Performance Analysis of Delayed Acknowledgment Scheme in UWB-Based High-Rate WPAN

Hongyuan Chen, Zihua Guo, *Member, IEEE*, Richard Yuqi Yao, *Member, IEEE*, Xuemin (Sherman) Shen, *Senior Member, IEEE*, and Yanda Li, *Senior Member, IEEE*

***Abstract*—The wireless personal area network (WPAN) is designed for short-range connectivity among fixed or portable moving devices. The ultra-wideband (UWB) technology is being defined as the physical-layer (PHY) support for the high-rate WPAN. At medium access control (MAC) layer of the WPAN, a delayed acknowledgment (Dly-ACK) or burst-ACK (B-ACK) scheme is introduced to improve the channel utilization by reducing the overhead of ACK. In this paper, the authors first study the delay performance of the Dly-ACK scheme. An analytical model is developed for the Dly-ACK mechanism, and the delay is decomposed into queuing delay and delivery delay. These delay metrics are derived, and some important observations are obtained. In particular, there exists an optimal burst size, which is determined by the input traffic load and is very insensitive to the channel error rate within a normal error-rate range. It is also demonstrated that Dly-ACK cannot work properly if the burst size is fixed. The authors then propose a dynamical Dly-ACK scheme that can adaptively change its burst size according to the queue buffer size. Simulation results show that the dynamical scheme can improve the delay performance significantly.**

***Index Terms*—Delay performance, delayed acknowledgment (Dly-ACK), ultra-wideband (UWB), wireless personal area network (WPAN).**

I. Introduction

THE wireless personal area network (WPAN) is designed for short-range *ad hoc* connectivity among portable moving devices, and it has gained much attention recently. To support high data-rate WPAN, in the physical layer (PHY), the industry is defining specifications based on ultra-wideband (UWB) technology. Currently, there are mainly two UWB camps. One is direct sequence UWB (DS-UWB), which utilizes the DS-UWB [3] as the PHY technology and IEEE 802.15.3 [1] as the medium access control (MAC) layer support. The other is the multiband orthogonal frequency division multiplexing (OFDM) alliance (MBOA), which takes the OFDM technology as the UWB PHY [4] and defines an alternative MAC specification [2]. In both camps, the UWB PHY can support a data rate up to several hundreds megabits per second or even gigabits per second, while the MAC layer specifications support *ad hoc* connections.

Since wireless channel usually is error prone due to various fading, certain error-control techniques have to be used to provide a reliable link in a WPAN [5]. Conventionally, the MAC layer always adopts ACK and retransmission mechanism to provide reliable data-frame delivery for higher layers. In general, once the sender transmits one MAC frame, the receiver immediately responds one ACK frame to indicate this frame's reception status (error or error free). Normally, this conventional mechanism is referred to as immediate ACK (Imm-ACK). However, in high data-rate WPAN, the Imm-ACK policy may be relatively too expensive because the overhead caused by ACK may consume significant bandwidth. Fig. 1 shows the ACK-overhead significance under Imm-ACK for various data rates. In the figure, we assume that the basic data rate is 100 Mb/s while the channel data rate varies, and the frame size is 1000 B. The other parameters, such as the MAC header, the PHY header, etc., are according to [1]. $t_{\rm p}$ and $t_{\rm ACK}$ denote the time to transmit the data frame and ACK frame, respectively. It can be seen that there is a significant consumption of ACK overhead on the bandwidth when using the Imm-ACK policy, in particular, when the data rate is high. To resolve this problem, a new ACK policy, called delayed acknowledgment (Dly-ACK), has been proposed in 802.15.3 [1], and a similar policy, called burst-ACK (B-ACK), is defined in MBOA MAC [2]. In the following, we do not distinguish them and simply use Dly-ACK for both of them. In the Dly-ACK mechanism, instead of acknowledging each data frame, a burst of frames is first transmitted by the sender, and only after the whole burst is received, the receiver sends one ACK frame to the sender to acknowledge the whole burst.

Compared with the Imm-ACK scheme, the Dly-ACK is always able to improve the bandwidth efficiency because the number of ACK frames can be reduced. Moreover, the bandwidth efficiency is higher with the increased number of frames transmitted in one burst (we call this number the burst size). On the contrary, for the delay performance, the effect of Dly-ACK policy is twofold. On one hand, due to the fact that it can send frames to channel more quickly, Dly-ACK policy can reduce the queuing time in the sender's buffer. On the other hand, it introduces an additional random waiting delay at the receiver's buffer (reordering buffer or R-buffer) since all the frames should be delivered to a higher layer (say, IP layer) in order, and the correctly received frames have to wait for those frames that have lower frame ID and have been unsuccessfully

Manuscript received January 24, 2005; revised April 28, 2005 and July 14, 2005. The review of this paper was coordinated by Prof. D. O. Wu.

H. Chen and Y. Li are with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: chen.hongy@gmail.com; daulyd@tsinghua.edu.cn).

Z. Guo is with the Lenovo Corporate R&D, Beijing 100085, China (e-mail: guozh@lenovo.com).

R. Y. Yao is with the Microsoft, Redmond, WA 98052 USA (e-mail: richyao@microsoft.com).

X. Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: xshen@bbcr.uwaterloo.ca).

Digital Object Identifier 10.1109/TVT.2005.863432

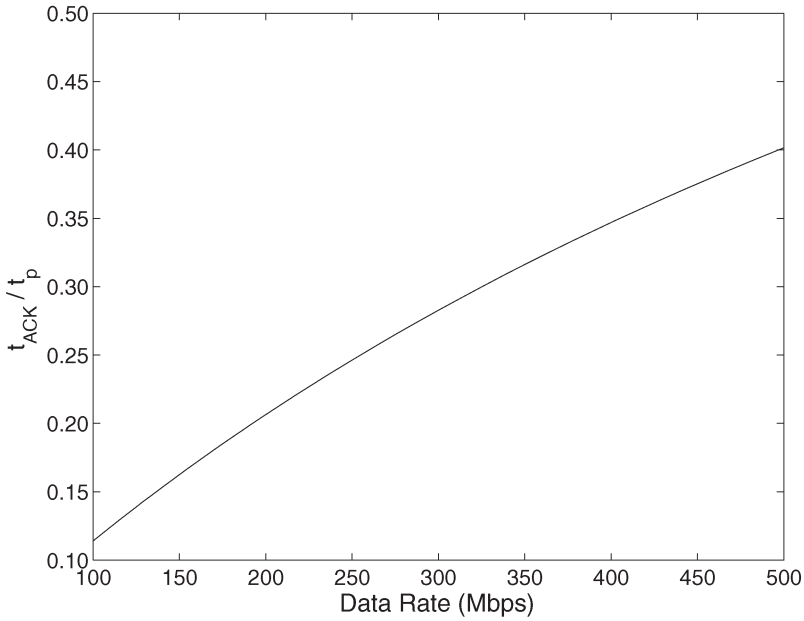


Fig. 1. Effects of data rate on $t_{\mathrm{ACK}}/t_{\mathrm{p}}$.

received by the receiver. Thus, the delay performance may be degraded when using the Dly-ACK in contrast with Imm-ACK. Moreover, some applications, such as video telephony, are delay sensitive, i.e., the frame will be useless and discarded if it is not delivered within a specified delay. For these applications, the delay degradation may result in unsatisfactory quality-of-service (QoS) [6]–[8]. Therefore, it is very important to correctly understand the delay behavior associated with Dly-ACK, which will enable us to gain useful insights on designing the Dly-ACK scheme for the WPAN. In this paper, we develop an analytical model for the Dly-ACK mechanism. The delay is decomposed into queuing delay and delivery delay. These delay metrics are derived, and some important observations are obtained. In particular, there exists an optimal burst size, which is determined by the input traffic load and is very insensitive to the channel error rate within a normal error-rate range. It is demonstrated that Dly-ACK cannot work properly if the burst size is fixed. We then propose a dynamical Dly-ACK scheme that can adaptively change its burst size according to the queue buffer size to improve the delay performance.

The rest of this paper is organized as follows. In Section II, the Dly-ACK policy is briefly introduced. In Section III, we develop an analytical model for the delay performance of Dly-ACK. Simulation results and discussions are given in Section IV. Our proposed dynamical Dly-ACK policy and performance evaluation are presented in Section V, and finally the conclusions are given in Section VI.

II. Delayed Acknowledgment Scheme

In both 802.15.3 MAC and MBOA MAC, three types of ACK policies are defined: no ACK (No-ACK), Imm-ACK, and Dly-ACK. When using the No-ACK policy, the destination device (DEV) will not acknowledge the received frame. Two successive frames are separated by minimum interframe space (MIFS). The No-ACK policy is appropriate for frames that do not require guaranteed delivery. The Imm-ACK policy provides an ACK process in which each frame is individually ACKed following the reception of the frame. All frames, including the data frames and the ACK frame, are separated by the short interframe space (SIFS), which is larger than MIFS. The Dly-ACK policy allows the source DEV to send a burst of frames without the intervening ACK frames. The source also adds Dly-ACK request information to a frame's MAC header when it is necessary. Once the destination receives this frame, which includes request information, it will send the Dly-ACK frame, which acknowledges those correctly received frames in current burst. The source will not start or resume the next burst transmission until a Dly-ACK frame is received. These frames that are not ACKed should be retransmitted in the next burst. The data frame and the ACK frame are separated by an SIFS, while there is an MIFS interval between two successive data frames. For convenience, we use n-Dly-ACK to represent the Dly-ACK mechanism that adopts n as its burst size. Obviously, the Imm-ACK is a special case of Dly-ACK with n being 1. In this paper, we say that one frame is in position i if this frame is the ith transmitted frame in one burst.

To demonstrate the advantage of Dly-ACK $(n \geq 2)$ over Imm-ACK in terms of bandwidth utilization, we define an index named maximum effect bandwidth (MEB), which is a fraction of time the channel is used to successfully transmit data frames versus the total channel time. Assuming that the sender is always busy, i.e., there are always pending MAC frames to be transmitted in the sender's buffer, then the MEB of n-Dly-ACK policy can be expressed by

$$\mathrm{MEB} = n \cdot \frac{t_{\mathrm{f}}(1-p)}{t_{\mathrm{b}}} \tag{1}$$

where t_{f} is the effective time used to transmit one frame (excluding the frame header), and t_{b} is the total time of transmitting one burst and equals $nt_{\mathrm{p}} + t_{\mathrm{ACK}} + (n-1)\mathrm{MIFS} +$

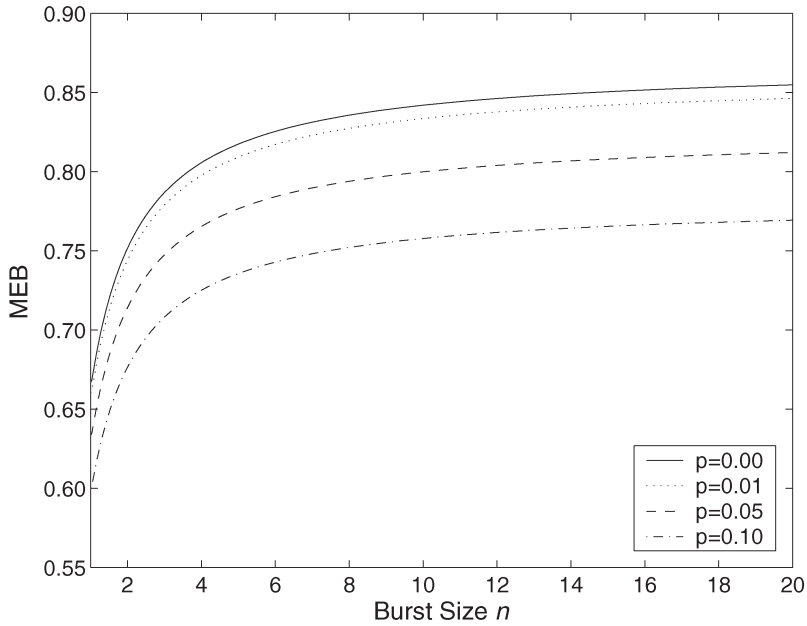


Fig. 2. MEB versus n under various p (data rate $=$ 100 Mb/s).

