We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Efficient Medium Access Control Protocols for Broadband Wireless Communications

Suvit Nakpeerayuth[1], Lunchakorn Wuttisittikulkij[1], Pisit Vanichchanunt[2],
Warakorn Srichavengsup[3], Norrarat Wattanamongkhol[1], Robithoh Annur[1],
Muhammad Saadi[4], Kamalas Wannakong[1] and Siwaruk Siwamogsatham[5]

[1]Chulalongkorn University
[2]King Mongkut's University of Technology North Bangkok
[3]Thai-Nichi Institute of Technology
[4]University of Management and Technology
[5]National Electronics and Computer Technology Center
[1,2,3,5]Thailand
[4]Pakistan

1. Introduction

In wireless communication systems, an efficient medium access control (MAC) protocol is usually required so that multiple wireless stations can efficiently share the scarcely-limited wireless channel. In a typical wireless environment, wireless stations are usually geographically distributed over a service area and are not synchronized. As a consequence, wireless stations are typically required to contend for transmission opportunities. In general, if the MAC protocol is not properly designed, channel contention may cause serious transmission collisions and can considerably degrade the system throughput performance.

Over the past several decades, numerous MAC protocols have been developed to smartly utilize the wireless channel, e.g., ALOHA (Abramson, 1970), carrier-sense multiple access (CSMA) (Kleinrock & Tobagi, 1975; Tobagi & Hunt, 1980), and many other variants (Tasaka & Ishibashi, 1984; Karn, 1990; Frigon, et al., 2001; Amitay & Greenstein, 1994). These conventional MAC protocols have been successfully deployed in practice for different applications and environments, including the widely adopted IEEE 802.11 a/b/g/n wireless local area network systems, the emerging IEEE 802.16 (WiMAX) wireless metropolitan area network, the IEEE 802.15.4 (Zigbee) wireless sensor networks, and various famous MAC protocols for satellite communication networks. In addition, the emerging multimedia technologies in recent years have continuously driven the requirements for higher and higher system transmission throughput. In such an environment, the round trip propagation delays between the base station and wireless stations have increasingly become relatively larger and larger compared with a packet transmission time. As a consequence, a fair deal of recent research work has been directed toward the renewed studies of efficient MAC schemes for systems with relatively large propagation delays.

This chapter overviews the existing MAC technologies and summarizes recent research advancements toward the improvements and analysis of various MAC protocols. In

particular, a class of efficient modified random channel contention and reservation schemes based on our proposed work (Sivamok, et al, 2001; Srichavengsup, et al, 2005) is presented with a complete discussion of mathematical performance evaluation and numerical results.

2. Pure ALOHA

In 1970, Norman Abramson and his colleagues at the University of Hawaii proposed a new medium access control, known as ALOHA or Pure ALOHA, as part of the ALOHA system, that aimed to interconnect a central computer at the university main campus near Honolulu to remote consoles at colleges and research institutes on several islands using UHF radio communications. Two 100 kHz channels at 407.350 MHz and 413.475 MHz are assigned for transmission in each direction, each operating at a bit rate of 24,000 baud. In the ALOHA system, information is transmitted in the form of packets, and all packets are of fixed length, i.e. 88 bytes (8 bytes for header and 80 bytes for data). Therefore, the packet transmission time is about 29 msec and this time becomes 34 msec when information for receiver synchronization is included.

The basic idea of the Pure ALOHA protocol is simple, but elegant: each station is allowed to send its packet whenever it has a packet ready for transmission. Since a common channel is shared among stations, collision between packets from different stations will result when they are sent at nearly the same time. Fig. 1 shows an example of packet transmissions and possible collisions of four stations contending for the same channel. Those packets that are overlapped in time are collided and destroyed. In this example, only two packet transmissions are successful, and the rest of them need to be retransmitted.
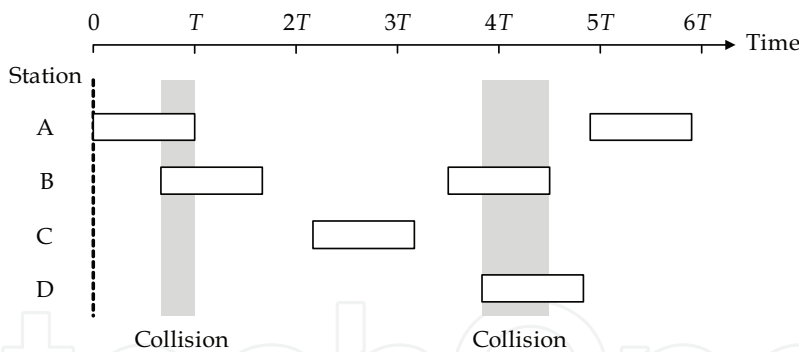

Fig. 1. Packet transmissions in a Pure ALOHA system

After a packet transmission, the sending station waits for an acknowledgement from the receiver to indicate successful transmission of the packet. However, if no acknowledgement is returned within a time-out period, the sending station assumes that the packet is destroyed and starts a retransmission procedure. In principle, the time-out period must be set at least equal to the maximum possible round trip delay between two most widely separated stations to ensure correct functioning of the protocol. Obviously, if the colliding stations try to retransmit their packets immediately, they will collide again. Therefore, each station is required to wait for a random amount of time, called back-off time, before resending the packet. This random back-off mechanism is intended to keep multiple stations from trying to transmit at the same time again which helps reduce probability of collisions.

The back-off time is randomly chosen from the range $[0, 2^k - 1]$ multiplied by the maximum

propagation delay (or alternatively the packet transmission time), where k is the number of previous unsuccessful transmission attempts. This means that the mean value of back-off time is doubled each time the packet is retransmitted. This retransmission is repeated until either the packet is acknowledged or a predetermined number of retransmissions, typically set as 15 attempts, is exceeded.

To see how well such a simple protocol will perform, a throughput analysis for the Pure ALOHA protocol is carried out with the following basic assumptions. There is an infinite number of stations that are generating new packets according to a Poisson process with an average of S packets per packet transmission time. All packets are of equal length and the packet transmission time is T seconds. Packets that fail to reach the intended receivers due to collisions are retransmitted. Since retransmitted packets are vulnerable to collisions too, they will also require retransmission again if not successful. Let us define G as the average number of packets both new and retransmitted combined per packet transmission time. Obviously, G is always greater than or equal to S. It is further assumed that generations of these combined packets during one packet transmission time also follow Poisson distribution. The ratio of S to G is essentially the probability of a successful packet, that is
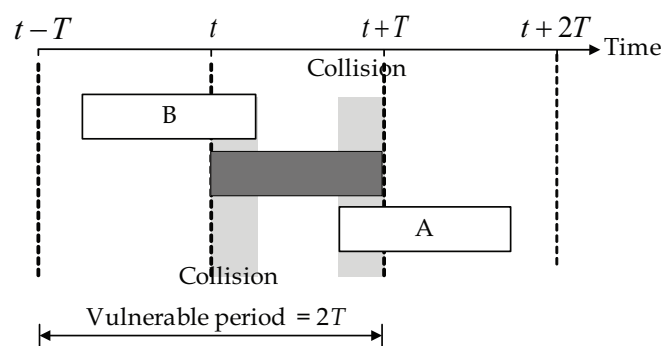
$$P_S = \frac{S}{G} \tag{1}$$

Fig. 2. Vulnerable time for Pure ALOHA

Fig. 2 shows the vulnerable time of a shaded packet, which starts its transmission at time t and finishes at $t+T$. This shaded packet is successfully transmitted, as long as no other packet is transmitted during the interval $t-T$ to $t+T$, so-called vulnerable period. If another packet begins a transmission within the interval $t-T$ to t, such as packet B, the end of this packet will collide with the start of the shaded packet. If another packet begins a transmission within the interval t to $t+T$, such as packet A, the start of this packet will collide with the end of the shaded packet. Based on this observation, it is clear that the shaded packet has a vulnerable period of $2T$, in which if no other packet starts any packet transmission, no collision will occur and the shaded packet will reach the receiver successfully. Therefore, the probability of a successful packet (P_s) in Pure ALOHA is equal to the probability of no generation of packet within $2T$ seconds. Since the probability of k packets are generated within 2 times the packet transmission time according to the Poisson distribution is given by:

$$\Pr[k] = \frac{(2G)^k e^{-2G}}{k!} \qquad (2)$$

the probability of no packet generated is

$$\Pr[k = 0] = e^{-2G} \qquad (3)$$

By combining Equations (1) and (3), we get

$$S = Ge^{-2G} \qquad (4)$$

This relation between G which represents the total offered traffic on the channel and S which represents the throughput of the Pure ALOHA system is plotted in Fig. 3. It shows that initially at low traffic load throughput increases with increasing offered traffic up to a maximum of $1/2e = 0.184$ occurring at a value of $G = 0.5$. A further increase of traffic leads to a higher collision probability due to more intense contention, causing a reduction of throughput.
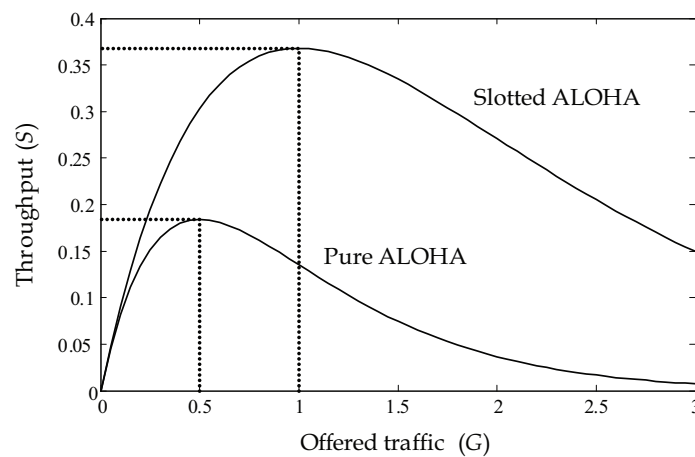

Fig. 3. Throughput versus offered traffic for Pure and Slotted ALOHA

3. Slotted ALOHA

In 1972, Robert introduced a simple modification to Pure ALOHA for improved performance. Time is divided into slots, where each time slot has a fixed size equal to the time required to transmit one packet. Unlike Pure ALOHA, a station is allowed to start a packet transmission only at the beginning of each time slot. If the station has a packet ready to send, it must wait until the beginning of the next time slot. If more than one packet are transmitted in the same slot, they are collided and retransmissions are required. In case of collision, each station involved retransmits its packet in each subsequent slot with probability p until success. Since a packet transmission is confined within the slot boundary, when a collision between packets from different stations occurs, they will overlap completely. This means that the vulnerable period for Slotted ALOHA is reduced by half compared to Pure ALOHA. This modified protocol is commonly known as Slotted ALOHA. Fig. 4 shows an example of packet transmissions and possible collisions in the Slotted ALOHA system. Notice that most packets are generated during a slot interval, and they are

kept waiting until the start of the next slot before transmitted. Indeed, the traffic pattern is deliberately selected to be the same as in Fig. 1 for comparison purpose with Pure ALOHA. Slotted ALOHA appears to reduce collision in this example; only two packets are collided compared to four in case of Pure ALOHA.

