updates these variables by counting only those related to the station itself. In other words, local variables are those related to a particular station and obtained from the viewpoint of this station, whereas global variables are related to all stations in the common wireless channel and obtained from the viewpoint of the AP. $TxUsed[i]$ counts the amount of time occupied on-air by transmissions, irrespective of success or not, from AC i of this station, including associated SIFS and ACK times if applicable. $TxSuccess[i]$ counts for the transmission time for successful transmissions. A station shall not transmit a data frame if doing so would result in the value in $TxUsed[i]$ exceeding the value in $TxLimit[i]$, where how to determine this value is presented below. If the QSTA is prevented from sending a frame for this reason, it may carry over the partial frame time remainder to the next beacon interval, by storing the remainder in $TxRemainder[i]$, where

$$TxRemainder[i] = TxLimit[i] - TxUsed[i].$$

Otherwise, $TxRemainder[i] = 0$. $TxMemory[i]$ "memorizes" the amount of resource that AC i of this station utilized during a beacon interval. Let f denote the damping factor whose function will be explained below. Let $B[i]$ denote a predefined threshold. Note that $B[i]$ is also referred as to an inside guard period, which will be discussed in a later section. $B[i]$ is also referred as to non-zero-budget in [20] to prevent overprovisioning, and it is essential to provide a stable quality of video and voice. It is similar to the guard period, and it has been well studied in [20] and its journal version in [22]. At each target beacon transmission time, the $TxMemory$, $TxLimit$, and $TxSuccess$ variables are updated according to the following procedure:

- If $TXOPBudget < B[i]$,
  - Both $TxMemory[i]$ and $TxRemainder[i]$ shall be set to zero for new QSTAs which start transmission with this AC in the next beacon interval. All other QSTAs' $TxMemory[i]$ remain unchanged;
- Else
  - For new QSTAs, which start transmission with this AC in the next beacon interval, an initial value for $TxMemory[i]$ is assigned a number between 0 and $TXOPBudget/SurplusFactor[i]$. All other QSTAs' $TxMemory[i]$ are updated according to the following procedure:
    - $TxMemory[i] = f \times TxMemory[i] + (1 - f) \times (TxSuccess[i] \times SurplusFactor[i] + TXOPBudget)$;
    - $TxSuccess[i] = 0$;
    - $TxLimit[i] = TxMemory[i] + TxRemainder[i]$.

Note that, in the above procedure, only $TXOPBudget$ and $SurplusFactor[i]$ are global variables, and the others are local variables. $TXOPBudget$ is shared by both the video AC and voice AC so that there is no "$[i]$." From the above procedure, when the transmission budget for an AC becomes zero:

- Its $TxLimit[i]$ will become zero for new STAs and, hence, AC i of any new QSTA will not be able to gain a transmission time in the next beacon interval.
- The existing QSTAs' $TxMemory[i]$ remain unchanged and, hence, the existing QSTAs' $TxLimit[i]$ remain basically unchanged. In other words, existing stations will not be able to increase the transmission time above what they are already using. Note that this mechanism protects existing flows.

From the above procedure, as long as the transmission budget is larger than zero, both $TxMemory[i]$ and $TxLimit[i]$ need be adjusted periodically. The new $TxMemory[i]$ value is a weighted average of the old $TxMemory[i]$ value and the sum of the successful transmission time and the budget. The value $TxSuccess[i] \times SurplusFactor[i] + TXOPBudget$ is the target to which $TxMemory$ converges. The $TxLimit$ is equal to $TxMemory$ plus a possible capped remainder, where $TxMemory$ "memorizes" the amount of time which a specific AC of the QSTA has been able to utilize per beacon interval. Once the budget is depleted (i.e., $TXOPBudget$ hovers around 0), $TxMemory$ converges to $TxSuccess$, which is the lower limit. This ensures that a QSTA can continue consuming the same amount of time in subsequent beacon intervals. The damping allows for some amount of fluctuation to occur. However, $TxMemory$ cannot grow any further in the saturated state. This prevents new flows from entering a specific AC when it is saturated.

The damping factor does not affect the entrance of a newly entered flow into the system when an enough budget is available because the decreased $TXOPBudget$ is offset by an increased $TxSuccess$ instantaneously, so $TxMemory$ does not change a lot. In other words, for a newly entered flow, $TXOPBudget$ is decreased due to this new entrance, and $TxSuccess$ is increased since it is changed from zero to a positive value so that the sum of these two in the algorithm above does not change a lot. The damping factor does affect $TxMemory$ when a new flow starts up in a QSTA, which does not have an existing flow of the corresponding AC. In such a case, the decreased $TXOPBudget$ is not offset by an increased $TxSuccess$, and the $TxMemory$ converges to the lower target value consequently. QSTAs shall not increase their $TxLimit[i]$ if they did not transmit traffic of AC i during the previous beacon interval.

For each video/voice flow, a Leaky-Bucket algorithm plus a Token-Bucket algorithm can be also implemented at the QSTA to control the flow rate.

4.2 Forward-Voice Reserved Sharing Scheme

The forward-voice reserved sharing scheme, shown in Fig. 1b, is similar to the complete sharing scheme. But, some revisions are needed. Next, we summarize those differences from the complete sharing scheme.

There are two different transmission budgets now: $TXOPBudget\_Voice$ and $TXOPBudget\_Shared$. The QAP announces these two transmission budget via beacon frames, the first budget is used for voice only, and the second budget is shared by both voice and video.

To implement the forward-voice reserved sharing scheme, we need to differentiate voice flows used for the reserved region and for the shared region. A flag bit is needed for frames of voice flows accepted by using the reserved region to distinguish those frames of voice flows accepted by the shared region. The flag bit can be implemented by using one of reserved/unused fields in QoS MAC header. For convenience of presentation/calculation, we rename voice flows accepted by using the reserved region as a different AC, such as $\mathrm{AC} = 4$, and voice flows accepted by using the shared region remain the same AC, i.e., $\mathrm{AC} = 3$.

The QPSE includes $TXOPBudget\_Voice$ and

$$TXOPBudget\_Shared$$

instead of $TXOPBudget$. $TXOPBudget\_Voice$ specifies the additional amount of time available during the next beacon interval for voice flows in the reserved region, and $TXOPBudget\_Shared$ specifies the additional amount of time available during the next beacon interval for voice and video flows in the shared region. The QAP shall measure the amount of time occupied by transmissions from each region (reserved or shared) during the beacon period, including associated SIFS and ACK times if applicable. The QAP shall maintain a set of counters $TxTime[i]$, which shall be set to zero immediately following the transmission of a beacon. For each data frame transmission (either uplink or downlink), the QAP shall add the time, equal to the frame transmission time and all overhead involved such as SIFS and ACK, to the $TxTime$ counter corresponding to the AC of that frame. The QAP determines $TXOPBudget\_Voice$ and $TXOPBudget\_Shared$ by

$$TXOPBudget\_Shared = \max\left(\alpha_1 T - \sum_{i=1}^{3} TxTime[i] \times SurplusFactor[i], 0\right), \quad (4)$$

$$TXOPBudget\_Voice = \max(\alpha_2 T - TxTime[4] \times SurplusFactor[4], 0), \quad (5)$$

where $\alpha_1 T$ and $\alpha_2 T$ are defined in Fig. 1b and associated explanations. Note that, in (5), $\mathrm{AC} = 4$ is just for presentation convenience, and it can be implemented by using a flag bit in a real system.

