Wireless Netw (2009) 15:87–98
DOI 10.1007/s11276-007-0023-z

Performance of a burst-frame-based CSMA/CA protocol: Analysis and enhancement

Kejie Lu · Jianfeng Wang · Dapeng Wu · Yuguang Fang

Published online: 16 February 2007

Abstract In this paper, we develop an analytical model to evaluate the delay performance of the burst-frame-based CSMA/CA protocol under unsaturated conditions, which has not been fully addressed in the literature. Our delay analysis is unique in that we consider the end-to-end packet delay, which is the duration from the epoch that a packet enters the queue at the MAC layer of the transmitter side to the epoch that the packet is successfully received at the receiver side. The analytical results give excellent agreement with the simulation results, which represents the accuracy of our analytical model. The results also provide important guideline on how to set the parameters of the burst assembly policy. Based on these results, we further develop an efficient adaptive burst assembly policy so as to optimize the throughput and delay performance of the burst-frame-based CSMA/CA protocol.

Keywords High data rate · MAC · CSMA/CA · Unsaturated · Throughput · Delay · Performance · Analysis

1 Introduction

In the past decade, wireless ad hoc networks, particularly wireless *local area networks* (WLANs) have been widely deployed and been studied extensively in both academia and industry. With the advances in wireless communication technologies such as *multi-input multi-output* (MIMO) [1] and *Ultra-Wide Band* (UWB) [2], the next generation wireless ad hoc networks are able to provide high data rate (>100 Mb/s) in the physical layer [3–5]. To efficiently utilize these high data rate, medium access control (MAC) protocols must be carefully designed. In this paper, we will focus on MAC protocols that are based on *carrier sense multiple access with collision avoidance* (CSMA/CA), since CSMA/CA is the most popular MAC scheme and has been standardized in IEEE 802.11.

In high data rate wireless ad hoc networks, the throughput of MAC protocols is significantly limited by the overhead, which becomes more serious with the increase of the physical layer data rate. In CSMA/CA, the overhead includes collision, control messages, backoff, and various inter-frame-spacing. To reduce these overheads, a common solution is to transmit multiple packets in a burst, instead of transmitting them one by one [6–9].

Since the burst-frame-based protocol is expected to become the essential component of MAC schemes in the next generation wireless networks, it is crucial to analyze the performance of the protocol under various traffic conditions. In our previous study [10], we have developed an unsaturated throughput analysis for a burst-frame-based CSMA/CA protocol, which shows that the proposed protocol can significantly improve the throughput performance by reducing overheads.

In this paper, we further study the delay performance of the burst-frame-based MAC protocol, since the burst assembly procedure may also introduce extra packet delay, which is undesirable for many applications. In addition, we address how to achieve the optimum throughput and delay performance through appropriately setting the burst assembly policy.

K. Lu (✉)
Department of Electrical and Computer Engineering, University of Puerto Rico at Mayagüez, Mayagüez, PR 00681, USA
e-mail: lukejie@ece.uprm.edu

J. Wang
Wireless Communications and Networking Department, Philips Research North America

D. Wu · Y. Fang
Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL-32611, USA

In our delay analysis, unlike all existing studies, we define the *end-to-end delay* as the time duration from the time epoch that a packet enters the queue at the source node to the epoch that the packet is successfully received. This means, the *queueing delay* is included in our analysis while only *successfully received* packets are considered. To the best of our knowledge, such kind of delay analysis has never been theoretically investigated in the literature. Extensive simulation and numerical results show that, the proposed analytical model is quite accurate in most cases. We also observe that, with an appropriate setting of the burst assembly policy, the burst-frame-based MAC protocol can significantly improve both the throughput performance and the average end-to-end delay performance. These results motivate us to develop an adaptive burst assembly policy such that the delay and throughout performance can both be optimized.

The rest of the paper is organized as follows. We first provide an overview of related works in Section 2. We then briefly describe the burst-frame-based MAC protocol in Section 3. In Section 4, we analyze the unsaturated delay performance of the proposed MAC protocol. Simulation and numerical results are shown in Section 5, followed by the enhanced burst assembly policy in Section 6. Finally, Section 7 concludes the paper.

2 Related works

2.1 Performance analysis for CSMA/CA

The delay performance of CSMA/CA protocols, particularly IEEE 802.11, has been analyzed in several recent studies [11–17]. In [11, 12], the authors derived the average MAC service time under saturated traffic condition, in which the MAC service time is the duration from the epoch that the packet is to be transmitted to the epoch that the transmission attempt is finished, regardless whether the packet is received or not (a packet may be dropped at the transmitter side after the transmitter has tried a certain number of transmissions). In addition, their model does not consider the queueing delay at the transmitter side and only applies to the saturated condition.

In Chen et al. [13], proposed an approach to calculate the MAC service time under unsaturated condition. Their analysis is based on the assumption that the MAC service system does not depend on the queue status. Although this assumption can simplify the analysis, it may not be valid in practice. To calculate the average delay of the queue, [13] uses a classic M/G/1 model, in which the mean and variance of the MAC service time are required. To avoid the complexity of analysis, the authors obtained the variance of MAC service time through simulation.

Recently, [14] developed an approximate model to evaluate the queue behavior of IEEE 802.11. The model is based on a G/G/1 queue; but an M/M/1 model is applied to estimate the probability that a node is busy. Consequently, the analytical results have a large derivation from the simulation results. A better approximation of the probability that a node is busy was provided in [15]; however, the results are still not accurate under moderate and high traffic load. Note that both the M/G/1 model and the G/G/1 model assume that the queue size is infinite, which may not be realistic and can lead to an infinite delay if the incoming traffic load is high (even if the load/utilization is much less than 1).

The models in [16] and [17] are similar in that both of them are based on the M/G/1/K queueing model. The difference between them is how to achieve the MAC service time distribution. Specifically, [17] uses a Markov-modulated general -distribution to model the service time distribution, while in [16] the service time distribution is directly calculated through a transfer-function approach.

To summarize the unsaturated delay analysis in existing works, we first note that the delay performance under burst-frame-based CSMA/CA protocol has not been addressed. Moreover, we notice that the delay in these analyses is defined as the duration from the epoch that the packet enters the queue to the epoch that the next transmission can be initiated. Clearly, this definition does not consider the fact that the delay of a successfully received packet will generally be smaller than the delay of a packet that is not successfully delivered. This is because in the later case, the packet must be re-transmitted for a pre-defined number of times before it is dropped; and the dropped packets should not be considered in the calculation of average end-to-end delay.

2.2 Performance analysis for bulk/batch service queueing system

Besides the analysis for CSMA/CA, the delay performance of some bulk (batch) service systems has also been studied in the literature [18–21]. However, we note that all these analyses assume that the service time distribution does not depend on the size of the batch, which is not applicable to our study, in which the service time of a burst frame depends on the number of packets in the burst and the method of channel access (e.g., RTS/CTS).

3 A burst-frame-based MAC protocol

In this section, we first summarize the framework in [9]. We then briefly describe a burst-frame-based MAC protocol within the framework.

3.1 A framework for high throughput MAC

The main idea of the framework is to aggregate multiple upper-layer packets into one burst frame at the MAC layer. Compared to the traditional approach, in which each upper-layer packet is delivered individually, transmitting multiple upper-layer packets in one frame will significantly reduce the overheads of physical layer and MAC layer.

The framework consists of five major components. The first component is a packet classification policy that determines how to classify incoming upper-layer packets according to their destination and quality of service (QoS) requirements.