Since the throughput of Slotted ALOHA can be analyzed in the same way as Pure ALOHA except that the vulnerable period is now equal to the packet transmission time, the probability of no other packet is sent in the same slot is

$$\Pr[k = 0] = e^{-G} \tag{5}$$

and thus the relation between throughput and offered traffic for Slotted ALOHA can be obtained as
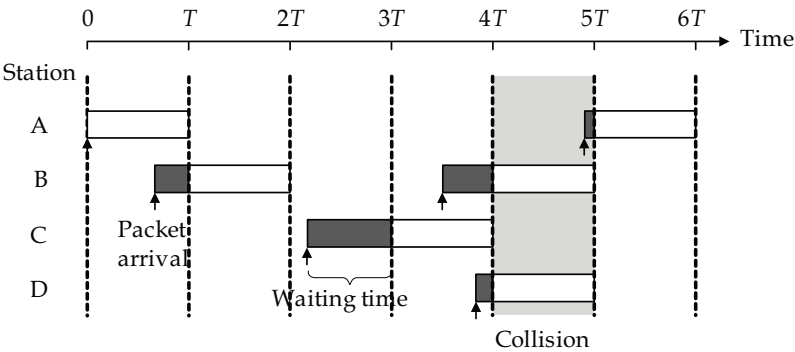
$$S = Ge^{-G} \tag{6}$$

Fig. 4. Packet transmissions in a Slotted ALOHA system

Fig. 3 illustrates the comparison of throughput performance of Pure and Slotted ALOHA. The maximum throughput of Slotted ALOHA is $1/e = 0.368$, which occurs at $G = 1$; this is doubled of that of Pure ALOHA. As we can see, the efficiency of Pure ALOHA can be improved by the introduced time slot structure. However, time synchronization is required to align stations to the slot structure. One possible solution is to have a central station send a kind of clock signal at a regular interval.

Both Pure and Slotted ALOHA have advantageous features. First, they are highly decentralized and quite simple to implement, especially Pure ALOHA. Second, when there is only one active station, the station can continuously transmit its packets at the maximum channel capacity. These two key features make the ALOHA system particularly useful for large population of stations each with light and burst traffic demand. However, due to their simplicity of operation, ALOHA makes inefficient use of channel capacity and is low in throughput performance.

4. Carrier Sense Multiple Access (CSMA)

Pure ALOHA has a shortcoming in that a station still transmits its packet even if the channel is already occupied by another station. Such collisions can be avoided, if only the sending station senses the channel before using it. This led to the development of an important class

of MAC protocols called Carrier Sense Multiple Access (CSMA). A station that wishes to send a packet is required to sense if the channel is busy or idle first. If the channel is sensed busy, the station must wait until the channel becomes idle again before making any transmission. Such a "listen before talk" strategy helps reduce unnecessary packet collisions, thereby increasing channel efficiency. Fig. 5 shows an example of possible packet transmissions in a CSMA system for the same traffic situation as in Fig. 1 of Pure ALOHA. As we can see, each packet waits until the channel becomes idle before transmission and in this particular example, no collisions occur at all; all packets are successfully transmitted.
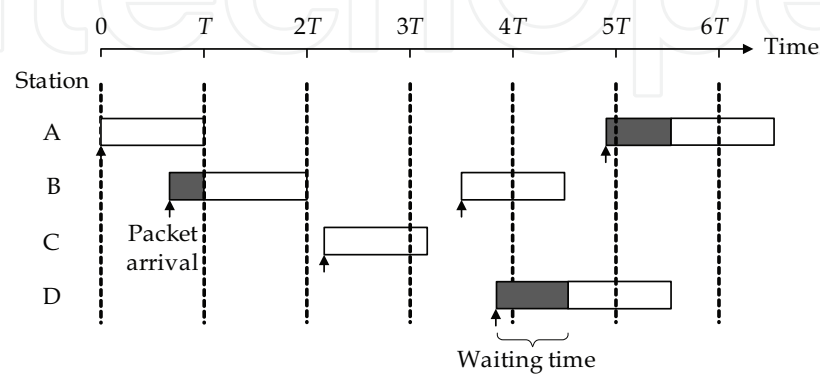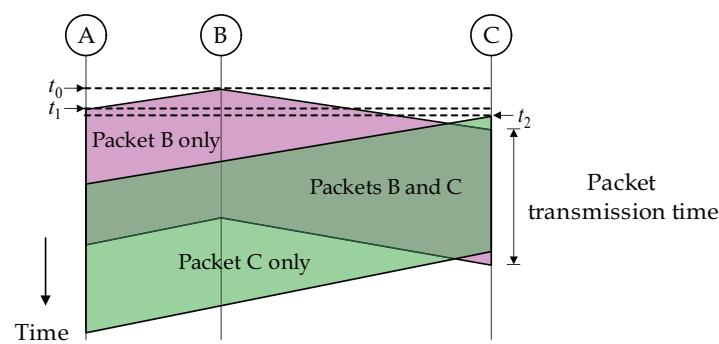
Fig. 5. Packet transmissions in a CSMA system

Fig. 6. An example of a collision in CSMA

Even if the channel is sensed by all stations before their transmissions, collisions can nonetheless occur in CSMA due to propagation delay. That is, when a station transmits a packet, it takes time equal to propagation delay before all other stations detect this transmission. During this period, if another station has a packet ready to send and not yet detect that transmitted packet, it will send its own packet and a collision will result. Fig. 6 shows an example of a possible collision of two packets in CSMA. Station B starts a packet transmission at time t_0. A moment later at time t_1, station A receives the first bit of the packet and thus refrains from transmission. However, at time t_2 station C has a packet ready to send and does not detect any signal on the channel, so it starts a packet transmission, which of course will collide with the packet from station B. Therefore, the vulnerable period of CSMA is equal to the propagation delay, which is the time required for a signal to traverse from one station to another at the opposite end. This means that the smaller the propagation delay between two most widely separated stations gets, the less the

collisions are, and the more the performance improvement can be achieved. Note however that even if the propagation delay is zero, collisions still occur. Consider two or more persistent stations, awaiting the channel to become idle. As soon as the ongoing packet transmission is ended, all persistent stations will transmit their packets immediately, and results in a definite collision.

(*a*) 1-persistent CSMA   (*b*) Nonpersistent CSMA
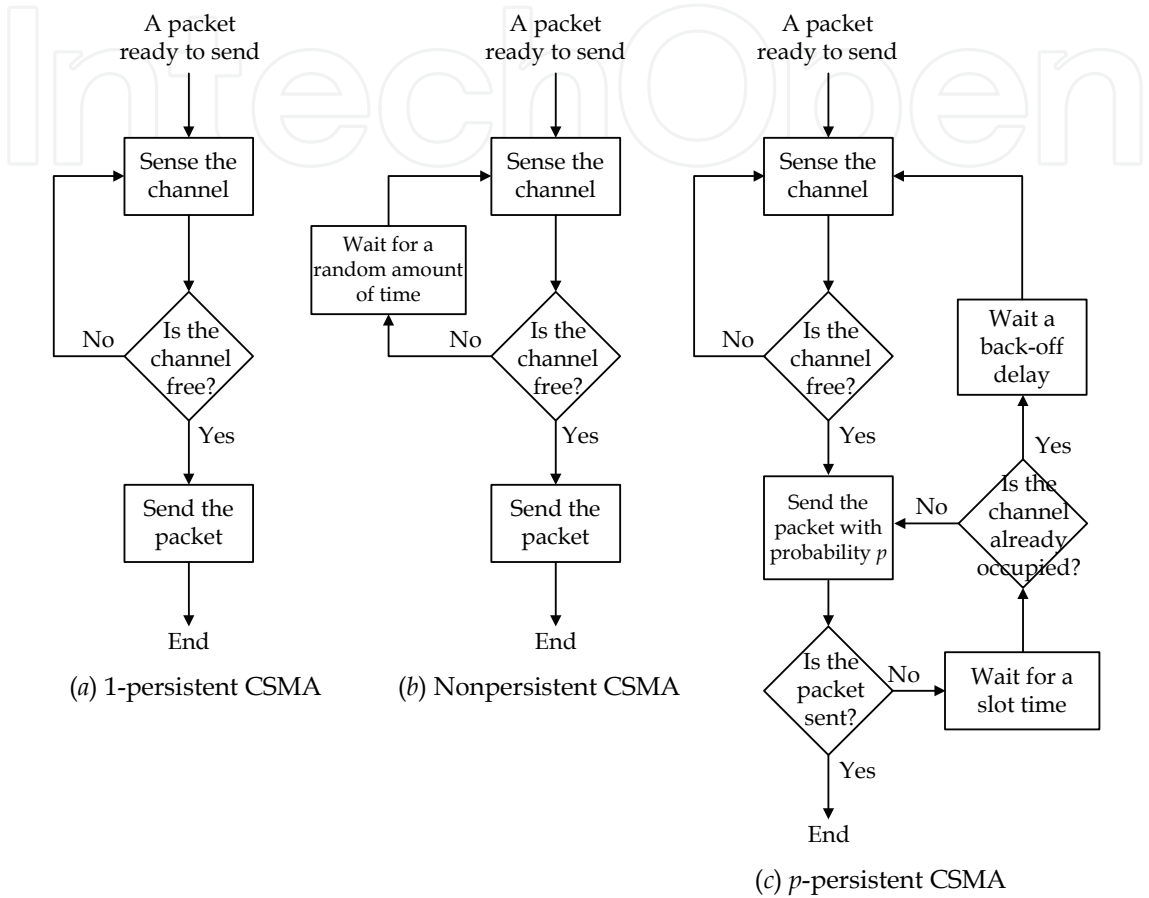
(*c*) *p*-persistent CSMA

Fig. 7. Flow diagrams for CSMA systems

CSMA has several variations which differ in the strategy used in waiting for the channel to become idle. Three most commonly known strategies, namely 1-persistent CSMA, nonpersistent CSMA and *p*-persistent CSMA, are considered below. Note that Fig. 7 shows flow diagrams for these three persistent strategies.

