Sequential Coordination Function for Throughput and Fairness Enhancement of Wireless LANs

Chaegwon Lim, Chong-Ho Choi, and Hyuk Lim

Abstract

High throughput and fair resource sharing are two of the most important objectives in designing a medium access control (MAC) protocol. Currently, most MAC protocols including IEEE 802.11 DCF adopt a random access based approach in a distributed manner in order to coordinate the wireless channel accesses among competing stations. In this paper, we first identify that a random access–based MAC protocol may suffer from MAC protocol overhead such as a random backoff for data transmission and a collision among simultaneously transmitting stations. Then, we propose a new MAC protocol, called sequential coordination function (SCF), which coordinates every station to send a data frame sequentially one after another in a distributed manner. By defining a service period and a joining period, the SCF eliminates unnecessary contentions during the service period, and by explicitly determining the sequence of frame transmission for each stations, it reduces collision occurrences and ensures fairness among stations in the service period. The performance of SCF is investigated through intensive simulations, which show that the SCF achieves higher throughput and fairness performances than other existing MAC protocols in a wide range of the traffic load and the number of stations.

Index Terms

Wireless LAN, medium access control, throughput, and fairness

A preliminary version of this paper has been presented at the IEEE Global Communications Conference, Nov. 29 – Dec. 3, 2004 [1].

C. Lim is with the Telecommunication R&D Center, Samsung Electronics Co., LTD., Suwon, Korea.

C.-H. Choi is with the School of Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea.

H. Lim is with the Department of Information and Communications, Gwangju Institute of Science and Technology, Gwangju, Korea. Email: hlim@gist.ac.kr

I. INTRODUCTION

In recent years, the demand for higher throughput in wireless networking is continuously increasing, because wireless network users expect better communication environment, and most network applications require wider bandwidth with low latency. The IEEE 802.11 (a.k.a. Wi-Fi) WLAN is one of the most promising technologies that provide last mile wireless connectivity for both enterprise and residential customers. The IEEE 802.11 working group has specified and published several standards such as IEEE 802.11b (up to 11Mb/s) [2], 802.11a/g (up to 54Mb/s) [3], [4], and 802.11n (up to 540 Mb/s) [5] to support data transmission at a higher rate. However, the IEEE 802.11 medium access control (MAC) scheme intrinsically has some inefficiency due to the MAC header overhead for backoff–based access mechanism and the waste of radio resource, especially when collisions occur among a number of competing stations. (We will investigate the efficiency issues of IEEE 802.11 DCF in detail in Section II.) One may think that a higher throughput performance can be obtained simply by increasing the transmission rate. Unfortunately, this is not the case because a simple increase of transmission rate cannot make transmission throughput higher than the throughput performance limit imposed by the MAC overhead of IEEE 802.11 as discussed in [6].

Several other schemes have been proposed to achieve higher throughput by reducing MAC overhead and collision occurrences. The frame aggregation [7] and the burst sending [8] are good examples for reducing the overhead. In the aggregation, several small frames are assembled into one big frame to reduce the header overhead. The burst sending using TXOP in IEEE 802.11e can significantly reduce the backoff overhead. For reducing collisions, adjusting the value of the contention window (CW) according to the network condition has been studied by many researchers [9]–[12]. Furthermore, some ideas that can eliminate collisions completely were presented in [13]. However, they may introduce extra overhead by using tokens or an extra field in the MAC header.

In addition to the throughput performance, fair resource sharing is another important objective for a MAC protocol. Under the assumption that all users are equal, the resource should be shared evenly by all the users, (i.e., ideally each of n stations should be given a fraction $1/n$ of the total resource). However, it is reported that there exists a tradeoff relationship between achieving high throughput and fair resource sharing [14], [15].

In this paper, we present a new MAC mechanism, called sequential coordination function (SCF), which coordinates every station to send a data frame sequentially one after another in a distributed manner, instead of competing with each other station for transmitting frames in order to improve the throughput and fairness performances of WLAN. By defining a service period and a joining period, the SCF eliminates unnecessary contentions during the service period. As a result, the backoff overhead and collision among stations are effectively reduced, and the aggregate throughput and fairness performance are significantly improved.

The rest of this paper is organized as follows. In Section II, we revisit the inefficiency of IEEE 802.11 DCF scheme, and in Section III we present related work. Then, we describe the SCF mechanism in detail in Section IV. We analyze the performance of SCF in Section V, and compare the performance of SCF to those of DCF, FCR [12], and Idle Sense [11] through intensive simulations in Section VI. Finally, in Section VII conclusions and future work are given.

II. Background

In this section, we consider the efficiency issues of random access–based MAC protocols. Specifically, the contention overhead and the impact of collisions are evaluated for the contention–based MAC protocol of IEEE 802.11 DCF.

A. Contention overhead

First, we compute the expected MAC protocol overhead of IEEE 802.11 DCF. Suppose that there are two stations, (i.e., a sender and a receiver). Under the assumption that the sender always has backlogged data, the expected time T_{BASIC} spent for sending a data frame successfully is

$$T_{BASIC} = T_{DIFS} + T_{\mathbb{E}[BC]} + T_{data} + T_{SIFS} + T_{ack} + 2\delta, \quad (1)$$

where T_{data} and T_{ack} represent the transmission times of a data frame and an ack frame, respectively, and δ denotes the propagation delay, which is assumed to be negligible. Note that T_{data} is the sum of the transmission time of the frame header (T_{header}) which is fixed and the transmission time of payload ($T_{payload}$) which is variable. $\mathbb{E}[BC]$ is the expected value of backoff counter (BC), and $T_{\mathbb{E}[BC]}$ is the expected contention period, (i.e., $\mathbb{E}[BC] \times T_{aSlotTime}$). When there is no other station, the contention window (CW) of the sender is maintained as CW_{min},

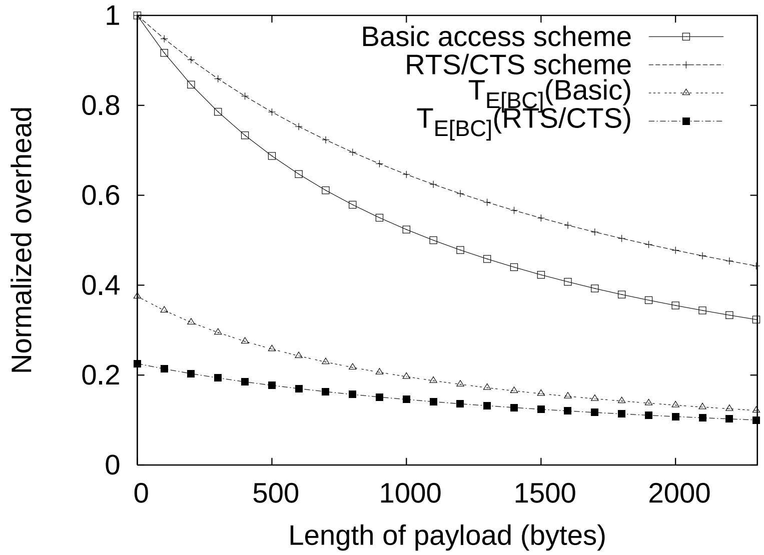


Fig. 1. Contention overhead including a fixed length of protocol overhead and a variable contention period in an IEEE 802.11b DCF wireless network.

and $\mathbb{E}[BC]$ becomes $\frac{CW_{min}-1}{2}$. When the sender uses the RTS/CTS scheme, the overhead of $T_{RTS/CTS}$ is calculated as

$$T_{RTS/CTS} = T_{RTS} + T_{CTS} + 2T_{SIFS} + T_{BASIC}.$$

We define the normalized overhead as

$$\mathcal{O} = 1 - \frac{T_{payload}}{T_{BASIC}}.$$

Fig. 1 depicts the overhead for IEEE 802.11b DCF, when only two stations communicate with each other at 11 Mb/s. We observe that when the length of payload is small, the overhead is quite dominant, and a large portion of overhead is due to the contention period. Even when the length of payload reaches its accepted maximum value (i.e., 2,304 bytes), the overhead is around 40 %. In order to improve the performance of DCF, this overhead should be reduced by minimizing $T_{\mathbb{E}[BC]}$ or maximizing $T_{payload}$, which is determined by the data transmission rate and the frame size. However, even in the case of an extremely high data transmission rate, the achievable throughput is still bounded in [6]. This implies that the contention based approach incurs the contention overhead in coordinating the contentions among multiple nodes, and it should be reduced for better throughput performance.