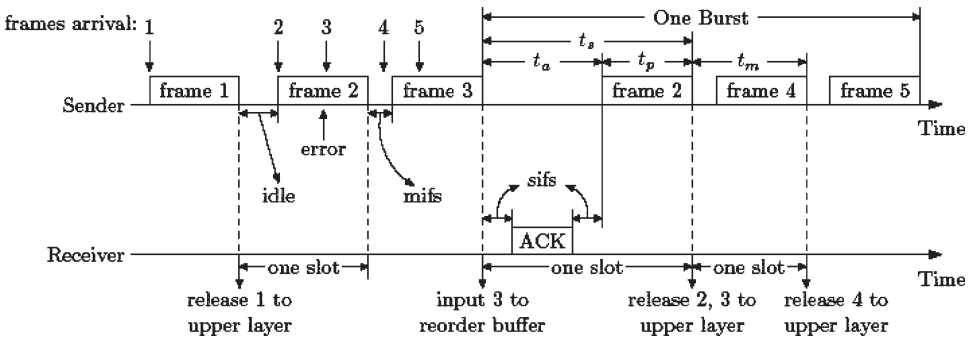


Fig. 3. Frame transmission process by using the Dly-ACK policy.

2SIFS, while p is the frame error rate. The curve of MEB versus n is given in Fig. 2. We can see that the Dly-ACK policy $(n > 1)$ results in a much larger MEB compared with Imm-ACK $(n = 1)$, which indicates the potential of achieving much higher channel unitization than Imm-ACK.

In contrast to the MAC throughput, the Dly-ACK effect on the delay performance is quite complicated. Fig. 3 shows one example of Dly-ACK policy with a burst size of 3. The sender assigns each frame a unique identifier (frame id), and the frames are transmitted according to their frame ids. Initially, Frame 1 is generated and transmitted in the first position of a burst. At the receiver, it will immediately release Frame 1 to a higher layer once it correctly receives Frame 1. After Frame 1 is transmitted, there is no frame queuing in traffic buffer (T-buffer) of the sender. Since the burst size is 3, and there is only one frame transmitted in this burst, the sender has to wait until Frame 2 is generated. We call this additional waiting time the idle time. When Frame 2 is being transmitted, Frame 3 arrives at T-buffer. Therefore, Frame 3 is immediately transmitted after an MIFS interval. After Frame 3 is transmitted, the receiver responds one ACK frame. Assume that Frame 2 is in error; thus, it is retransmitted in the next burst. Moreover, even if Frame 3 has been correctly received, it is not released to a higher layer due to the failure of Frame 2. Frame 3 must wait in the R-buffer until Frame 2 is correctly received. For convenience, we define some variables: $t_a = t_{\mathrm{ACK}} + 2\mathrm{SIFS}$, $t_s = t_{\mathrm{p}} + t_a$, and $t_m = t_{\mathrm{p}} + \mathrm{MIFS}$.

III. Delay Analysis

In this section, we analyze the delay performance of Dly-ACK scheme. In all analysis, we aim at one desired frame, called tagged frame. Fig. 4 shows all the delay components that the tagged frame undergoes in its entire life time at MAC layer [8], [9]. The total delay consists of the queuing delay and the delivery delay. The queuing delay is the duration from the time the tagged frame arrives at the T-buffer of sender to the time that it is transmitted for the first time. The delivery delay is divided into transmission and reordering delays. Transmission delay is the time interval elapsed between the first transmission and the correct reception of the tagged frame. The reordering delay is the time spent in the R-buffer.

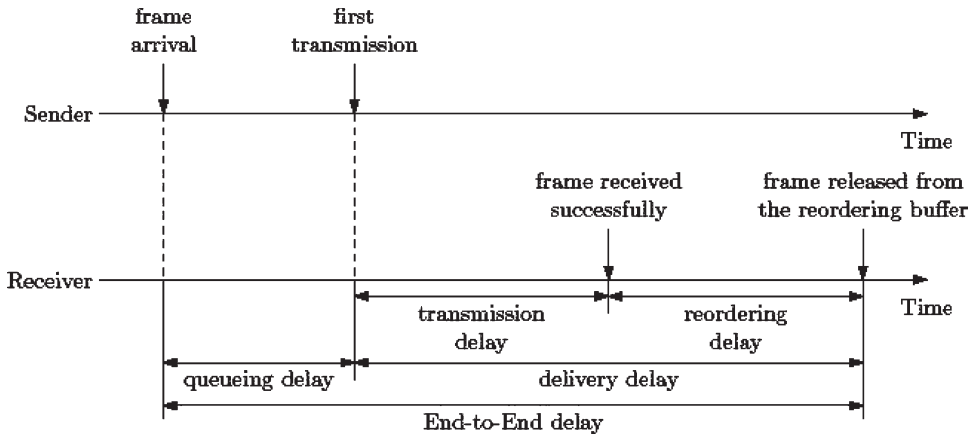


Fig. 4. Time diagram for end-to-end delay of a tagged frame transmitted with Dly-ACK policy.

TABLE I
POSSIBLE ONE-STEP STATE TRANSITION FROM STATE $X_1 = (q_1, i_1)$ TO STATE $X_2 = (q_2, i_2)$

State X_1	Successor X_2	Probability $P\{X_2\|X_1\}$	Condition
(q_1, i_1)	$(q_1 + k - 1, i_1 + 1)$	$\alpha_k(t_m)$	$q_1 > 0, i_1 < n, k \geq 0$
(q_1, n)	$(q_1 + r + k - 1, 1)$	$\alpha_k(t_s)\varphi(r, n)$	$q_1 > 0, 0 \leq r \leq n, k \geq 0$
$(0, i_1)$	$(k, i_1 + 1)$	$\alpha_k(t_p)$	$i_1 < n, k \geq 0$
$(0, n)$	$(r + k - 1, 1)$	$\varphi(r, n)\alpha_k(t_s)$	$0 < r \leq n, k \geq 0$
$(0, n)$	$(l + k - 1, 1)$	$\varphi(0, n)\alpha_l(t_a)\alpha_k(t_p)$	$l > 0, k \geq 0$
$(0, n)$	$(k, 1)$	$\varphi(0, n)\alpha_0(t_a)\alpha_k(t_p)$	$k \geq 0$

When constructing the analytical model, the following assumptions have been made.

1) The probability of a frame failure is p and the failures of frames are mutually independent. Furthermore, the ACK frames are to be error free. This common assumption is reasonable since the ACK frames are much smaller in size and are often protected by forward error correction (FEC) [9].
2) Each frame has the same size and is equal to f B.
3) In the analysis, for simplicity, input traffic with parameter Poisson λ is assumed because of the following reasons.
   a) It has been widely used as the basis for network modeling and analysis [10].
   b) Recent findings [11]–[14] have revealed that the Poisson process may be a good approximation of frame interarrival time distribution for backbone IP traffic.
   c) Our focus is to study the delay performance of Dly-ACK over different traffic loads, and the analysis can be readily extended to other traffic models. Let G be the ratio of input traffic load to the channel data rate H Mb/s. The Poisson parameter λ (in frames/second) can be computed by $\lambda = G \cdot H \cdot 10^6/(8f)$.
4) We concentrate on only a single pair of communicating users who have access to the full channel bandwidth. Note that a time-division multiple access (TDMA) MAC is an example for this scenario in the assigned time intervals, which is the basic medium-access scheme in IEEE 802.15.3 MAC [1] or MBOA MAC [2].

We divide the channel time into some logical slots. As shown in Fig. 3, a logical slot is the time interval between the end time of one frame transmission and that of the successive frame transmission. Note that the length of each slot may be variable, since the T-buffer may be empty sometimes. By using this logical slot, we can ensure that each slot corresponds to one frame transmission. Hence, if the mean value of slot length T_{slot} and the mean number of slots N_{slot} experienced by one frame are given, the mean end-to-end delay $E[D] = E[T_{\text{slot}}]E[N_{\text{slot}}]$.

A. Computing the Probability Distribution $D_{q,i}$

First, we introduce probability distribution $D_{q,i}$, which represents the joint probability that there are q frames waiting for transmission in T-buffer at the end of one time slot, and this slot is in position i of the current burst. This distribution is important for the analysis of end-to-end delay. A Markov chain can be built to compute $D_{q,i}$. We only consider the system state of those time points that are at the end of one time slot and are just before the start of the next slot. Accordingly, let the system state of slot t be defined by the vector $X(t) = (q, i)$, where $q(q \geq 0)$ and $i(1 \leq i \leq n)$ have the same meaning as that of $D_{q,i}$. Therefore, $X(t)$ is a two-state Markov chain. Let $P\{X_2|X_1\}$ denote the one-step transition probability from state $X_1 = (q_1, i_1)$ in slot t to state $X_2 = (q_2, i_2)$ in slot $(t + 1)$. The nonnull one-step transition probabilities are listed in Table I, which are obtained from the following detailed derivation.

In Table I, the fourth column denotes the condition that X_1 has to satisfy in order to move to X_2. Moreover, $\varphi(r, n)$ is the probability to correctly transmit $n-r$ frames over n frames in any order and is computed as follows:

$$\varphi(r, n) = \binom{n}{r} p^r (1 - p)^{n-r} \tag{2}$$

and $\alpha_k(t_l)$ is the probability that there are k new frames generated in interval t_l under Poisson arrival process with parameter λ. $\alpha_k(t_l)$ is given by

$$\alpha_k(t_l) = \mathrm{e}^{-\lambda t_l}\frac{(\lambda t_l)^k}{k!}. \tag{3}$$

In the state transition from (q_1, i_1) to (q_2, i_2), the value of i_2 is only dependent on the value of i_1 according to frame transmission procedure of Dly-ACK. If i_1 is smaller than n, i_2 must be $i_1 + 1$ because one burst consists of n frame transmissions, while $i_1 = n$ means that the current burst ends and the next burst starts. Hence, $i_2 = 1$ if $i_1 = n$.

The T-buffer dynamics (the value q) are governed by the frame-generation process and the frame-transmission process of Dly-ACK. According to the values of q_1 and i_1, there are four cases that should be considered.

The first case is that q_1 is larger than 0 and i_1 is smaller than n. Then, the length of slot $t + 1$ is t_m because frame transmission in slot $t + 1$ starts immediately after an MIFS. Consider that the frame transmitted in slot $i + 1$ will be removed from T-buffer and put into burst buffer (B-buffer) regardless of this frame's transmission status (failure or error free). Furthermore, there are k new frames generated in slot $t + 1$. Therefore, the state variable q_2 is $q_1 + k - 1$ with probability $\alpha_k(t_m)$.

The second case is that $q_1 > 0$ and $i_1 = n$. Frames that failed in the previous burst are again injected into T-buffer at the beginning of slot $t + 1$. There may be r error frames in one burst with probability $\varphi(r, n)$. Supposing the length of slot $t + 1$ is t_s and k new frames are generated in slot $t + 1$, state $X_1 = (q_1, n)$ moves to $X_2 = (q_1 + k + r - 1, 1)$ with probability $\alpha_k(t_s)\varphi(r, n)$.