1-persistent CSMA   When a station is ready for a packet transmission, it first senses whether the channel is busy or idle. If the channel is idle, it sends the packet immediately. If the channel is busy, the station keeps on sensing the channel until it becomes idle and then sends the packet immediately. The problem of 1-persistent CSMA is that if two stations have a packet ready to send in the middle of another packet transmission. Both stations will wait until the end of the transmission and start their packet transmissions at the same time, guaranteeing a collision. Thus, 1-persistent CSMA can be perceived as a greedy strategy.

Nonpersistent CSMA   When a station wishes to transmit a packet, it first senses the channel to see if it is idle, if so the station sends the packet immediately. If the channel is busy, instead of continuing to listen for the channel to become idle and transmitting immediately,

it waits a random amount of time, then tries again. In contrast to 1-persistent CSMA, nonpersistent CSMA is much less greedy. Therefore, in high load situations, there is less chance of collisions occurring. On the other hand, in low load conditions, the channel capacity is left unused despite some ready stations.

p-**Persistent CSMA**   When a station has a packet ready to send, it first senses the channel. If the channel is sensed busy, the station keeps on sensing the channel until it becomes idle and uses the following procedure. Note that this same procedure is applied if the channel is sensed idle right from the start. The station transmits its packet with probability p, and delays one time slot with probability $1 - p$, where the duration of a time slot is set equal to or greater than the maximum propagation delay. If the station decides to delay one time slot, it checks whether the skipped slot has been occupied by another station. If it is, the station assumes as if there is a collision and starts its back-off procedure. Otherwise, the station repeats the same procedure as before. That is, it transmits the packet with probability p, and delays one time slot with probability $1 - p$, and so on.

Fig. 8 shows an example of packet transmissions in each of the three CSMA systems for the same packet arriving scenario. Station A is the first to have a packet ready to send and then followed by stations B and C. For 1-persistent CSMA as shown in Fig. 8(a), a packet from station A is transmitted immediately as it arrives because the channel is sensed idle. Packets from stations B and C arrive while the channel is busy, hence they wait until the end of packet A transmission and start their transmissions immediately, resulting a collision. Both of them start a back-off procedure, by delaying their next attempt by a random amount of time. In this example, station C selects a shorter back-off time, so it begins a packet transmission before station B. As a result, when station B wishes to retransmit its packet, the channel is already occupied by station C. So station B waits until the end of transmission and then sends its packet. For nonpersistent CSMA as shown in Fig. 8(b), similar to the previous case packet A is successfully transmitted as it arrives first and finds the channel idle. Packets from stations B and C arrive a little while later when the channel is already occupied by station A, they are rescheduled for later transmission. After a random delay, the packet from C is transmitted. During its transmission, the packet from B tries again, but unfortunately finds the channel busy, so it is rescheduled again. After another random delay later, the packet from B is finally transmitted successfully. For *p*-persistent CSMA as shown in Fig. 8(c), unlike the previous two schemes station A that has a packet ready to send first and finds the channel idle does not transmit its packet immediately. Instead it waits until the beginning of the next slot and makes a decision based on the *p*-persistent CSMA's rule. That is, it transmits its packet with probability p, and delays one time slot with probability $1 - p$. In this example, station A does not send its packet in the first slot, but it does in the second. When packets from stations B and C arrive, the channel is already used by station A, so they wait until the end of the packet transmission. Then the same *p*-persistent CSMA's rule as before is applied. In this example, packet B decides to send its packet in the third slot. Once station C learns at the end of the third slot that the channel is already taken by other station it assumes as if there is a collision and starts its back-off procedure. After a random back-off time later, station C try to retransmit with the same rule and it sends its packet in the second slot. It should be noted that the packet transmission time is assumed to be a multiple integer number of the propagation delay.
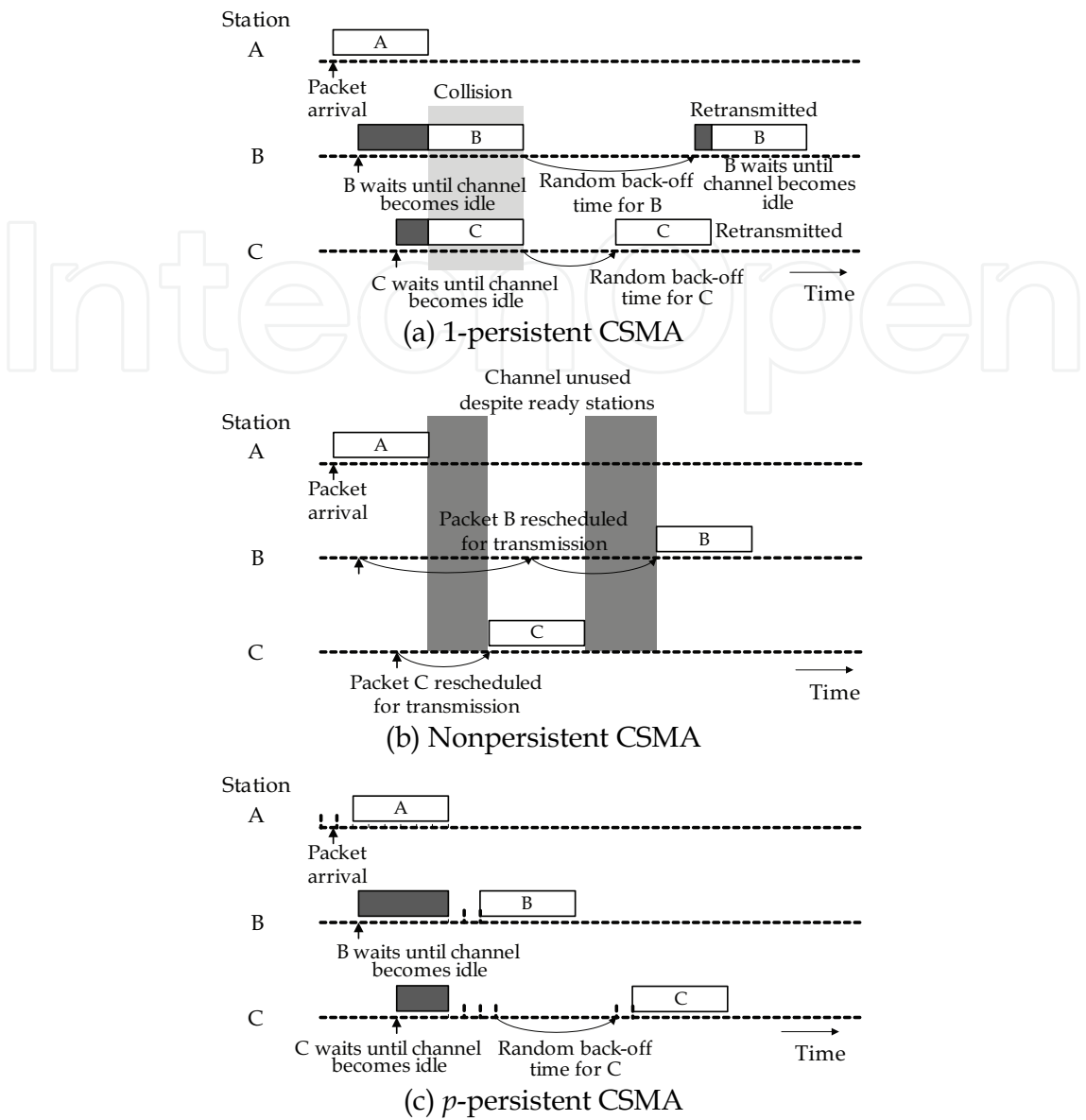
(a) 1-persistent CSMA

(b) Nonpersistent CSMA

(c) *p*-persistent CSMA

Fig. 8. An example of packet transmissions in different CSMA systems

5. Performance analysis of nonpersistent CSMA

Like other CSMA protocols, the nonpersistent CSMA reduces interferences from collision by listening to the channel before packet transmission. If the channel is busy, the stations reschedule the packet transmission to some random time in the future. To analyze the performance, we assume that the offered traffic rate which is the sum of the new arrival rate and the retransmission rate is constant and follows the Poisson point process. The average retransmission delay is also assumed to be large compared to the transmission time of each packet. From Fig. 9, a packet from station A arrives at time t and is immediately sent out through the channel, because the channel is sensed idle. As it takes another τ seconds for the packet to reach other stations, if there are other stations that have a packet ready to send during t to $t + \tau$, then they send their packets, causing inevitable collisions. So, the nonpersistent CSMA has a vulnerable period of τ. In this example, two other stations B and

C each send a packet a moment later with station C being the last station to send during this vulnerable period. As a result, all these three packets need to be retransmited. The duration that there is one or more packet transmitted is referred to as the transmission period (TP). A transmission period can be a successful transmission period or an unsuccessful transmission period depending on whether there is collision or not. The time duration between two TPs will be referred to as an idle period. A cycle of the transmission along the time axis consists of a busy period B and an idle period I, where the busy period can be a successful or unsuccessful TP. Let us define a useful transmission period U as the time duration that the channel carries useful information without collision in a cycle. From the renewal theory, the average channel utilization can be expressed as

$$S = \frac{\bar{U}}{\bar{B} + \bar{I}} \tag{7}$$

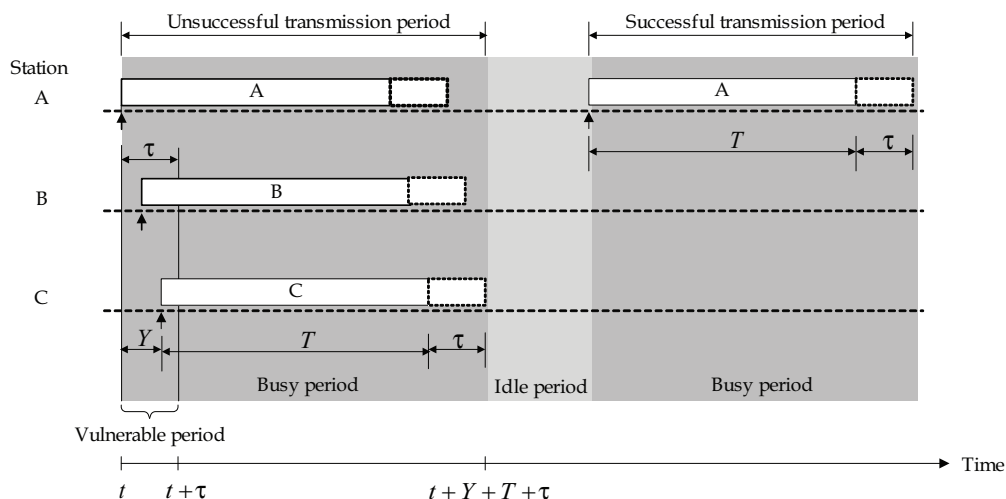where " $\bar{}$ " stands for the average.