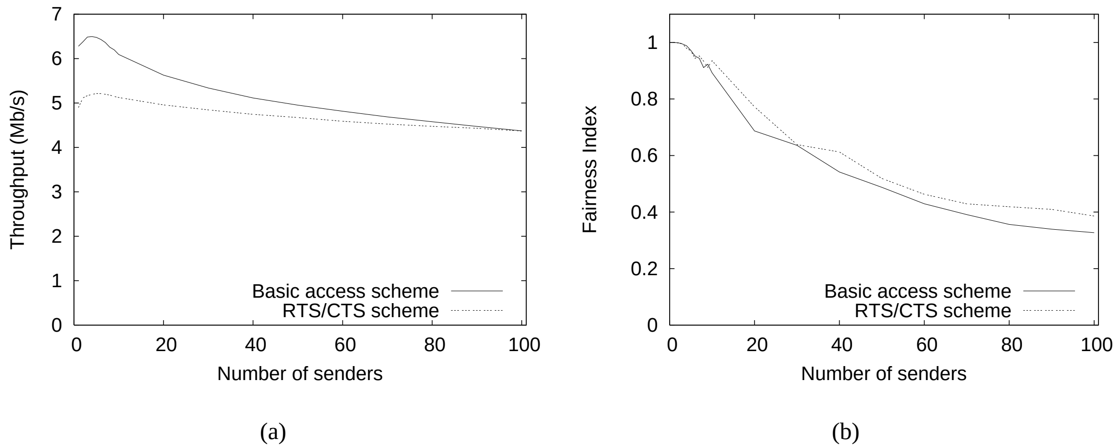


Fig. 2. Performance evaluation for an IEEE 802.11b DCF wireless network in terms of (a) aggregate throughput and (b) fairness index.

B. Impact of collisions

We evaluate the throughput and fairness performance for IEEE 802.11 DCF operated networks where the length of payload is 1,500 bytes, and the channel is lossless. Each sender always has backlogged data in ad hoc mode.

Fig. 2(a) shows the aggregate throughput as the number of competing stations increases. As the number of senders increases up to six, the throughput for both basic and RTS/CTS schemes increases. After reaching its maximum, the throughput begins to decrease gradually. The reason is that when the throughput has not yet reached its maximum, the backoff time is longer than that needed in avoiding collisions among stations, and thus as the number of senders increases, the throughput increases. However, after reaching its maximum, collisions occur more frequently because the contention period is not sufficiently long enough to avoid collisions. In this case, whenever perceiving that a collision has occurred, the senders double their CWs to reduce further potential collisions. This binary exponential backoff operation of the IEEE 802.11 DCF may degrade the fairness performance [16]. Fig. 2(b) shows that the Jain's fairness index [17]. The result shows that the binary backoff algorithm does not provide fair throughput sharing among users [18], [19].

In this simulations, we observe that as the number of senders increases, the aggregate throughput and the fairness performances become worse due to the increased collisions among stations. It is mainly due to that the random access contention mechanism of IEEE 802.11 DCF is not efficient, especially when the offered network load dynamically varies [9], [10], [20].

III. Related Work

There have been many studies to improve the throughput performance of IEEE 802.11 networks. Methods to improve the throughput of 802.11 networks can fall into two categories: reducing overhead and reducing collisions.

First, increasing payload by aggregating several service packets into a single frame [7], [21] could be effective in improving throughput performance especially when there are many small-sized packets, such as voice over IP packets or TCP ack packets. Another method is sending several frames consecutively without backoff by using the TXOP bursting mechanism of IEEE 802.11e [8] to reduce the backoff overhead. The block ack mechanism also can be used for reducing the overhead induced by sending ack frames for each transmission [22]. The MAC header compression is another alternative method that can be used to reduce the MAC header overhead [23].

Adjusting the value of contention window in IEEE 802.11 DCF is one of the most popular methods to reduce collisions. The adjustments are made by either estimating the current congestion level of the channel [9]–[11], [20], [24], [25] or changing the rule of choosing the next value of CW [12], [26]–[29]. These schemes avoid collisions by reducing the transmission opportunities of senders when the level of collision is high. Penalty due to collisions can be reduced by adopting the RTS/CTS mechanism because the length of an RTS frame is much shorter than that of a data frame in most cases.

Collisions can be eliminated entirely by perfectly scheduling the order of transmissions. An example of this approach is the PCF of IEEE 802.11, which uses a polling mechanism during the contention free period. A similar idea was presented in [13], where the polling mode and the contention mode alternate based on the on-line estimation of the network load. However, these mechanisms need a supervisor node, so they are inapplicable to ad hoc networks. In order to be used in ad hoc networks, the distributed token ring mechanism [30] and the distributed ordering protocol [31] were proposed. However, these mechanisms may introduce additional overhead such as tokens or an extra field in the MAC header. In addition, they increase the system complexity.

Regarding the fairness issue, there exists a tradeoff relationship between achieving high throughput and sharing resource fairly [14], [15]. In order to increase channel utilization, stations need to be more aggressive, however this behavior attributes to unfair resource sharing in general [18], [19].

In a recent study, the basic concept of the sequential coordination has been proposed to achieve high throughput performance by mitigating the collision overhead among contending nodes in a distributed manner [1]. This paper is an extended version of the work done in [1]. The contributions of this study are as follows: (i) We have modified the state transition as shown in Fig. 4 in order to make SCF nodes more responsive to unsuccessful transmissions. (ii) We have analyzed the the joining latency and the utilization performance in steady state. (iii) We have implemented another algorithm of ”Idle Sense” [11] in our comparative study, and compared their performances against that of SCF under more various network scenarios.

IV. Sequential Coordination Function

In order to provide high throughput and fair resource sharing, we propose a new MAC mechanism, called sequential coordination function (SCF), in which, instead of competing with each other for transmitting frames, every sender is coordinated to send a data frame sequentially one after another in a distributed manner. This operation is regulated automatically by a simple counting operation, which has a little similarity with the backoff operation of IEEE 802.11 DCF. As a result, the senders can eliminate unnecessary collisions among themselves and share the channel fairly. The operation of SCF is based on the assumption that each sender is in the carrier sensing range of the other stations. (We will show that the operation of SCF is quite robust to the hidden stations problems once the network is established in Section VI.)

The SCP defines a **service period** (SP) and a **joining period** (JP). Because the SCP explicitly determines the sequence of frame transmission for each stations during the service period, it can eliminate both unnecessary contentions and collision occurrences. A new station is allowed to join the network only during the joining period, and is coordinated to begin to send a data frame at the next SP. For the operation of SCF, each station maintains three system parameters listed below.

- N_{AS}: the estimated number of senders in the network counted by each station in each period. If a station detects a transmission, it increases N_{AS} by one.

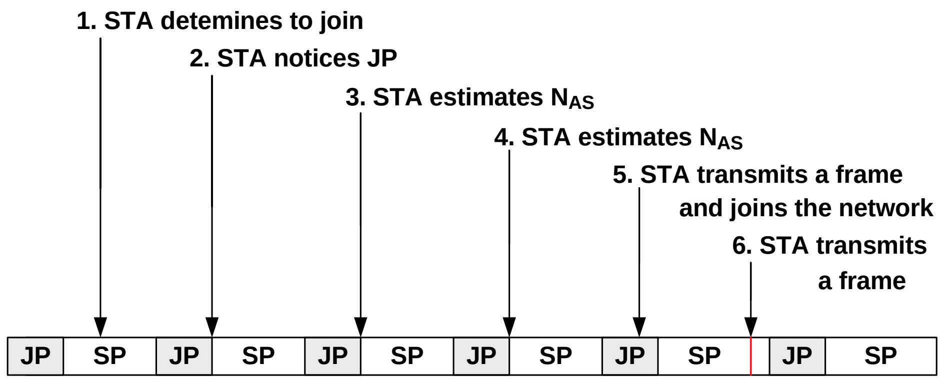


Fig. 3. Operation of SCF, in which service and joining periods alternate.

- N_{JP}: the number of slots for a JP. N_{JP} is a design parameter and is assumed to be known to all stations in the network in advance. In this paper, N_{JP} is set to five, but is desirable to be adaptively adjusted depending on the arrival rate of newly joining stations. It would be a part of our future work.
- N_{BC}: the value of the backoff counter. It is decreased by one for each DIFS after a transmission or for each idle $T_{aSlotTime}$. A station can transmit a frame only when its N_{BC} becomes zero.

A. Basic operation of SCF

A basic period of SCF is composed of one service period (SP) and one joining period (JP), and an SP and a JP are repeated alternately as shown in Fig. 3. During an SP, each sender in a network is given an opportunity to transmit one frame in an SP slot. During a JP, a new station is given an opportunity to join the network. The duration of a JP (N_{JP}) is fixed and is known to all stations in advance. If a new station detects an idle time that lasts longer than N_{JP}, it implies that the preceding SP period includes some idle slots, and the end of idle slots is the end of a JP period.