At each target beacon transmission time, the $TxMemory$, $TxLimit$, and $TxSuccess$ variables for video AC in each QSTA are updated according to the following procedure: $(i = 1, 2)$

- If TXOPBudget_Shared $<$ B$[i]$,
  - Both $TxMemory[i]$ and $TxRemainder[i]$ shall be set to zero for new QSTAs which start transmission in the next beacon interval. All other QSTAs' $TxMemory[i]$ remains unchanged;
- Else
  - For new QSTAs, which start transmission with this AC in the next beacon interval, an initial value for $TxMemory[i]$ is assigned a number between 0 and $TXOPBudget\_Shared/SurplusFactor[i]$. All other QSTAs' $TxMemory[i]$ are updated according to the following procedure:
    - $TxMemory[i] = f \times TxMemory[i] + (1 - f) \times (TxSuccess[i] \times SurplusFactor[i] + TXOPBudget\_Shared)$.
- $TxSuccess[i] = 0$.
- $TxLimit[i] = TxMemory[i] + TxRemainder[i]$;

At each target beacon transmission time, the $TxMemory$, $TxLimit$, and $TxSuccess$ variables for voice AC in each QSTA are updated according to the following procedure: $(i = 3, 4)$

- If $TXOPBudget\_Reserved < B[4]$,
  - If $TXOPBudget\_Shared < B[3]$,
    - Each of $TxMemory[3]$, $TxMemory[4]$, $TxRemainder[3]$, and $TxRemainder[4]$ shall be set to zero for new QSTAs which start transmission in the next beacon interval. In other words, this voice flow is rejected. All other QSTAs' $TxMemory[3]$ and $TxMemory[4]$ remain unchanged;
  - Else, if $TXOPBudget\_Shared \geq B[3]$,
    - For new QSTAs, which start transmission with this AC in the next beacon interval, an initial value for $TxMemory[3]$ is assigned a number between 0 and

      $$TXOPBudget\_Shared/SurplusFactor[3].$$

    - All other QSTAs' $TxMemory[3]$ and $TxMemory[4]$ are updated according to the following procedure:
      - $TxMemory[3] = f \times TxMemory[3] + (1 - f) \times (TxSuccess[3] \times SurplusFactor[3] + TXOPBudget\_Shared)$;
      - $TxMemory[4] = f \times TxMemory[4] + (1 - f) \times (TxSuccess[4] \times SurplusFactor[4] + TXOPBudget\_Reserved)$;
  - $TxSuccess[i] = 0$;
  - $TxLimit[i] = TxMemory[i] + TxRemainder[i]$;
- Else If $TXOPBudget\_Reserved \geq B[4]$,
  - For new QSTAs, which start transmission with this AC in the next beacon interval, an initial value for $TxMemory[4]$ is assigned a number between 0 and

    $$TXOPBudget\_Reserved/SurplusFactor[4].$$

    From now on, this voice flow is accepted in the reserved region and a flag bit is set for each frame from this flow.

- All other QSTAs' $TxMemory[3]$ and $TxMemory[4]$ are updated according to the following procedure:
  - $TxMemory[3] = f \times TxMemory[3] + (1 - f) \times (TxSuccess[3] \times SurplusFactor[3] + TXOPBudget\_Shared)$;
  - $TxMemory[4] = f \text{ x } TxMemory[4] + (1 - f) \times (TxSuccess[4] \text{ x } SurplusFactor[4] + TXOPBudget\_Reserved)$;
- $TxSuccess[i] = 0$;
- $TxLimit[i] = TxMemory[i] + TxRemainder[i]$;

4.3 Backward-Voice Reserved Sharing Scheme

The backward-voice reserved sharing scheme is the same as the forward-voice reserved sharing scheme except that the order of using budgets for voice flows. We explain the difference the forward-voice reserved sharing scheme as follows. At each target beacon transmission time, the $TxMemory$, $TxLimit$, and $TxSuccess$ variables for voice AC in each QSTA are updated according to the following procedure: $(i = 3, 4)$

- If $TXOPBudget\_Reserved < B[3]$,
  - If $TXOPBudget\_Shared < B[4]$,
    - Each of $TxMemory[3]$, $TxMemory[4]$, $TxRemainder[3]$, and $TxRemainder[4]$ shall be set to zero for new QSTAs which start transmission in the next beacon interval. In other words, this voice flow is rejected. All other QSTAs' $TxMemory[3]$ and $TxMemory[4]$ remain unchanged;
  - Else if $TXOPBudget\_Shared \geq B[4]$,
    - For new QSTAs, which start transmission with this AC in the next beacon interval, an initial value for $TxMemory[4]$ is assigned a number between 0 and

      $$TXOPBudget\_Reserved/SurplusFactor[4].$$

      From now on, this voice flow is accepted in the reserved region, and a flag bit is set for each frame from this flow.
    - All other QSTAs' $TxMemory[3]$ and $TxMemory[4]$ are updated according to the following procedure:
      - $TxMemory[3] = f \times TxMemory[3] + (1 - f) \times (TxSuccess[3] \times SurplusFactor[3] + TXOPBudget\_Shared)$;
      - $TxMemory[4] = f \times TxMemory[4] + (1 - f) \times (TxSuccess[4] \times SurplusFactor[4] + TXOPBudget\_Reserved)$;
  - $TxSuccess[i] = 0$;
  - $TxLimit[i] = TxMemory[i] + TxRemainder[i]$;
- Else If $TXOPBudget\_Reserved \geq B[3]$,
  - For new QSTAs, which start transmission with this AC in the next beacon interval, an initial value for $TxMemory[3]$ is assigned a number between 0 and

    $$TXOPBudget\_Shared/SurplusFactor[3].$$

  - All other QSTAs' $TxMemory[3]$ and $TxMemory[4]$ are updated according to the following procedure:
    - $TxMemory[3] = f \text{ x } TxMemory[3] + (1 - f) \times (TxSuccess[3] \times SurplusFactor[3] + TXOPBudget\_Shared)$;
    - $TxMemory[4] = f \times TxMemory[4] + (1 - f) \times (TxSuccess[4] \times SurplusFactor[4] + TXOPBudget\_Reserved)$;
  - $TxSuccess[i] = 0$;
  - $TxLimit[i] = TxMemory[i] + TxRemainder[i]$;

4.4 Other Sharing Schemes

The previous two schemes provide two examples of how to extend the complete sharing schemes to other sharing schemes. Similarly, we can define other seven algorithms shown in Fig. 1. Due to limited space, we do not plan to explain each algorithm in detail. Instead, we provide some guidelines as follows.

4.4.1 Number of Budgets

For the forward-video reserved sharing scheme shown in Fig. 1d and the backward-video reserved sharing scheme shown in Fig. 1e, we need to define two different budgets for the shared region and the reserved region similar to the forward-voice reserved sharing scheme. An $\alpha_1 T$ portion of the total bandwidth is reserved for video flows only, an $\alpha_2 T$ portion of the total bandwidth is for complete sharing among voice and video flows, and a βT portion of the total bandwidth is for the guard period, where $\beta + \alpha_1 + \alpha_2 = 1$. Equations (4) and (5) need to be redefined accordingly and similarly.

For the forward-reserved sharing scheme shown in Fig. 1f, the backward-reserved sharing scheme shown in Fig. 1g, the backward-voice forward-video reserved sharing scheme shown in Fig. 1h, and the forward-voice backward-video reserved sharing scheme shown in Fig. 1i, we need to define three different budgets for the shared region and the reserved regions similar to the forward-voice reserved sharing scheme. An $\alpha_1 T$ portion of the total bandwidth is reserved for video flows only, an $\alpha_2 T$ portion of the total bandwidth is for complete sharing among voice and video flows, an $\alpha_3 T$ portion of the total bandwidth is reserved for voice flows only, and a βT portion of the total bandwidth is for the guard period, where $\beta + \alpha_1 + \alpha_2 + \alpha_3 = 1$. Equations (4) and(5) need to be redefined into three equations accordingly and similarly. Note that parameters, such as α, β, etc., can be tuned via experiments by service providers. Since traffic patterns of voice, video, and data traffic may change based on the users' requirements, they may be adjusted by service providers. How to optimize these parameters is a difficult task and will be our future work.

4.4.2 Difference of Voice or Video in Different Regions

A flag bit is needed to distinguish frames of video flows accepted by the reserved region from frames of video flows accepted by the shared region for the forward-video reserved sharing scheme shown in Fig. 1d and the backward-video reserved sharing scheme shown in Fig. 1e.

Two flag bits are needed for the forward reserved sharing scheme shown in Fig. 1f, the backward reserved sharing scheme shown in Fig. 1g, the backward-voice forward-video reserved sharing scheme shown in Fig. 1h, and the forward-voice backward-video reserved sharing scheme shown in Fig. 1i. A flag bit is needed to distinguish frames of video flows accepted by the reserved region from frames of video flows accepted by the shared region, and another flag bit is needed to distinguish frames of voice flows accepted by the reserved region from frames of voice flows accepted by the shared region.

4.4.3 Forward versus Backward

Similar to the forward/backward-voice reserved sharing scheme proposed in the previous sections, in a forward scheme, budget is used first in the reserved region if having any, and then in the shared region; in a backward scheme, budget is used first in the shared region, and then in the reserved region if having any. Admission control algorithms are changed accordingly and similarly.

4.4.4 Local Calculations

At each QSTA, the calculations in algorithms are adjusted accordingly by considering at least three aspects: 1) number of different budgets, 2) forward/backward, and 3) different regions.