The second component is a buffer management policy that provides QoS and/or fairness among different flows. With this policy, each queue in the system can be controlled through a number of parameters, for example, the maximum number of packets in the queue, the maximum value of the total length of all packets in the queue, and the arrival time and the expected departure deadline of each packet.

The third component is a packet assembly policy that determine how to assemble packets into a burst frame, which should take into account synchronization overhead, physical layer constraints, QoS, and fairness among different nodes. For example, we can define the maximum and minimum size of a burst frame, the maximum and minimum number of packets in a burst, delay constraints that could trigger a burst assembly, and the destinations of packets in a burst (a burst may include packets to different destinations if an omni-directional antenna is used in the system).

The fourth component is an acknowledgement policy that specifies the acknowledgement procedure at the receiver side. For instance, if a burst contains packets to multiple destinations, then the destination nodes must be able to coordinate their ACK messages. Another important policy is to indicate the delivery status of each packet in a burst to avoid the retransmission of the whole burst if transmission errors occur.

The last component is a packet error control policy, which describes the method to mitigate packet errors.

In summary, our framework provides a guideline to design MAC protocols for high data rate wireless ad hoc networks.

3.2 A burst-frame-based MAC protocol

To facilitate the discussion in the following sections, we define a burst-frame-based MAC protocol as the following. In this protocol, we consider only one quality-of-service (QoS) class of traffic for each destination, i.e., all packets for the same destination have the same QoS requirements. Incoming packets are first classified based on its destination, and then put into a corresponding packet queue. Suppose there are N nodes in an ad hoc network; then we can implement $N-1$ packet queues in each node, where the $N-1$ queues are used for buffering packets destined to other $N-1$ nodes. For each queue, we use tail-dropping when there is a buffer overflow.

A burst frame will be generated if the total number of packets in the queue exceeds a threshold $B_{\min}$ and the server is idle (i.e., there is no other burst waiting for transmission). In addition, we assume that the total number of packets in a burst must be smaller than or equal to a preset value $B_{\max}$. In this protocol, we require that all the packets in a burst frame have the same destination so that most existing functions of IEEE 802.11 can be re-used. To achieve the fairness among destinations, a simple round-robin scheme will be employed for the $N-1$ queues in a node. When a burst assembly is finished, the burst frame will be stored in a buffer and waiting for transmission. If a burst frame is correctly received, the receiver will send one ACK frame to the transmitter.

4 Unsaturated delay analysis

In this section, we develop an analytical model to evaluate the end-to-end delay performance of the MAC protocol. Here we define the end-to-end delay of a packet as the time duration from the epoch that the packet arrives at the MAC layer of the source node to the epoch that the packet is successfully received by the MAC layer of the destination node. Note that we only consider the delay of packets that are successfully received.

The organization of this section is as follows. We first provide an overview for the unsaturated throughput analysis in our previous study [10]. We then discuss a general relationship between the steady-state probability distributions of the buffer condition at the packet arrival time and the packet departure time, for a G/G$^{[B_{\min},B_{\max}]}$/1/K queue. Based on this relationship, we then derive the average queueing delay of the M/G$^{[B_{\min},B_{\max}]}$/1/K queue. Finally, we analyze the average end-to-end delay of a packet.

4.1 Unsaturated throughput analysis

In the unsaturated throughput analysis in our previous study [10], we assume that packet arrivals in each node follow a Poisson process with the same rate λ; and that there is no packet transmission error due to bit errors. We consider the whole MAC system at any node as an M/G$^{[B_{\min},B_{\max}]}$/1/K queue (Fig. 1), where K is the capacity of the queue and the superscription $[B_{\min}, B_{\max}]$ means that the total number of packets in a burst is an integer in the range of $[B_{\min}, B_{\max}]$. To analyze this queue, we assume that the service time is a multiple of a pre-defined time unit, denoted as τ; and that the service time distribution, denoted as

$$q_{bi} = \Pr[\text{service time} = i\tau \mid b \text{ packets in the burst}],$$

is known.

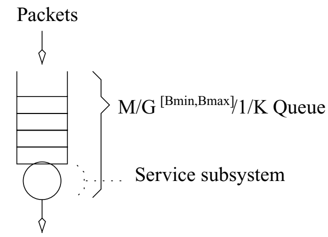


Fig. 1 The queueing model

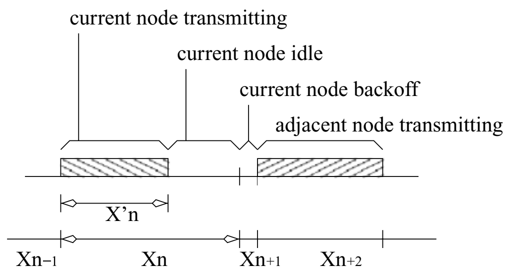


Fig. 2 The service time model

To analyze the MAC service time, we first derive the probability that a node is idle at any time instance, which is denoted as p_I, from the queueing analysis. Similar to [22], we partition the continuous time axis into slots, where two consecutive slots are delimited by the event of a value change in the backoff counter (shown in Fig. 2). Note that in the n-th slot with length X_n, the duration that the service is busy (denoted as X'_n) will be smaller than X_n if the node has no burst to transmit. We can then formulate an embedded Markov chain at the end of every slot. With the probability of p_I, we obtain the probability of transmission and collision in each slot, which is similar to the saturated analysis in [22,23].

By using a transfer-function approach similar to [16], we can derive the probability generating function (PGF) of q_{bi}, as illustrated in Fig. 3. In Fig. 3, $H(z)$ denotes the PGF of X'_n if current node backs off; $C_b(z)$ denotes the PGF of X'_n if a burst transmission fails in the slot and one of the collided burst has b packets; $S_b(z)$ denotes the PGF of X'_n if a burst transmission succeeds in the slot and the burst has b packets.

Finally, the throughput performance is calculated through a recursive algorithm. Particularly, we initialize p_I as 0, which is the saturated condition. After calculating the PGF of the service time, we update p_I through the queueing analysis. Although the convergence of the recursive algorithm has not been proved, the algorithm always achieves convergence in our numerical calculations.

4.2 Relationship between the arrival and departure steady-state distributions in a G/G$^{[B_{\min},B_{\max}]}$/1/K queue

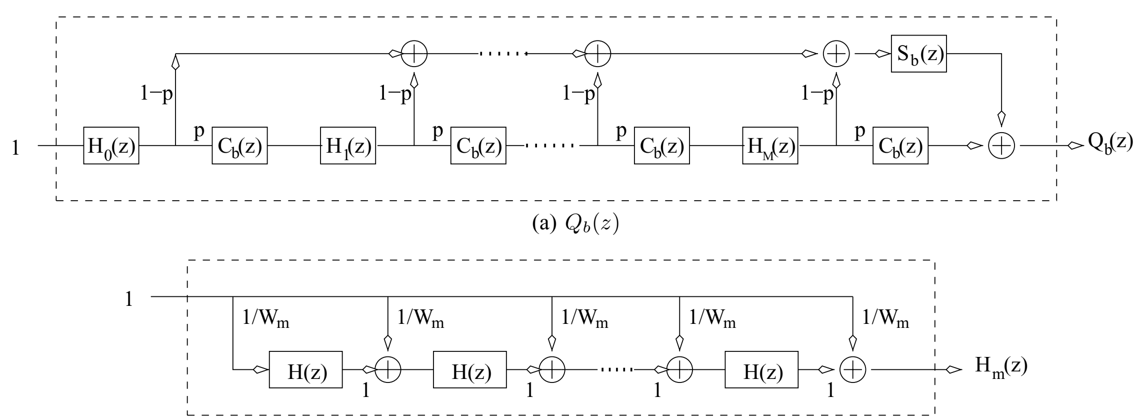
Let $\xi(t)$ be the state of a G/G$^{[B_{\min},B_{\max}]}$/1/K queueing system at time t and

$$\xi(t) \in \mathbf{S} = \{I_0, I_1, \ldots, I_{B_{\min}-1}, A_0, A_1, \ldots, A_K\}$$

where I_k means that the server is idle and there are k customers waiting in the queue; A_k means that the server is busy and there are k customers waiting in the queue. Let α_n and δ_n be the epoch of the n-th packet arrival and the n-th burst departure, respectively. Similar to [21], we define the following steady state probabilities:

- p_k^d denotes the steady-state probability that $\xi(\delta_n^-) = A_k$, where t^- is the epoch just before t;
- p_s^a denotes the steady-state probability that $\xi(\alpha_n^-) = s$, $s \in \mathbf{S}$;
- p_s^e denotes the steady-state conditional probability that $\xi(\alpha_n^-) = s, s \in \mathbf{S}' = \mathbf{S} - \{A_K\}$, given that the queue is not full, i.e., $\xi(\alpha_n^-) \neq A_K$.

The purpose of this section is to develop the relationship between p_k^d, p_s^a, and p_s^e.



(a) $Q_b(z)$

(b) $H_m(z)$

Fig. 3 Service system diagram

4.2.1 Analysis for entrances and departures of states

Let $E_s(t)$ and $D_s(t)$ be the total number of entrances and departures of state s, $s \in \mathbf{S}$, in $[0, t]$. For any queueing system, we can claim that

$$D_s(t) = E_s(t) + \Delta_s(t) \tag{1}$$

where

$$\Delta_s(t) = \begin{cases} 1 & \xi(0) = s \text{ and } \xi(t) \neq s \\ -1 & \xi(0) \neq s \text{ and } \xi(t) = s \\ 0 & \text{Otherwise.} \end{cases} \tag{2}$$

Now let $d_k(t)$ be the total number of departures in $[0, t]$ such that there are k packets in the queue just before the departure; let $a_s(t)$, $s \in \mathbf{S}$ be the total number of arrivals in $[0, t]$ such that the state of the queue is s just before the arrival. We can then derive $E_s(t)$ and $D_s(t)$ as follows.

$$\begin{aligned} E_{I_0} &= d_0(t), \\ E_{I_k} &= d_k(t) + a_{(I_{k-1})}(t),\ 1 \le k \le B_{\min} - 1, \\ E_{A_0} &= \sum_{k=B\min}^{B\max} d_k(t) + a_{(I_{(B_{\min}-1)})}(t), \\ E_{A_k} &= d_{(B_{\max}+k)}(t) + a_{(A_{k-1})}(t),\ 1 \le k \le K - B_{\max} \\ E_{A_k} &= a_{A_{k-1}}(t),\ K - B_{\max} < k < K; \end{aligned} \tag{3}$$

and

$$\begin{aligned} D_{I_k} &= a_{(I_k)}(t),\ 0 \le k \le B_{\min} - 1, \\ D_{A_k} &= a_{(A_k)}(t) + d_k(t),\ 0 \le k \le K - 1 \\ D_{A_K} &= d_K. \end{aligned} \tag{4}$$

With Eqs. (1), (3), and (4), we can derive the relationship between $a_s(t)$ and $d_k(t)$ as

$$\begin{aligned} a_{I_k}(t) &= \sum_{l=0}^{k} \left[d_l(t) + \Delta_{I_l}(t) \right], \quad 0 \le k \le B_{\min} - 1; \\ a_{A_k}(t) &= \sum_{l=k+1}^{B_{\max}+k} d_l(t) + \sum_{l=0}^{B_{\min}-1} \Delta_{I_l}(t) + \sum_{l=0}^{k} \Delta_{A_l}(t), \\ &\qquad 0 \le k \le K - B_{\max}; \\ a_{A_k}(t) &= \sum_{l=k+1}^{K} d_l(t) + \sum_{l=0}^{B_{\min}-1} \Delta_{I_l}(t) + \sum_{l=0}^{k} \Delta_{A_l}(t), \\ &\qquad K - B_{\max} + 1 \le k \le K - 1. \end{aligned} \tag{5}$$

4.2.2 Relationship between p_s^e and p_k^d

Let $D(t) = \sum_{\forall k} d_k(t)$ be the total number of burst departure in $[0, t]$; let $C(t) = \sum_{\forall s \in \mathbf{S}'} a_s(t)$ be the total number of packets that enter the queueing system. Therefore, we have

$$p_k^d = \lim_{t\to\infty} \frac{d_k(t)}{D(t)} \tag{6}$$

and

$$p_s^e = \lim_{t\to\infty} \frac{a_s(t)}{C(t)}. \tag{7}$$

Notice that all packets that enter the queue will eventually leave the system through bursts. Therefore, the average number of packets in a burst can be derived as

$$E(B) = \lim_{t\to\infty} \frac{C(t)}{D(t)} = \sum_{k=0}^{K} B_k \cdot p_k^d. \tag{8}$$

where B_k is the number of packets in the first burst after a burst departure, before which the state of queue is A_k.

From Eqs. (7) and (8), we have

$$p_s^e = \lim_{t\to\infty} \frac{a_s(t)}{D(t)} \times \frac{1}{E(B)}. \tag{9}$$

With Eqs. (5) and (9), and notice that $\Delta_s(t)$ can only be $0, \pm 1$, we can finally get

$$p_{I_k}^e = \frac{1}{E(B)} \times \sum_{l=0}^{k} p_l^d \tag{10}$$

$$p_{A_k}^e = \frac{1}{E(B)} \times \sum_{l=k+1}^{\min(K,k+B_{\max})} p_l^d \tag{11}$$

4.2.3 Relationship of p_s^e and p_s^a

Let $A(t)$ be the total number of packet arrivals in $[0, t]$. Then, for any $s \in \mathbf{S}'$, we have

$$p_s^a = \lim_{t\to\infty} \frac{a_s(t)}{A(t)} = p_s^e \times \lim_{t\to\infty} \frac{C(t)}{A(t)} = p_s^e \times \left(1 - p_{A_K}^a\right) \tag{12}$$

where $p_{A_K}^a$ is the packet blocking probability.

4.3 Queueing delay of the M/G$^{[B_{\min}, B_{\max}]}$/1/K queue

We first derive the packet loss probability for the M/G$^{[B_{\min}, B_{\max}]}$/1/K queueing system. Since packet arrivals are a Poisson process with rate λ, we have

$$P_{A_K}^a = 1 - \lim_{t\to\infty} \frac{C(t)}{\lambda t} = 1 - \frac{E(B)}{\lambda} \times \lim_{t\to\infty} \frac{D(t)}{t} \tag{13}$$

Since the calculation of $\lim_{t\to\infty}\frac{D(t)}{t}$ has been derived in [10], we have

$$P^a_{A_K} = 1 - \frac{E(B)}{\lambda T^s + \sum_{k=0}^{B_{\min}-1} p^d_k \cdot (B_{\min} - k)} \tag{14}$$

where T^s is the average burst service time, which can also be derived as in [10].