A SCF station is always in one of the three states; STANDBY, JOIN, or ACTIVE. Fig. 4 shows the transition diagram among these states. When a station is turned on, it first enters the STANDBY state. In the STANDBY state, it does nothing except receiving frames from other stations. When a station in the STANDBY state receives data to send from its upper layer, it enters the JOIN state and actively observes the network to obtain information on the value of

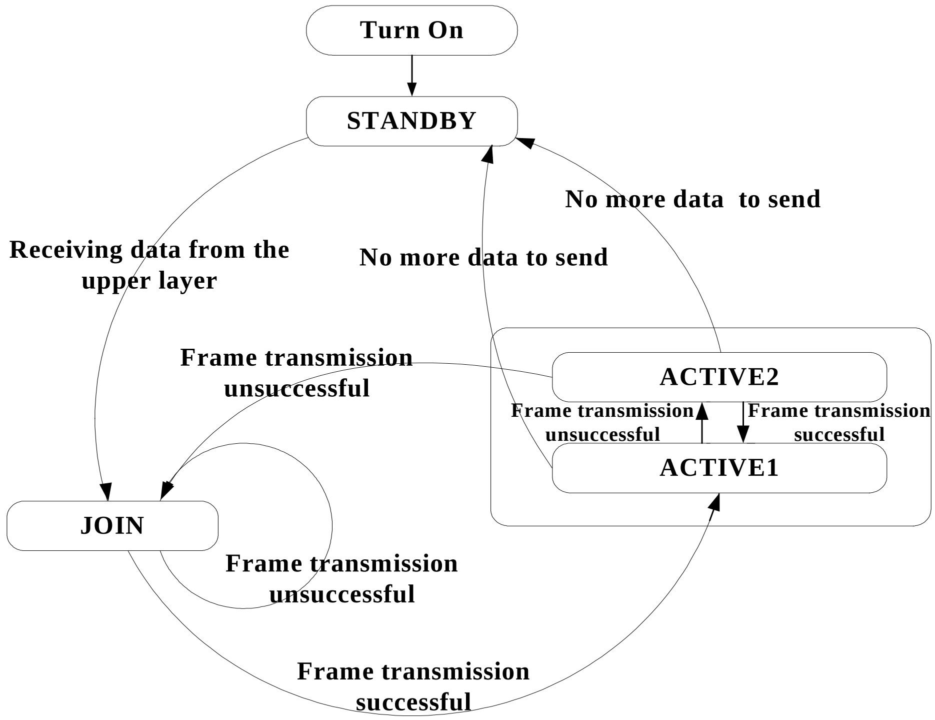


Fig. 4. State transition diagram of SCF.

current N_{AS} and when the next JP starts. After obtaining the information, it sends a data frame during the next JP. If the sender receives an ack frame, which indicates that the data frame has been transmitted successfully, it goes into the ACTIVE state. After this, the sender sends a frame in each SP. After sending all data, it enters the STANDBY state again.

The ACTIVE state is composed of two sub-states; ACTIVE1 and ACTIVE2. When a station enters the ACTIVE state, it goes into the ACTIVE1 state. When a sender notices a collision, it enters the ACTIVE2 state. If a station in the ACTIVE2 state sends a data frame successfully in the following SP, it goes back to the ACTIVE1 state. When a station in the ACTIVE2 state becomes involved in another collision, it enters the JOIN state again in order to refresh its network information. When a station in the ACTIVE state does not have data to send anymore, it enters the STANDBY state.

B. Sequential coordination

As shown in Fig. 4, when a station in the STANDBY state has data to send, it enters the JOIN state. Upon entering this state, the station attempts to find out when the next JP starts. First, the station estimates N_{AS} by counting the number of transmissions between two consecutive JPs as shown in Fig. 3. If two successive estimates of N_{AS} are equal, the station assumes that it has counted N_{AS} correctly and sends a data frame during the following JP. If not, it has to delay its transmission until two successive estimates of N_{AS} are equal. When two successive estimates of N_{AS} are equal, at the beginning of the following SP (in Fig. 3, the fourth SP), the station sets its N_{BC} and N_{AS} as

$$\begin{aligned} N_{BC} &= N_{AS} + K, \qquad (2) \\ N_{AS} &= 0. \end{aligned}$$

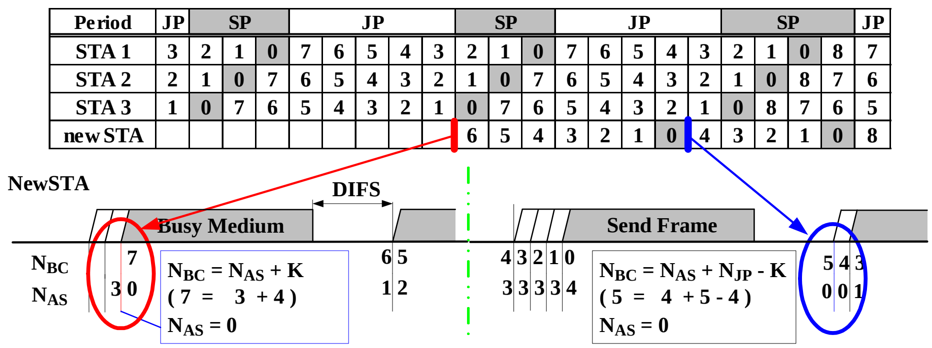
where $K = rand(N_{JP})$ and $rand(n)$ function returns an integer which is chosen between one and n with equal probability. This randomness helps avoid collisions among multiple new stations. As time goes on, N_{BC} is decreased to zero, and the new station attempts to transmit a frame. In the case of a successful transmission, the station sets its N_{BC} and N_{AS} as follows:

$$\begin{aligned} N_{BC} &= N_{AS} + N_{JP} - K, \qquad (3) \\ N_{AS} &= 0. \end{aligned}$$

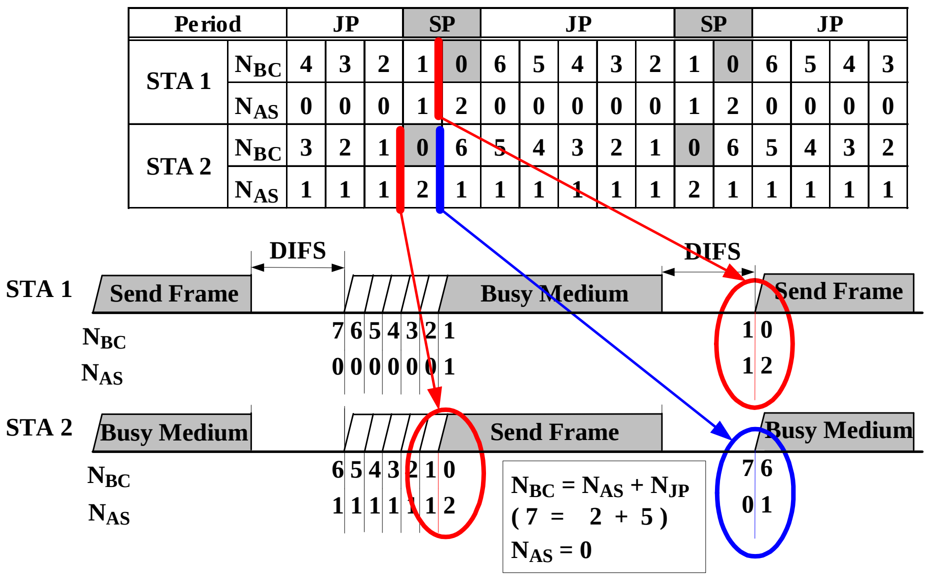
Note that by setting the value of N_{BC} as above, the last station whose transmission has been successful in the last JP becomes the last sender in the following SP. N_{AS} is reset, and then begins to increase by one for each detected transmission. Eventually, it will be larger by one than that in the preceding basic period.

After a successful transmission during a JP, the station enters the ACTIVE state and is given an opportunity to transmit in each SP that follows. If the transmission is not successful (i.e., the sender does not receive an ack frame within a pre-determined time), it behaves as if it has just entered the JOIN state.

Fig. 5(a) shows the changes of N_{BC} for each station when it joins the network. There are three active stations that are in the ACTIVE state and one new station that newly joins the network. The slots in white color simply denote idle slots, and the slots in grey color indicate that a station is sending a frame because its N_{BC} becomes zero. Thus, the duration of a slot in white color is equal to $T_{aSlotTime}$, while that in grey color is $T_{data} + T_{SIFS} + T_{ack}$, although all slots are depicted with the same length. The time corresponding to T_{DIFS} has been omitted for simplicity. As shown in Fig. 5(a), each station decreases its N_{BC} after waiting for T_{DIFS} or



(a) A station joins the network ($N_{JP} = 5, K = 4$).



(b) Stations are in the ACTIVE state.