4.5 Data Control

Since too many data transmissions can degrade the performance of existing voice and video flows, we propose a *retry-based* data control mechanism in this paper to dynamically adjust data traffic parameters based on traffic conditions. In the proposed approach, stations dynamically adjust the EDCA data parameters based on the behavior of one or more frame transmission(s). During each frame transmission, whenever the number of retries ever reaches a threshold K, the next frame's initial window size is increased by $CW_{\min}[0] = \theta \times CW_{\min}[0]$; whenever there are L consecutive successful transmissions, the next frame's initial window size is decreased by $CW_{\min}[0] = CW_{\min}[0]/\theta$. Note that the above changes should be within the data EDCA parameter's range, i.e., $CW_{\min}[0] \geq CW_{\min}[1]$ holds all the time and, otherwise, no change should be made.

4.6 Outside Guard versus Inside Guard

We explained the guard period in the previous sections. We further refer a guard period shown in Fig. 1 to an outside guard. We define an inside guard period as B[i], which is used in Sections 5.1, 5.2, 5.3, and 5.4 as an admission control threshold. Fig. 4 illustrates the concept of inside guard and outside guard. An outside guard occupies a separation portion of the total bandwidth, and an inside guard period is a budget threshold for a shared/reserved region beyond which new voice/video flows are rejected. If an inside guard is not used, then we have $B[\mathrm{i}] = 0$. Intuitively, an outside guard period and an inside guard have similar functions, but they are not exactly the same. They can be used at the same time. We will further evaluate them in the next section.

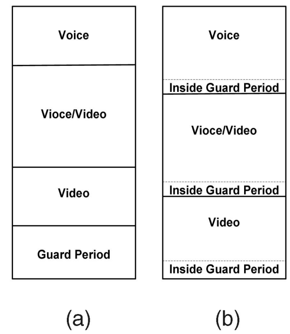

Fig. 4. (a) Outside guard (b) Inside guard.

5 Performance Evaluation

In this section, we conduct a performance evaluation for the proposed sharing schemes via extensive simulations. We adopted IEEE 802.11a [1] and IEEE 802.11e draft [2] in these simulations. Three traffic types are considered in our simulations: voice (AC 3), video (AC 2), and data (AC 0). Section 5.1 will introduce performance metrics and simulation setup. In Section 5.2, we compare a complete sharing scheme with our Partition scheme proposed in [20]. In Section 5.3, we compare the complete sharing scheme with the forward-voice reserved sharing scheme. In Section 5.4, we compare the forward-voice reserved sharing scheme with the backward-voice reserved sharing scheme. In Section 5.5, we compare the complete sharing scheme with the forward-video reserved sharing scheme. In Section 5.6, we study the effects of the guard period and compare outside guard period with inside guard period in the complete sharing scheme.

5.1 Performance Metrics and Simulation Setup

We adopt the following performance metrics in our simulations:

1. *average throughput* per voice flow, video flow, or data station,
2. *total throughput*,
3. *Txlimit*,
4. *TxBudget*,
5. *number for accepted and active flows (NAAF)* in the system per AC, and
6. *throughput square relative difference (SRD)*.

Throughput SRD is proposed to characterize the normalized difference of achieved throughput and required throughput. Let $k(t)$ and $T_i(t)$ denote the number of flows in an AC $(\mathrm{AC} > 0)$ and the average throughput, respectively, at the tth measurement interval. Let T_i denote the required throughput for flow $i (i = 1, 2, \ldots, k(t))$. Throughput SRD at the tth measurement interval for this AC is defined

$$SRD_T(t) = \sum_{i=1}^{k(t)} \left((T_i(t) - T_i)/T_i\right)^2. \tag{6}$$

We assume that the transmitted traffic is not larger than required throughput on average; otherwise, a token bucket algorithm can be also implemented to control the traffic rate. Throughput SRD can be only applied to voice and video traffic, but not data traffic.

The defaults EDCA access parameters used for our simulations are listed as follows:

$$CW_{min}[3] = 16, \quad CW_{max}[3] = 256, \qquad AIFS[3] = 25\mu s,$$
$$CW_{min}[2] = 32, \quad CW_{max}[2] = 2{,}048, \quad AIFS[2] = 25\mu s,$$
$$CW_{min}[0] = 256, \; CW_{max}[0] = 51{,}200, \; AIFS[0] = 34\mu s,$$

and queue size is 30 frames for each AC (voice, video, and data). For other parameters, the following values are adopted unless stated otherwise: the beacon interval is 100 ms, the damping factor is 0.9, and each voice flow is 0.0832 Mbps, which is generated by a constant interarrival time 20 ms with a fixed payload size of 208 bytes, corresponding to G.711-coded VoIP over RTP/UDP/IP/SNAP [19]. Each video flow is 4.68 Mbps, which is generated by a constant interarrival time 2.5 ms with a mean payload size of 1464 bytes. It corresponds to a traffic-shaped CBR video flow. Each station generates data frames with an exponential distribution with a mean interarrival time 12 ms and a fixed payload size of 1,500 bytes. IEEE 802.11a is adopted and parameters are listed as follows: the data rate is 54 Mbps, the control rate is 24 Mbps, the retry limit is 7, the SIFS time is 16 μs, the Slot time is 9 μs, the physical layer's preamble is 16 μs, the physical header time is 4 μs, and a symbol time is 4 μs. We assume that all the stations are within the transmission range. Data control parameter $\theta = 1.3$. An inside voice guard is *4 ms* and an inside video guard is *20 ms* unless stated otherwise.

5.2 Complete Sharing versus Partition

In this section, we compare the complete sharing (CS) scheme with our Partition scheme proposed in [20]. For the CS scheme, we have $\alpha_1 = 0.8$ and $\beta = 0.2$. For the Partition scheme [20], voice traffic, video traffic, and guard (data) have separate partitions, and we have $\alpha_2 = 0.2$ (voice), $\alpha_1 = 0.6$ (video), and $\beta = 0.2$ (data or guard). We conduct two case studies. The simulation time is 300 seconds. The simulation results are summarized in Table 1. As indicated in the table, for both case studies and both schemes, QoS are well achieved. For both case studies, the CS scheme accepts more voice/video flows than the Partition scheme [20] with guaranteed QoS requirement. In other words, the proposed CS scheme outperforms than our Partition scheme in [20].

5.3 Complete Sharing versus Forward-Voice Reserved Sharing

In this section, we compare the complete sharing (CS) scheme with Forward-VOice Reserved Sharing (FVORS) scheme. Traffic arrivals are listed as follows: At the beginning of the simulation, for every 5 seconds, there is one video flow arrival until the number of arrived video flows reaches 5; after then, for every 5 seconds, there is one voice flow arrival until the number of arrived voice flows reaches 25. At the beginning of the simulation, for every 5 seconds, there is one data station arrival until the number of arrived data stations reaches 10. Note that some video/voice flows are accepted, but others are rejected. Voice and data traffic stays in the system throughout the simulation. The lifetime of a video flow is 150 seconds. For the CS scheme, we have $\alpha_1 = 0.8$ and $\beta = 0.2$. For the FVORS scheme, we have $\alpha_2 = 0.2$, $\alpha_1 = 0.6$, and $\beta = 0.2$. The simulation time is 200 seconds.

TABLE 1
Comparison

Case studies	Case study 1		Case study 2	
Traffic Generated	50 voice flows arrive in 5s interval; 10 data stations arrive in 5s interval;		10 video flows arrive in 5s interval; 10 data stations arrive in 5s interval;	
Schemes	CS	Partition	CS	Partition
Accepted	50 voice flows; 10 data stations	20 voice flows; 10 data stations	5 video flows; 10 data station	4 video flows; 10 data stations
Average voice/video throughput	Good	Good	Good	Good
Throughput SRD	<0.04	<0.08	<0.02	<0.08

Fig. 5 shows the number of accepted and active flows (NAAF) for voice, video, and data, where Figs. 5a, 5b, and 5c are for the CS scheme, and Figs. 5d, 5e, and 5f are for the FVORS scheme. As illustrated in figure, the CS scheme totally accepts five video flows, 12 voice flows, and 10 data stations, and the FVORS scheme totally accepts four video flows, 21 voice flows, and 10 data stations. Although the two schemes have the same traffic pattern, five video flows are accepted for the CS scheme, whereas four video flows are accepted in the FVORS scheme. This indicates that the CS scheme is more efficient to accommodate more video traffic. However, 21 voice flows are accepted in the FVORS scheme, whereas only 12 voice flows are accepted in the CS scheme. One reason is that the CS scheme accepts one more video flow so that there is not much budget left. Another reason is that the FVORS scheme reserves bandwidth for the voice traffic so that more voice traffic can be accommodated

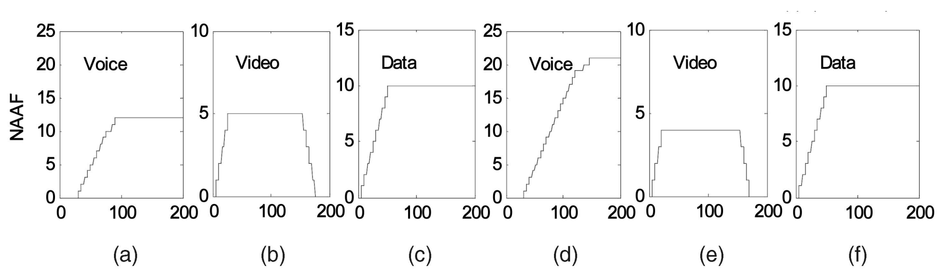

Fig. 5. (a), (b), and (c) NAAF versus time (s) (CS). (d), (e), and (f) NAAF versus time (s) (FVORS).