The third case is that q_1 is equal to 0 and i_1 is not equal to n. Here, $q_1 = 0$ means that the sender must wait until one new frame is generated. Once this new frame is injected into the T-buffer, it will be transmitted immediately because there are no any frames waiting for transmission. Thus, we can divide slot $t + 1$ into two parts: idle time waiting for one new frame and frame-transmission time t_p of this new frame. At the end of idle time, one frame is generated. In the interval t_p, there are k new frames generated. Also, the frame transmitted in slot $t + 1$ will be removed from the T-buffer. Then, at the end of slot $t + 1$, the probability that there are k frames in the T-buffer is $\alpha_k(t_\mathrm{p})$.

The forth case is that $q_1 = 0$ and $i_1 = n$. There are three possible scenarios.

1) There are $r(r > 0)$ error frames, which are queued in the B-buffer at the end of slot t, and these frames will be injected into the T-buffer at the beginning of slot $t + 1$.
2) All frames are correctly transmitted in the current burst (i.e., $r = 0$), and there are some new frames generated in interval t_a of slot $t + 1$ (refer to Fig. 3 for definition of t_a).
3) There is no frame queued in T-buffer at the end time of t_a, i.e., r is equal to 0 and no frame arrives in interval t_a.

Hence, the sender should wait for one new frame that will be transmitted in position 1 of the next burst, and there may be some new frames generated in the transmission interval t_p of this new frame. In the first scenario, it is similar to the case $(q_1, n)(q_1 > 0)$ because r larger than 0 means that the T-buffer is not empty. Also, consider that there may be k new frames generated in t_s of slot $t + 1$. Therefore, one-step transition probability that state $(0, n)$ moves to $(r + k - 1, 1)$ is $\alpha_k(t_s)\varphi(r, n)$. In the second scenario, there are $l(l > 0)$ new frames injected into the T-buffer in interval t_a, and this means that one new frame transmission will immediately start once t_a ends. At the end of slot $t + 1$, the frame transmitted in slot $t + 1$ will be removed from the T-buffer. In the transmission interval t_p, k new frames are generated. Hence, there are $l + k - 1$ frames queued in the T-buffer at the end of slot $t + 1$, with probability $\varphi(0, n)\alpha_l(t_a)\alpha_k(t_\mathrm{p})$. For the third scenario, it is similar to the case $(0, i_1)(i_1 \neq n)$. After the interval t_a, there is an idle interval, which is used to wait for one new frame. During this frame transmission, there may be other k new frames arriving at the T-buffer. Since the frame transmitted in slot $t + 1$ must be removed from T-buffer, there are k frames queued in the T-buffer at the end of slot $t + 1$. Also, we can obtain that the transition probability is $\varphi(0, n)\alpha_0(t_a)\alpha_k(t_\mathrm{p})$ in this scenario. Note that in these three scenarios, the same state X_2 may be reached from different state X_1. For example, if r is equal to l in the first two scenarios, the state $(0, n)$ may move to the same state $(r + k - 1, 1)$. We list them separately because these transitions are due to different conditions.

We compute the steady-state probabilities from the Markov chain by an approximate approach. Let B be the maximum number of frames that T-buffer can accommodate, and it is set to be sufficiently large so that the probability of having B frames in the T-buffer is almost 0. Thus, when computing the steady-state probabilities, the states (q, i) with $q \geq B$ are omitted, and the total number of states is given by $M = B \cdot n$. Note that one appropriate value B is determined by the traffic load G. If the traffic load is higher, the value B should be larger so that the numerical results are more accurate. After an ordering of states, denoted by $f(X)$, which maps state X into an integer value in the range $[1, M]$, the transition probabilities can be collected in the matrix $\mathbf{P}$ in which the size is $M \times M$ [15]. The transition probability $P\{X_2|X_1\}$ corresponds to the element in row $f(X_1)$ and column $f(X_2)$ of matrix $\mathbf{P}$. Under the same state ordering, vector $\Pi = (\pi_1, \pi_2, \ldots, \pi_\mathrm{M})$ denotes the steady-state probabilities. Vector Π can be obtained by solving the following linear equations:

$$\begin{cases} \Pi\mathrm{P} = \Pi \\ \sum\limits_{j=1}^{M} \pi_j = 1 \cdot \end{cases} \tag{4}$$

Note that $D_{q,i}$ represents the joint probability that there are q frames waiting for transmission in the T-buffer at the end time of one slot, and this slot is in position i of the current burst. Thus, Π_j is the reordering of $D_{q;i}$.

To validate our proposed Markov model, we compare the results obtained from simulations with those from the above analysis. As shown in Table II, our analytical results match well with the simulation results. Although we only present one simulation case here, this model has also been validated by many other simulations. This will be further supported by more comprehensive results in Section IV.

TABLE II
CALCULATION OF $D_{q,i}(B = 100, n = 5, p = 0.1, G = 0.2, H = 100$ Mb/s) (SIM: SIMULATION; APPROX: OUR APPROXIMATION)

State	Sim	Approx	State	Sim	Approx
(0,1)	0.12451	0.12697	(1,1)	0.05492	0.05339
(0,2)	0.14221	0.14382	(1,2)	0.04627	0.04470
(0,3)	0.15026	0.15036	(1,3)	0.04169	0.04140
(0,4)	0.15347	0.15295	(1,4)	0.04015	0.04011
(0,5)	0.15418	0.15401	(1,5)	0.03894	0.03961
(2,1)	0.01634	0.01528	(3,1)	0.00382	0.00354
(2,2)	0.00990	0.00929	(3,2)	0.00161	0.00178
(2,3)	0.00711	0.00697	(3,3)	0.00102	0.00107
(2,4)	0.00583	0.00603	(3,4)	0.00013	0.00010
(2,5)	0.00572	0.00565	(3,5)	0.00007	0.00007

B. Analysis of Queuing Delay

1) Distribution of $R_{q,i}$: Let $R_{q,i}$ be the joint probability distribution that there are q frames queuing in the T-buffer, and the slot is in position i just when a new frame (observed frame) arrives at the T-buffer. Note that the frame that is being transmitted at the arriving instant of the observing frame is included in these q frames, and the q frames exclude the observed frame. Moreover, we say that this new frame is being in state $\{q, i\}$, where q and i have the same meaning as that of $R_{q,i}$. Considering that $R_{q,i}$ is actually the ratio between the number of new frames belonging to state $\{q, i\}$ and the total number of new frames in a large-enough interval, the following relationship can be derived:

$$R_{q,i} = \lim_{k\to\infty} \left\{ \frac{\sum_{t=1}^{k} \xi_{t,q,i}}{\sum_{t=1}^{k} \beta_t} \right\} \tag{5}$$

where $\xi_{t,q,i}$ is the number of new frames belonging to state $\{q, i\}$ in one burst t, and β_t represents the total number of new frames arriving in burst t. Let $W_{q,i}$ be the mean number of new frames belonging to state $\{q, i\}$ in one burst, i.e., $W_{q,i} = E[\xi_{t,q,i}]$, and U be the mean number of new frames arriving in one burst, i.e., $U = E[\beta_t]$. From (5), we have

$$R_{q,i} = \frac{\lim_{k\to\infty} \sum_{t=1}^{k} \frac{\xi_{t,q,i}}{k}}{\lim_{k\to\infty} \sum_{t=1}^{k} \frac{\beta_t}{k}} = \frac{W_{q,i}}{U}. \tag{6}$$

To compute $R_{q,i}$, we have the following proposition.

Proposition 1: $W_{q,i}$ is defined by the equation shown at the bottom of the page, where

$$P\{g|(k,r)\} = \begin{cases} \alpha_g(t_s), & k+r>0 \\ \alpha_0(t_\alpha)\alpha_{g-1}(t_\mathrm{p}) + \sum_{g_1=1}^{g} \alpha_{g_1}(t_\alpha)\alpha_{g-g_1}(t_\mathrm{p}), & k+r=0 \end{cases} \tag{7}$$

and U is defined as

$$U = \sum_{i=2}^{n} \left\{ \lambda t_m \sum_{q=1}^{\infty} D_{q,i-1} + (\lambda t_\mathrm{p} + 1) D_{0,i-1} \right\} + \lambda t_s \sum_{q=0}^{\infty} D_{q,n} + \mathrm{e}^{-\lambda t_\alpha} \varphi(0,n) D_{0,n}. \tag{8}$$

Proof: See Appendix A. ■

2) Queuing Delay t_q: Let $\psi_{qi}(t_q)$ be the probability that the tagged frame waits t_q slots before it is transmitted the first time, given that there are q frames in the T-buffer, and the slot is in position i at the arriving instant of this tagged frame. Let $L_{qi}(t_q)$ be the corresponding waiting time of this tagged frame. Then, the average value of queuing delay $E[t_q]$ can be given by

$$E[t_q] = \sum_{t_q=0}^{\infty} \sum_{q=0}^{\infty} \sum_{i=1}^{n} L_{qi}(t_q)\psi_{qi}(t_q)R_{q,i} \tag{9}$$

$\psi_{qi}(t_q)$ can be calculated as follows. As shown in Fig. 5, l is the number of bursts experienced by this tagged frame before it is first transmitted and can be obtained by

$$l = \left\lceil \frac{t_q - n + i}{n} \right\rceil + 1 \tag{10}$$

where $\lceil x \rceil$ represents the smallest integer larger than x. This tagged frame is transmitted for the first time in position n_l, which is given by $n_l = t_q - (l-1)n + i$. If the tagged frame can be transmitted in the burst where it arrives, i.e., $l = 1$, this means that the tagged frame is immediately transmitted once these q queued frames have been transmitted. Assume each frame consumes one slot; hence, t_q, which is the number of slots the tagged frames experienced before its first transmission, should be equal to q. Therefore, we have

$$\psi_{qi}(t_q) = \begin{cases} 1, & \text{for } l = 1,\ t_q = q \\ 0, & \text{for } l = 1,\ t_q \neq q \end{cases}. \tag{11}$$

When $l > 1$, the tagged frame can be transmitted if and only if it is at the head of the T-buffer. This means that all q frames have

$$W_{q,i} = \begin{cases} \sum_{k=1}^{q} \sum_{l=q+1-k}^{\infty} \alpha_l(t_m) D_{k,i-1} + \sum_{l=q+1}^{\infty} \alpha_{l-1}(t_\mathrm{p}) D_{0,i-1}, & 1 < i \le n \\ \sum_{r=0}^{n} \sum_{k=0}^{q-r} \sum_{g=q+1-k-r}^{\infty} P\{g|(k,r)\} \varphi(r,n) D_{k,n}, & i = 1 \end{cases}$$

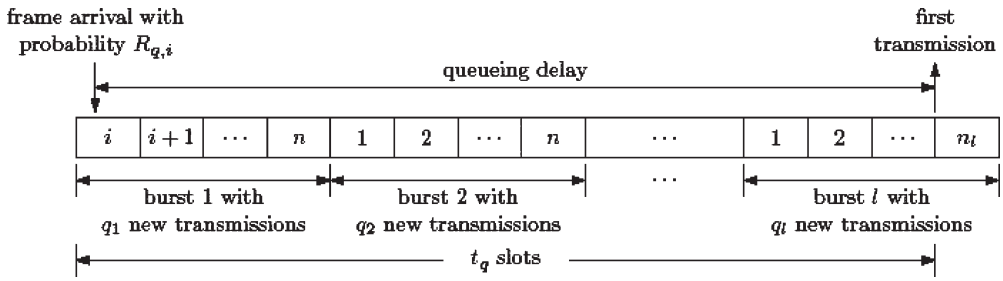


Fig. 5. Queuing delay.