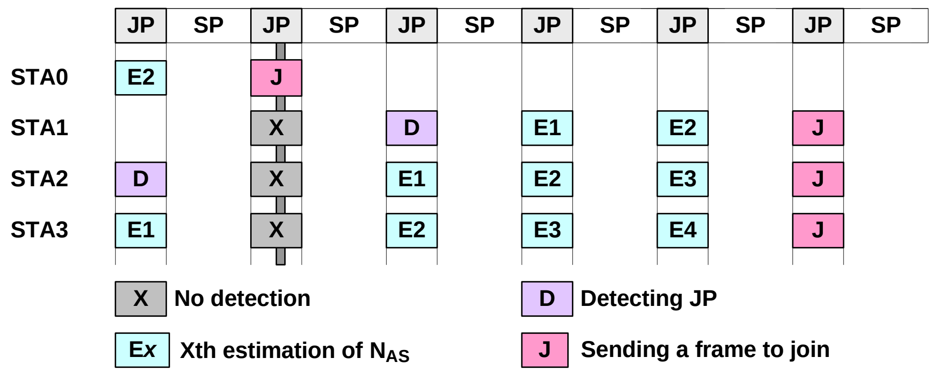
Fig. 5. An illustrative example for N_{AS} and N_{BC} in JOIN and ACTIVE states.

after one idle $T_{aSlotTime}$. Suppose the new station selects four for K in Fig. 5(a). Then, at the beginning of the first SP period, N_{BC} and N_{AS} are set to be 7 and 0, respectively, by (2), and they are immediately decreased by one due to the transmission of STA 3. At the fourth slot of the second JP, the new station transmits a frame, and sets N_{BC} = 5 and N_{AS} = 0 by (3). Then, in the second SP, it is scheduled to transmit a frame at the fourth slot.

After successfully transmitting a frame, the station in the ACTIVE state sets its N_{BC} and N_{AS}

Period		JP	SP		JP					SP				JP					SP			
STA 1	N_{BC}	2	1	0	6	5	4	3	2	1	0	8	7	6	5	4	3	2	1	0	8	7
	N_{AS}	0	1	2	0	0	1	2	2	3	4	1	2	2	2	2	2	2	3	4	1	2
STA 2	N_{BC}	1	0	6	5	4	3	2	1	0	8	7	6	5	4	3	2	1	0	8	7	6
	N_{AS}	1	2	1	1	1	2	3	3	4	1	2	3	3	3	3	3	3	4	1	2	3
New STA 3	N_{BC}		4	3	2	1	0	4	3	2	1	0	8	7	6	5	4	3	2	1	0	8
	N_{AS}	2						1	1	2	3	4	1	1	1	1	1	1	2	3	4	1
New STA 4	N_{BC}		5	4	3	2	1	0	4	3	2	1	0	8	7	6	5	4	3	2	1	0
	N_{AS}	2							0	1	2	3	4	0	0	0	0	0	1	2	3	4

(a) N_{BC} and N_{AS} for two joining stations.



(b) Estimation of N_{AS} for four joining stations.

Fig. 6. An illustrative example for N_{AS} and N_{BC} when stations simultaneously join the wireless network.

as follows:

$$N_{BC} = N_{AS} + N_{JP}, \tag{4}$$
$$N_{AS} = 0.$$

Recall that N_{AS} is increased by one whenever there is a transmission during the basic period. Fig. 5(b) depicts the changes in N_{AS} and N_{BC} of each station in the ACTIVE state as time goes by.

C. Simultaneous joins and leaves

We consider a special case when multiple stations join and leave the network simultaneously. The transmission of stations in SCF is automatically coordinated simply by counting N_{AS} even when the number of senders changes. Fig. 6(a) shows the changes of N_{BC} and N_{AS} when two stations join the network simultaneously. Note that STA1 and STA2 increase their N_{AS}s by two after two new stations join.

On the other hand, if there is a transmission in a JP, the other joining stations will fail to detect the JP correctly, because they cannot observe the consecutive N_{JP} idle slots. After missing a JP, the consecutive estimations of N_{AS} will not be the same. Thus, after missing a JP, the joining stations should find out the starting point of the next SP and JP, and estimate again the current N_{AS} correctly. Fig. 6(b) shows the joining process for four stations. Due to the transmission of STA0, the other three stations fail to identify the second JP correctly. As a result, STA1 is able to successfully identify a JP in one basic period later. STA2 will realize that the first estimation (E1) and the second estimation (E2) of N_{AS} are different, and it needs another N_{AS} estimation (E3). For STA3, it needs two more N_{AS} estimations (E3, E4). After these operations, all joining stations transmit their frames in one of N_{JP} slots in the sixth SP, depending on the randomly selected N_{BS} in (2).

Fig. 7 shows the case when two stations leave the network simultaneously. In this case, N_{AS} in each of the remaining stations is decreased accordingly. Consequently the remaining stations can send data frames with a shorter basic period as shown in Fig. 7. However, when several senders leave the network simultaneously, the stations in the JOIN state may determine the start of the next JP incorrectly, and send a frame during an SP. In such a case, it is possible that collisions may occur among the stations in the ACTIVE and JOIN states. By adopting the two sub-states of ACTIVE1 and ACTIVE2 in Fig. 4, the former stations will attempt to send a frame in the next SP again, while the latter stations will behave as if it have just entered the JOIN state.

V. Performance Analysis

In this section, we analyze two key properties of (i) the latency to join the network and (ii) the utilization when the network is operating in steady state. For simplicity of the analysis, we first assume that collisions are the only cause for corrupted frames. Second, we assume that the number of joining stations and the number of leaving stations are the same in average (i.e., the number of station in the ACTIVE state (N_{AS}) is constant.).

Period		SP		JP					SP		JP							SP		JP			
STA 1	N_{BC}	8	7	6	5	4	3	2	1	0	8	7	6	5	4	3	2	1	0	6	5	4	3
	N_{AS}	1	2	2	2	2	2	2	3	4	0	0	0	0	0	0	0	1	2	0	0	0	0
STA 2	N_{BC}	7	6	5	4	3	2	1	0	8	7	6	5	4	3	2	1	0	6	5	4	3	2
	N_{AS}	2	3	3	3	3	3	3	4	1	1	1	1	1	1	1	1	2	1	1	1	1	1
STA 3	N_{BC}	0	8	7	6	5		STA 3 exits															
	N_{AS}	1	2	2	2	2																	
STA 4	N_{BC}	1	0	8	7	6	5	4		STA 4 exits													
	N_{AS}	4	1	1	1	1	1	1															

Fig. 7. An illustrative example for N_{AS} and N_{BC} when two stations simultaneously leave the wireless network.

A. Latency to join the network

We define the joining latency D_{join} as the time delay experienced by a station until it successfully joins the network. It implies that it takes D_{join} for each station to switch its state from the JOIN state to the ACTIVE state.

Recall that a basic period is composed of an SP and a JP, and they are alternately repeated as shown in Fig. 3. Let T_{SP} denote the duration of an SP, when the RTS/CTS scheme is not used. During an SP, it takes T_{BASIC} with $E[BC] = 0$ in (1) for each sender station to transmit a frame. Therefore, T_{SP} is simply calculated as follows:

$$T_{SP} = N_{AS} \cdot T_{SCF-BASIC}, \tag{5}$$

where $T_{SCF-BASIC}$ is equal to T_{BASIC} with $T_{E[BC]} = 0$ in (1). Let T_{JP} denote the duration of a JP. When there is no station joining the network during a JP, T_{JP} is given by $T_{DIFS} + N_{JP} \cdot T_{aSlotTime}$. However, if there are a number of stations that attempt to join the network, all the slots of the JP are occupied with the transmission attempts. In such a case, the duration of the JP is given by $N_{JP} \cdot T_{SCF-BASIC}$. The number of cases that there exist k non-idle slots among N_{JP} slots in a JP is given by

$$E_k(N_{JP}, N_{JS}) = \begin{cases} 0 & \text{for} \quad k = 0 \\ N_{JP} & \text{for} \quad k = 1 \\ C(N_{JP}, k) \cdot \left((k)^{N_{JS}} - \sum_{i=1}^{k-1} p_i(i, N_{JS}) \right) & \text{for} \quad k = 2, \cdots, N_{JP}, \end{cases}$$

where N_{JS} is the number of stations in the JOIN state, and $C(n, k)$ is the binomial coefficient (i.e., $C(n,k) = \frac{n!}{k!(n-k)!}$). Consequently, the expectation of T_{JP} is given by

$$\mathbb{E}[T_{JP}] = \sum_{k=1}^{N_{JP}} \left(\frac{E_k(N_{JP}, N_{JS})}{(N_{JP})^{N_{JS}}} \cdot ((N_{JP}-k)(T_{DIFS} + N_{JP} \cdot T_{aSlotTime}) + k \cdot T_{SCF-BASIC}) \right).$$

Suppose a data frame is received from the upper layer at $t = t_{arrive}$. If the frame arrives during the SP (i.e., $t_{arrive} \le T_{SP}$), the station can successfully detect the end of the JP. In this case, the latency for detecting the end of the JP is $(T_{SP} + T_{JP} - t_{arrive})$. On the contrary, if the frame arrives during the JP, it fails to detect the end of the JP, and has to wait for another basic period. Therefore, the latency in this case is $(2T_{SP} + 2T_{JP} - t_{arrive})$. Under the assumption that the arrival of a data frame is uniformly distributed within the basic period, the expected latency is given by

$$\mathbb{E}[D_{detect}] = \frac{1}{2} T_{SP} + \frac{3}{2} \mathbb{E}[T_{JP}].$$