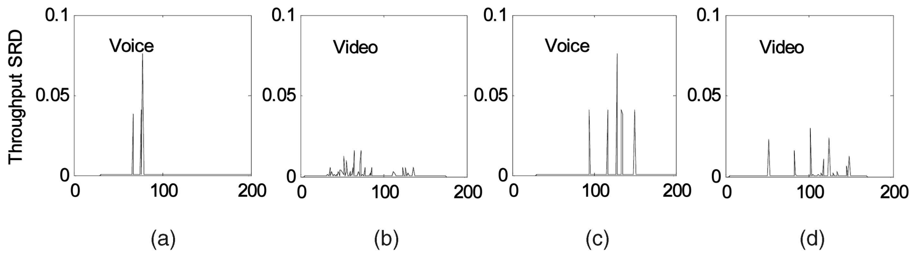

Fig. 6. (a) and (b) Throughput SRD versus time (s) (CS). (c) and (d) Throughput SRD versus time (s) (FVORS).

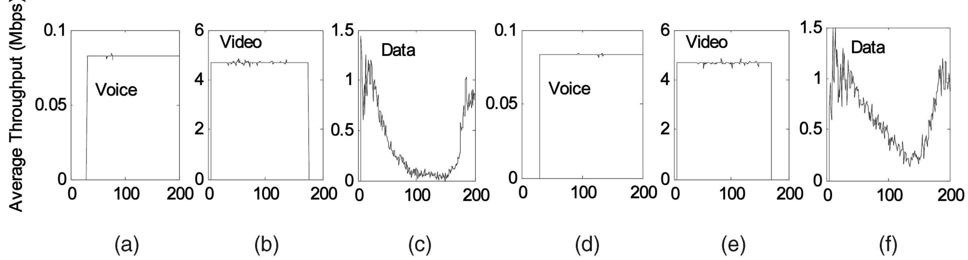

Fig. 7. (a), (b), and (c) Average throughput (Mbps) per flow versus time (s) (CS). (d), (e), and (f) Average throughput (Mbps) per flow versus time (s) (FVORS).

in the FVORS scheme. Therefore, if the system has the requirement that some number of voice flows should be reserved so that guaranteed voice traffic are available if needed, the FVORS scheme should be adopted. Otherwise, the CS scheme should be adopted.

Fig. 6 shows throughput SRDs for voice and video, where Figs. 6a and 6b are for the CS scheme and Figs. 6c and 6d are for the FVORS scheme. We observe that, for both schemes, the throughput SRDs are small, good, and bounded by 0.1, which is a predefined QoS requirement in this simulation. In other words, QoS is guaranteed.

Fig. 7 shows average throughput (Mbps) per flow, where Figs. 7a, 7b, and 7c are for the CS scheme, and Figs. 7d and 7f are for the FVORS scheme. We observe that, for both schemes, voice and video flows have very stable average throughputs, although traffic is changed and more flows are accepted with the time. A voice flow maintains 0.0832 Mbps throughput pretty well, and a video flow maintains 4.68 Mbps throughput pretty well. We also observe that, when more video flows and voice flows are accepted in the system, the throughput per data station decreases, and when video flows leave, the throughput per station increases. These are caused by the data control mechanism in which, when voice and video traffic is heavy, data parameters are controlled by the data control mechanism automatically. Note that data throughput is not guaranteed since it is best effort traffic. We also observe that data traffic is degraded more for the CS scheme than that for the FVOSR scheme since there is one more video accepted in the system for the CS scheme.

Fig. 8 shows total throughput, where Fig. 8a is for the CS scheme and Fig. 8b is for the FVORS scheme. As illustrated in the figure, for both schemes, the total throughput increases as more video flows are accepted in the system and decreases as video flows leave. The CS scheme has a little better throughput for the most of the simulation time since it accepts one more video flow. At the nearly end of the simulation, the FVORS scheme has a little better throughput than the CS scheme since it has more voice flows in the system.

Fig. 9 shows Budget, i.e., $TXOPBudget$, where Fig. 9a is for the CS scheme and Figs. 9b and 9c are for the FVORS scheme. Fig. 9a shows that, at the beginning of the simulation for the CS scheme, the total budget is 80 ms. As video flows and voice flows are accepted in the system, the budget becomes less. As the budget becomes very less, the newly arrived video/voice flows are all rejected by the admission control. As video flows leave, the budget comes back around to 70 ms but not to 80 ms since there are still 12 voice flows in the system. Fig. 9c shows that, at the beginning of the simulation for the FVORS scheme, the budget for the shared region is 60 ms. As voice and video

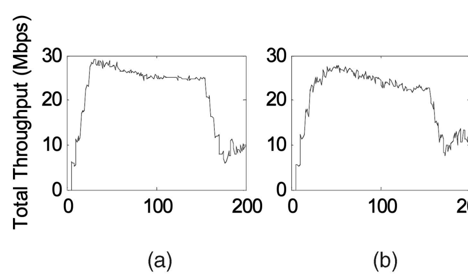

Fig. 8. (a) Total throughput versus time (s) (CS). (b) Total throughput versus time (s) (FVORS).

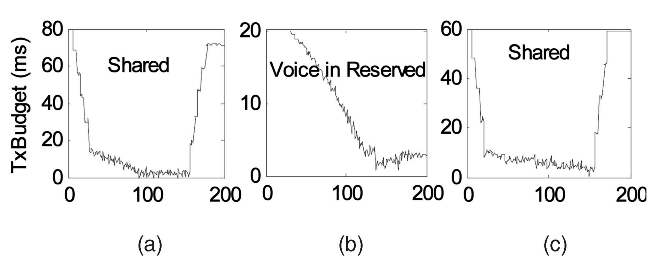

Fig. 9. (a) Budget versus time (s) (CS). (b) and (c) Budget versus time (s) (FVORS).

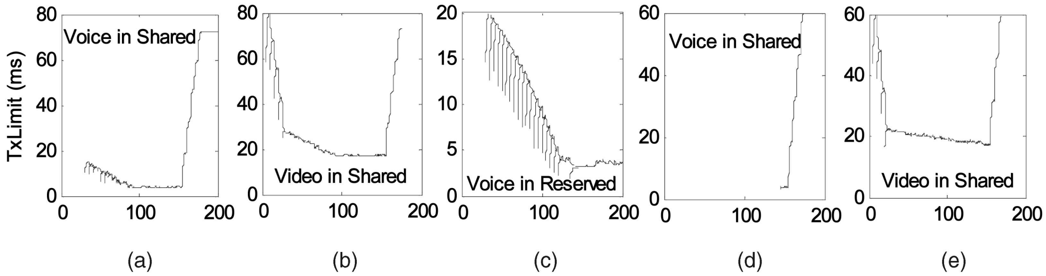

Fig. 10. (a) and (b) *Txlimit* versus time (s) (CS). (c), (d), and (e) *Txlimit* versus time (s) (FVORS).