been transmitted (successfully or falsely) at least once. Thus, transmissions of these q new frames can be distributed in all l bursts. According to the policy of Dly-ACK, there are two categories of frames transmitted in one burst: Retransmission frames that have been transmitted in previous bursts, and new frames that are transmitted in current burst for the first time. However, from the viewpoint of the tagged frame, all the q frames in the T-buffer upon its arrival need to be transmitted before its transmission. Thus, the tagged frame regards all these q frames as "new" frames. We actually ignore the probability that, seldom, the retransmitted frame will be discarded due to very many retransmissions, which is usually extremely small. Let q_j be the number of new frames transmitted in burst $j(1 \le j \le l)$. In burst 1, q_1 should be equal to $n - i + 1$ since frames are transmitted in each slot from position i to position n. For burst $j(1 < j < l)$, q_j is in $[0, n]$ by considering that there may be some retransmissions. In burst l, since the tagged frame is transmitted in position n_l, q_l should be distributed in $[0, n_l - 1]$. For convenience, we collect all possible q_j in the set $\Omega_{qi} = \{(q_1, q_2, \ldots, q_l) | \sum_{j=1}^{l} q_j = q\}$, where all q_j satisfy the conditions discussed above. Note that if there are q_j new frame transmissions in burst j, this means that $n - q_j$ frames are transmitted falsely in burst $j - 1$. Therefore, the probability that q_j new frame transmissions happen in burst j is $\varphi(n - q_j, n)$. There are two special cases that should be considered. For burst 1, there are q_1 new transmissions with probability 1 because in each transmission one new frame is transmitted. Note that the tagged frame is actually a new frame for burst l. Then, the retransmission of those frames falsely transmitted in burst $l - 1$ must happen before the transmission of the tagged frame. If q_l new transmissions before the tagged frame happen in burst l, this means that there are $n_l - 1 - q_l$ error transmissions in burst $l - 1$. Therefore, the corresponding probability is $\varphi(n_l - q_l - 1, n)$. For every possible element $(q_1, q_2, \ldots, q_l)$ in set Ω_{qi}, the probability that it happens should be the product of probabilities that each q_j happens consecutively. Finally, $\psi_{qi}(t_q)$ can be obtained by summing the probabilities of each element in Ω_{qi}

$$\psi_{qi}(t_q) = \begin{cases} \sum\limits_{\Omega_{qi}} \varphi(n_l - q_l - 1, n) \prod\limits_{j=2}^{l-1} \varphi(n - q_j, n), & l > 2 \\ \sum\limits_{\Omega_{qi}} \varphi(n_l - q_l - 1, n), & l = 2 \end{cases}. \tag{12}$$

When computing $L_{qi}(t_q)$, there are four scenarios to be considered according to the different values of q and i: 1) $q = 0$, $i \neq 1$; 2) $q = 0$, $i = 1$; 3) $q > 0$, $i \neq 1$; and 4) $q > 0$; $i = 1$. When $q = 0$, this means that the tagged frame is at the head of the T-buffer. If $i \neq 1$, it can be immediately transmitted just after arriving at the T-buffer. Therefore, for the first scenario, $L_{qi}(t_q)$ is equal to 0. Moreover, based on (11), only $t_q = 0$ is meaningful in this scenario, but for $i = 1$, there is a little difference from $i \neq 1$. If the tagged frame arrives after the interval t_a, it can be immediately transmitted. Thus, $L_{qi}(t_q) = 0$. If the tagged frame is generated in interval t_a, it has to wait until the end time of t_a and then can be transmitted. The probability that there is at least one frame that is generated in interval t_a is $1 - \alpha_0(t_a)$. Assuming that the arriving instant of the tagged frame is uniformly distributed in interval t_a, then, if $i = 1$, $L_{qi}(t_q) = ((1 - \alpha_0(t_a) \cdot t_a)/2)$. In summary, when $q = 0$, we have

$$L_{qi}(t_q) = \begin{cases} \frac{t_a}{2} \cdot (1 - \mathrm{e}^{-\lambda t_a}), & i = 1,\ t_q = 0 \\ 0, & \text{otherwise} \end{cases}. \tag{13}$$

Let us consider the case $q > 0$. As shown in Fig. 5, the interval t_q consists of l bursts. For burst $j(1 < j < l)$, its length l_j is equal to t_{b}. For burst l, its length l_l is based on the value of n_l: if $n_l = 1$, l_l should be t_a; if $n_l > 1$, l_l is $t_s + (n_l - 2)t_m + \mathrm{MIFS}$. For burst 1, the arriving instant of the tagged frame needs to be first considered. Since it may arrive at any time in the slot, for simplicity, we approximate the arriving time to be uniformly distributed in the slot, where the tagged frame is generated. Moreover, the length of the slot in position 1 is different from that of the slot in other positions. Hence, when $i = 1$, l_1 is $(n - 1)t_m + (t_s/2)$, and when $i \neq 1$, l_1 is $(n - i)t_m + (t_m/2)$. $L_{qi}(t_q)$ is given by summing all l_j over $1 \le j \le l$. When $q > 0$, by using the relations among t_{b}, t_s, t_m, t_{p} and t_a, the following equation can be obtained:

$$L_{qi}(t_q) = \begin{cases} (l-1)t_{\mathrm{b}} + (n_l - 1)t_m - t_{\mathrm{p}} + \frac{t_s}{2}, & i = 1 \\ (l-1)t_{\mathrm{b}} + (n_l - i)t_m - t_{\mathrm{p}} + \frac{t_m}{2}, & i \neq 1 \end{cases}. \tag{14}$$

C. Delivery Delay t_d

Before computing the delivery delay, we introduce the condition that the tagged frame can be released to a higher layer. The burst in which the tagged frame is transmitted for the first time will be indicated as the fundamental burst.

Definition 1 [block frame (bf)]: If the tagged frame is transmitted in position i of the fundamental burst, its BFs are those frames that are transmitted in position $j(1 \le j \le i)$ of the fundamental burst. For instance, the tagged frame is assumed to be with frame-id 5 and is first transmitted in position 3. While in the fundamental burst, frame 2 and frame 4 are transmitted in positions 1 and 2, respectively. Then, for frame 5, its BFs are frames 2, 4, and 5. According to the policy of Dly-ACK in Section II, we have the following important proposition.

Proposition 2: The tagged frame can be released sequentially to a higher layer if and only if all its BFs have been received correctly by the receiver.

Proof: Since frames are transmitted in increasing order in one burst, the tagged frame's BFs are characterized by a frame id that should not be larger than the one assigned to the tagged frame. If the tagged frame is released to a higher layer, this means that all BFs have been released because all frames need to be released sequentially to a higher layer. In the following, we show that other frames are unable to affect the delivery of the tagged frame other than its BFs. Let us first focus on those frames with frame ids larger than the tagged frame. These frames must be transmitted for the first time after the first transmission of the tagged frame. Moreover, they are unable to affect the release of the tagged frame even if they are transmitted in fundamental burst, because their frame ids are larger than that of the tagged frame. Second, we analyze those frames that have been transmitted in the bursts that are previous to the fundamental burst. If some frames among those frames have been transmitted successfully before the fundamental burst, then all of them are received correctly by the receiver when the tagged frame is transmitted for the first time. Hence, they do not affect the release of the tagged frame. But for others among those frames that have not been transmitted successfully, they have to be retransmitted in fundamental burst. Therefore, only BFs are able to affect the release of the tagged frame. When there is at least one BF that is not successfully received, the tagged frame is not released even if it has been received by the receiver correctly. Once all BFs are received correctly, the tagged frame is immediately released to the higher layer. In summary, only if all BFs of the tagged frame have been successfully received, the tagged frame can be released. ■

We use the same example above to explain this proposition. For Frame 5, it can be released if it is correctly received and Frames 1–4 have been released. In the fundamental burst, Frames 2 and 4 are transmitted in front of frame 5, and this means that Frames 1 and 3 have been correctly received. Hence, once Frames 2 and 4 are correctly received, Frames 1–4 will be released to the higher layer. At this time, if Frame 5 is correctly received, it can be released to the higher layer. Therefore, Frames 1 and 3 do not affect the release of Frame 5, and only Frames 2, 4, and 5 are needed to be considered when analyzing the delivery delay.

For the sake of simplicity, we denote burst 1 as the fundamental burst. Burst 2 is the burst that is successive to the fundamental burst. Bursts 3, 4, etc., may be deduced on the analogy of burst 2. Let $\delta_i(y,k)$ be the probability that there are k BFs that have been incorrectly received at the end of burst y, given that the tagged frame is transmitted in position i for the first time. Note that $0 \le k \le i$, $1 \le i \le n$, and $\delta_i(y,k)$ satisfy the recursion

$$\begin{cases} \delta_i(y,k) = \sum_{j=k}^{i} \delta_i(y-1,j)\varphi(k,j), & y > 0 \\ \delta_i(0,k) = \sigma_{ki} \end{cases} \tag{15}$$

where $\sigma_{ki} = 1$ for $k = i$, and 0 otherwise. $\delta_i(0,k)$ is the initial condition of this function. We assume that there are i BFs at the end of burst 0 (i.e., the beginning of burst 1). Moreover, the probability of having k BFs at the end of burst y is evaluated as the probability of having $j(\in [k,\ldots,i])$ BFs in the previous burst $y-1$ [denoted by term $\delta_i(y-1,j)$] and that exactly k frames of these j BFs are falsely transmitted in burst y [denoted by term $\varphi(k,j)$].

Let $\phi_i(t_d)$ be the probability that the tagged frame experiences t_d slots from the instant of the first time transmission by the sender to the instant of its release time at the receiver, given that the tagged frame is transmitted in position i for the first time. Similar to the analysis in Section III-B2, the interval t_d consists of m bursts, and m is given by

$$m = \left\lceil \frac{t_d - n + i - 1}{n} \right\rceil + 1 \tag{16}$$

and the slot where the tagged frame is released to the higher layer is in the position $c_m = t_d - (m-1)n + i - 1$.

If the tagged frame is released to the higher layer in position c_m of burst m, this means that there are exactly c_m BFs at the end of burst $m-1$ [with probability $\delta_i(m-1,c_m)$], and all c_m BFs are successfully transmitted in burst m [with probability $\varphi(0,c_m)$]. Hence, the probability $\phi_i(t_d)$ can be calculated by

$$\phi_i(t_d) = \delta_i(m-1,c_m)\varphi(0,c_m). \tag{17}$$

Next, we study the length $L_i(t_d)$ of the interval t_d, given that the first transmission of tagged frame is in position i. Since the T-buffer may be empty in the delivery process of tagged frame, there is idle time in some slots. For these slots, their lengths are different from that of the slot in the queuing process of the tagged frame. Let $E[t_i]$ be the mean value of one slot, which is in position i, and we have the following proposition.