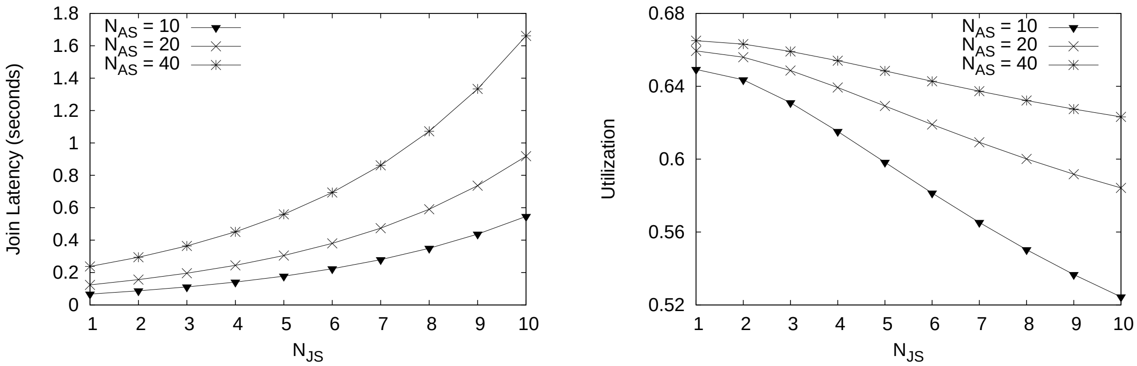
After detecting the end of the JP, the station estimates N_{AS} for two successive basic period, and at the next period, it transmits a frame. Therefore, each attempt fail causes a latency of $D_{transmit} = 3T_{SP} + 3T_{JP}$. The expected joining latency is obtained as follows:

$$\begin{aligned} \mathbb{E}[D_{join}] &= \mathbb{E}[D_{detect}] + \sum_{i=1}^{\infty} \big(i\ p_s (1-p_s)^{i-1}\big) \mathbb{E}[D_{transmit}] \qquad (6) \\ &= \left(\frac{1}{2} + \frac{3}{p_s}\right) T_{SP} + \left(\frac{3}{2} + \frac{3}{p_s}\right) \mathbb{E}[T_{JP}], \end{aligned}$$

where p_s is the probability that a station enters the ACTIVE state successfully, (i.e., the probability of sending a frame successfully during a JP). The probability p_s is calculated as

$$\begin{aligned} p_s(N_{JP}, N_{JS}) &= \frac{N_{JP} \cdot (N_{JP}-1)^{N_{JS}-1}}{(N_{JP})^{N_{JS}}} \\ &= \left(1 - \frac{1}{N_{JP}}\right)^{N_{JS}-1}. \end{aligned}$$

Fig. 8(a) shows the numerical result of the joining latency in (6) with respect to N_{JS} when the N_{JP} is set 5. Because the number of slots for newly joining stations is fixed at 5, the joining latency becomes larger as the number of nodes in the JOIN state increases. It is seen that as the number of nodes in the ACTIVE state increases, the latency also increases due to the longer duration of the service period.



(a) Joining Latency (b) Utilization

Fig. 8. Numerical results for joining latency and the utilization with respect to N_{JS} when N_{JP} = 5.

B. Utilization

When there exist N_{AS} active stations and N_{JS} joining stations, the utilization is defined as the ratio of the time taken to transmit a payload with respect to the total time, and is calculated as

$$U = \frac{T_{payload} \cdot (N_{AS} + N_{JS} \cdot p_s)}{T_{SP} + \mathbb{E}[T_{JP}]}. \tag{7}$$

Note that $N_{AS} \cdot T_{payload}$ and $N_{JS} \cdot p_s \cdot T_{payload}$ are the expected time spent in sending payload during an SP and a JP, respectively. When there is no joining station, the utilization is re-written as

$$U = \frac{N_{AS} \cdot T_{payload}}{N_{AS} \cdot T_{SCF-BASIC} + T_{DIFS} + N_{JP} \cdot T_{aSlotTime}}.$$

Even though the effects of a collision in contention based approaches are assumed to be negligible in lightly loaded traffic, the proposed scheme can still achieve the higher utilization if

$$N_{JP} \leq N_{AS} \cdot \frac{\text{CW}_{min} - 1}{2} - \frac{T_{DIFS}}{T_{aSlotTime}}.$$

Fig. 8(b) shows the numerical result of the utilization in (7) with respect to N_{JS}. It is seen that the number of nodes in the ACTIVE state increases, the higher throughput is achieved because the effect of the overhead due to the fixed duration of joining period is relatively reduced. Also, because p_s becomes smaller as the number of nodes in the JOIN state increases, the throughput performance becomes slightly lower. From this numerical analysis, we observe that the duration of N_{JP} significantly affects the throughput and delay performance of SCF. How to adjust N_{JP}

and how to distribute the adjusted value of N_{JP} to all the contending nodes will be one of our future research work.

VI. Performance Evaluation

To evaluate the performance of SCF, we have performed intensive simulations by using the ns-2 (ver. 2.27) and have compared the performance of SCF with that of IEEE 802.11 DCF, fast collision reduction (FCR) [12], and Idle Sense [11] in terms of the throughput, fairness, and delay.

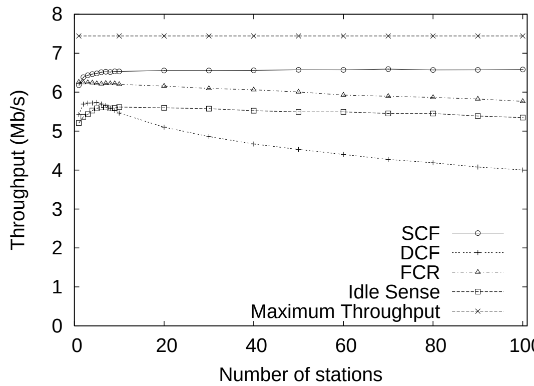
- SCF: The implementation os SCF is based on that of IEEE 802.11 DCF. To make the stations in SCF operate synchronously even in cases of frame corruption by a collision or a channel error, we simply set $CTSTimeout$ and $ACKTimeout$ to be equal to EIFS.
- FCR: In order to reduce the idle slots, FCR uses a small value of the initial CW_{min} for successful transmissions and reduces the backoff counter exponentially for successive idle slots. FCR increases the contention window size for a busy state as well as a collision state of wireless channel. In our simulations, CW_{min} for FCR is set to three.
- Idle Sense: Each sender counts and averages the number of idle slots between two successive transmissions, and compares it to the optimal number of idle slots. If the average number is larger, the sender decreases its contention window in order to increase channel utilization. Otherwise, it increases its contention window to avoid potential collisions.

The performances of these schemes are investigated under five different scenarios: a) heavily loaded network, b) lightly loaded network, c) load-varying network, d) network with TCP traffic flows, and e) network with hidden stations.

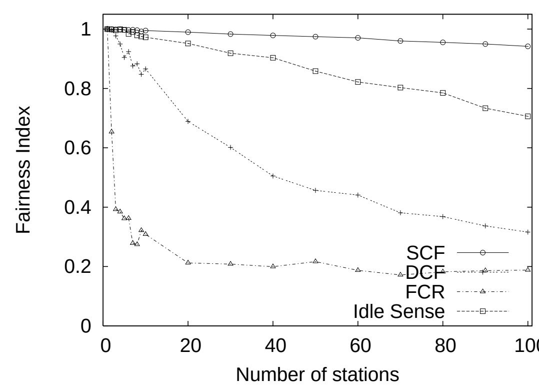
In the following simulations, the transmission range of each station is 100 meters, and all the stations are in a region of 70 meters x 70 meters (i.e., there is no hidden station) except the fifth scenario with hidden stations in Section VI-E. The length of a data frame is 1,500 bytes, and the RTS/CTS mechanism is disabled. The transmission rate of each station is fixed at 11 Mb/s. The reported results are obtained by averaging 20 runs of the simulations.