Based on the Poisson arrivals see time average (PASTA) property, we know that the steady state queue length distribution is equivalent to the queue length distribution at the epoch of arrivals. Therefore, with Eq. (12), we can calculate the average number of packets in the queue, denoted as N_q, through

$$N^q = \left[\sum_{k=0}^{B_{\min}-1} k \cdot p^e_{I_k} + \sum_{k=0}^{K-1} k \cdot p^e_{A_k}\right] \times \left(1 - p^a_{A_K}\right) + K \times p^a_{A_K} \tag{15}$$

Finally, using the Little's Law, we can calculate the average queueing delay of a packet by

$$T^q = \frac{N_q}{\lambda \times (1 - p^a_{A_K})} \tag{16}$$

4.4 The average service time for a successfully received packet

Let q'_{bi} be the steady state probability that the burst service time is $i\tau$, given that there are b packets in the burst and that the packet is successfully received. Let T^a be the average service time of a successfully received packet. Similar to the calculation of the average service time in [10], we have

$$T^a = \sum_{k=0}^{K} p^d_k \left[\sum_{\forall i} q'_{B_k i} \times (i\tau)\right]. \tag{17}$$

Now denote $Q'_b(z)$ the PGF of q'_{bi}, which is

$$Q'_b(z) = \sum_i z^i \cdot q'_{bi}. \tag{18}$$

Following [10], we can derive

$$Q'_b(z) = \frac{1-p}{1-p^{M+1}} S_b(z) \sum_{m=0}^{M} \left[(pC_b(z))^m \prod_{i=0}^{m} H_i(z)\right] \tag{19}$$

where

$$H_i(z) = \frac{1}{W_i} \sum_{j=0}^{W_i-1} [H(z)]^i \tag{20}$$

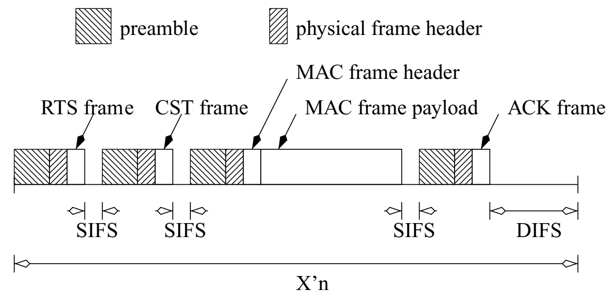


Fig. 4 Timing in a slot of successful transmission with the RTS/CTS access scheme

4.5 End-to-end delay

We now consider the timing in a slot that has a successful burst transmission, shown in Fig. 4.[1] From Fig. 4, we can observe that, X'_n is larger than Y_n, where Y_n is defined as the duration between the start of the RTS frame and the end of the DATA frame. Particularly, the difference between X'_n and Y_n, denoted as T^o, can be calculated by

$$T^o = X'_n - Y_n = T_{\text{SIFS}} + T_{\text{DIFS}} + T_{\text{sync}} + \frac{1}{R}(L_{\text{PH}} + L_{\text{ACK}}) \tag{21}$$

where T_{SIFS} denotes the time duration of SIFS, T_{DIFS} denotes the time duration of DIFS, T_{sync} denotes the synchronization time, L_{PH} denotes the length of physical frame header in bits (excluding the synchronization preamble), and L_{ACK} denotes the length of ACK frame in bits.

Finally, the average end-to-end delay T^d can be achieved by

$$T^d = T^q + T^a - T^o, \tag{22}$$

where T^q and T^a can be calculated by Eqs. (16) and (17), respectively.

5 Numerical and simulation results

In this section, we evaluate the delay performance of the burst-frame-based MAC protocol through simulation and analysis. The settings of experiments are summarized in Table 1.

In addition, we also make the following assumptions:

- All nodes are located in a 10 m × 10 m area.
- There are no bit errors in transmission.
- The synchronization time T_{sync} is identical for all messages.

[1] Here we ignore the propagation delay.

Table 1 Setting of the MAC protocol

Minimum contention window size	8
Maximum contention window size	256
σ	2 μs
SIFS	1 μs
DIFS	5 μs
Retry limit	4
Access scheme	RTS/CTS
Packet size	1000 Bytes
Buffer size	50 packets

- Except the preamble portion, all data fields in a frame (including physical header and MAC header) are transmitted with channel data rate R.
- Packet arrivals to any node i are a Poisson process with the same rate λ (packets/s). We further define the incoming traffic load as

$$\rho = \frac{N \times 8000 \times \lambda}{R} \text{ Erlang}$$

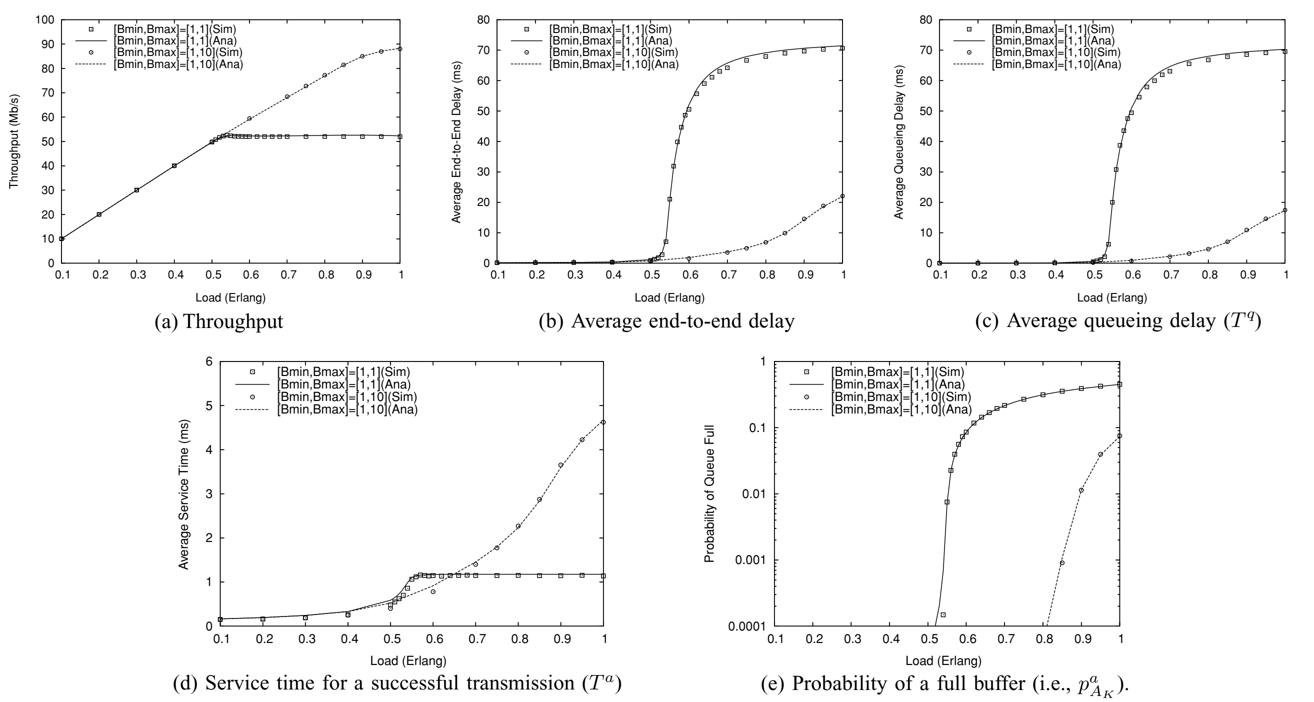
where 8000 is the packet size in bits.