Fig. 9. The busy and idle periods of the nonpersistent CSMA

Let T be the packet time and g be the offered traffic rate (the number of packets per second). A TP is successful if there is no other packet transmitted in the vulnerable period $(t, t+\tau)$ and the useful transmission time is T. This occurs with the probability of $e^{-g\tau}$ and we get

$$\bar{U} = Te^{-g\tau} \tag{8}$$

Let $t + Y$ be the time that the last packet arrives in the vulnerable period (which is the packet from station C in Fig. 9). For an unsuccessful transmission period, the busy period B includes a packet time T, Y, and τ which is the time for the last bit of the packet to leave the channel.

$$B = T + Y + \tau \tag{9}$$

and the cumulative distribution function of Y is

$$F_Y(y) = \Pr\{Y \le y\} = \Pr\{\text{no arrival occurs in an interval } (t+Y, t+\tau)\}$$
$$= e^{-g(\tau - y)}, \quad y \le \tau \tag{10}$$

The average of Y is

$$\overline{Y} = \tau - \frac{1}{g}(1 - e^{-g\tau}) \tag{11}$$

Since the mean of inter-arrival time is $1/g$, and it is assumed to be large compared to T, the average of the idle period is

$$\overline{I} = \frac{1}{g} \tag{12}$$

Substituting $\overline{U}$, $\overline{B}$ and $\overline{I}$ into (7), we obtain

$$S = \frac{Te^{-g\tau}}{T + 2\tau - \frac{1}{g}(1 - e^{-g\tau}) + \frac{1}{g}}$$

$$= \frac{gTe^{-g\tau}}{gT(1 + 2\tau/T) + e^{-g\tau}} \quad , \tag{13}$$

$$= \frac{Ge^{-aG}}{G(1 + 2a) + e^{-aG}}$$

where $a = \tau/T$ is the propagation time relative to the packet time and $G = gT$ is the offered traffic rate per packet time.
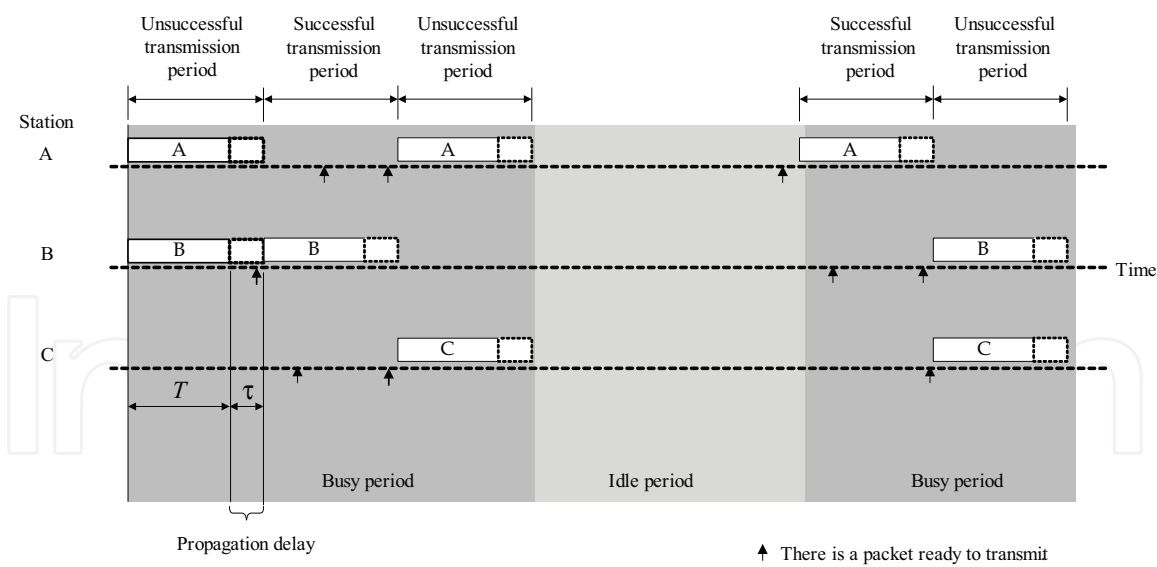

Fig. 10. The busy and idle periods of the slotted nonpersistent CSMA

For slotted nonpersistent CSMA, the time duration of each slot is set to τ and the packet time T is an integer multiple of τ (see Fig. 10 for details). When there is a packet ready to send, each station waits for the beginning of the next slot and senses whether the channel is idle. If so, the packet will be sent otherwise the packet is rescheduled for transmission later on. The probability mass function (PMF) of the idle period is a geometric function of the form

$$\Pr\{I = k\tau\} = (e^{-g\tau})^{k-1}(1 - e^{-g\tau}), \quad k = 1, 2, \dots \tag{14}$$

This gives us

$$\bar{I} = \sum_{k=1}^{\infty} k\tau \Pr\{I = k\tau\} = \frac{\tau}{1 - e^{-g\tau}} \tag{15}$$

In the slotted nonpersistent CSMA, both successful and unsuccessful TPs are $T + \tau$. The busy period B contains k TPs if at least one arrival occurs at the last slot of $(k-1)^{\text{th}}$ TPs, and no arrival occurs at the last slot of the kth TP.

$$\Pr\{B = k(T + \tau)\} = (1 - e^{-g\tau})^{k-1} e^{-g\tau}, \quad k = 1, 2, \dots \tag{16}$$

The average busy period is

$$\bar{B} = \sum_{k=1}^{\infty} k(T + \tau) \Pr\{B = k(T + \tau)\} = \frac{T + \tau}{e^{-g\tau}} \tag{17}$$

The average number of TPs per cycle is $\bar{B} / (T + \tau)$. When the transmission is successful, the useful transmission period is T. The average of the useful transmission period is

$$\bar{U} = T \left(\frac{\bar{B}}{T + \tau} \right) P_{success} \tag{18}$$

where $P_{success}$ is the probability that a TP is successful.

$P_{success} = \Pr\{\text{Only one arrival in the last slot before a TP} \,|\, \text{At least one arrival}$

$\text{in the last slot before a TP}\}$

$$\tag{19}$$

$$= \frac{\Pr\{\text{Only one arrival in the last slot before a TP}\}}{\Pr\{\text{At least one arrival in the last slot before a TP}\}}$$

$$= \frac{g\tau e^{-g\tau}}{1 - e^{-g\tau}}$$

Now we obtain

$$S = \frac{\bar{U}}{\bar{B} + \bar{I}} = \frac{T \left(\frac{(T + \tau) / e^{-g\tau}}{T + \tau} \right) \left(\frac{g\tau e^{-g\tau}}{1 - e^{-g\tau}} \right)}{\frac{T + \tau}{e^{-g\tau}} + \frac{\tau}{1 - e^{-g\tau}}} = \frac{g T \tau e^{-g\tau}}{T + \tau - T e^{-g\tau}} = \frac{g T (\tau / T) e^{-g\tau}}{1 + (\tau / T) - e^{-g\tau}} \,. \tag{20}$$

$$= \frac{aG e^{-aG}}{1 + a - e^{-aG}}$$

If a approaches zero, we obtain

$$\lim_{a \to 0} S = \frac{G}{1 + G} \,. \tag{21}$$

The unity throughput can theoretically be obtained when the offered traffic rate G approaches infinity. The throughputs S versus the offered traffic rate per packet time G of the nonpersistent CSMA and the slotted nonpersistent CSMA are plotted for various values of a in Figs. 11 and 12, respectively.
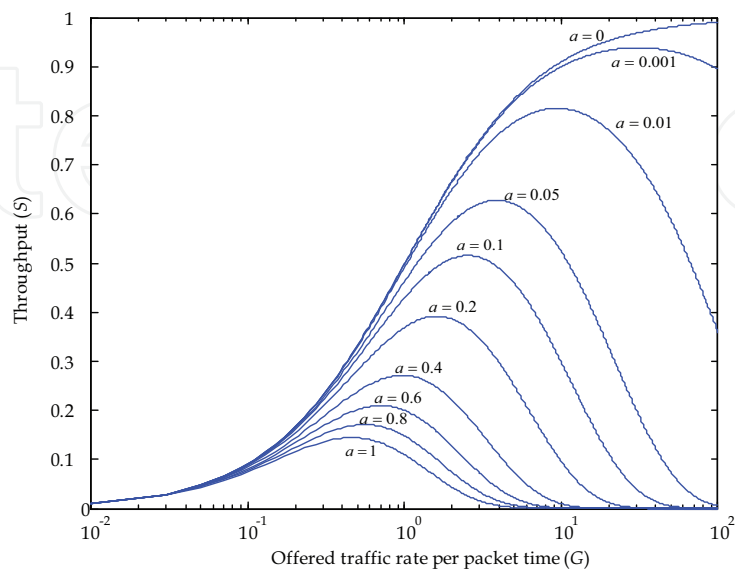