A. Static network with backlogged nodes

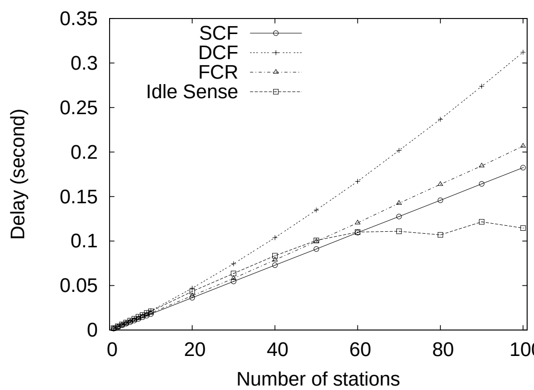
We investigate the network performance of SCF in a static network where all the senders are always backlogged and the contention level is extremely high. First, we study the aggregate



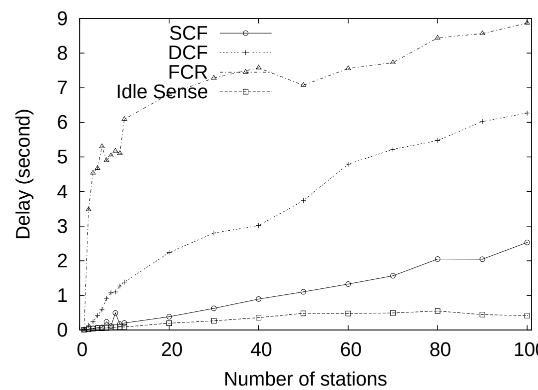
(a) Aggregate throughput



(b) Fairness index



(c) Average delay



(d) Worst case delay

Fig. 9. Simulation results in a static network with backlogged nodes.

throughput, which is defined as the total amount of data received divided by the simulation time. In this saturated scenario, the maximum achievable aggregate throughput on lossless channel is 7.44 Mb/s, which is calculated from (1) by setting $T_{\mathbb{E}[BC]}$ to zero. Fig. 9(a) shows that the aggregate throughput of DCF decreases rapidly as the number of stations increases as mentioned in Section II. The throughput of FCR also decreases, but the decrease is much smaller than that of DCF, because FCR has a small CW_{min}, which contributes to reducing overhead of the idle period. On the other hand, the aggregate throughput of SCF and Idle Sense remain constant as the number of senders increases due to their novel schemes to avoid or reduce collisions. We observe that the SCF improves the aggregate throughput by 65 % in comparison with DCF. It is worth noting that on lossless channel, the SCF achieves the highest throughput for all the cases.

We examine the fairness performance among the senders. We use the Jain's fairness index $\mathcal{F}$, which is defined as

$$\mathcal{F} = \frac{\left(\sum_i x_i\right)^2}{n \cdot \sum_i {x_i}^2} ,$$

where n is the number of flows, and x_i is the measured throughput of the ith flow in [17]. As $\mathcal{F}$ becomes larger, it can be said that the network resource is shared more evenly among stations. We calculated $\mathcal{F}$ from the data collected for 500 ms in steady state. Fig. 9(b) shows that SCF gives the highest fairness index. As the number of senders increases, the fairness index of SCF decreases slightly because SCF gives each sender at least one chance to transmit in every basic period. On the other hand, FCR shows the worst performance because it uses the binary backoff algorithm in both increasing and decreasing CW, whereas DCF uses the binary backoff algorithm only in increasing CW. Note that the binary backoff algorithm may degrade the fairness performance among stations [18], [19]. In order to overcome this unfairness, FCR may adopt the distributed version of the SCFQ algorithm [32]. Fig. 9(b) also shows that Idle Sense remarkably improves the fairness performance, even though it does not outperform SCF.

The transmission delay is defined as the time elapse from the instant a sender first attempts to send a data frame to the instant it receives an ack frame from the receiver. In SCF, when a sender has frames to send, it has to join the network first. One might expect that this joining latency becomes too large for a heavily loaded network. In (6), the joining latency increases as the number of stations in the JOIN state (N_{JS}) increases because p_s becomes smaller. Fig. 9(c) indicates that SCF achieves smaller average transmission delay than DCF and FCR in a reasonably wide range of the number of stations. In comparison with Idle Sense, it achieves the better performance until the number of stations reaches sixty. For the worst case delay, SCF and Idle Sense show better performance than the others as shown in Fig. 9(d). We observe that the unfairness among stations significantly contributes to the worst case delay, because when a station does not have a chance to transmit for sometime, the worst case delay increases. In the case of SCF, the worst case delay is mostly due to a joining delay.

B. Dynamic network with light traffic load

We evaluate the performance of SCF in a dynamic network with light traffic load, where there exist ten stations that are not backlogged. In this scenario, the stations keep silent for a certain time interval depending on their sending rate. Under SCF, when a station does not have data

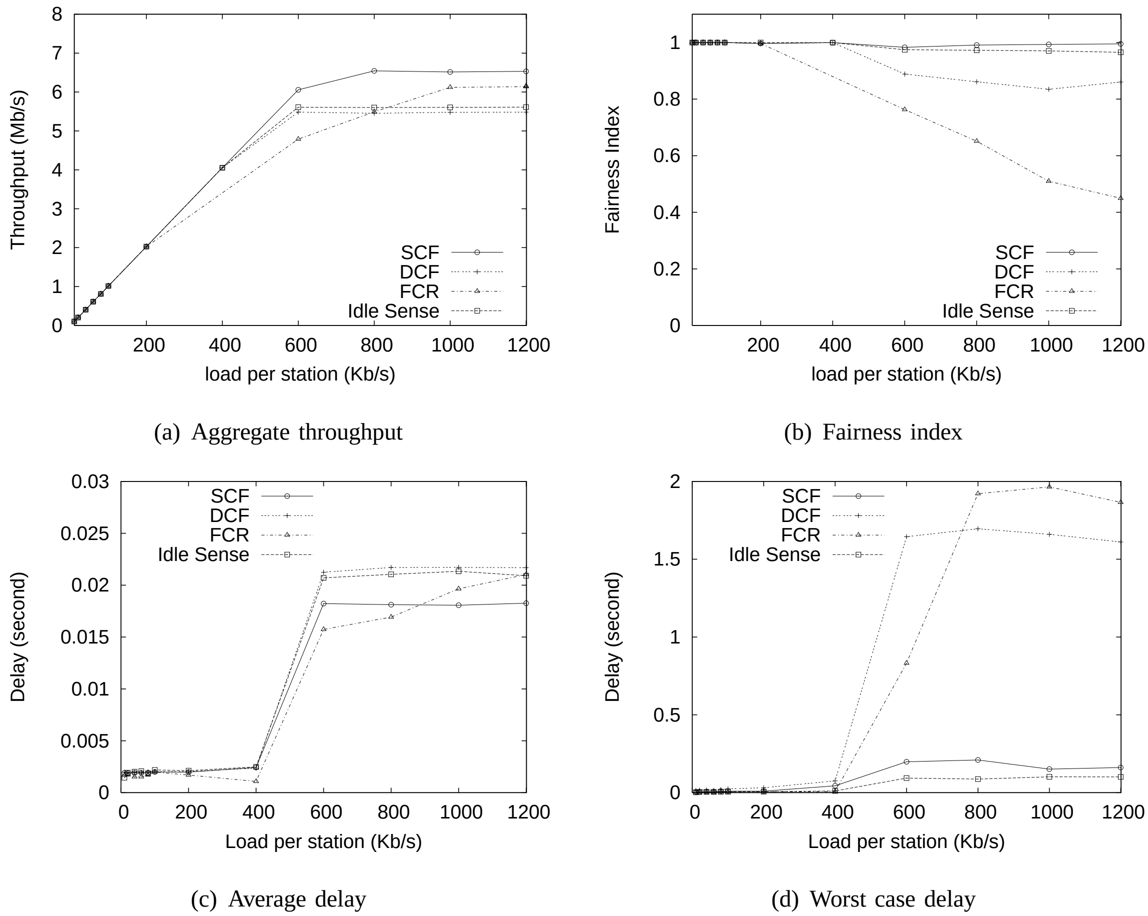


Fig. 10. Simulation results in a dynamic network with light traffic load.

to send, it transits from ACTIVE state to STANDBY state, and such a node is regarded as a leaving station. When it has a data to send, it joins the network again. Consequently, the stations that are not backlogged repeatedly join and leave the network under SCF.

Here, we investigate the aggregate throughput, fairness index, and joining delay with respect to the offered load per station. Fig. 10(a) indicates that the aggregate throughput of SCF is higher than those of the other schemes in the whole range. It demonstrates that SCF can work properly and efficiently under a dynamic network condition. In terms of fairness, FCR is the worst, and SCF is the best as shown in Fig. 10(b). Thus, it implies that SCF utilizes the network resource efficiently and fairly in the dynamic network with light traffic load. Fig. 10(c) and 10(d) also show the average transmission delay and the worst case delay of SCF are better than or comparable to the others in all the ranges of traffic load. Compared with the case for the

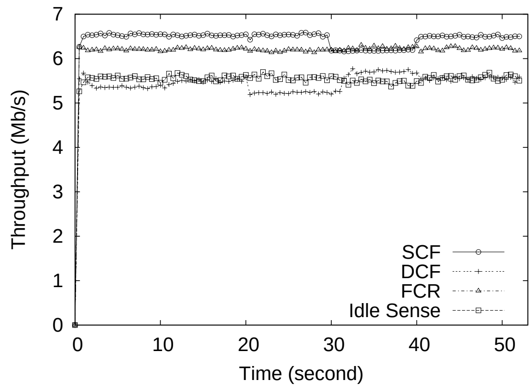


Fig. 11. Simulation results in a mobile network with joining and leaving nodes.