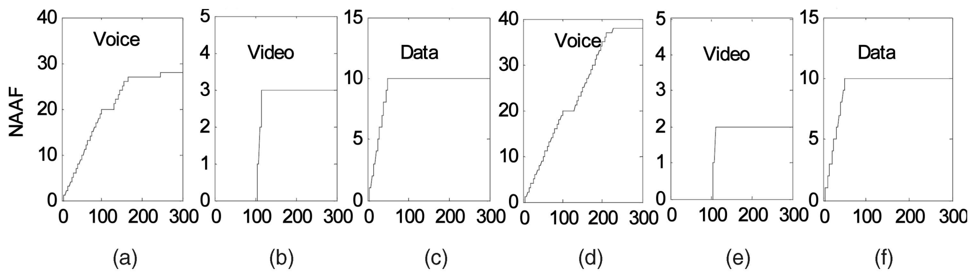

Fig. 11. (a), (b), and (c) NAAF versus time (s) (FVORS). (d), (e), and (f) versus time (s) (BVORS).

flows arrive, the budget in the shared region decreases, and as the budget in the shared region becomes low enough, newly arrived video flows are rejected. Fig. 9b shows that, at the beginning of the simulation for the FVORS scheme, the budget for the reserved region has 20 ms. The budget for the reserved region is used when the first voice flow arrives since the scheme is a "Forward" scheme in the sense that the reserved region is used first. The budget decreases when more voice flows are accepted in the system. As the budget in the reserved region becomes very less and at this moment the budget in the shared region is very less too, new voice flows are all rejected.

Fig. 10 shows *Txlimit*, where Figs. 10a and 10b are for the CS scheme, and Figs. 10c, 10d, and 10e are for the FVORS scheme. Fig. 10b shows that video *Txlimit* for the CS scheme is 80 ms at the beginning of the simulation in the shared region and drops five times in big steps since five video flows are accepted in the system; then, video *Txlimit* further decreases since more voice flows are accepted and both voice and video share the same region. As the video flows leave, video *Txlimit* increases back again, but is still less than 80 ms since there are still some voice flows in the system. Fig. 10a shows that voice *Txlimit* for the CS scheme drops many times in small steps in the shared since many voice flows are accepted in the system; as the video flows leave, video *Txlimit* increases back again. Fig. 10e shows that video *Txlimit* for the FVORS scheme is 60 ms in the shared region at the beginning of the simulation and drops four times in big steps since four video flows are accepted in the system. As the video flows leave, video *Txlimit* increases back to 60 ms again since there are no voice flows in the shared region. Fig. 10c shows voice *Txlimit* for the CS scheme drops many times in small steps in the reserved region since many voice flows are accepted in the system. As the video flows leave, video *Txlimit* does not increase since the departed video is in the different region. Fig. 10 also shows the effects of convergence of *TxLimit*: After several seconds of acceptance, *TxLimits* of different video/voice flows converge to almost the same value fairly.

5.4 Forward-Voice Reserved Sharing versus Backward-Voice Reserved Sharing

In this section, we compare the Forward-VOice Reserved Sharing (FVORS) scheme with the Backward-VOice Reserved Sharing (BVORS) scheme. Traffic arrivals are listed as follows: At the beginning of the simulation, for every 5 seconds, there is one voice flow arrival until the number arrived voice flows reaches 20. After then, for every 5 seconds, there is one video flow arrival until the number of arrived video flows reaches 5. After then, for every 5 seconds, there is one voice flow arrival until the number of arrived voice flows reaches total 45. At the beginning of the simulation, for every 5 seconds, there is one data station arrival until the number of arrived data stations reaches 10. Note that some video/voice flows are accepted, but others are rejected. Voice, video, and data traffic stays in the system throughout the simulation. For both the FVORS scheme and the BVORS scheme, we have $\alpha_2 = 0.2$, $\alpha_1 = 0.6$, and $\beta = 0.2$. The simulation time is 300 seconds.

Fig. 11 shows the number of accepted and active flows (NAAF) for voice, video, and data, where Figs. 11a, 11b, and 11c are for the FVORS scheme, and Figs. 11d and 11f are for the BVORS scheme. As illustrated in figure, the FVORS scheme totally accepts 20 voice flows, three video flows, an additional seven voice flows, and 10 data stations, and the BVORS scheme totally accepts 20 voice flows, two video flows, an additional 18 voice flows, and 10 data stations. Although the two schemes have the same traffic pattern, three video flows are accepted for the FVORS scheme, whereas two video flows are accepted in the BVORS scheme. This indicates that the FVORS scheme is more efficient to accommodate more video traffic. However, 38 voice flows are accepted in the BVORS scheme, whereas only 27 voice flows are accepted in the FVORS scheme. One reason is that the FVORS scheme accepts one more video flow so that there is not much budget left. Another reason is that the BVORS scheme uses budget in the shared region first, and then reserved region so that more voice traffic can

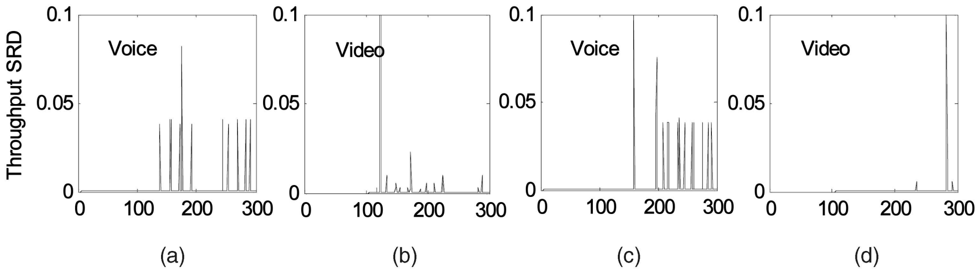

Fig. 12. (a) and (b) Throughput SRD versus time (s) (FVORS). (c) and (d) Throughput SRD versus time (s) (BVORS).

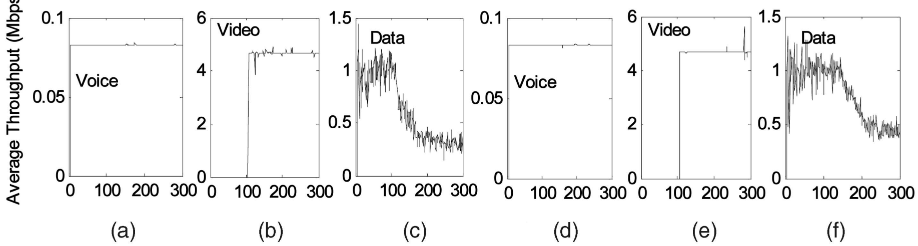

Fig. 13. (a), (b), and (c) Average throughput (Mbps) per flow versus time (s) (FVORS). (d), (e), and (f) Average throughput (Mbps) per flow versus time (s) (BVORS).

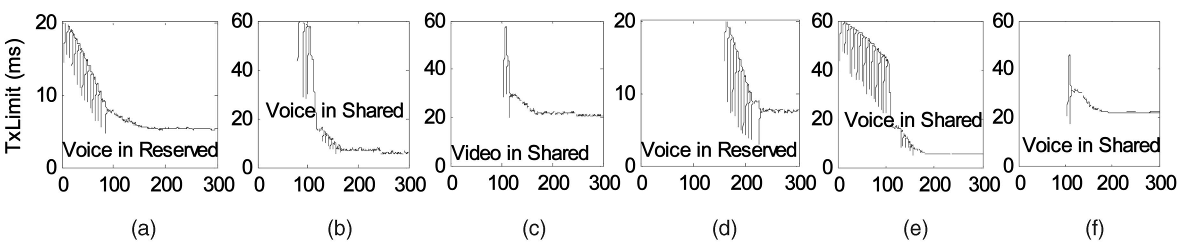

Fig. 14. (a), (b), and (c) *Txlimit* versus time (s) (FVORS). (d), (e), and (f) *Txlimit* versus time (s) (BVORS).

be accommodated in the BVORS scheme. In other words, the BVORS scheme is more "selfish" on voice budget.

Fig. 12 shows throughput SRDs for voice and video, where Figs. 12a and 12b are for the FVORS scheme, and Figs. 12c and 12d are for the BVORS scheme. We observe that, for both schemes, the throughput SRDs are small, good, and bounded by 0.1, which is a predefined QoS requirement in this simulation. In other words, QoS is guaranteed.

Fig. 13 shows average throughput (Mbps) per flow, where Figs. 13a, 13b, and 13c are for the FVORS scheme, and Figs. 13d and 13f are for the BVORS scheme. We observe that, for both schemes, voice and video flows have very stable average throughputs, although traffic is changed and more flows are accepted with the time. We also observe that, when more video flows and voice flows are accepted in the system, the throughput per data station decreases. These are caused by the data control mechanism. We also observe that data traffic is degraded more for the FVOSR scheme than for the BVOSR scheme since there is one more video accepted in the system for the FVOSR scheme. Although, for the BVOSR scheme, there are 11 more accepted voice flows, a video flow needs much more bandwidth than 11 voice flows.