Proposition 3: $E[t_i]$ can be obtained from $D_{q,i}$, as follows:

$$\begin{cases} E[t_i] = \left(\frac{1}{\lambda} + t_{\mathrm{p}} - t_m\right) \frac{D_{0,i-1}}{\sum_{q=0}^{\infty} D_{q,i-1}} + t_m, & i > 1 \\ E[t_1] = \frac{1}{\lambda}\varphi(0,n)\alpha_0(t_a)\frac{D_{0,n}}{\sum_{q=0}^{\infty} D_{q,n}} + t_s \end{cases}. \tag{18}$$

Proof: See Appendix B. ■

If b_j is the average length of burst $j(1 \le j \le m)$, then $L_i(t_d) = \sum_{j=1}^{m} b_j$. For burst 1, it consists of $n - i + 1$ slot from position i to position n. Because the tagged frame is transmitted in position i, the length of this slot is t_{p}. Moreover, for those slots whose positions are from $i+1$ to n, they may

TABLE III
SOME SYSTEM PARAMETERS FOR SIMULATIONS

Packet length	1000 bytes
MAC header	10 bytes
PHY header & PLCP preamble	9.4 μs
Imm-ACK frame length	MAC header + PHY header & PLCP preamble
n-Dly-ACK frame length	Imm-ACK frame length + (2n +7)bytes
Basic Data Rate	100 Mbps
Channel rate	100 Mbps
MIFS	2 μs
SIFS	10 μs

have idle time by considering that the T-buffer may be empty. Then, b_1 is given by

$$b_1 = t_{\rm p} + \sum_{k=i+1}^{n} E[t_k]. \tag{19}$$

For burst m, those frames that are transmitted in position 1 to c_m are BFs. This means that they are retransmission frames. Hence, in these slots, where they are transmitted, the T-buffer always has at least one frame, i.e., the T-buffer is nonempty. Therefore, there is no any idle time in these slots. From Fig. 3, we have

$$b_m = t_s + \sum_{k=2}^{c_m} t_m. \tag{20}$$

For burst $j(1 < j < m)$, there are two kinds of slots: one is those slots where retransmission occurs, and another is those slots where new frame is transmitted. For the former, they have no any idle time, and then their lengths are t_s (in position 1) or t_m (in other position). For the latter, since they may include some idle times, their lengths are $E[t_i]$, which depended on their position i. Moreover, in these bursts, we divide the retransmission frames into two groups: those belonging to BFs of tagged frame and those not belonging to BFs. Let s_j be the number of BFs at the end of burst j, and r_j be the number of those frames that are not BFs and are falsely transmitted in burst j. Obviously, the following relation holds: $s_m \le s_{m-1} \le s_{m-2} \le \cdots \le s_1 \le s_0 = i$. Moreover, $0 \le r_j \le n - s_{j-1}$ holds for $1 \le j \le m-1$. If there are s_j BFs and r_j error transmissions (which are not BFs) in burst j, this means that there are $s_j + r_j$ retransmission slots in burst $j+1$. Hence, $b_{j+1} = t_s + \sum_{k=2}^{s_j+r_j} t_m + \sum_{k=s_j+r_j+1}^{n} E[t_k]$ with probability $\varphi(s_j, s_{j-1}) \cdot \varphi(r_j, n - s_{j-1})$. Let Y_i be the set whose elements are all possible permutations of $(s_1, r_1, \ldots, s_{m-1}, r_{m-1})$. Finally, the following equation is obtained:

$$\sum_{j=2}^{m-1} b_j = \sum_{Y_i} \left\{ \sum_{j=1}^{m-2} \left(t_s + \sum_{k=2}^{s_j+r_j} t_m + \sum_{k=s_j+r_j+1}^{n} E[t_k] \right) \times \prod_{j=1}^{m-2} \varphi(s_j, s_{j-1}) \cdot \varphi(r_j, n - s_{j-1}) \right\}. \tag{21}$$

Let $\eta(i)$ be the probability that the tagged frame is transmitted in position i for the first time. Note that the first transmission of tagged frame is a new frame transmission from the viewpoint of fundamental burst. Thus, the probability $\eta(i)$ is also evaluated as the probability that there is one new transmission in position i. Then, $\eta(i)$ should be the ratio, that among all the new frame transmissions, how many frames are transmitted in position i for the first time. Therefore, we have

$$\eta(i) = \lim_{k\to\infty} \frac{\sum_{j=1}^{k} \theta_{j,i}}{\sum_{j=1}^{k} f_j} = \frac{\sum_{r=0}^{i-1} \varphi(r,n)}{\sum_{z=0}^{n} (n-z)\varphi(z,n)} \tag{22}$$

where f_j is the number of new frames transmitted in burst j and $\theta_{j,i} = 1$ for there is a new frame transmission in position i of burst j; otherwise, $\theta_{j,i} = 0$. Note that if there are f_j new transmissions in burst j, this means that there are $n - f_j$ error transmissions in burst $j-1$ [with probability $\varphi(n - f_j, n)$]. Therefore, if $0 \le f_j \le n$, the mean value of f_j should be $\sum_{z=1}^{n} z\varphi(n-z,n)$. If one new transmission occurs in position i of burst j, this means that there are at most $i-1$ error transmissions in burst $j-1$. This is due to the fact that there must be a retransmission in position i of burst j if $r(r \ge i)$ frames are falsely transmitted in burst $j-1$. Thus, $\theta_{j,i} = 1$ occurs with probability $\sum_{r=0}^{i-1} \varphi(r,n)$. The mean value of $\theta_{j,i}$ is $\sum_{r=0}^{i-1} \varphi(r,n)$. Let F be the mean number of new frames transmitted in one burst and V_i be the mean number of new frames transmitted in position i for the first time. Obviously, F and V_i are the mean values of f_j and $\theta_{j,i}$, respectively. Since $F = \lim_{k\to\infty} \sum_{j=1}^{k} f_j/k$, $V_i = \lim_{k\to\infty} \sum_{j=1}^{k} \theta_{j,i}/k$, and $\eta(i) = F/V_i$, (22) can be obtained.

Finally, the average value $E[t_d]$ for delivery delay can be expressed as

$$E[t_d] = \sum_{i=1}^{n} \sum_{t_{d=0}}^{\infty} \eta(i) L_i(t_d) \phi_i(t_d). \tag{23}$$

IV. SIMULATION RESULTS AND DISCUSSIONS

In this section, we present simulation results including the end-to-end delay, the queuing delay, and the delivery delay. The results help to validate our analysis, and understand the key factors that impact the delay performance of Dly-ACK, such

as the traffic load G, the frame error probability p, and various values of burst size n.

To verify the analytical results, they are compared with those obtained from the well-known simulator ns $-$ 2 [16]. In the simulation, the typical WPAN system configurations are used. Some important parameters are listed in Table III, and others are chosen according to [1] and [4].

As we mentioned earlier, from the viewpoint of the throughput, the Dly-ACK policy is superior to the Imm-ACK. However, it is not necessarily true for the delay performance. To understand the impact of traffic load G on the delay performance, Fig. 6(a) shows the end-to-end delay in a UWB system when using n-Dly-ACK policy under various traffic load G, given that $p = 0.05$. In the figure, the real lines represent our analytic results while the simulation results are denoted by various separated symbols. Unless stated otherwise, we use the same representation for the comparison of analytical and simulation results in the rest of this section. From Fig. 6(a), we can see that a perfect agreement is observed between analysis and simulation. For $G = 0.2$, the 1-Dly-ACK policy, i.e., Imm-ACK, has the best performance. Moreover, the delay performance becomes worse with the increase of burst size n. When $G = 0.4$, the best performance occurs at the point $n = 2$. For $G = 0.5$, 1-Dly-ACK policy becomes the worst, while 3-Dly-ACK is the best. Furthermore, for $G = 0.6$ and 0.7, delay becomes worse when using 1-Dly-ACK because the system is more saturated. The 5-Dly-ACK and 8-Dly-ACK are the optimal policies for $G = 0.6$ and $G = 0.7$, respectively. In conclusion, for a given p, there is one corresponding optimal Dly-ACK policy for a certain traffic input load G. Moreover, the optimal burst size n becomes larger with the increase of G. The similar behaviors can be observed in Fig. 6(b) and (c), where $p = 0.01$ and $p = 0.10$, respectively. In fact, for other values of p, there is also a similar relationship between the traffic input load G and the optimal burst size n.

Fig. 7 shows how the traffic load G affects the queuing delay, while Fig. 8 is for the delivery delay. Both figures indicate that our analysis matches well with the simulations. For all values of G, the Dly-ACK policy that has a larger burst size n will have a less queuing delay. This is due to the fact that when n is larger, the frames queued in the T-buffer will be fewer because more frames can be transmitted in one burst. Therefore, the sender is able to transmit frames faster so that each frame stays in the sender's traffic queue for less time as the burst size increases. Furthermore, from Fig. 7, we can see that with the increase of G, the reduction of the queuing delay is more remarkable when using larger n. However, as shown in Fig. 8, the delivery delay will become larger with the increase of n. When n is larger, the probability that one frame needs to wait for those frames with lower frame ids is larger. Moreover, the larger n means that more time is needed to feedback one frame's transmission result (failure or success). Specially, for larger n (for example, $n = 9$) and smaller $G(G = 0.2)$, most of the delivery delay is due to waiting to fill in one burst in the sender. Thus, for 1-Dly-ACK, it has the fewest deliver delay. Since the impact of burst size n on the queuing delay is completely opposite to that on the delivery delay, there exists an optimal n, which minimizes the summation of $E[t_d] + E[t_q]$ for a given G, and the optimal n depends on G.

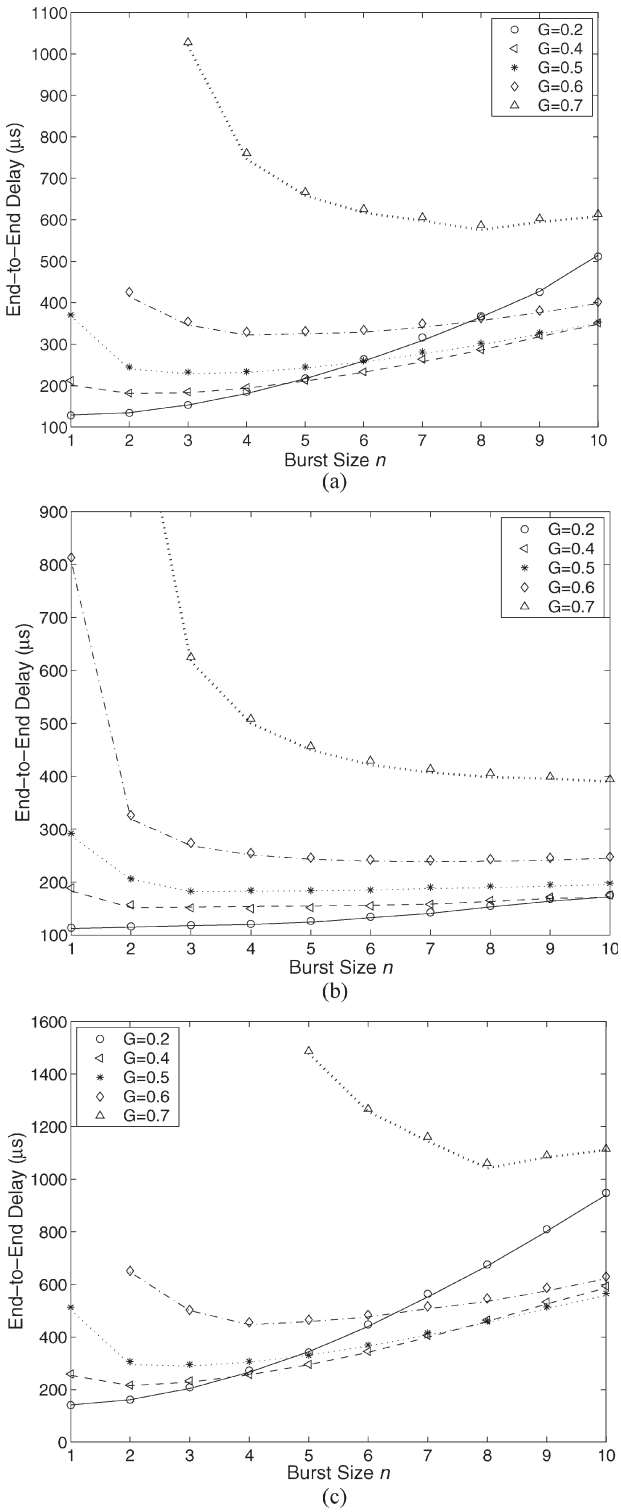


Fig. 6. (a) End-to-end delay versus various traffic load $G(p = 0.05; H =$ 100 Mb/s). (b) End-to-end delay versus various traffic load $G(p = 0.01; H =$ 100 Mb/s). (c) End-to-end delay versus various traffic load $G(p = 0.10; H =$ 100 Mb/s).