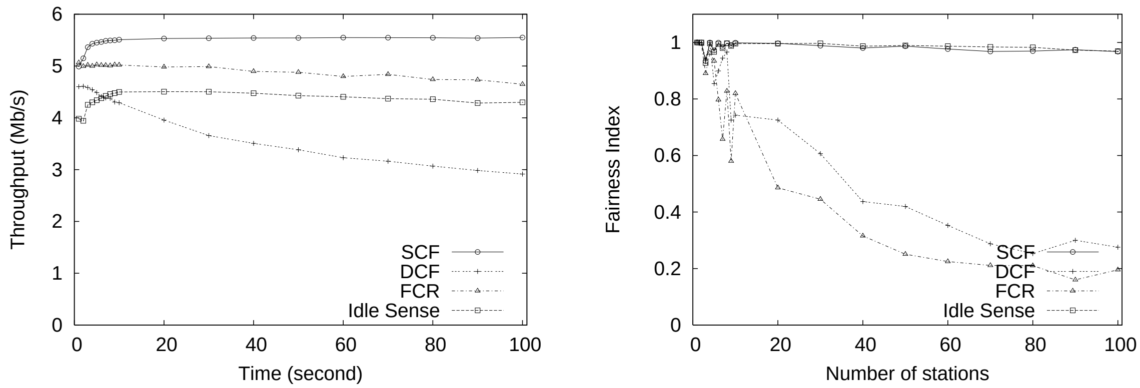
static network with backlogged nodes, the probability that a station enters the ACTIVE station successfully is higher, and as a result, its latency in a lightly loaded network is much smaller than that in the static network.

C. Mobile network with joining and leaving nodes

We investigate the adaptiveness of DCF, FCR, Idle Sense, and SCF in a mobile network, where the number of stations varies due to the joining and leaving of mobile nodes. In the beginning of this simulation, there are initially ten stations, and the number of stations changes to 7 $\rightarrow$ 13 $\rightarrow$ 1 $\rightarrow$ 5 at every ten seconds. In Fig. 11, we see that SCF is able to adapt well to the changes in the number of stations and yields the highest aggregate throughput. During the time interval between 30 and 40 seconds, the number of senders is just one. In this case, SCF regularly transmits frames every $5 \times T_{aSlotTime}$ because N_{JP} is five. However, FCR does every $3 \times T_{SlotTime}$ in this worst case. Therefore, the throughput of FCR is slightly higher than that of SCF during only this time duration.

D. Network with TCP traffic flows

In order to investigate the performance when the traffic is elastic, we simulate the case where packets are transmitted under the TCP New Reno [33]. We assume that the length of packets is 1,460 bytes, and the channel is lossless. Fig. 12 shows that SCF outperforms the other schemes in terms of the aggregate throughput and fairness performance. Compared with Fig. 9(b), Fig.



(a) Aggregate throughput (b) Fairness index

Fig. 12. Simulation results in a network with TCP traffic flows.

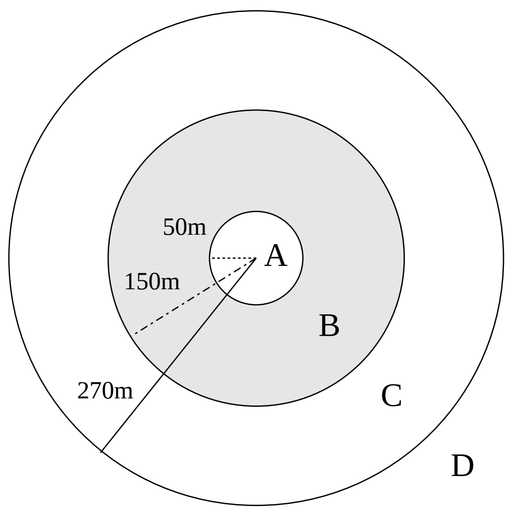


Fig. 13. Network topology with four regions in hidden station scenario.

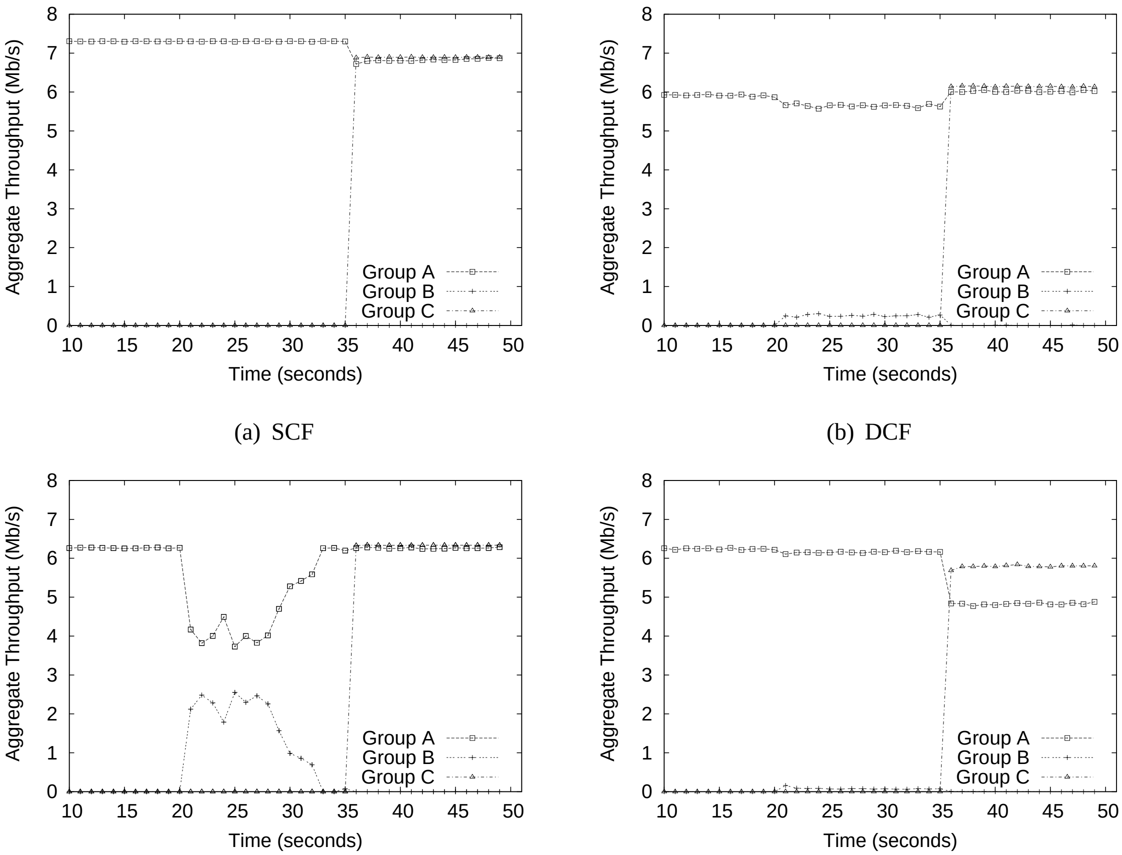
12(b) indicates that the fairness of SCF, FCR, and Idle Sense is improved to some extent with elastic traffic flows, while that of DCF degrades.

E. Network with hidden stations

The performance of SCF may be affected by collisions during a joining period unless all the stations are within the transmission range of each other. Here, we examine the effect of hidden stations to the operation of SCF. We consider a network topology with four regions as shown in Fig. 13. Suppose that the transmission range and the signal sensing range are 100 and 220 meters, respectively. All the stations in the region A (Group A) are within the transmission range of each other, and all the stations in the region B (Group B) are either within the signal sensing range or within the transmission range of some of the stations in Group A. Some stations in the region C (Group C) are within the signal sensing range of some stations in Group A, and the stations in the region D do not interfere with any of the stations in Group A.

In our simulation, the channel is assumed to be lossless in order to concentrate on the effect of hidden stations. We consider the following scenario. In region A, there exist ten stations running each corresponding protocol, and they begin to send frames to their corresponding stations in the beginning of the simulation. After 20 seconds, an interfering station with a simple CSMA/CA protocol in region B wakes up and starts to send frames. After 15 seconds, another interfering station in region C begins to send frames. Note that the station in region C is a hidden station to the stations in region A.

Fig. 14 shows that the aggregate throughput of SCF, DCF, FCR, and Idle Sense. In Fig. 14(a), we observe that the transmissions from stations in Group A are not affected by the station in Group B under SCF, but are affected by the stations in Group C. It is because the stations in Group A leave the network due to the interference from stations in Group C, but they shortly transit from the ACTIVE2 state to the JOIN state to join the network again as shown in Fig. 4. Even in this case, we observe that the throughput decrease due to the hidden station in Group C is not so severe. This result implies that, once a network is operating under SCF, the stations in the network can communicate with each other properly in spite of the interference from the hidden stations. The interfering station in Group B cannot transmit any packet successfully because the CAMA/CA protocol defers its transmission. Fig. 14(b) and 14(d) show the aggregate throughput under DCF and the Idle Sense. The station in Group B can transmit quite a small amount of data at the expense of the throughput decrease at the stations in Group A. In the case of FCR, the network is seriously affected by the station in Group B as shown in Fig. 14(c). Among the four protocols, the Idle Sense experiences the severest throughput decrease due to hidden station in Fig. 14(d). Fig. 15 shows that the fairness performances of SCF, DCF, and Idle Sense are very similar to each other except FCR.