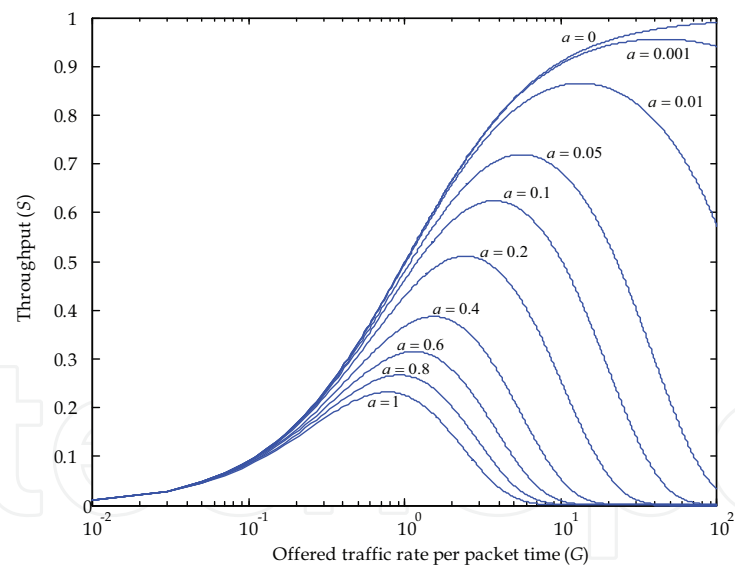Fig. 11. Throughput versus offered traffic for nonpersistent CSMA

Fig. 12. Throughput versus offered traffic for slotted nonpersistent CSMA

6. Proposed channel reservation algorithms

This section presents a class of MAC protocols that organizes the channel bandwidth into a frame structure consisting of two alternate periods, namely contention period and information transfer period, see Fig. 13. The contention period consists of a fixed number of contention slots, which are used by all users to reserve for channel bandwidth on a contention basis. Users who succeed in the reservation process will be assigned data slots by

the base station in the information transfer period for their information transmission. Since the overall system performance very much depends on the efficiency during reservation period, we have proposed a number of efficient channel reservation algorithms to meet the required performance. They include CFP, CAP, COP, COP+SPL, CFP+SPL, UNI, UNI+LA, MT-CFP, MT-CFP+SPL, MT-UNI, MT-UNI+LUA and MT-UNI+LUT. We will explain each algorithm in turn together with their performance analysis.
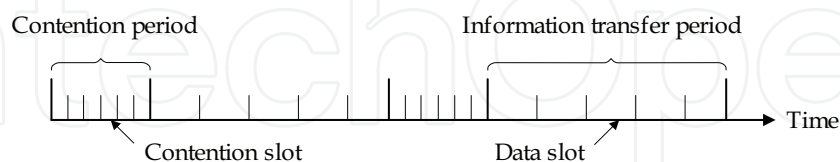

Fig. 13. A frame structure of MAC protocols

6.1 Cascade Fixed Probability (CFP)

In the first algorithm, each user will attempt to make reservation on each contention slot in sequence from the first slot to the last. In each slot, the user will decide whether it will access the present slot with a certain probability (p) and the value of this probability is the same for all users and fixed throughout all contention slots. As a result, this algorithm will be referred to as Cascade Fixed Probability (CFP). It is apparent that the value of probability p is the key parameter to the system performance, hence must be chosen with care. We shall now derive the average number of successful users as a function of the number of active users and the number of available slots.

Let $S[M,N]$ be the average number of successful users for the system with M users and N contention slots and $b[M,i,p]$ be the binomial probability that i out of M users access a particular contention slot with permission probability p, which is expressed as:

$$b[M,i,p] = \binom{M}{i} p^i (1-p)^{M-i}, \text{ where } \binom{M}{i} = \frac{M!}{i!(M-i)!} \tag{22}$$

In each contention slot, only a single user can succeed in reservation, which will occur only when no other users access the slot. A detailed analysis of $S[M,N]$ is formulated in the following recursive formula:

$$S[M,N] = b[M,0,p]S[M,N-1] + b[M,1,p](1+S[M-1,N-1]) + \sum_{i=2}^{M} b[M,i,p]S[M-i,N-1]$$
$$= b[M,1,p] + \sum_{i=0}^{M} b[M,i,p]S[M-i,N-1] \tag{23}$$

where $M \geq 0, N \geq 0$.

The boundary conditions of (23) are $S[a,0] = S[0,b] = 0$ where $a = 0,1,2,....,M$ and $b = 0,1,2,....,N$. We can then find an appropriate permission probability $p_{CFP}[M,N]$ of each frame by differentiating (23) with respect to p, setting it to 0, i.e. $\frac{\partial}{\partial p}S[M,N] = 0$ and determining p that gives the maximum average number of successful users $S_{CFP}[M,N]$.

6.2 Cascade Adaptive Probability (CAP)

In the CFP algorithm, it is seen that an appropriate value of p exists and can be formulated as a function of the number of active users at the start of each frame (M) and the number of slots in each frame (N). It is interesting to further explore this finding to improve the system performance by introducing an idea of adaptive probability. Like the CFP algorithm, all users still use the same value of probability at each slot, but the permission probability may change from one slot to another by considering the current number of remaining users and slots. At the beginning of each contention slot, each user must somehow acquire the present system conditions, i.e. the current number of remaining users and slots. Note that this requirement contradicts with the fundamental system assumption made here. Nevertheless, its analysis provides and interesting new aspect to this study. Once the user knows both parameters the user will choose the value of p based on these values using the formulation derived in the CFP algorithm. Since the permission probability is properly selected in response to the current system scenarios, an improved system performance can be intuitively expected. This algorithm will be known as Cascade Adaptive Probability (CAP). The model for throughput analysis of this algorithm is similar to that of the CFP algorithm, though details may differ.

Let $S_{CAP}[M,N]$ be the average number of successful users of the CAP system with M users and N contention slots and $p_{CFP}[M,N]$ is the optimal permission probability derived from the CFP system with M users and N contention slots. $S_{CAP}[M,N]$ is computed as a recursive formula.

$$\begin{aligned}
S_{CAP}[M,N] &= b[M,0,p_{CFP}[M,N]]S_{CAP}[M,N-1] + b[M,1,p_{CFP}[M,N]](1+S_{CAP}[M-1,N-1]) \\
&\quad + \sum_{i=2}^{M} b[M,i,p_{CFP}[M,N]]S_{CAP}[M-i,N-1] \\
&= b[M,1,p_{CFP}[M,N]] + \sum_{i=0}^{M} b[M,i,p_{CFP}[M,N]]S_{CAP}[M-i,N-1]
\end{aligned} \tag{24}$$

where $M \geq 0, N \geq 0$.

The boundary conditions of (24) are $S[a,0] = S[0,b] = 0$ where $a = 0,1,2,....,M$ and $b = 0, 1,2,....,N$.

6.3 Cascade Optimal Probability (COP)

The adaptive algorithm described above can indeed enhance the system performance, see the comparative results in the next section. However, there exists a more effective way to adapt the permission probability in accordance with the present system status, which can provide truly optimal results. That is, instead of using $p_{CFP}[M,N]$ in the adapting process, the optimal value of permission probability that maximizes $S_{COP}[M,N]$ is determined for given system parameters, i.e., M users and N slots. This algorithm is referred to as Cascade Optimal Probability (COP) and its mathematical analysis is given by:

$$\begin{aligned}
S_{COP}[M,N] &= b[M,0,p_{COP}[M,N]]S_{COP}[M,N-1] + b[M,1,p_{COP}[M,N]](1+S_{COP}[M-1,N-1]) \\
&\quad + \sum_{i=2}^{M} b[M,i,p_{COP}[M,N]]S_{COP}[M-i,N-1] \\
&= b[M,1,p_{COP}[M,N]] + \sum_{i=0}^{M} b[M,i,p_{COP}[M,N]]S_{COP}[M-i,N-1]
\end{aligned} \tag{25}$$

The boundary conditions of (25) are the same as in the CFP system, except that in each step of recursion, the appropriate permission probability $p_{COP}[M,N]$ is selected such that it yields the maximum average number of successful users.

6.4 Cascade Optimal Probability + Split (COP+SPL)

This algorithm is further developed from the COP algorithm. The concept of this algorithm is based on the observations that the success rate for the system with small number of users and slots tends to be superior to the system with an increased number of users and slots with the same factors. As a result, it may be useful and effective to split the number of slots into halves and randomly divide users into two groups. Users in one group will make reservation in the first half of contention slots and users in the other group utilize the second half. Each user determines which group it belongs to by simply flipping a coin. If users can be grouped perfectly, i.e. equally split between the two groups, improvement of the overall system performance will result. However, since users are split in a random manner, it is not known what pattern of grouping will appear. In the worst case half of the slots are heavily loaded with all users while the other half are left totally unused. Under this condition, the overall performance will clearly degrade. The uncertainty in various grouping possibilities raises the concern whether such an idea will really offer benefit or it may actually make things even worse. To answer this problem, we shall derive its performance analytically as follows. It is noted that the number of groups can be set to an arbitrary values, not neccessarily limited to two.

Let g be the number of groups and N/g is the number of contention slots in each group which must be an integer number. The average number of successful users of the COP+SPL system can be expressed as follows:

$$S_{COP+SPL}[M,N] = g\sum_{i=0}^{M}b[M,i,\frac{1}{g}]S_{COP}[i,\frac{N}{g}] \tag{26}$$

6.5 Cascade Fixed Probability + Split (CFP+SPL)

This algorithm can be considered as a simplified version of the previously described COP+SPL algorithm. It functions in the same manner as the COP+SPL except for the permission probability used in each group split. The permission probability for this algorithm is set to a fixed value for all the groups, not optimized for individual group separately as in the COP+SPL algorithm. We shall call this technique as the Cascade Fixed Prob + Split (CFP+SPL) algorithm. The average number of successful users of the CFP+SPL algorithm can be expressed as follows:

$$S_{CFP+SPL}[M,N] = g\sum_{i=0}^{M}b[M,i,\frac{1}{g}]S[i,\frac{N}{g}] \tag{27}$$

where $S[i,\frac{N}{g}]$ is the recursive formula as in (23).