For the analysis, we let the time unit be $\tau = \sigma$, let the maximum service time be 60000 time units, and run the recursive algorithm described in [10] until the result converges.

Figure 5 shows various performance metrics versus incoming traffic load for two burst assembly policies: (1) $[B_{\min}, B_{\max}] = [1, 1]$ (benchmark) and (2) $[B_{\min}, B_{\max}] = [1, 10]$. Here we assume that $N = 10$, $R = 100$ Mb/s, and $T_{\text{sync}} = 10\ \mu$s, where $T_{\text{sync}} = 10\ \mu$s is a typical assumption in UWB networks [4, 5]. Figure 5(a) and (b) show that policy 2 can significantly improve both the throughput and the delay performance, especially when the traffic load is high. From Fig. 5(a) and (b), we also observe that the throughput and delay performance of the benchmark policy become *saturated* if the traffic load is high. Particularly, under the benchmark policy, the throughout is saturated if the load is larger than 0.52; the delay increases sharply when the load increases from 0.5 to 0.7 and gradually increases if load is greater than 0.7. In contrast, the average end-to-end delay under policy 2 increases much slower.

To better understand the delay performance of these two policies, we plot the queueing delay versus load, and the service time for successfully received packets versus load in Fig. 5(c) and (d), respectively. We can see that $T^q < T^a$ if the load is less than 0.5, while $T^q > T^a$ if the load is greater than 0.5. That is, for both policies, the queueing delay T^q is dominant in the end-to-end delay (see Eq. (22)) when the load is high, while the service time T^a contributes the most in the end-to-end delay when the load is small.

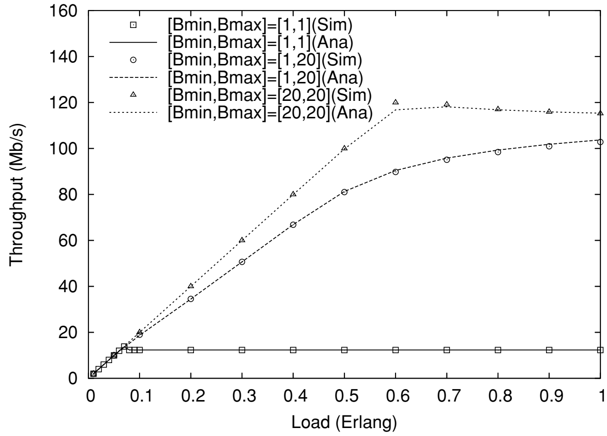
We show the probability of a full buffer versus traffic load in Fig. 5(e). It can be observed that the packet loss due to buffer overflow is negligible when the load is small for both policies. For policy 1, we note that $p^a_{A_K}$ increases dramatically as soon as the throughput becomes saturated. For policy 2, we see that the loss is much smaller than that of policy 1, even if the load is very high. For example, under policy 2, the probability of a full buffer is only 8% when load is 1; in contrast, the probability is nearly 50% for policy 1



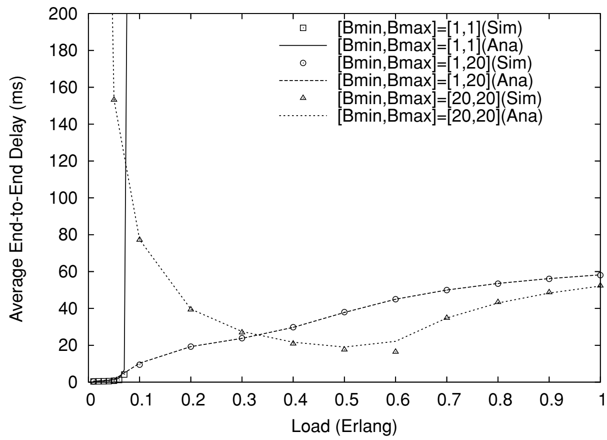
(a) Throughput (b) Average end-to-end delay (c) Average queueing delay (T^q)

(d) Service time for a successful transmission (T^a) (e) Probability of a full buffer (i.e., $p^a_{A_K}$).

Fig. 5 Performance versus incoming traffic load ($N = 10$, $T_{\text{sync}} = 10\ \mu$s, $R = 100$ Mb/s)

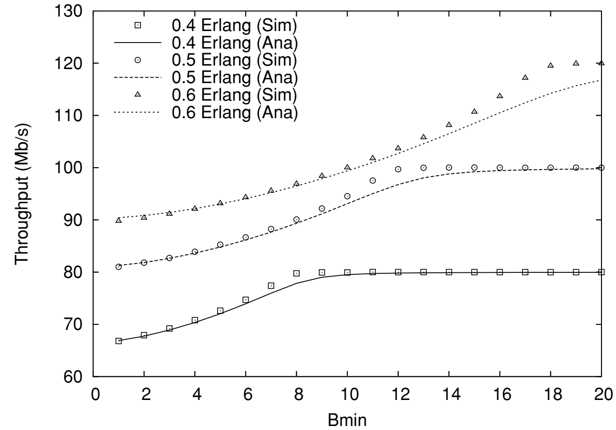


(a) Throughput

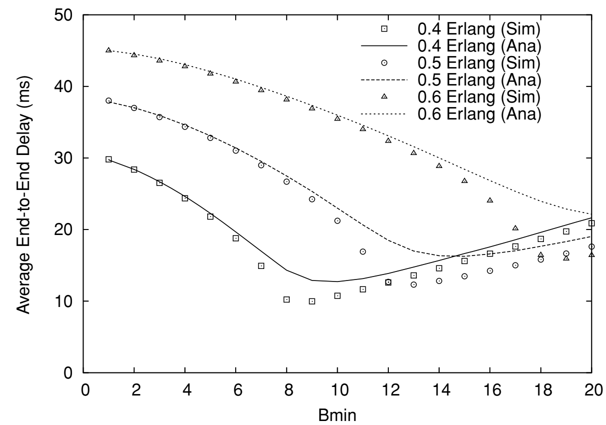


(b) Average end-to-end delay

Fig. 6 Performance versus incoming traffic load ($N = 20$, $T_{sync} = 100\ \mu s$, $R = 200$ Mb/s)



(a) Throughput



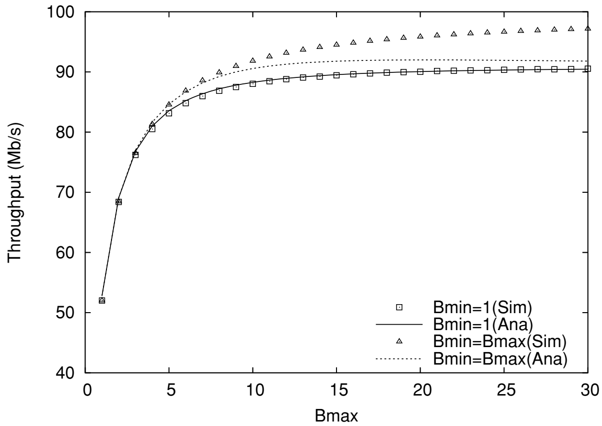
(b) Average end-to-end delay

Fig. 7 Performance versus $B_{\min}$ ($N = 20$, $T_{sync} = 100\ \mu s$, $R = 200$ Mb/s, $B_{\max} = 20$)

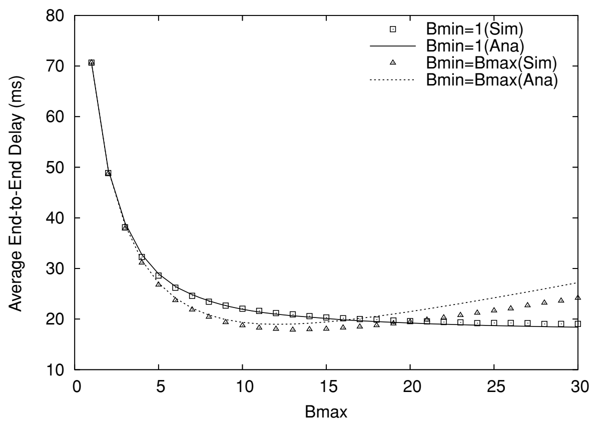
when load is 1. From Fig. 5, we also note that the analytical results give excellent agreement with the simulation results under different traffic load.