The impact of p on the end-to-end delay performance is shown in Fig. 9 with $G = 0.2$. It can be seen that the end-to-end delay becomes larger with the increase of p, considering that one frame needs to be retransmitted more times before successfully received by the receiver. From this figure, an interesting

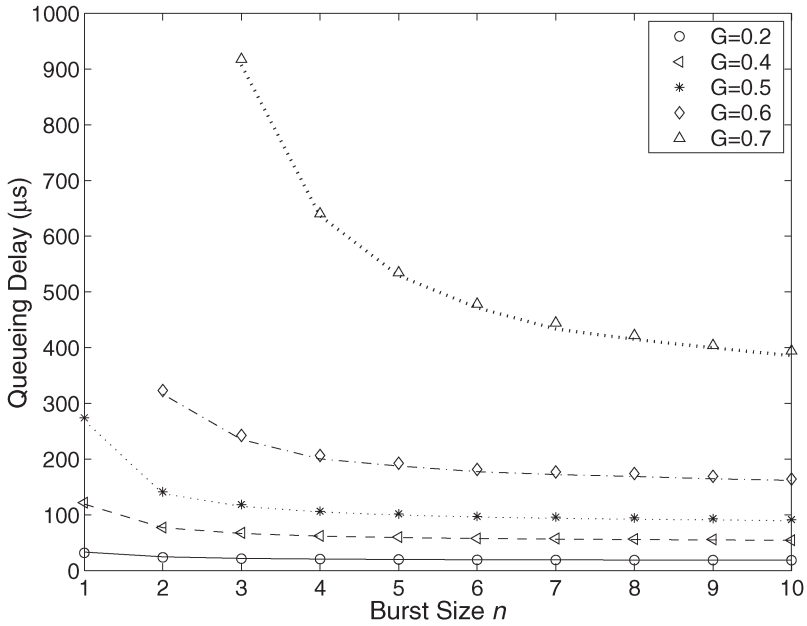


Fig. 7. Queuing delay versus various traffic load $G(p = 0.05; H = 100$ Mb/s).

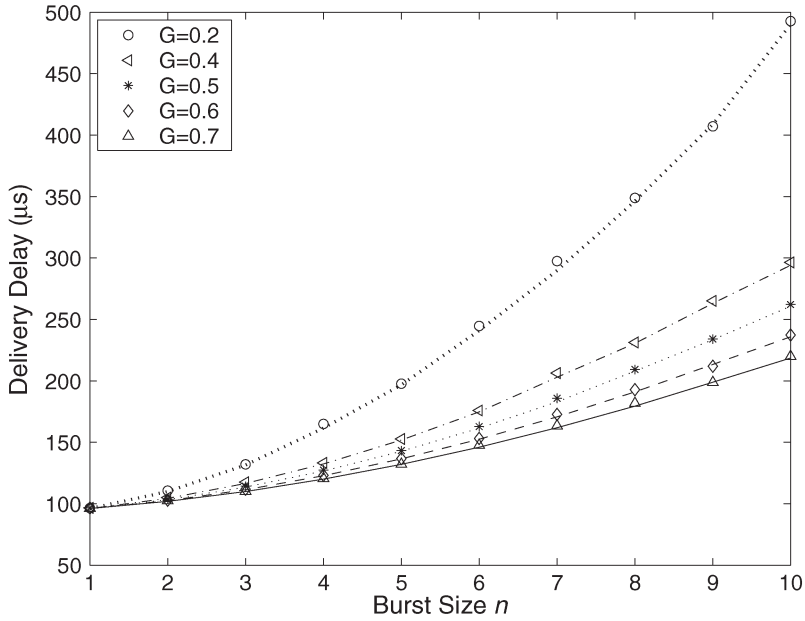


Fig. 8. Delivery delay versus various traffic load $G(p = 0.05; H = 100$ Mb/s).

behavior can be observed: Given the traffic load G, the optimal burst size $n(n = 1)$ is almost independent of p within a reasonable frame error-rate range, say, $p = [0.01 - 0.2]$. Moreover, the curves of delay versus n have the same trend under various p. This means that the optimal burst size is heavily dependent on the input traffic load G but very insensitive to the error probability p. In Fig. 10, where $G = 0.5$, we can find exactly the same phenomenon (here, the optimal $n = 3$). Furthermore, we illustrate the impacts of p on the queuing delay and the delivery delay separately in Fig. 11 with $G = 0.5$. We can see that the delivery delay increases with the increase of n, while the queuing delay decreases with the increase of n. This trend is irrespective of the choice of error probability p.

V. Dynamic Tuning of Burst Size

As discussed in the previous section, the optimal burst size n versus G can be shown approximately in Fig. 12. There are two difficulties to determine the optimal burst size n. First, the points $g_1, \ldots, g_{n_{\max}}$ are difficult to obtain. Second, even if these points are determined by our simulation or analysis, the input traffic load G is difficult to be obtain online because it may vary with time. Furthermore, when the input traffic is not Poisson, it will be more complex to determine the optimal burst size. As a result, we present a simple but very effective heuristic method to determine the selection of the burst size. In fact, the number of frames queued in the T-buffer is able to reflect the input traffic load to some extent. When G is

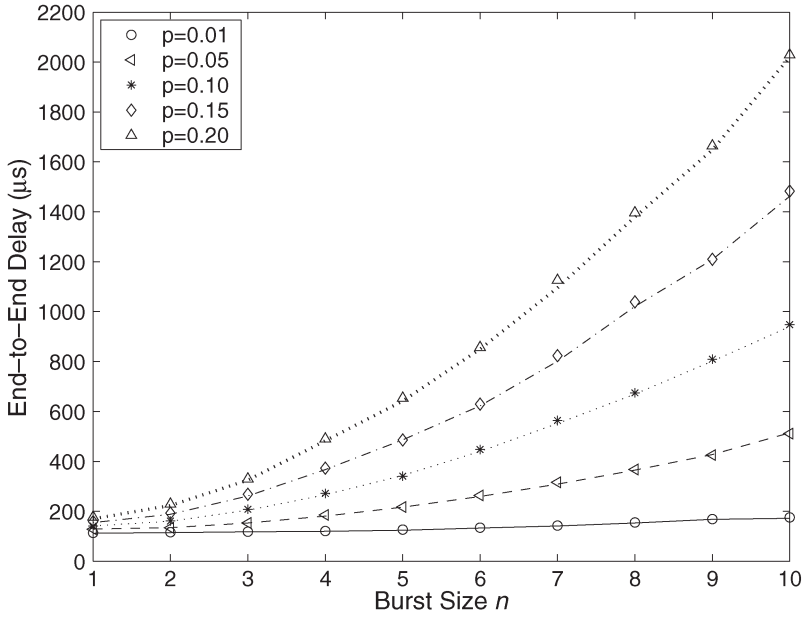


Fig. 9. End-to-end delay versus various error probabilities $p(G = 0.2; H = 100$ Mb/s$)$.

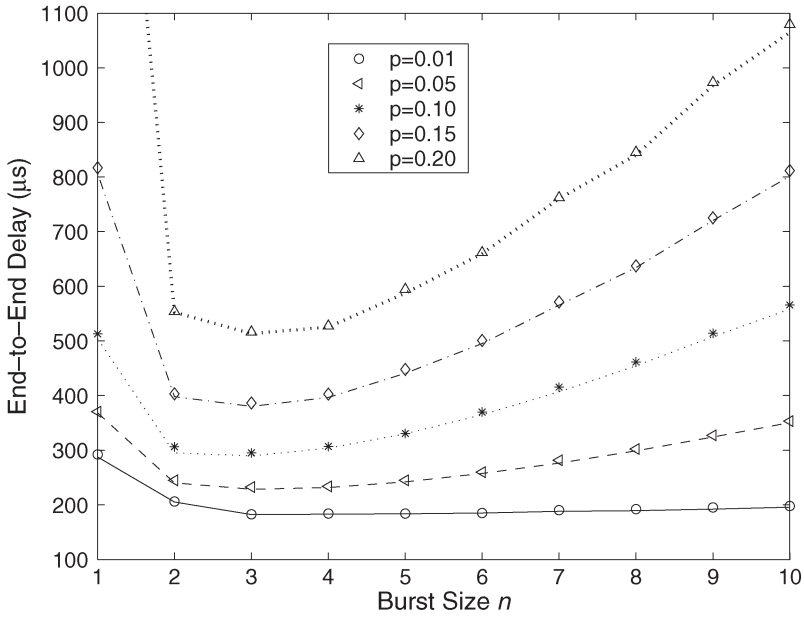


Fig. 10. End-to-end delay versus various error probabilities $p(G = 0.5; H = 100$ Mb/s$)$.

low, this number should be small, whereas the queuing frame number in the T-buffer will be large if the traffic load is high. In a realistic network, the arrival of traffic may be bursty. At some time instances, there may be lots of frames queued in the T-buffer, or the T-buffer may be empty. Therefore, the selection of burst size must reflect this fluctuation. In other words, we can determine the burst size based on the number of frames queued in the T-buffer, rather than on the value G, which represents a macroscopical quantity of the whole traffic. We also do not need to estimate the points g_1, etc. Since a higher G needs a larger n to be optimized, the sender should transmit frames and does not need feedback information as long as there are frames in the T-buffer. More importantly, from the throughput point of view, the n-Dly-ACK policy is always superior to $(n-1)$-Dly-ACK. Therefore, we should select a large value n when it is possible. The proposed dynamic Dly-ACK policy is as follows.

1) Before transmitting one frame, the sender detects whether the T-buffer is empty or not after this frame. If the T-buffer is empty, the sender requests a Dly-ACK frame by setting the Dly-ACK request bit in the MAC header [1]; else, go to 2).
2) The sender possibly transmits more frames as long as the T-buffer is not empty. Once the number of frames transmitted in one burst is larger than a given value $n_{\max}$, the sender requests a Dly-ACK frame.

We call this Dly-ACK policy the Dynamic Dly-ACK (D-Dly-ACK). In fact, it acts as the $n_{\max}$-Dly-ACK policy at most

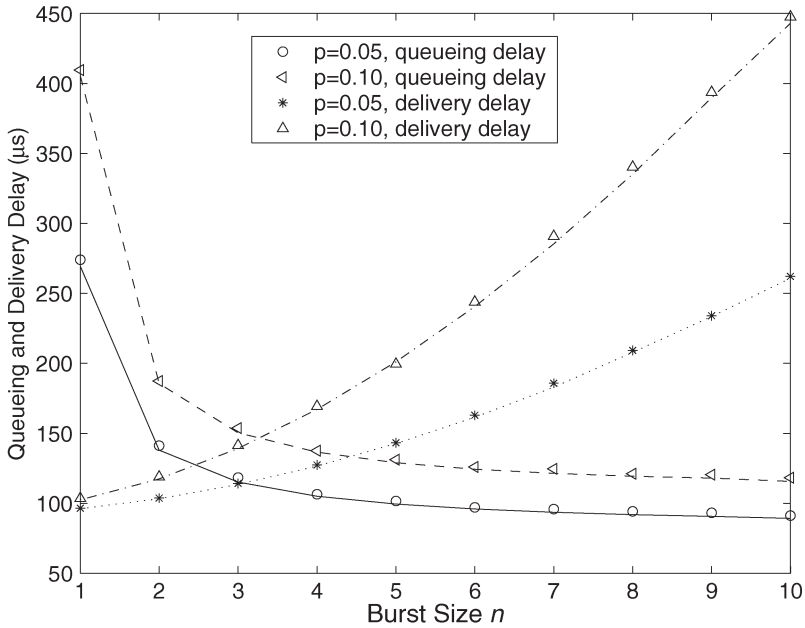


Fig. 11. Queuing and delivery delays versus various error probabilities $p(G = 0.5; H = 100$ Mb/s).