The maximum average number of successful users of the CFP+SPL algorithm can be determined in a similar fashion as in the CFP algorithm. The same boundary conditions as in (23) can be applied here.

6.6 Uniform (UNI)

All four previous algorithms have one feature in common: users consider reservation on each contention slot in sequence. This is a common method adopted in most well-known access control algorithms, as it fits well with the conventional system environment where users can repeatedly make reservation attempts on consecutive contention slots. In some systems, the round trip propagation delay between the base station and users is relatively larger than a packet transmission time. In this situation, users will not receive such a chance, i.e. only a single attempt is possible at each frame. Under this system condition, the order of contention slots becomes irrelevant. Users need not consider each slot in sequence. They may simply select one slot for reservation out of the available slots uniformly. Therefore this technique will be called the Uniform (UNI) algorithm. This UNI algorithm poses some interesting properties. First, the system no longer needs to know the number of active users at the start of each frame, making this algorithm more practical. Second, unlike the previous algorithms where early slots tends to experience greater reservation demands than later slots, all contention slots can now be uniformly loaded and thus better utilized. The average number of successful users can be computed as follow:

$$
\begin{aligned}
S_{UNI}[M,N] &= b[M,0,\frac{1}{N}]S_{UNI}[M,N-1] + b[M,1,\frac{1}{N}](1 + S_{UNI}[M-1,N-1]) \\
&\quad + \sum_{i=2}^{M} b[M,i,\frac{1}{N}]S_{UNI}[M-i,N-1] \\
&= b[M,1,\frac{1}{N}] + \sum_{i=0}^{M} b[M,i,\frac{1}{N}]S_{UNI}[M-i,N-1]
\end{aligned}
\tag{28}
$$

where $M \geq 0, N \geq 0$.

The boundary conditions of (28) are $S[a,0] = S[0,b] = 0$ where $a = 0,1,2,....,M$ and $b = 0, 1,2,....,N$.

6.7 Uniform with Limited Access (UNI+LA)

One problem associated with the *Uniform* algorithm is that it does not take the number of users into account. Accordingly, its performance can be significantly deteriorated when the number of users is relatively much higher than the number of slots available. This is because all users will definitely place a reservation in one of the slots, collision will most likely be hard to avoid. For example, if only two slots are available for ten active users, it is better for most users not to make request. Otherwise, collisions will inevitably take place in both slots. As all users will access the slots, the maximum number of successful users is one and this occurs with a very small chance, i.e. nine users access one slot and one of them accesses the other slot. Clearly the UNI algorithm is not at all effective in this situation. To eliminate such shortcomings, it is essential to find some means to limit the user attempts in accordance with the number of users and available slots. This is achieved by introducing a probability (p) that limits each user in contending for a slot. That is, each user will not access in the current contention period and wait till the next period with probability of $1-p$, and with probability of p users will follow exactly the same step as the Uniform algorithm. We shall refer to this algorithm as Uniform + Limited Access (UNI+LA). The average number of successful users can be derived as follow:

$$S_{UNI+LA}[M,N] = \sum_{i=0}^{M} b[M,i,p]S_{UNI}[i,N] \qquad (29)$$

The boundary conditions of (29) are the same as in the CFP system.

6.8 Multi-Token Cascade Fixed Probability (MT-CFP)

All seven proposed algorithms described earlier have one feature in common, i.e. each user is entitled to make reservation only once in each frame. For the remaining five algorithms, users are no longer limited by such a constraint. It is possible for users to send multiple requests. Nonetheless, as before the outcome of each of their requests do not return immediately. In principle, giving users more chances of making reservations should enable them to achieve greater success. This more flexible mechanism will be referred to as Multi-Token, where the number of tokens defined as T represents the number of accesses allowed per frame. The first Multi-Token algorithm to describe here is MT-CFP (Multi-Token CFP), which is the extension to the CFP algorithm. This MT-CFP is almost the same as the CFP except that each user may repeat reservation attempts as long as the number of attempts remains less than or equal to the predefined number of tokens. In a special case where the number of tokens is set to 1 the MT-CFP becomes the CFP.

Let us first define the following variables which are to be used correspondingly in the mathematical analysis:

T_i = the number of tokens for user i.

B_i = the successful status bit for user i, where "0" means user i has not succeeded yet and "1" means user i has already succeeded.

M = the number of users.

N = the number of reservation slots.

R = the number of remaining users in the system.

In terms of the performance analysis, we can calculate the probability that there are k successful users given the number of tokens T_i, successful status bit B_i and the number of slots N by using the following equation:

$$P_{MT-CFP}[k \mid T_1,T_2,...,T_M,B_1,B_2,...,B_M,N] = (1-p)^R \times P_0 + p(1-p)^{R-1} \times P_1 + \sum_{i=2}^{R} p^i(1-p)^{R-i} \times P_i \quad (30)$$

where
$R = M$ - summation of zero bit of $T_1,T_2,...,T_M$

$$P_0 = P[k \mid T_1,T_2,T_3,...,T_{M-1},T_M,B_1,B_2,B_3,...,B_{M-1},B_M,N-1]$$

$$P_1 = P[k-x \mid T_1-1,T_2,T_3,...,T_{M-1},T_M,B_1+x,B_2,B_3,...,B_{M-1},B_M,N-1]$$

$$\quad + P[k-x \mid T_1,T_2-1,T_3,...,T_{M-1},T_M,B_1,B_2+x,B_3,...,B_{M-1},B_M,N-1]$$

$$\quad + P[k-x \mid T_1,T_2,T_3-1,...,T_{M-1},T_M,B_1,B_2,B_3+x,...,B_{M-1},B_M,N-1]$$

$$\vdots$$

$$\quad + P[k-x \mid T_1,T_2,T_3,...,T_{M-1}-1,T_M,B_1,B_2,B_3,...,B_{M-1}+x,B_M,N-1]$$

$$\quad + P[k-x \mid T_1,T_2,T_3,...,T_{M-1},T_M-1,B_1,B_2,B_3,...,B_{M-1},B_M+x,N-1]$$

$$x = \begin{cases} 0 \ \text{if repeated success} \\ 1 \ \text{if new success} \end{cases}$$

$$\begin{aligned} P_2 = \ & P[k|T_1-1,T_2-1,T_3,T_4,...,T_{M-1},T_M,B_1,B_2,...,B_{M-1},B_M,N-1] \\ & + P[k|T_1-1,T_2,T_3-1,T_4,...,T_{M-1},T_M,B_1,B_2,...,B_{M-1},B_M,N-1] \\ & + P[k|T_1-1,T_2,T_3,T_4-1,...,T_{M-1},T_M,B_1,B_2,...,B_{M-1},B_M,N-1] \\ & \qquad\qquad\qquad\qquad\qquad \vdots \\ & + P[k|T_1,T_2,T_3,...,T_{M-2}-1,T_{M-1}-1,T_M,B_1,B_2,...,B_{M-1},B_M,N-1] \\ & + P[k|T_1,T_2,T_3,...,T_{M-2}-1,T_{M-1},T_M-1,B_1,B_2,...,B_{M-1},B_M,N-1] \\ & + P[k|T_1,T_2,T_3,...,T_{M-2},T_{M-1}-1,T_M-1,B_1,B_2,...,B_{M-1},B_M,N-1] \\ & \qquad\qquad\qquad\qquad\qquad \vdots \\ & \qquad\qquad\qquad\qquad\qquad \vdots \\ & \qquad\qquad\qquad\qquad\qquad \vdots \\ P_R = \ & P[k|T_1-1,T_2-1,...,T_{M-1}-1,T_M-1,B_1,B_2,...,B_{M-1},B_M,N-1] \end{aligned}$$

The boundary conditions are set according to the following:

$$P_{MT-CFP}[k|T_1,T_2,...,T_M,B_1,B_2,...,B_M,N] = \begin{cases} 0 & if \ k<0, T_i \ge 0, B_i \ge 0, N \ge 0 \\ 1 & if \ k=0, T_i \ge 0, B_i \ge 0, N=0 \\ 1 & if \ k=0, T_i=0, B_i \ge 0, N \ge 0 \\ 0 & if \ k>N, T_i \ge 0, B_i \ge 0 \end{cases}$$

The average number of successful users of the MT-CFP system can be expressed as follows:

$$S_{MT-CFP}[M,N,T] = \sum_{k=0}^{M} k \times P_{MT-CFP}[k|T_1,T_2,...,T_M,B_1,B_2,...,B_M,N] \tag{31}$$

6.9 Multi-Token Cascade Fixed Probability with Split (MT-CFP+SPL)

This algorithm is further developed from the MT-CFP algorithm by applying Split mechanism to MT-CFP algorithm. We shall call this technique as the Multi-Token Cascade Fixed Prob + Split (MT-CFP+SPL) algorithm. Let g be the number of groups and N/g be the number contention slots in each group which must be an integer number. In this algorithm, each user randomly chooses any group from g groups with equal probability $1/g$. Then each user will attempt to make a reservation in sequence from the first slot to the last slot in that group until there is no remaining token for making a request. The average number of successful users of the MT-CFP+SPL algorithm can be expressed as follows:

$$S_{MT-CFP+SPL}[M,N,T] = g\sum_{i=0}^{m} b[M,i,\frac{1}{g}]S_{MT-CFP}[i,\frac{N}{g},T] \tag{32}$$

The boundary conditions of (32) are the same as in the MT-CFP system.

6.10 Multi-Token Uniform (MT-UNI)

This algorithm is an alteration of UNI by applying the Multi-Token mechanism which allows each user to randomly choose any T slots from N slots for reservation with equal probability. Therefore, we can calculate $P[k|M,N,T]$, the probability that k successful users given the number of users M, the number of slots N and the number of tokens T by using the following equation:

$$P[k|M,N,T] = \frac{C[k|M,N,T]}{\sum_{k=0}^{M} C[k|M,N,T]}. \tag{33}$$

$C[k|M,N,T]$ is now given as the number of cases that k successful users given the number of users M, the number of slots N and the number of tokens T.