Fig. 14 (Fig. 15) shows *Txlimit*(Budget), where Figs. 14a, 14b, and 14c are for the FVORS scheme, and Figs. 14d and 14f are for the BVORS scheme. According to the traffic pattern, the FVORS scheme first uses the budget in the reserved region for accepted voice flows (Fig. 14a or Fig. 15a). Second, it uses the budget in the shared region for accepted voice flows (Fig. 14b or Fig. 15b) when the budget in the reserved region is smaller enough. Third, it uses the budget in the shared region for accepted video flows (Fig. 14c or Fig. 15b). Finally, it uses the budget in the shared region for accepted voice flows (Fig. 14b or Fig. 15b). On the other hand, according to the traffic pattern, the BVORS scheme first uses the budget in the shared region for accepted voice flows (Fig. 14e or Fig. 15d). Second, it uses the budget in the shared region for accepted video flows (Fig. 14f or Fig. 15d). Third, it uses the budget in the shared region for accepted voice flows (Fig. 14e or Fig. 15c). Finally, it uses the budget in the reserved region for accepted voice flows (Fig. 14d or Fig. 15b) when the budget in the shared region is smaller enough. The above sequences exactly show the difference between "forward" and "backward."

5.5 Complete Sharing versus Forward-Video Reserved Sharing

In this section, we compare the complete sharing (CS) scheme with the Forward VIdeo Reserved Sharing (FVIRS) scheme. Traffic arrivals are listed as follows: At the beginning of the simulation, for every 5 seconds, there is one voice flow arrival until the number of arrived voice flows reaches 50; after then, for every 5 seconds, there is one video flow arrival until the number of arrived video flows reaches 5. At the beginning of the simulation, for every

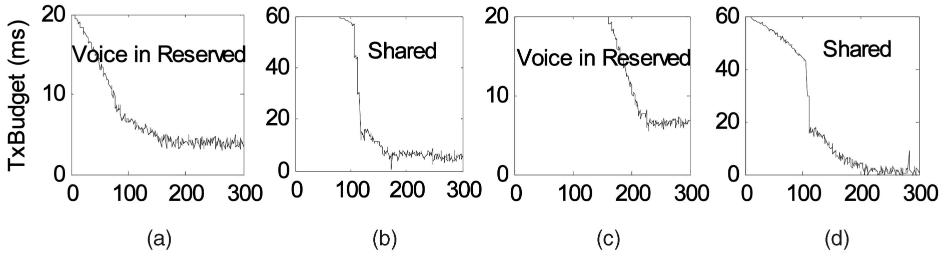

Fig. 15. (a) and (b) Budget versus time (s) (FVORS). (c) and (d) Budget versus time (s) (BVORS).

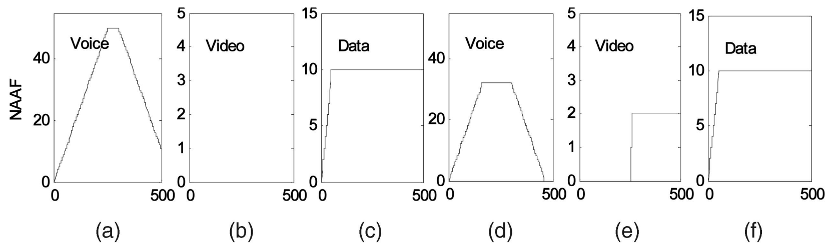

Fig. 16. (a), (b), and (c) NAAF versus time (s) (CS). (d), (e), and (f) NAAF versus time (s) (FVIRS).

5 seconds, there is one data station arrival until the number of arrived data stations reaches 10. Video and data traffic stays in the system throughout the simulation. The lifetime of a voice flow is 300 seconds. For the CS scheme, we have $\alpha_1 = 0.8$ and $\beta = 0.2$. For the FVIRS scheme, we have $\alpha_2 = 0.4$, $\alpha_1 = 0.4$, and $\beta = 0.2$. The simulation time is 500 seconds.

Fig. 16 shows the number of accepted and active flows (NAAF) for voice, video, and data, where Figs. 16a, 16b, and 16c are for the CS scheme and Figs. 16d and 16f are for the FVIRS scheme. As illustrated in figure, the CS scheme totally accepts 50 voice flows, zero video flow, and 10 data stations, and the FVIRS scheme totally accepts 32 voice flows, two video flows, and 10 data stations. Although the two schemes have the same traffic pattern, no video flow is accepted for the CS scheme, whereas two video flows are accepted in the FVIRS scheme. However, the CS scheme accepts 50 voice flows, whereas the FVIRS scheme accepts only 32 voice flows. Since the voice flows come first, the CS scheme accepts all the voice flows, and then has no budget for any new video. On the other hand, the FVIRS scheme reserves some bandwidth for the video flows so that two video flows are accepted when they arrive, but the bandwidth for voice flows is limited so that only 32 voice flows are accepted, and other voice flows are rejected. Therefore, if the system has the requirement that some number of video flows should be reserved so that guaranteed video traffic are available if needed, the FVIRS scheme should be adopted.

Fig. 17 shows Budget, where Fig. 17a is for the CS scheme and Figs. 17b and 17c are for the FVIRS scheme. Fig. 17a shows that, for the CS scheme, as more voice flows are accepted in the system, the budget decreases, and as more voice flows leave, the budget increases again. Fig. 17b shows that, for the FVIRS scheme, the budget for video in the reserved region decreases with two big steps when two video are accepted and, at a later time, when voice flows leave, the budget for video in the reserved increases a little. Note that all other video flows are rejected. Fig. 17c shows that, for the FVIRS scheme, the budget for voice in the shared region decreases as more voice flows are accepted and increases as more voice flows leave.

Fig. 18 shows *Txlimit*, where Fig. 18a is for the CS scheme and Figs. 18b and 18c are for the FVIRS scheme. According to the traffic pattern, the CS scheme accepts 50 voice flows, as shown in Fig. 18a, and the FVIRS scheme accepts 32 voice flows, as shown in Fig. 18c, and two video flows, as shown in Fig. 18b.

5.6 Outside versus Inside Guard Period in CS

In this section, we compare the outside guard with the inside guard and study the effects of the guard period. As we discussed earlier, a guard period has two functions: 1) as a guard threshold reserved partially for collision and idle time and 2) as a period reserved partially for best-effort data. The inside guard has one additional function as the admission control threshold. Traffic arrivals are listed as

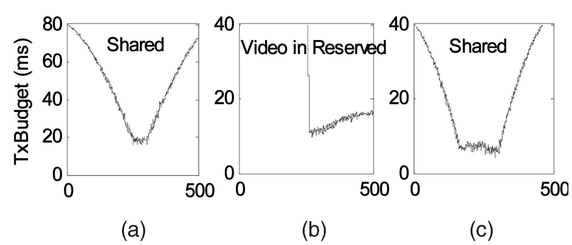

Fig. 17. (a) Budget versus time (s) (CS). (b) and (c) Budget versus time (s) (FVIRS).

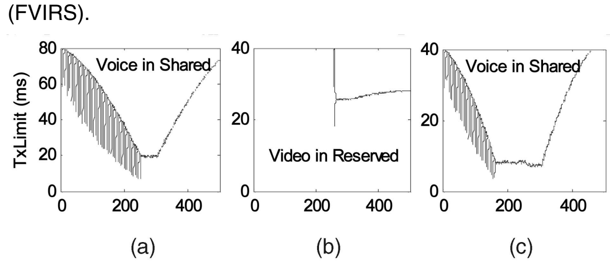

Fig. 18. (a) *Txlimit* versus time (s) (CS). (b) and (c) *Txlimit* versus time (s) (FVIRS).

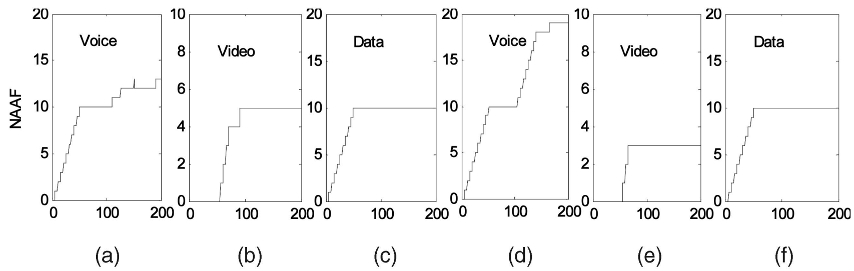

Fig. 19. (a), (b), and (c) NAAF versus time (s), $guard = 20$ ms. (d), (e), and (f) NAAF versus time (s), $guard = 40$ ms.