Figure 6 plots the throughput and the delay performance versus traffic load, where we assume $N = 20$, $R = 200$ Mb/s, and $T_{sync} = 100\ \mu s$, which might also be a typical scenario for UWB networks or WLANs [24, 25]. In this experiment, we examine three policies: (1) $[B_{\min}, B_{\max}] = [1, 1]$ (benchmark), (2) $[B_{\min}, B_{\max}] = [1, 20]$, and (3) $[B_{\min}, B_{\max}] = [20, 20]$. We can see that, policy 3 has the best throughput performance amongst the three policies. We can also observe that, the benchmark policy performs very poorly. Particularly, we see that the saturated throughput of the benchmark is only about 12 Mb/s. In contrast, policy 3 can achieve 120 Mb/s throughput. From Fig. 6, we can observe that, while policy 3 has the best throughput performance, its delay performance is worse than the performance of policy 2 if the load is less than 0.33. The reason for this is that, the packet assembly delay (e.g., a packet may have to wait for other $B_{\min} - 1$ packets to arrive before it can be sent) in policy 3 is very large when the incoming traffic load is small. It is interesting to note that, for policy 3, there exists a certain traffic load (i.e., around 0.6 Erlang), which leads to a maximum throughput and a minimum average end-to-end delay.

In Fig. 6, we have demonstrated that the delay performance of policy 3 is worse than policy 2 in some situations, although it performs better from the throughput perspective. To better understand the impact of the burst assembly policy on the throughput and delay performance, we plot the performance versus $B_{\min}$ in Fig. 7 with various traffic load, where we let $B_{\max} = 20$ and apply the same setting as that in Fig. 6. From Fig. 7(a), we can see that, the increase of $B_{\min}$ can improve the throughput performance when $B_{\min}$ increases from 1. However, the throughput converges to a certain value if $B_{\min}$ is larger than a certain *threshold*, denoted by $B'_{\min}$. For example, the throughput converges to 80 Mb/s when $B_{\min} \geq B'_{\min} = 8$ under a traffic load of 0.4 Erlang. In Fig. 7(b), we observe that the delay performance is also optimum near the same threshold $B'_{\min}$. These results indicate that the performance can be optimized by choosing appropriate parameters of the burst assembly policy. In Fig. 6, we can also



(a) Throughput



(b) Average end-to-end delay

Fig. 8 Performance versus $B_{\max}$ ($N = 10$, $T_{\text{sync}} = 10$ μs, $R = 100$ Mb/s)

see that the proposed analytical model may under-estimate the throughput (and over-estimate the delay) performance if $B_{\min}$ is near the threshold $B'_{\min}$. However, in most other cases, the analytical results match the simulation ones.

Finally, we show the performance versus $B_{\max}$ in Fig. 8, with the same setting as in Fig. 5. In this figure, we compare the performance under two policies: (1) $B_{\min} = 1$, and (2) $B_{\min} = B_{\max}$. From Fig. 8(a), we can observe that, for both policies, the throughput increases with the increase of $B_{\max}$; and we can also see that policy 2 always performs better than policy 1. However, Fig. 8(b) shows that, the delay decreases with the increase of $B_{\max}$ under policy 1; in contrast, under policy 2, there exists a certain value of $B_{\max}$, which leads to a minimum average end-to-end delay.

6 Enhanced burst assembly policy

In the previous section, we can clearly observe that a non-adaptive burst assembly policy cannot perform well under different traffic loads. For instance, if we assume that $B_{\max}$ is fixed, then a smaller $B_{\min}$ (e.g., $B_{\min} = 1$) will under-utilize the channel capacity if the traffic load is high; on the other hand, a larger $B_{\min}$ (e.g., $B_{\min} = B_{\max}$) will result in significant packet delay when the traffic load is small. This observation motivates us to design an adaptive burst assembly policy to optimize the throughput and delay performance of the burst-frame-based MAC protocol under different traffic loads.

6.1 Adaptive burst assembly policy

In this subsection, we propose a simple adaptive burst assembly policy. In this protocol, we assume that $B_{\max}$ is a fixed value due to physical layer constraints, which is valid in practice. The key idea of the policy is to keep $B_{\min}$ as small as possible when the traffic load is low and allow it to increase convexly with the increase of the traffic load.

Apparently, if each node in the network knows the overall traffic load of the network, then the task can be easily achieved. However, in an ad hoc network scenario, the overall traffic load information is not available. Therefore, we introduce another parameter to represent the channel utilization.

From a tagged node's perspective, we let the channel be *busy* if one of the following conditions holds.

1. The tagged node is sending or receiving a message.
2. The tagged node senses that there are signals transmitting through the channel.
3. The tagged node's network allocation vector (NAV) indicates that the channel is reserved by an ongoing communication.

We let the channel be *idle* if none of the above conditions is true. We then define the channel utilization ratio u as the following

$$u = \frac{T_{\text{busy}}}{T_{\text{busy}} + T_{\text{idle}}} \tag{23}$$

where T_{busy} denotes the total amount of time that the channel is busy in a certain period of length T_p, and T_{idle} denotes the total amount of time that the channel is idle in the same period. Clearly, $T_p = T_{\text{busy}} + T_{\text{idle}}$.

To measure the channel utilization ratio in a realistic case, we can use exponential averaging method or Kalman filtering method. In this paper, as the first step of our study, we provide a simple algorithm as below.

Suppose the time axis is partitioned into intervals where two consecutive intervals is delimited by an event that the channel status changes from busy to idle. Let k be the index of an interval, and let $T_{\text{busy}}(k)$ and $T_{\text{idle}}(k)$ be the amount of time that the channel is busy and idle, respectively, in the kth interval. We also denote $\overline{T}_{\text{busy}}(k)$ and $\overline{T}_{\text{idle}}(k)$ as the average busy time and the average idle time in an interval, respectively, where k means being calculated up to the kth interval.

$\overline{T}_{\text{busy}}(k)$ and $\overline{T}_{\text{idle}}(k)$ can then be calculated in a recursive manner by

$$\begin{cases} \overline{T}_{\text{busy}}(k) = (1 - w_k) \times \overline{T}_{\text{busy}}(k-1) + w_k \times T_{\text{busy}}(k) \\ \overline{T}_{\text{idle}}(k) = (1 - w_k) \times \overline{T}_{\text{idle}}(k-1) + w_k \times T_{\text{idle}}(k) \end{cases} \tag{24}$$

where w_k is an adjusting weight and is set to $\frac{1}{k}$ in this study; we set $\overline{T}_{busy}(0) = \overline{T}_{idle}(0) = 0$.