$\sum_{k=0}^{M} C[k|M,N,T]$ is now given as the number of all cases that each of M users uses T tokens to reserve T slots from N slots. Then

$$\sum_{k=0}^{M} C[k|M,N,T] = \left[N \cdot (N-1) \cdot (N-2) \cdot \ldots \cdot (N-(T-1)) \right]^{M}. \tag{34}$$

The average number of successful users of the MT-UNI system can be expressed as follows:

$$S_{MT-UNI}[M,N,T] = \sum_{k=0}^{M} k \times P[k|M,N,T] \tag{35}$$

6.11 Multi-Token Uniform with Limited User's Access (MT-UNI+LUA)

The MT-UNI algorithm described earlier can become ineffective when there are high traffic. To improve the MT-UNI performance, we introduce Limited Access (LA) mechanism to the MT-UNI algorithm. In this paper, we propose 2 types of LA for MT-UNI. The first one is Limited User's Access (LUA) mechanism. We use LUA to limit the user attempts by introducing a probability (p). Users that find themselves not to access the slots will do nothing whereas other users will follow exactly the same step as the Multi-Token Uniform algorithm. This technique is referred to as Multi-Token Uniform + Limited User's Access (MT-UNI+LUA). The average number of successful users of the MT-UNI+LUA system can be expressed as follows:

$$S_{MT-UNI+LUA}[M,N,T] = \sum_{i=0}^{M} b[M,i,p] \, S_{MT-UNI}[i,N,T] \tag{36}$$

We can identify the appropriate permission probability $p_{MT-UNI-LUA}[M,N]$ of the MT-UNI+LUA algorithm by differentiating (36) with respect to p, setting it to 0, and finding p that gives maximum average number of successful users $S_{MT-UNI+LUA}[M,N,T]$.

6.12 Multi-Token Uniform with Limited User's Token (MT-UNI+LUT)

Another type of LA is Limited User's Token (LUT) mechanism. If we choose LUT mechanism for the MT-UNI algorithm, we shall refer to this algorithm as Multi-Access

Uniform + Limited User's Token (MT-UNI+LUT). In this algorithm, each user's token will be decided to use or not by a probability (p). As a result, each user has a limited use of his available number of tokens. The value of p certainly plays an important role to the system performance and we will now illustrate how the optimal value of p can be analytically determined.

The average number of successful users of the MT-UNI+LUT system can be expressed as follows:

$$S_{MT-UNI+LUT}[M,N,T] = \frac{\sum_{k=0}^{M} k \sum_{u=0}^{MT} (p^u(1-p)^{MT-u})C[k,u|M,N,T]}{\sum_{k=0}^{M} \sum_{u=0}^{MT} (p^u(1-p)^{MT-u})C[k,u|M,N,T]} \qquad (37)$$

$C[k,u|M,N,T]$ is now given as the number of cases that k successful users use u tokens from all of user's tokens (MT) given the number of users M, the number of slots N and the number of tokens for each user T.

We can identify the appropriate permission probability $p_{MT-UNI+LUT}[M,N,T]$ of the MT-UNI+LUA algorithm by differentiating (37) with respect to p, setting it to 0, and finding p that gives maximum average number of successful users $S_{MT-UNI+LUT}[M,N,T]$.

The relationship among CFP, CAP, COP, COP+SPL, CFP+SPL, UNI, UNI+LA, MT-CFP, MT-CFP+SPL, MT-UNI, MT-UNI+LUA and MT-UNI+LUT protocols can be shown in Fig. 14.
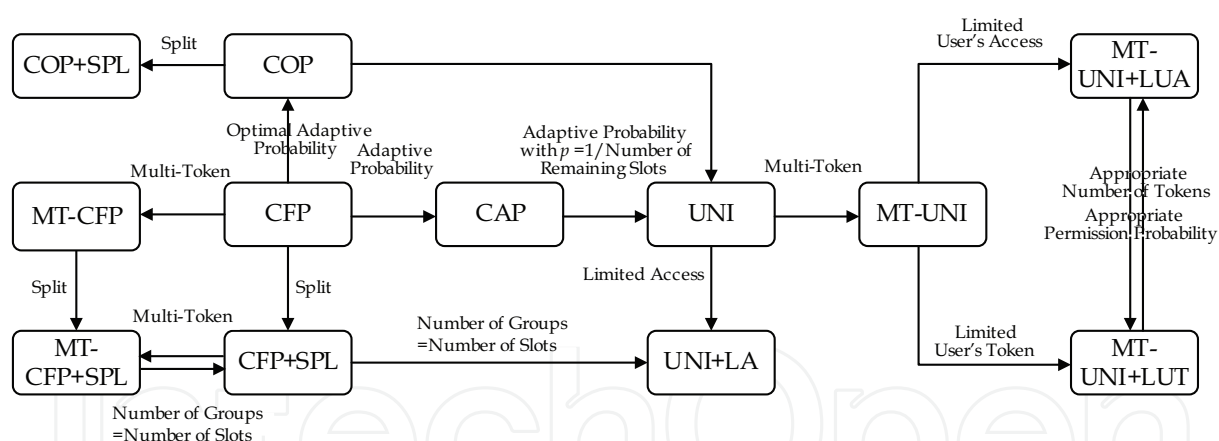

Fig. 14. The relationship among CFP, CAP, COP, COP+SPL, CFP+SPL, UNI, UNI+LA, MT-CFP, MT-CFP+SPL, MT-UNI, MT-UNI+LUA and MT-UNI+LUT protocols

The required information for each algorithm can be illustrated in Table I, it appears that only the CFP, CFP+SPL, MT-CFP, MT-CFP+SPL, UNI, UNI+LA, MT-UNI, MT-UNI+LUT and MT-UNI+LUA algorithms are practically applicable to the system assumption that the base station can obtain the number of active users at the start of each reservation period. Other algorithms CAP, COP and COP+SPL require additional information, i.e. the number of remaining users at each contention slot or the number of users in each split group. Such information is hard to acquire instantly in the system where the round trip propagation delay between the base station and users is relatively larger than a packet transmission time. Therefore, these algorithms will not be practical in this situation.

Table II shows the applied mechanisms of each algorithm, for example, starting with the CAP algorithm, it can be seen that CAP applies Adaptive Probability. The COP and COP+SPL algorithms employ the mechanism namely Optimally Adaptive Probability. The applied mechanisms for other algorithm are fully illustrated in Table II.

Algorithm	Know the number of active users at the start of each reservation period	Know the number of remaining users at each contention slot
CFP	√	
CAP	√	√
COP	√	√
CFP+SPL	√	
COP+SPL	√	√
MT-CFP	√	
MT-CFP+SPL	√	
UNI	√	
UNI+LA	√	
MT-UNI	√	
MT-UNI+LUT	√	
MT-UNI+LUA	√	

Table 1. The Required Information for Each Algorithm

Algorithm	Multi-token	Split	Adaptive Probability	Optimally Adaptive Probability
CFP				
CAP			√	
COP				√
CFP+SPL		√		
COP+SPL		√		√
MT-CFP	√			
MT-CFP+SPL	√	√		
UNI				
UNI+LA				
MT-UNI	√			
MT-UNI+LUT	√			
MT-UNI+LUA	√			

Table 2. The Applied Mechanisms of Each Algorithm

6.13 Numerical results

All results given here are obtained from the mathematical formulations described in the previous sections. We shall first illustrate how the permission probability has an effect on the system performance, which is measured in terms of the average number of successful users. The CFP algorithm is specifically selected for discussion, as it is ideal for this purpose.

By using equation (23), it is possible to obtain a relation between the average number of successful users and the permission probability p; this is depicted in Fig. 15. In this figure, the number of slots N is fixed at 16 and the total number of users M varied from 1 to 16. As we can see, at small values of permission probability the average number of successful users increases with the permission probability. This is simply because under this condition users do not access the contention slots frequently enough; a lot of time these slots are idle. Therefore, an increase in the permission probability will reduce the number of idle slots and thus improving the system throughput. When increasing the permission probability up to a certain value, the number of successful users begins to decline. This performance degradation is due to an increase in the number of collisions caused by too many accessing attempts. A further increment of the permission probability beyond this will only generate more collisions and results in the reduction of the number of successful users.

Fig. 15. The average number of successful users vs the permission probability with $N = 16$ for CFP

All of the above investigations indicate that the permission probability is a key factor to the system performance and to determine an appropriate permission probability it is essential to take account of both the total number of users and the number of slots available into consideration simultaneously. Notice that when the number of contention slots is large, the appropriate permission probability tends to be small and will approach zero in the extreme case where the number of slots is infinite. This is because when there are an increased number of contention slots, users gain greater opportunity for access. Therefore, they can access using the lower permission probability to avoid collision. In other words, in the system with a small number of contention slots, the users must attempt to increase their success opportunity by increasing their permission probability.

Fig. 16 illustrates the performance comparison of the CFP, CFP+SPL, MT-CFP and MT-CFP+SPL algorithms. These numerical results are obtained by using the appropriate number of tokens and appropriate permission probability. It is clear that MT-CFP algorithm generally performs better at small number of users. On the other hand, in case of heavy loads the CFP+SPL with 16 groups and MT-CFP+SPL with 16 groups offer relatively superior performance. Moreover, it can be noticed that at the large number of users the performance of MT-CFP algorithm is equal to the performance of CFP algorithm. This is because at the large number of users, the best value of the number of tokens is equal to 1.
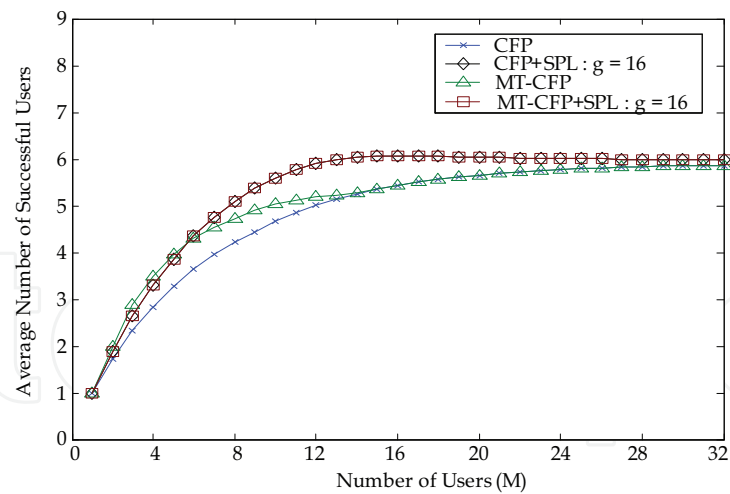
Fig. 16. The average number of successful users vs the number of users (M) with $N = 16$ using the appropriate probability of limitation and appropriate number of tokens for CFP, CFP+SPL, MT-CFP and MT-CFP+SPL