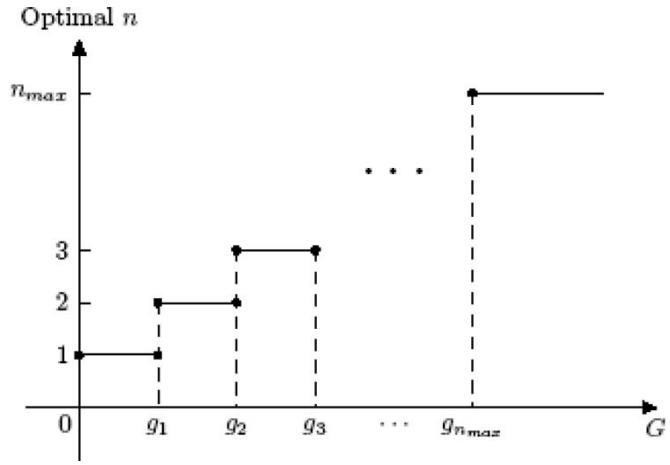
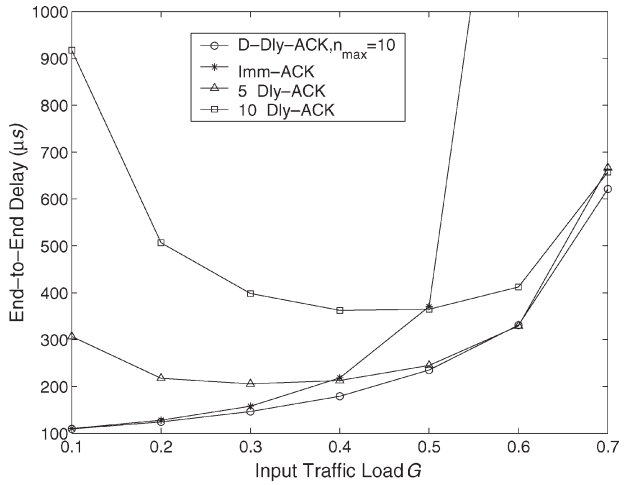


Fig. 12. Optimal burst size versus input traffic load.

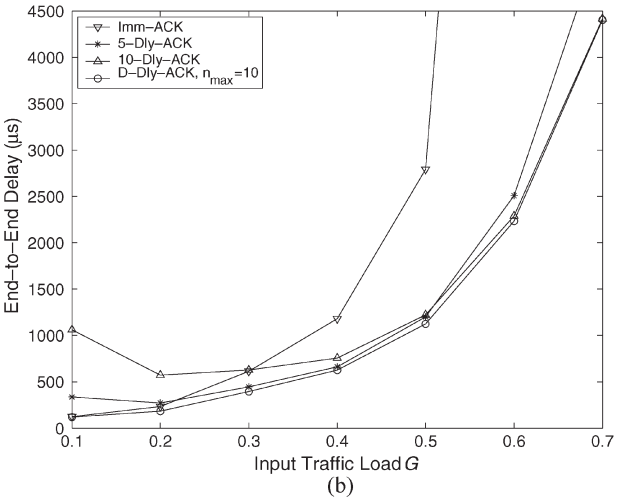
of the time if the input traffic load is high. When G is low, its burst size is changed from 1 to $n_{\max}$ adaptively. While it is able to make the best use of the bandwidth, it can also achieve a pretty good delay performance. Here, we can choose a proper parameter $n_{\max}$ so that the system improvement is trivial when the burst size is larger than $n_{\max}$, particularly when the traffic load is large. This can be seen from the results shown in the previous sections. For example, in Fig. 2, the MEB is almost constant when $n > 10$. In general, we select the parameter $n_{\max}$ being around 10. Fig. 13(a) shows that our scheme is superior over all other Dly-ACK schemes with fixed burst size in terms of the delay performance when the Poisson traffic is used. In order to further demonstrate our D-Dly-ACK performance under non-Poisson traffic, we present the delay-performance comparison for self-similarity traffic in Fig. 13(b). Here, we use the traffic generator SS in [17] to generate self-similarity traffic. The generator SS is based on superposition of a number of Markov modulated Poisson processes (MMPPs) and has five associated parameters: rate_, packet_size_, correlation_, burst_, and time_scales_. In Fig. 13(b), the parameters correlation_, burst_, and time_scales_ are set to the default values 0.6, 0.82, and 0.5, respectively; the parameter packet_size_ is set to f B, and rate_ is set to a corresponding value based on G. It can be seen that D-Dly-ACK performs best. Note that D-Dly-ACK reduces the delay without the throughput penalty. The throughput comparison between D-Dly-ACK and fixed-burst-size Dly-ACK is shown in Fig. 13(c) for Poisson traffic. From Fig. 13(c), the throughput improvement with Dly-ACK over Imm-ACK can also be observed. That is, when $G > 0.6$, the system begins to saturate with Imm-ACK, while it can still afford more injected traffic with Dly-ACK. For self-similarity traffic, the results are similar. To further investigate D-Dly-ACK, we illustrate the distribution of burst size of D-Dly-ACK with $p = 0.05$ and Poisson traffic in Table IV. It can be seen that D-Dly-ACK can adaptively change the burst size according to the traffic load. As to $n_{\max}$, its effects are shown in Fig. 14 with Poisson traffic $G = 0.7$, and $p = 0.05$ and 0.1, respectively. We can see that in the two scenarios, the delay becomes smaller with the increase of $n_{\max}$ when $n_{\max}$ is lower than 8. However, when $n_{\max}$ is larger than 8, the delay is almost stable. Therefore, it is unnecessary to select $n_{\max}$ to be larger than 10.

VI. Conclusion

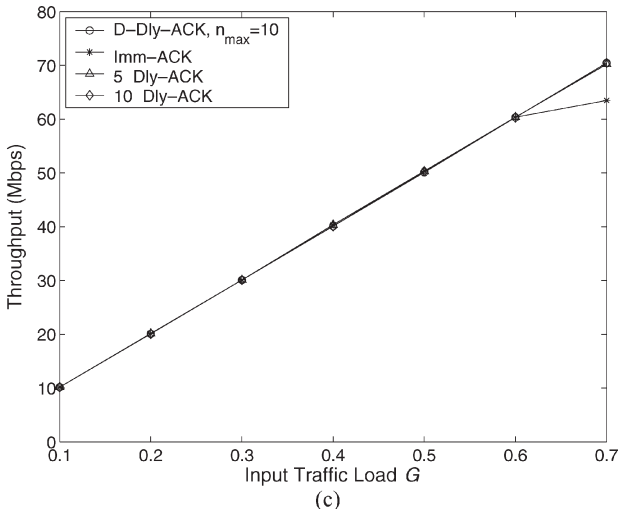
We have comprehensively studied the delay performance of Dly-ACK for high-speed WPAN. It is concluded that the optimal burst size is heavily dependent on the input traffic volume and is not sensitive to the channel error rate. Therefore, the proposed dynamic Dly-ACK scheme, which tunes the burst size adaptively, can significantly improve the delay performance.



(a)



(b)



(c)

Fig. 13. (a) Delay comparison by using D-Dly-ACK policy ($p = 0.05$; $H = 100$ Mb/s, Poisson traffic). (b) Delay comparison by using D-Dly-ACK policy ($p = 0.05$; $H = 100$ Mb/s, self-similarity traffic). (c) Throughput comparison by using D-Dly-ACK ($p = 0.05$; $H = 100$ Mb/s, Poisson traffic).

APPENDIX A
PROOF OF PROPOSITION 1

We know that the new frames that belong to a state are generated only in position i. Thus, $W_{q,i}$ can be computed only by analyzing what will happen in position i. Since the slot in position 1 includes the ACK frame, it should be considered individually. First, we compute $W_{q,i}$ for $1 < i \leq n$. Assume that the tagged frame arrives in position i of slot t. Let $W_{q,i,k}$ be the mean number of frames that belong to state $\{q, i\}$, given that there are k frames queued in the T-buffer at the end of slot $t-1$. Obviously, the slot $t-1$ is in position $i-1$. According to the definition $D_{k,i}$, we have

$$W_{q,i} = \sum_{k} W_{q,i,k} \cdot D_{k,i-1}. \tag{24}$$

If the tagged frame is the first arrival among those frames generated in slot t, it should belong to state $\{k, i\}$. Accordingly, the tagged frame belongs to state $\{k+j-1, i\}$ if it is the jth arriving frames in slot t. Then, if the tagged frame belongs to state $\{q, i\}$, this means that it is the $(q+1-k)$th frame generated in slot t. Let g be the number of new frames generated in slot t, then, only when $g \geq q+1-k$, the state $\{q, i\}$ may occur. Considering that the state $\{q, i\}$ occurs only once in slot t, we only need to know the probability that state $\{q, i\}$ occurs in slot t when computing $W_{q,i,k}$. If state $\{q, i\}$ occurs in slot t, this means that there are at least $q+1-k$ frames generated in slot t. Thus, $W_{q,i,k}$ can be obtained by summing the probability $P\{g|k\}$ over $g \geq q+1-k$, where $P\{g|k\}$ denotes the probability that there are g new frames generated in slot t, given that there are k frames queuing in the T-buffer at the end of slot $t-1$ (note that the buffer status, i.e., empty or not, determines the length of slot t). When $k = 0$, the probability $P\{g|k\}$ is $\alpha_{g-1}(t_{\mathrm{p}})$, since the first new frame is transmitted in slot t. Accordingly, the probability $P\{g|k\}$ is $\alpha_g(t_m)$ if $k > 0$. Thus, $W_{q,i,k}$ can be calculated as

$$W_{q,i,k} = \begin{cases} \sum\limits_{g=q+1-k}^{\infty} \alpha_g(t_m), & k > 0 \\ \sum\limits_{g=q+1}^{\infty} \alpha_{g-1}(t_{\mathrm{p}}), & k = 0 \end{cases}. \tag{25}$$

Actually, if $k > q$, any new frame will not belong to state $\{q, i\}$. Finally, by substituting (25) into (24), we can obtain the first equation of (7).