The channel utilization ratio until the k^{th} interval can then be defined by

$$u(k) = \frac{\overline{T}_{\text{busy}}(k)}{\overline{T}_{\text{busy}}(k) + \overline{T}_{\text{idle}}(k)} \tag{25}$$

Similar to $u(k)$, $B_{\min}$ will also be updated in an recursive manner. Specifically, given the channel utilization ratio $u(k)$, the minimum burst size in the $(k+1)^{th}$ interval is defined as

$$B_{\min}(k+1) = B_0 + (u(k))^{\alpha} (B_{\max} - B_0), \tag{26}$$

where B_0 is the minimum possible burst size and α is an adjusting parameter not less than 1, which can keep $B_{\min}$ convex to the channel utilization ratio.

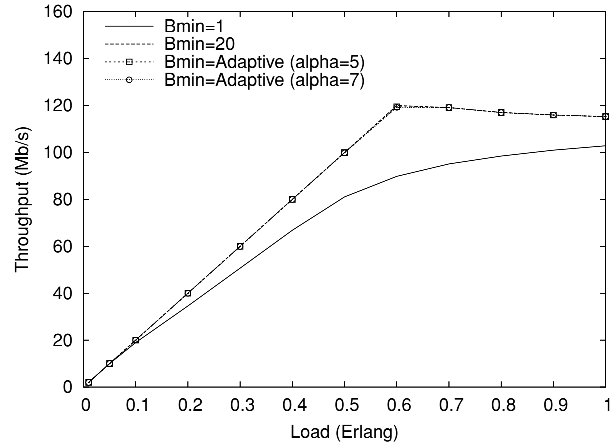
The intuition of Eq. (26) is that, given the same $B_{\max}$ and traffic arrival rate, a larger $B_{\min}$ will lead to a smaller channel utilization ratio, and the smaller channel utilization ratio will pull down $B_{\min}$. On the other hand, a smaller $B_{\min}$ will lead to a larger channel utilization ratio, and the larger channel utilization ratio will pull up $B_{\min}$. Thus $B_{\min}$ will eventually convergence to a certain value through the recursive process.

6.2 Simulation results

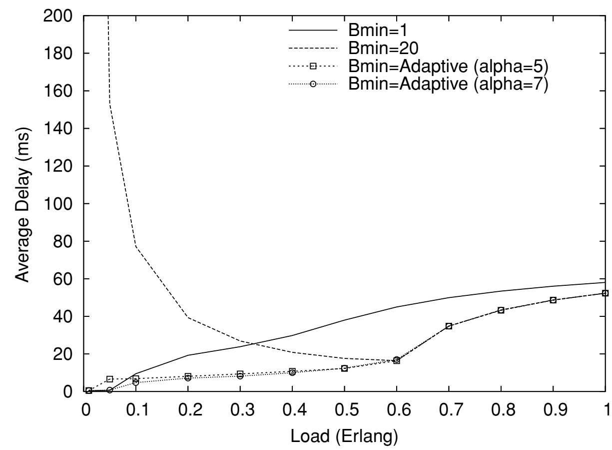
In this subsection, we evaluate the proposed adaptive burst assembly policy through simulation. All settings are basically the same as those in Section 5.

Figure 9 demonstrates the performance of the proposed adaptive burst assembly policy, where the setting are the same as that in Fig. 6. In this experiment, we set $B_{\max} = 20$ and compare four assembly policies: (1) fixed $B_{\min} = 1$, (2) fixed $B_{\min} = 20$, (3) adaptive $B_{\min}$ with $\alpha = 5$, and (4) adaptive $B_{\min}$ with $\alpha = 7$. We can observe from Fig. 9(a) that policies 2, 3, and 4 achieve larger throughput than policy 1, under a traffic load larger than 0.1 Erlang. Figure 9(b) shows that both adaptive policies can lead to lower end-to-end delay in most traffic conditions and setting $\alpha = 7$ can achieve the lowest delay performance. The above results demonstrate that the proposed adaptive burst assembly policy can provide better throughput and delay performance, compared to the non-adaptive one.

Finally, we investigate the convergence performance of the recursive algorithm in Fig. 10, where the settings of the experiment are the same as in Fig. 9 and we choose the



(a) Throughput



(b) Average end-to-end delay

Fig. 9 Performance of the adaptive burst assembly policy

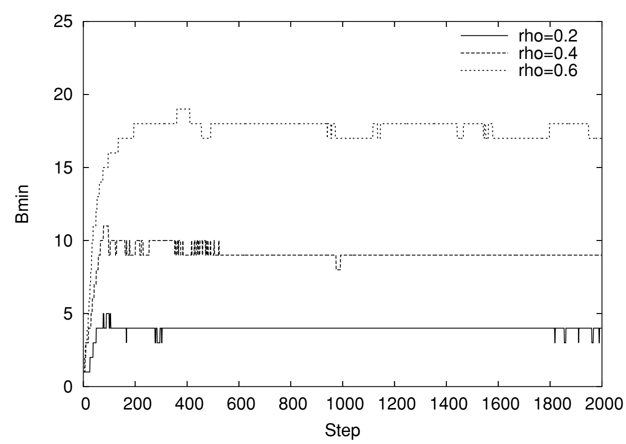


Fig. 10 Convergence of the proposed adaptive scheme ($\alpha = 7$)

adaptive burst assembly policy with $\alpha = 7$. It can be clearly observed that the proposed method can make $B_{\min}$ converge in a few hundreds of steps.

7 Conclusions

In this paper, we have developed an accurate analytical model to calculate the end-to-end delay under the burst-frame-based CSMA/CA protocol. In this model, we consider the end-to-end delay from the epoch that a packet enters the queue at the transmitter node to the epoch that it is successfully received by the receiver node, which has not been studied theoretically in the literature. Extensive simulation and numerical results show that, the burst-frame-based CSMA/CA protocol can significantly improve both the throughput performance and the delay performance. Our analytical results give excellent agreement with the simulation results, which represents the accuracy of our analytical model. Our results also provide important guideline on how to set the parameters of the burst assembly policy. Based on these results, we have further developed an efficient adaptive burst assembly policy so as to optimize the throughput and delay performance of the burst-frame-based CSMA/CA protocol.

Acknowledgments This work was supported in part by the US National Science Foundation (NSF) under EPSCoR start-up grant in Puerto Rico, and was supported in part by the US NSF under grant 0424546, ANI-0093241, and DB-0529012.