Fig. 17 illustrates the performance comparison of the UNI, UNI+LA, MT-UNI, MT-UNI+LUA and MT-UNI+LUT algorithms. These numerical results are obtained by using the appropriate probability of limitation and the appropriate number of tokens. It can be noticed that under the light load condition, when the number of users is not more than the number of slots divided by 2, the average number of successful users of MT-UNI algorithm is comparatively equal to MT-UNI+LUT and MT-UNI+LUA algorithms and the average number of successful users of UNI algorithm is comparatively equal to UNI+LA algorithm. This is because at the small number of users, the appropriate probability of limitation is equal to 1. In case of heavy load condition, when the number of users is more than the number of slots, the average number of successful users of UNI algorithm is comparatively equal to MT-UNI algorithm and the average number of successful users of UNI+LA algorithm is comparatively equal to MT-UNI+LUT and MT-UNI+LUA algorithms. This is because at the large number of users, the best value of the number of tokens is equal to 1. In this case, limiting the number of user's token is the same meaning as limiting the user's access.
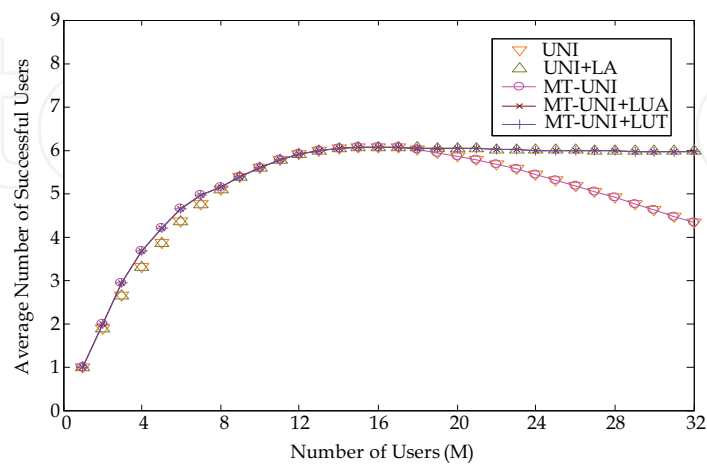

Fig. 17. The average number of successful users vs the number of users (M) with $N = 16$ using the appropriate probability of limitation and appropriate number of tokens for UNI, UNI+LA, MT-UNI, MT-UNI+LUT and MT-UNI+LUA
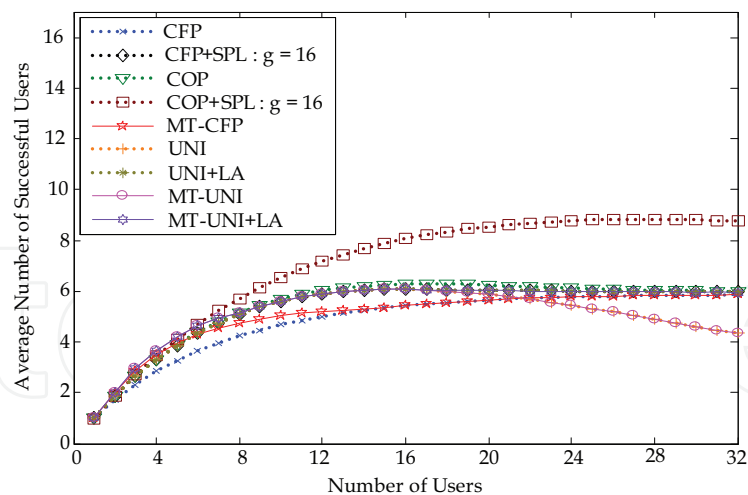
Fig. 18. The number of successful users vs the number of users with $N = 16$

From the above results, it can be noticed that when using the appropriate probability of limitation and appropriate number of tokens the MT-UNI+LUT algorithm is completely identical to the MT-UNI+LUA algorithm under any load condition. Thus, we shall call MT-UNI+LUT and MT-UNI+LUA algorithms as the Multi-Token Uniform + Limited Access (MT-UNI+LA) algorithm for the following discussion.

The performance comparison of all algorithms is depicted in Fig. 18. It is clear that the MT-CFP, MT-UNI and MT-UNI+LA algorithms are effective at systems with light to medium loads. In case of heavy load condition, the COP+SPL algorithm offers relatively superior performance.
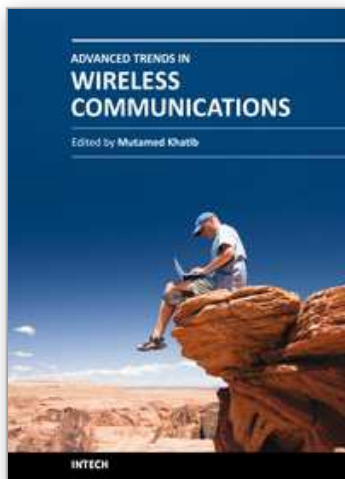
7. Conclusions

In this chapter, several well known MAC protocols for the wireless networks are overviewed such as ALOHA, slotted ALOHA, CSMA including 1-pesistent, non-persistent, and p-persistent. Performance analyses for some of these MAC protocols are given in details. Due to the nature of randomness in ALOHA systems, packets can easily collide. In order to minimize collisions, carrier sensing technique, i.e. stations monitor the channel status before transmission, can be applied to improve the throughput performance. In addition, a class of MAC protocols that organizes the channel bandwidth into a frame structure consisting of two alternate periods, namely contention period and information transfer period, are presented. For contention period, we have proposed a number of efficient channel reservation algorithms, namely CFP, CAP, COP, COP+SPL, CFP+SPL, UNI, UNI+LA, MT-CFP, MT-CFP+SPL, MT-UNI, MT-UNI+LUA and MT-UNI+LUT, which are designed for systems where the round trip propagation delays between the base station and wireless stations is relatively larger than the packet transmission time. Mathematical analyses of these algorithms are described and some numerical results are given to compare their performance.

Due to many newly emerging wireless applications, such as entertainment applications, interactive games, medical applications and high speed data transmission, the global demand for multimedia services such as data, speech, audio, video, and image are growing at rapid pace. Future MAC protocols are therefore required not only to handle high speed transmission, but also support various different Quality of Services (QoS). In addition,

misbehaviors at the MAC layer, such as DoS attack, have become another concern, as it can potentially cause serious damages to the entire networks. Much ongoing research work in the literature has also been active toward these emerging directions.

8. References

Abramson, N. (1970). The ALOHA System - Another Alternative for Computer Communications. *AFIP Conf. Proc Fall Joint Computing Conf.*, pp. 281–285, 1970.

Amitay, N. & Greenstein, L. J. (1994) Resource Auction Multiple Access (RAMA) in the Cellular Environment. *IEEE Trans. Veh. Technol.*, Vol. 43, No. 4, (January 1994) pp. 1101–1111.

Frigon, J. F.; Leung V.C.M. & Chan, H.C.B. (2001) Dynamic Reservation TDMA Protocol for Wireless ATM networks. *IEEE J. Select. Areas Commun.*, Vol. 19, No. 2, (February 2001) pp. 370–383.

Karn, P. (1990) MACA: a new channel access method for packet radio. *Proceedings of the ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pp. 134-140, September 1990, Ontario, Canada.

Kleinrock, L. & Tobagi, F. A. (1975). Packet switching in radio channels: part I-carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Trans. on Commun.*, Vol. COM-23, No. 12, (December 1975) pp. 1400–1416.

Sivamok, N.; Wuttisttikulkij, L. & Charoenpanitkit, A. (2001). New channel reservation techniques for media access control protocol in high bit-rate wireless communication systems. *IEEE Proc. of Globecom*, vol.6, pp. 3558–3562, 2001.

Srichavengsup, W.; Sivamok, N.; Suriya, A. & Wuttisttikulkij, L. (2005). A design and performance evaluation of a class of channel reservation techniques for medium access control protocols in high bit-rate wireless communications. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E88-A, No.7, (July 2005) pp. 1824–1835.

Tasaka, S. & Ishibashi, Y. (1984) A Reservation Protocol for Satellite Packet Communication – A Performance Analysis and Stability Considerations. *IEEE Trans. Wireless Commun.*, Vol. COM-32, No. 8, (Aug. 1984) pp. 920–927.

Tobagi, F. A. & Hunt, V.B. (1980) Performance analysis of carrier sense multiple access with collision detection. *Comput. Netw.*, Vol. 4, (October/November 1980) pp.245–259.

Yang, Y. & Yum, T-S. P., Delay distributions of slotted ALOHA and CSMA. *IEEE Trans. on Commun.*, Vol. 51, No. 11, (November 2003) pp. 1846–1858.

Advanced Trends in Wireless Communications

Edited by Dr. Mutamed Khatib

Physical limitations on wireless communication channels impose huge challenges to reliable communication. Bandwidth limitations, propagation loss, noise and interference make the wireless channel a narrow pipe that does not readily accommodate rapid flow of data. Thus, researches aim to design systems that are suitable to operate in such channels, in order to have high performance quality of service. Also, the mobility of the communication systems requires further investigations to reduce the complexity and the power consumption of the receiver. This book aims to provide highlights of the current research in the field of wireless communications. The subjects discussed are very valuable to communication researchers rather than researchers in the wireless related areas. The book chapters cover a wide range of wireless communication topics.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Suvit Nakpeerayuth, Lunchakorn Wuttisittikulkij, Pisit Vanichchanunt, Warakorn Srichavengsup, Norrarat Wattanamongkhol, Robithoh Annur, Muhammad Saadi, Kamalas Wannakong and Siwaruk Siwamogsatham (2011). Efficient Medium Access Control Protocols for Broadband Wireless Communications, Advanced Trends in Wireless Communications, Dr. Mutamed Khatib (Ed.), ISBN: 978-953-307-183-1, InTech, Available from: http://www.intechopen.com/books/advanced-trends-in-wireless-communications/efficient-medium-access-control-protocols-for-broadband-wireless-communications

INTECH
open science | open minds