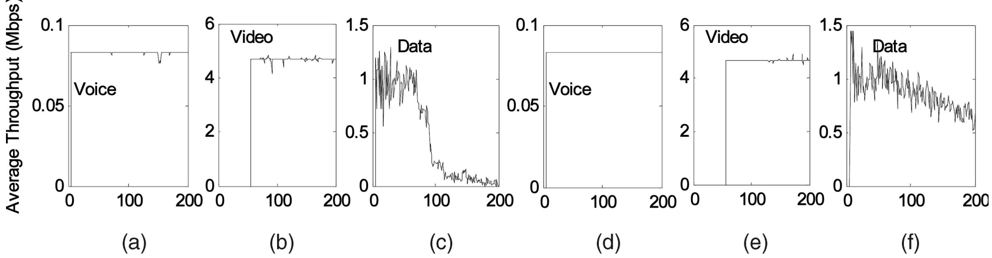

Fig. 20. Average throughput (Mbps) per flow versus time (s). (a), (b), and (c) $guard = 20$ ms. (d), (e), and (f) $guard = 40$ ms.

follows; At the beginning of the simulation, for every 5 seconds, there is one voice flow arrival until the number of arrived voice flows reaches 10. After then, for every 5 seconds, there is one video flow arrival until the number of arrived video flows reaches 10. After then, for every 5 seconds, there is one voice flow arrival until the total number of arrived voice flows reaches 30. At the beginning of the simulation, for every 5 seconds, there is one data station arrival until the number of arrived data stations reaches 10.

5.6.1 *Effects of Outside Guard*

In this section, we study the effects of the outside guard for the complete sharing scheme. We study two scenarios: 1) $guard(outside) = 20$ ms and 2) $guard = 40$ ms.

Fig. 19 shows the number of accepted and active flows (NAAF) for voice, video, and data, where Figs. 19a, 19b, and 19c are for $guard = 20$ ms and Figs. 19d, 19e, and 19f are for $= 40$ ms. As illustrated in figure, when $guard = 20$ ms, the scheme totally accepts 10 voice flows, five video flows, an additional two voice flows, and 10 data stations; when $\mathrm{guard} = 40$ ms, the scheme totally accepts 10 voice flows, three video flows, an additional eight voice flows, and 10 data stations. In other words, as the guard increases, the bandwidth for voice and video decreases. The number of accepted voice flows increases when the guard period is *40 ms* since fewer video flows are accepted due to the larger guard period and, on the other hand, more voice flows can be accepted due to the decreased number of accepted video flows.

Fig. 20 shows the average throughput (Mbps) per flow, where Figs. 20a, 20b, and 20c are for $guard = 20$ ms and Figs. 20d, 20e, and 20f are for $guard = 40$ ms. We observe that, for both schemes, voice and video flows have very stable average throughputs although traffic is changed and more flows are accepted with the time. We also observe that, when the guard period is larger, the best-effort data's average throughput per station is larger (Fig. 20c versus Fig. 20f). Since there are total data throughputs are about 0.5 Mbps and 6.5 Mbps at the end of the simulation for $guard = 20$ ms and $guard = 40$ ms, respectively.

Furthermore, both cases have good throughput SRD (the figure is omitted).

5.6.2 *Effects of Inside Guard Period*

We study the effects of inside guard period without an outside guard. We have two scenarios: 1) no inside guard and 2) a voice inside guard is *40 ms* and a video inside guard is *60 ms*. In scenario 1, the scheme totally accepts 10 voice flows, 10 video flows, an additional 19 voice flows, and 10 data stations; in scenario 2, the scheme totally accepts 10 voice flows, three video flows, an additional 12 voice flows, and 10 data stations. However, as shown in Fig. 21, scenario 1 has an unacceptably bad average throughput for video since too many videos are accepted. Furthermore, as shown in Fig. 22, throughput SRDs are very bad for both voice and video in scenario 1. On the other hand, scenario 2 has very a good average throughput for both video and voice, as shown in Fig. 21, and a very good throughput SRD for both voice and video, as shown Fig. 22. From Fig. 21 and Fig. 22, we also observe that the data throughput per station is better than inside guards are large and becomes very small when there are no inside guards under heavy video and voice traffic.

In summary, either the inside guard period or the outside guard period is necessary in order to guarantee QoS. However, several inside guards without an outside guard period may not be very efficient. Based on our many simulations and experiences, our recommendation is to use both an inside guard and an outside guard, and inside guards should be chosen as relatively small values, where an outside guard should be chosen as a relatively large value.

An adaptive mechanism to adjust channel access parameters based on traffic load or the number of contending stations can be easily incorporated with the proposed admission control algorithms to use channel more efficiently, i.e., accommodating more voice or video flow via adaptively reducing guard period. This is our future work.

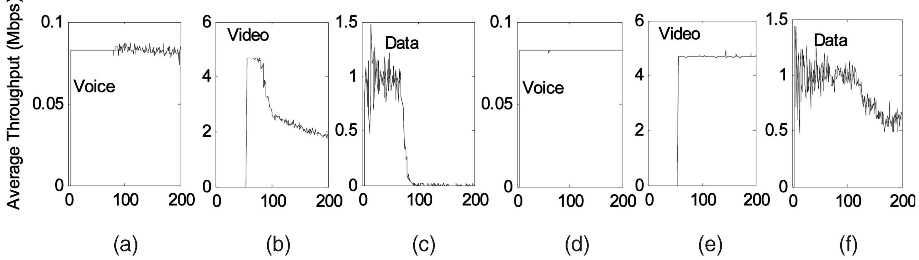

Fig. 21. Average throughput (Mbps) per flow without an outside guard versus time (s). (a), (b), and (c) *No inside guard*. (d), (e), and (f) A voice inside $guard = 40$ ms and a video inside $guard = 60$ ms.

6 Conclusion and Future Work

In this paper, we addressed very challenging issues, i.e., how to guarantee QoS and conduct efficient bandwidth allocations in the IEEE 802.11e contention-based distributed WLANs. We proposed nine novel sharing schemes for bandwidth allocation that all can provide guaranteed QoS with admission controls for voice and video and a data control mechanism. The complete sharing scheme is the best in terms of utilization among the nine sharing schemes; however, with a reserved region for an AC, some minimum traffic for this AC can be guaranteed and more traffic for the reserved AC may be accepted in the system depending on the need for this AC. A forward scheme is more efficient than a backward scheme since the backward scheme is a selfish and greedy scheme for the reserved AC; however, in a backward scheme, more traffic for the reserved AC may be accepted in the system. The complete sharing scheme is shown to be better than our Partition scheme proposed in [20].

Furthermore, a (outside) guard period concept is proposed to prevent bandwidth allocation from over provisioning and is for data traffic. Our study shows that the guard period is one key aspect for QoS guarantees. We further propose an inside guard period concept. Either aninside guard period or an outside guard period is necessary in order to guarantee QoS. However, several inside guards without an outside guard period may be not very efficient. Based on our many simulations and experiences, our recommendation is to use both inside guard and outside guard, and inside guards should be chosen as relatively small values, where an outside guard should be chosen a relatively large value. A larger guard period can allow more data traffic in the system.

Note that our work does not mean to follow the exact changing of the IEEE 802.11e standard, but provide different approaches and solutions. Our future work will look into more details of the final version of the published IEEE 802.11e standard. Our future work also include proposing adaptive mechanisms to adjust channel access parameters based on traffic load or the number of contending stations coupled with the proposed admission control algorithms to use channels more efficiently, i.e., accommodating more voice or video flow via adaptively reducing the guard period.

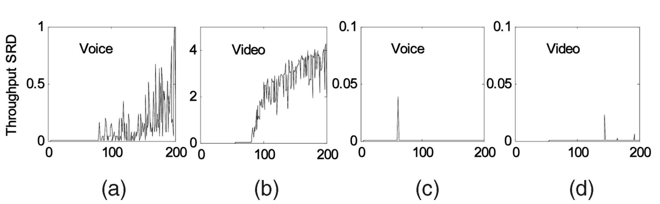

Fig. 22. Throughput SRD without an outside guard versus time (s). (a) and (b) *No inside guard*. (c) and (d) A voice inside $guard = 40$ ms and a video inside $guard = 60$ ms.

Acknowledgments

Bo Li's research was supported in part by grants from RGC under contracts 616505 and 616406.