(a) SCF (b) DCF (c) FCR (d) Idle Sense

Fig. 14. Simulation results for aggregate throughput in a network with hidden stations.

VII. Conclusions and Future work

In this paper, we have investigated the efficiency of contention based approaches such as IEEE 802.11 DCF for coordinating the contentions among multiple stations in terms of the MAC protocol overhead and collision occurrences. We have proposed a new MAC mechanism, called sequential coordination function (SCF), which coordinates every station to send a data frame sequentially one after another in a distributed manner, instead of competing with each other station for transmitting frames in order to improve the throughput and fairness performances of WLAN. By defining a service period and a joining period, the SCF eliminates unnecessary contentions during the service period. As a result, the backoff overhead and collision among stations are effectively reduced, and the aggregate throughput and fairness performance are significantly improved.

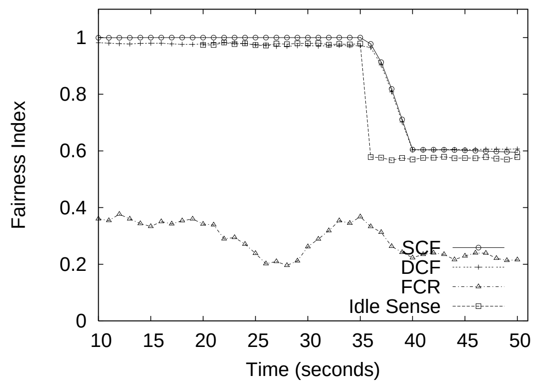


Fig. 15. Simulation results for fairness index in a network with hidden stations.

As future work, we will attempt to apply the SCF to various scenarios. First of all, it would be interesting to study how to support different levels of quality of service (QoS) requirement under the operation of SCF. While the current SCF only guarantees the same level of QoS by providing the users with the same opportunity of transmission, it can be readily extended to support different levels of QoS constraints. Second, we will study how to use the SCF in the infrastructure mode of WLAN, where an access point should have more opportunities to transmit than mobile stations. In addition, it would be a challenge to consider the co-existence issues for near-by multiple SCF–operated networks as well as SCF and CSMA/CA–operated networks.

REFERENCES

[1] C. Lim and C. Choi, "TDM-based Coordination Function (TCF) in WLAN for High Throughput," in *Proceedings of IEEE Globecom 2004*, November 2004.

[2] IEEE, "Part 11: Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specification: Higher-Speed Physical Layer Extension in the 2.4 GHz Band," *IEEE Standard 802.11b-1999*, 1999.

[3] ——, "Part 11: Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specification: Higher-Speed Physical Layer Extension in the 5 GHz Band," *IEEE Standard 802.11a-1999*, 1999.

[4] ——, "Part 11: Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specification: Amendment 4: Further Higher Data Rate Extention in the 2.4 GHz Band," *IEEE Standard 802.11g-2003*, 2003.

[5] ——, "Draft PAR for High Throughput Study Group," *11-02-798r7-HT-SG-Draft-PAR.doc*, March 2003.

[6] Y. Xiao and J. Rosdahl, "Throughput and Delay Limit of IEEE 802.11," *IEEE Communication Letters*, vol. 6, no. 8, pp. 355 – 357, August 2002.

[7] Y. Kim, S. Choi, K. Jang, and H. Hwang, "Throughput Enhancement of IEEE 802.11 WLAN via Frame Aggregation," in *Proceedings of IEEE VTC 2004-Fall*, September 2004.

[8] IEEE, "Part 11: Wireless LAN Medium Access (MAC) and Physical Layer (PHY) Specifications, Amendment 8: Medium Access Cntorl (MAC) Quality of Service Enhancement," *IEEE Standard 802.11e*, 2005.

[9] G. Bianchi, L. Fratta, and M. Oliveri, "Performance Evaluation and Enhancement of of the CSMA/CA MAC Protocol for 802.11 Wireless LANs," in *Proceedings of IEEE PIMRC 1996*, October 1996.

[10] F. Calì, M. Conti, and E. Gregori, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve Theoretical Throughput Limit," *IEEE/ACM Transactions on Networking*, vol. 8, no. 6, pp. 785 – 799, December 2000.

[11] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle Sense: An Optimal Access Method for High Throughput and Fairness in Rate Diverse Wireless LANs," in *Proceedings of ACM SIGCOMM 2005*, August 2005.

[12] Y. Kwon, Y. Fang, and H. Latchman, "A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs," in *Proceedings of IEEE INFOCOM 2003*, April 2003.

[13] D. Qiao and K. G. Shin, "UMAV: A Simple Enhancement to the IEEE 802.11 DCF," in *Proceedings of HICSS 2003*, January 2003.

[14] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," in *Proceedings of ACM MOBICOM 2000*, April 2000.

[15] X. Wang and K. Kar, "Throughput Modelling and Fairness Issues in CSMA/CA Based Ad-Hoc networks," in *Proceedings of IEEE INFOCOM 2005*, March 2005.

[16] Y. C. Tay and K. C. Chua, "A Capacity Analysis for the IEEE 802.11 MAC Protocol," *ACM/Baltzer Wireless Networks*, vol. 7, no. 2, pp. 159 – 171, March 2001.

[17] R. Jain, A. Durresi, and G. Babic, "Throughput Fairness Index: An Explanation," in *ATM Forum/99-0045*, February 1999.

[18] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LAN's," in *Proceedings of ACM SIGCOMM 1994*, September 1994.

[19] Y. Wang and B. Bensaou, "Achieving Fairness in IEEE 802.11 DFWMAC with Variable Packet Lengths," in *Proceedings of IEEE Globecom 2001*, May 2001.

[20] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Fucnition," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535 – 547, March 2000.

[21] Y. Xiao, "Packing Mechanisms for the IEEE 802.11n Wireless LANs," in *Proceedings of IEEE Globecom 2004*, November 2004.

[22] I. Tinnirello and S. Choi, "Efficiency Analysis of Burst Transmissions with Block ACK in Contention-Based 802.11e WLANs," in *Proceedings of IEEE ICC 2005*, May 2005.

[23] C. Bormann, "Robust Header Compression (ROHC) over 802 networks," *IETF draft-bormann-rohc-over-802-01.txt*, February 2005.

[24] L. Bononi, M. Conti, and L. Donatiello, "Design and Performance Evaluation of a Distributed Contention Control (DCC) Mechanism for IEEE 802.11 Wireless Local Area Networks," *Elsevier Journal of Parallel and Distributed Computing*, vol. 60, pp. 407 – 430, April 2000.

[25] H. Kim and J. C. Hou, "Improving Protocol Capacity with Model-based Frame Scheduling in IEEE 802.11-operated WLANs," in *Proceedings of ACM MOBICOM 2003*, September 2003.

[26] P. Gupta, Y. Sankarasubramaniam, and A. Stolyar, "Random-Access Scheduling with Service Differentiation in Wireless Networks," in *Proceedings of IEEE INFOCOM 2005*, March 2005.

[27] X. Yang and N. Vaidya, "A Wireless MAC Protocol Using Implicit Pipelining," *IEEE Transactions on Mobile Computing*, vol. 5, no. 3, pp. 258 – 273, March 2006.

[28] Q. Pang, S. Liew, J. Lee, and V. Leung, “Performance Evaluation of an Adaptive Backoff Scheme for WLAN,” *Wiley Wireless Communications and Mobile Computing*, vol. 4, pp. 867 – 879, June 2004.

[29] C. Wang, B. Li, and L. Li, “A New Collision Resolution Mechanism to Enhance the Performance of IEEE 802.11 DCF,” *IEEE Transactions on Vehicular Technology*, vol. 53, no. 4, pp. 1235 – 1245, July 2004.

[30] M. Ergen, D. Lee, R. Sengupta, and P. Varaiya, “Wireless Token Ring Protocol-Performance Comparison with IEEE 802.11,” in *Proceedings of IEEE ISCC 2003*, June 2003.

[31] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly, “Ordered Packet Scheduling in Wireless Ad hoc Networks: Mechanisms and Performacne Analysis,” in *Proceedings of ACM MOBICOM 2004*, June 2004.

[32] S. J. Golestani, “A Self-Clocked Fair Queueing for Broadband Applications,” in *Proceedings of IEEE INFOCOM 1994*, April 1994.

[33] S. Floyd, T. Henderson, and A. Gurtov, “The NewReno Modification to TCP’s Fast Recovery Algorithm,” *IETF RFC 3782*, April 2004.