References

1. D. Gesbert, M. Shafi, D. Shan Shiu, P.J. Smith and A. Naguib, From theory to practice: an overview of MIMO space-time coded wireless systems, IEEE Journal of Selected Areas of Communication 21(3) (April 2003) 281–302.
2. S. Roy, J.R. Foerster, V.S. Somayazulu and D.G. Leeper, Ultra-wideband radio design: The promise of high-speed, short-range wireless connectivity, Proceedings of the IEEE 92(2) (February 2004) 295–311.
3. Status of Project IEEE 802.11n (Web Page). [Online]. Available: http://grouper.ieee.org/groups/802/11/Reports/tgn_update.htm
4. IEEE P802.15-04/0137r1, DS-UWB physical layer submission to 802.15 Task Group 3a, March 2004, Project: IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs).
5. IEEE P802.15-04/268r3, Multi-band OFDM physical layer submission to 802.15 Task Group 3a, March 2004, Project: IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs).
6. Y. Xiao, MAC performance analysis and enhancement over 100 Mbps data rates for IEEE 802.11, in: *Proc IEEE VTC Fall* (October 2003) pp. 1869–1873.
7. Y. Xiao, Concatenation and piggyback mechanisms for the IEEE 802.11 MAC, in: *Proc IEEE WCNC* (March 2004) pp. 1642–1647.
8. Y. Yuan, D. Gu, W. Arbaugh and J. Zhang, High-performance MAC for high-capacity wireless LANs, in: *Proceedings of the IEEE ICCCN* (2004).
9. K. Lu, D. Wu and Y. Fang, A novel framework for medium access control in ultra-wideband ad hoc networks, Dynamics of Continuous, Discrete and Impulsive Systems, An International Journal for Theory and Applications, Series B: Applications and Algorithms 12(3) (June 2005) 427–441, special issue on UWB (Guest Editors: Robert C. Qiu, Sherman Shen).
10. K. Lu, D. Wu, Y. Fang and R.C. Qiu, Performance analysis of a burst-frame-based MAC protocol for ultra-wideband ad hoc networks, in: *Proc IEEE ICC* vol. 5 (May 2005) pp. 2937–2941.
11. P. Chatzimisios, V. Vitsas and A.C. Boucouvalas, Throughput and delay analysis of IEEE 802.11 protocol, in: *Proceedings of the IEEE International Workshop on Networked Appliances*, Liverpool (October 2002) pp. 168–174.
12. P. Chatzimisios, A.C. Boucouvalas and V. Vitsas, Packet delay analysis of IEEE 802.11 MAC protocol, Electronics Letters 39(18) (September 2003) 1358–1359.
13. Y. Chen, Q.-A. Zeng and D.P. Agrawal, Performance evaluation for IEEE 802.11e enhanced distributed coordinated function, John Wiley & Sons, Wireless Communications and Mobile Computing 4(6) (2004) 639–653.
14. O. Tickoo and B. Sikdar, Queueing analysis and delay mitigation in IEEE 802.11, in: *Proc IEEE INFOCOM*, Hong Kong, P.R. China (April 2004) pp. 1–1.
15. O. Tickoo and B. Sikdar, A queueing model for finite load IEEE 802.11 random access MAC, in: *Proc IEEE ICC*, Paris, France (June 2004) pp. 175–179.
16. H. Zhai, Y. Kwon, and Y. Fang, Performance analysis of IEEE 802.11 MAC protocol in wireless LAN, Wiley Journal of Wireless Communications and Moble Computing (WCMC) 4(8) (December 2004) 917–931.
17. M. Ozdemir and A.B. McDonald, A queuing theoretic model for IEEE 802.11 DCF using RTS/CTS, in: *IEEE Workshop on Local and Metropolitan Area Networks (LANMAN 2004)* (April 2004) pp. 33–38.
18. H. Gold and P. Tran-Gia, Performance analysis of a batch service queue arising out of manufacturing system modelling, Institute of Computer Science, University of Wurzburg, Technical Report #16 (July 1990).
19. M. Dummler, Analysis of the departure process of a batch service queueing system, Institute of Computer Science, University of Wurzburg, Technical Repeport #210 (September 1998).
20. T. Ryden, Waiting time distribution in buffers with batch service, IEEE Transaction on Communication 41(7) (July 1993) 1027–1030.
21. G. Hebuterne and C. Rosenberg, Arrival and departure state disributions in the general bulk-service queue, Naval Research Logistics 46(1) (January 1999) 107–118.
22. G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function, IEEE Journal of Selected Areas of Communication 18(3) (March 2000) 535–547.
23. H. Wu, Y. Peng, K. Long, S. Cheng and J. Ma, Performance of reliable transport protocol over IEEE 802.11 wireless LAN: analysis and enhancement, in: *Proc IEEE INFOCOM* (June 2002) pp. 599–607.
24. TGn Sync Proposal for IEEE 802.11n (Web Page). [Online]. Available: http://www.tgnsync.org
25. World-Wide Spectrum Efficiency (Web Page). [Online]. Available: http://www.wwise.org

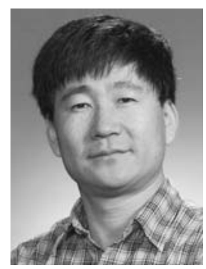
Kejie Lu received the B.E. and M.E. degrees in Telecommunications Engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1994 and 1997, respectively. He received the Ph.D. degree in Electrical Engineering from the University of Texas at Dallas in 2003. In 2004 and 2005, he was a postdoctoral research associate in the Department of Electrical and Computer Engineering, University of Florida. Currently, he is an assistant professor in the Department of Electrical and Computer Engineering, University of Puerto Rico at Mayagüez. His research interests include architecture and protocols design for computer and communication networks, performance analysis, network security, and wireless communications.

Jianfeng Wang received the B.E. and M.E. degrees in electrical engineering from Huazhong University of Science and Technology, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering from University of Florida in 2006. From January 2006 to July 2006, he was a research intern in wireless standards and technology group, Intel Corporation. In October 2006, he joined Philips Research North America as a senior member research staff in wireless communications and networking department. He is engaged in research and standardization on wireless networks with emphasis on medium access control (MAC).

Dapeng Wu received B.E. in Electrical Engineering from Huazhong University of Science and Technology, Wuhan, China, in 1990, M.E. in Electrical Engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1997, and Ph.D. in Electrical and Computer Engineering from Carnegie Mellon University, Pittsburgh, PA, in 2003.

Since August 2003, he has been with Electrical and Computer Engineering Department at University of Florida, Gainesville, FL, as an Assistant Professor. His research interests are in the areas of networking, communications, multimedia, signal processing, and information and network security. He received the IEEE Circuits and Systems for Video Technology (CSVT) Transactions Best Paper Award for Year 2001, and the Best Paper Award in International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine) 2006.

Currently, he serves as the Editor-in-Chief of Journal of Advances in Multimedia, and an Associate Editor for IEEE Transactions on Wireless Communications, IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Vehicular Technology, and International Journal of Ad Hoc and Ubiquitous Computing. He is also a guest-editor for IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Cross-layer Optimized Wireless Multimedia Communications. He served as Program Chair for IEEE/ACM First International Workshop on Broadband Wireless Services and Applications (BroadWISE 2004); and as a technical program committee member of over 30 conferences. He is Vice Chair of Mobile and wireless multimedia Interest Group (MobIG), Technical Committee on Multimedia Communications, IEEE Communications Society. He is a member of the Best Paper Award Committee, Technical Committee on Multimedia Communications, IEEE Communications Society.



Yuguang Fang received a Ph.D. degree in Systems Engineering from Case Western Reserve University in January 1994 and a Ph.D. degree in Electrical Engineering from Boston University in May 1997. He was an assistant professor in the Department of Electrical and Computer Engineering at New Jersey Institute of Technology from July 1998 to May 2000. He then joined the Department of Electrical and Computer Engineering at University of Florida in May 2000 as an assistant professor and got an early promotion to an associate professor with tenure in August 2003 and to a full professor in August 2005. He has published over 200 papers in refereed professional journals and conferences. He received the National Science Foundation Faculty Early Career Award in 2001 and the Office of Naval Research Young Investigator Award in 2002. He has served on several editorial boards of technical journals including IEEE Transactions on Communications, IEEE Transactions on Wireless Communications, IEEE Transactions on Mobile Computing and ACM Wireless Networks. He have also been actively participating in professional conference organizations such as serving as The Steering Committee Co-Chair for QShine, the Technical Program Vice-Chair for IEEE INFOCOM'2005, Technical Program Symposium Co-Chair for IEEE Globecom'2004, and a member of Technical Program Committee for IEEE INFOCOM (1998, 2000, 2003–2007). He is a senior member of the IEEE.