References

[1] *IEEE 802.11-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE, Aug. 1999.
[2] *IEEE 802.11e/D4.3, Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)*, IEEE, May 2003.
[3] *IEEE 802.11b-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification: High-Speed Physical Layer Extension in the 2.4 GHz Band*, IEEE, Sept. 1999.
[4] *IEEE 802.11a-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification: High-Speed Physical Layer in the 5GHz Band*, IEEE, Sept. 1999.
[5] D.-J. Deng and R.-S. Chang, "A Priority Scheme for IEEE 802.11 DCF Access Method," *IEICE Trans. Comm.*, vol. E82-B, no. 1, pp. 96-102, Jan. 1999.
[6] X. Pallot and L.E. Miller, "Implementing Message Priority Policies over an 802.11 Based Mobile Ad Hoc Network," *Proc. IEEE Military Comm. Conf. (MILCOM '01)*, 2001.
[7] A. Veres, A.T. Campbell, M. Barry, and L.-H. Sun, "Supporting Service Differentiation in Wireless Packet Networks Using Distributed Control," *IEEE J. Selected Areas in Comm. (JSAC)*, vol. 19, no. 10, pp. 2081-2093, Oct. 2001.
[8] S. Mangold, S. Choi, P. May, O. Kein, G. Hiertz, and L. Stibor, "IEEE 802.11e Wireless LAN for Quality of Service," *Proc. European Wireless Conf.*, 2002.
[9] Y. Xiao, "A Simple and Effective Priority Scheme for IEEE 802.11," *IEEE Comm. Letters*, vol. 7, no. 2, pp. 70-72, Feb. 2003.
[10] Y. Xiao, "Enhanced DCF of IEEE 802.11e to Support QoS," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '03)*, Mar. 2003.
[11] L. Romdhani, Q. Ni, and T. Turletti, "Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad Hoc Networks," *Proc. IEEE Int'l Conf. Comm. (ICC '03)*, May 2003.
[12] S. Choi, J. Prado, S. Mangold, and S. Shankar, "IEEE 802.11e Contention-Based Channel Access (EDCF) Performance Evaluation," *Proc. IEEE Int'l Conf. Comm. (ICC '03)*, May 2003.
[13] Y. Xiao, "Performance Analysis of Priority Schemes for IEEE 802.11 and IEEE 802.11e Wireless LANs," *IEEE Trans. Wireless Comm.*, vol. 5, no. 4, pp. 1506-1515, July 2005.
[14] G. Bianchi and I. Tinnirello, "Analysis of Priority Mechanisms Based on Differentiated Inter Frame Spacing in CSMA-CA," *Proc. IEEE Vehicular Technology Conf. (VTC '03)*, Oct. 2003.

[15] Y. Xiao and H. Li, "QoS Enhancement for the IEEE 802.11e Distributed Wireless LANs," *Resource Management in Wireless Networking,* M. Cardei, I. Cardei, and D.-Z. Du, eds., Kluwer Academic, 2004.
[16] S. Tanenbaum, *Computer Networks,* fourth ed. Prentice Hall, 2002.
[17] Y. Xiao, "Backoff Strategies for Demand Re-Registration in PCS Database Failure Recovery," *Computer Comm.,* vol. 27, no. 5, pp. 400-411, Mar. 2004.
[18] Y. Xiao, "IEEE 802.11 Performance Enhancement via Concatenation and Piggyback Mechanisms," *IEEE Trans. Wireless Comm.,* vol. 4, no. 5, pp. 2182-2192, Sept. 2005.
[19] D. Collins, *Carrier Grade Voice over IP,* second ed. McGraw-Hill, Sept. 2002.
[20] Y. Xiao, H. Li, and S. Choi, "Protection and Guarantee for Voice and Video Traffic in IEEE 802.11e Wireless LANs," *Proc. INFOCOM,* 2004.
[21] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE J. Selected Areas in Comm.,* vol. 18, no. 3, pp. 535-547, Mar. 2000.
[22] Y. Xiao, H. Li, and S. Choi, "Two-Level Protection and Guarantee for Multimedia Traffic in IEEE 802.11e Distributed WLANs," *Wireless Networks,* Feb. 2007, DOI 10.1007/s11276-007-0018-9.
[23] Y. Xiao and H. Li, "Voice and Video Transmissions with Global Data Parameter Control for the IEEE 802.11e Enhance Distributed Channel Access," *IEEE Trans. Parallel and Distributed Systems,* vol. 15, no. 11, pp. 1041-1053, Nov. 2004.
[24] Y. Xiao and H. Li, "Local Data Control and Admission Control for Ad Hoc Wireless Networks," *IEEE Trans. Vehicular Technology,* vol. 53, no. 5, pp. 1558-1572, Sept. 2004.

Yang Xiao (SM '04) worked at Micro Linear as a MAC (Medium Access Control) architect involving the IEEE 802.11 standard enhancement before he joined the Department of Computer Science at The University of Memphis in 2002. He joined the Department of Computer Science at The University of Alabama in 2006. He was a voting member of the IEEE 802.11 Working Group from 2001 to 2004. Dr. Xiao is a member of the American Telemedicine Association. He currently serves as editor-in-chief for the *International Journal of Security and Networks* (IJSN), the *International Journal of Sensor Networks* (IJSNet), and the *International Journal of Telemedicine and Applications* (IJTA). He serves as an associate editor or on the editorial board for the following refereed journals: the *International Journal of Communication Systems* (Wiley), *Wireless Communications and Mobile Computing* (WCMC), *EURASIP Journal on Wireless Communications and Networking* (WCN), the *International Journal of Wireless and Mobile Computing* (IJWMC), and *Recent Patents on Engineering*. He has served as a guest editor for the *IEEE Network* special issue on advances on broadband access networks in 2007, a guest editor for the *IEEE Wireless Communications* special issue on radio resource management and protocol engineering in future broadband and wireless networks in 2006, a (lead) journal guest editor for the *IJSN* special issue on security issues in sensor networks in 2005, a (lead) journal guest editor for the *EURASIP WCN* special issue on wireless network security in 2005, a (sole) journal guest editor for the *Computer Communications* (Elsevier) special issue on energy efficient scheduling and MAC for sensor networks, WPANs, WLANs, and WMANs in 2005, a (lead) journal guest editor for the *WCMC* (Wiley) special issue on mobility, paging, and quality of service management for future wireless networks in 2004, and as a (lead) journal guest editor for the *IJWMC* special issue on medium access control for WLANs, WPANs, ad hoc networks, and sensor networks in 2004. He serves as a referee/reviewer for many funding agencies, as a panelist for the US National Science Foundation, and as a member of the Canada Foundation for Innovation (CFI)'s Telecommunications Expert Committee. He has served on technical program committees for more than 90 conferences such as INFOCOM, ICDCS, ICC, GLOBECOM, WCNC, etc. His research areas are in wireless networks, mobile computing, and network security. He has published more than 200 papers in major journals and refereed conference proceedings related to these research areas, including more than 90 journal papers (more than 50 in various IEEE journals and magazines). He is a senior member of the IEEE.

Frank Haizhon Li received the MS and PhD degrees in computer science from The University of Memphis in 2003 and 2005, respectively. He is an assistant professor in the University of South Carolina Upstate. His main research interests are wireless networks and computer security. He also serves as a reviewer and technical program committee member for a number of journals and conferences. He is a student member of the IEEE.

Bo Li (S '89, M '92, SM '99) received the BEng (summa cum laude) and MEng degrees in computer science from Tsinghua University, Beijing, in 1987 and 1989, respectively, and the PhD degree in electrical and computer engineering from the University of Massachusetts at Amherst in 1993. Between 1993 and 1996, he worked on high performance routers and ATM switches in the IBM Networking System Division, Research Triangle Park, North Carolina. Since 1996, he has been with the Department of Computer Science, Hong Kong University of Science and Technology. He has held an adjunct researcher position at Microsoft Research Asia (MSRA), Beijing, China. His current research interests are on adaptive video multicast, packet scheduling, capacity planning in mobile wireless systems, scheduling and energy efficient routing in ad hoc networks, across layer design for sensor networks, and content distribution and replication. He has published 80 journal papers and holds several patents. He received the Outstanding Overseas Young Investigator Award from Natural Science Foundation of China in 2004. He has been on the editorial boards for 16 journals and involved in organizing more than 40 conferences, especially IEEE INFOCOM since 1996. He was the co-technical program committee chair for IEEE INFOCOM 2004. He is currently a Distinguished Lecturer for the IEEE Communications Society. He is a senior member of the IEEE.