TABLE IV
DISTRIBUTION OF BURST SIZE N OF Dly-ACK

burst-size n	G		
	0.3	0.5	0.7
1	15685 (91.53%)	16284 (79.90%)	7044 (55.61%)
2	841 (4.91%)	1581 (7.76%)	908 (7.17%)
3	304 (1.77%)	790 (3.86%)	529 (4.18%)
4	151 (0.88%)	448 (2.20%)	364 (2.87%)
5	65 (0.38%)	315 (1.55%)	308 (2.43%)
6	38 (0.22%)	210 (1.03%)	255 (2.01%)
7	24 (0.14%)	154 (0.76%)	198 (1.56%)
8	14 (0.08%)	132 (0.65%)	198 (1.56%)
9	7 (0.04%)	99 (0.49%)	153 (1.21%)
10	7 (0.04%)	369 (1.81%)	2710 (21.39%)
Total	17136	20382	12667

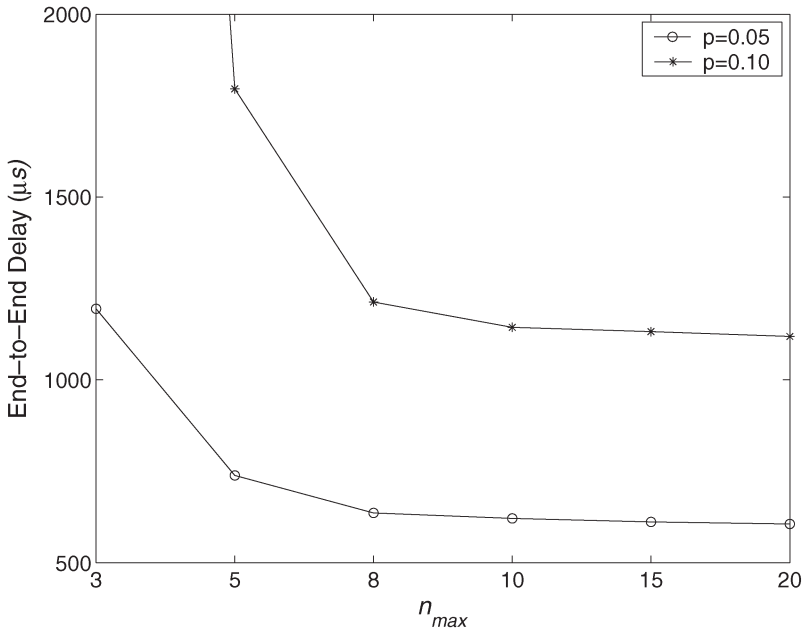


Fig. 14. Effects of parameter $n_{\max}$ ($G = 0.7$; $H = 100$ Mb/s, Poisson traffic).

On the other hand, for $i = 1$, we need to consider the number of frames r that is falsely transmitted. At the beginning of slot t, there should be $k + r$ frames queued in the T-buffer. Similar to the case $i \neq 1$, we only need to compute $W_{q,i,k,r}$, given k and r. Also, $P\{g|k, r\}$ is the probability that there g new frames generated in slot t, given k and r. If $k + r > 0$, g new frames are distributed in interval t_s, and $P\{g|k, r\}$ is $\alpha_g(t_s)$. When $k + r = 0$, we should handle these two interval t_a and $t_{\rm p}$ separately. If $g_1 (g_1 > 0)$ new frames are generated in interval t_a [with probability $\alpha_{g_1}(t_a)$], this means that there are no any idle time between t_a and $t_{\rm p}$. The other $g - g_1$ new frames are generated in $t_{\rm p}$ [with probability $\alpha_{g-g_1}(t_{\rm p})$]. If there are no any new frame generated in t_a, i.e., $g_1 = 0$, this means that one new frame must be generated between t_a and $t_{\rm p}$. Thus, there are $g - 1$ new frames generated in $t_{\rm p}$. Since $P\{g|k, r\}$ can be obtained by summing up $0 \leq g_1 \leq g$, given that there are r error transmissions with probability $\varphi(r, n)$, we have the second equation of (7).

Next, we compute U, which is the sum of mean number of new frames generated in each position i. For $1 < i \leq n$, if $k > 0$ (with probability $\sum_{k=1}^{\infty} D_{k,i-1}$), this means that slot t has no idle time and is equal to t_m. Thus, there are on average λt_m frames generated in slot t. If $k = 0$ (with probability $D_{0,i-1}$), slot t consists of idle time and interval $t_{\rm p}$. There is only one new frame generated at the end of idle time, while on average $\lambda t_{\rm p}$ frames are generated in $t_{\rm p}$. For $i = 1$, there are λt_s frames generated in slot t if $k + r > 0$. For $k + r > 0$, there are two cases: One is $k > 0$ (with probability $\sum_{k=1}^{\infty} D_{k,n}$); the other is $k = 0$, and $r > 0$ [with probability $D_{0,n}(1 - \varphi(0, n))$]. When $k + r = 0$ [with probability $D_{0,n}\varphi(0, n)$], t_a and $t_{\rm p}$ should be analyzed individually. For t_a, there are λt_a new frames. However, the average number of new frames generated in $t_{\rm p}$ is dependent on whether there are new frames generated in t_a or not. If there are some new frames generated in t_a [with probability $1 - \alpha_0(t_a)$], on average, $\lambda t_{\rm p}$ new frames will be generated in $t_{\rm p}$. If no frame is generated in t_a, there are on average $\lambda t_{\rm p} + 1$ new frames generated in $t_{\rm p}$, since one new frame must be generated just before the beginning of $t_{\rm p}$. By considering all the above conditions, (8) can be obtained. Actually, by some algebraic manipulations, it can be obtained that $U = \sum_{i=1}^{n} \sum_{q=0}^{\infty} W_{q,i}$.

APPENDIX B
PROOF OF PROPOSITION 3

When computing $E[t_i]$, we only need to consider the mean length of idle time in one slot. Since λ represents the average frame arrival rate, the average length of one idle time should be equal to $1/\lambda$. For the slots in position $i (i \neq 1)$, if there are slots that include some idle time, this means that T-buffer is empty just before the beginning of these slots. Thus, these slots consist of two parts: idle time with average length $1/\lambda$ and one frame transmission $t_{\rm p}$. If the T-buffer has some frames, the lengths of these slots are t_m, because one frame is immediately transmitted after one MIFS. Also, the probability that there is no frame in the T-buffer just before the beginning of these slots is $D_{0,i-1}$. Because $E[t_i]$ only aims at the slots that are in position i but not all slots, we must normalize the probability $D_{0,i-1}$ [the term $\sum_{q=0}^{\infty} D_{q,i-1}$ in (18)].

On the other hand, for those slots in position 1, if they include some idle time, their mean lengths should be $1/(\lambda + t_s)$. Otherwise, the lengths are equal to t_s. When some idle time occurs in these slots, three conditions must be satisfied: 1) There is no frame queued in the T-buffer at the end of position n of the previous burst (with probability $D_{0,n}$); 2) no frame is falsely transmitted in the previous burst [with probability $\varphi(0, n)$]; and 3) there is no new frame generated in t_a [with probability $\alpha_0(t_a)$]. In addition, the probability $D_{0,n}$ is normalized. We then have (18).

REFERENCES

[1] IEEE, *Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPAN)*, Sep. 2003.

[2] *MBOA MAC Specification for High Rate WPANs*, Draft 0.65, Oct. 15, 2004.

[3] UWB forum. [Online]. Available: http://www.uwbforum.org/

[4] Multiband OFDM Alliance. [Online]. Available: http://www.multibandofdm.org/

[5] H. Liu, H. Ma, M. El Zarki, and S. Gupta, "Error control schemes for networks: An overview," *Mob. Netw. Appl.*, vol. 2, no. 2, pp. 167–182, Jun. 1997.

[6] M. Zorzi, "Some results on error control for burst-error channels under delay constraints," *IEEE Trans. Veh. Technol.*, vol. 50, no. 1, pp. 12–24, Jan. 2001.

[7] M. Zorzi and R. R. Rao, "Lateness probability of a retransmission scheme for error control on a two-state Markov channel," *IEEE Trans. Commun.*, vol. 47, no. 10, pp. 1537–1548, Oct. 1999.

[8] M. Rossi and M. Zorzi, "Analysis and heuristics for the characterization of selective repeat ARQ delay statistics over wireless channels," *IEEE Trans. Veh. Technol.*, vol. 52, no. 5, pp. 1365–1377, Sep. 2003.

[9] J. G. Kim and M. M. Krunz, "Delay analysis of selective repeat ARQ for a Markovian source over a wireless channel," *IEEE Trans. Veh. Technol.*, vol. 49, no. 5, pp. 1968–1981, Sep. 2000.

[10] L. Kleinrock, *Queuing Systems, Volume II: Computer Applications*. New York: Wiley, 1976.

[11] S. Ata, M. Murata, and H. Miyahara, "Analysis of network traffic and its application to design of high-speed routers," *IEICE Trans. Inf. Syst.*, vol. E83-D, no. 5, pp. 988–995, May 2000.

[12] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun, "On the non-stationary of Internet traffic," in *Proc. ACM SIGMETRICS*, Cambridge, MA, Jun. 2001, pp. 102–112.

[13] ——, "Internet traffic tends to Poisson and independent as the load increases," Bell Labs, Murray Hill, NJ, Bell Labs Tech. Rep., 2001.

[14] Svigelj, M. Mohorcic, G. Kandus, A. Kos, M. Pustisek, and J. Bester, "Routing in ISL networks considering empirical IP traffic," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 2, pp. 261–272, Feb. 2004.

[15] F. Chiasserini and M. Meo, "Impact of ARQ protocols on QoS in 3GPP systems," *IEEE Trans. Veh. Technol.*, vol. 52, no. 1, pp. 205–215, Jan. 2003.

[16] Network Simulator 2. [Online]. Available: http://www.isi.edu/nsnam/ns/

[17] M. Yuksel, B. Sikdar, K. S. Vastola, and B. Szymanski, "Workload generation for ns simulations of wide area networks and the Internet," in *Proc. Communication Networks and Distributed Systems Modeling and Simulation Conf.*, San Diego, CA, 2000, pp. 93–98.

Hongyuan Chen was born in Hunan, China, in 1980. He attended in the Special Class for the Gifted Young, Xi'an Jiaotong University (XJTU), Xi'an, China, in 1995. He received the B.S. degree in information engineering from XJTU in 2000 and the M.S. and the Ph.D. degrees from Tsinghua University, Beijing, China, in 2002 and 2005, respectively.

His current research interests include mesh networks, WLAN, and wireless personal area network (WPAN).

Zihua Guo (S'99–M'01) received the B.S. and M.S. degrees from the University of Science and Technology, Beijing, China, in 1995 and 1998, respectively. He received the Ph.D degree from the Hong Kong University of Science and Technology, Hong Kong, in 2001.

His research interests include wireless communications and networking, 3G and beyond technologies, multimedia communications, signal processing, etc. He has published more than 30 IEEE journal papers and conference papers. He is now with Lenovo Corporate Research, Beijing.

Richard Yuqi Yao (S'94–M'96) received the M.S. and Ph.D. degrees in electrical engineering from Polytechnic University, Brooklyn, NY, in 1994 and 1996, respectively.

In 1996, he joined the Bell Labs of Lucent Technologies, Whippany, NJ, as a member of Technical Staff, where he worked on the system performance analysis of 2G, 2.5G, and 3G CDMA wireless networks. From 2000 to 2002, he was a Technical Manager in PacketVideo Corporation, Rochelle Park, NJ, where he performed system analysis and field trial of MPEG-4 video and audio over various 2.5G and 3G wireless networks. Since 2002, he has been with the Microsoft Research Asia, where he is a Researcher and Project Lead in Wireless and Networking Group. His current research areas include ultra-wideband (UWB) technology, 3G and B3G wireless technologies and networks, and multimedia communications and applications. He has published more than 50 conference, journal papers, and book chapters.

Dr. Yao is a member of Sigma Xi.

Xuemin (Sherman) Shen (M'97–SM'02) received the B.Sc. degree from Dalian Maritime University, Dalian, China, in 1982 and the M.Sc. and Ph.D. degrees from Rutgers University, Piscataway, NJ, in 1987 and 1990, respectively, all in electrical engineering.

From September 1990 to September 1993, he was first with the Howard University, Washington, DC, and then with the University of Alberta, Edmonton, AB, Canada. Since October 1993, he has been with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, where he is a Professor and the Associate Chair for Graduate Studies. He is the author or coauthor of two books and more than 200 papers and book chapters in wireless communication and network control and filtering. His research interests focus on the mobility and resource management in interconnected wireless/wireline networks, ultra-wideband (UWB) wireless communications systems, wireless security, and *ad hoc* and sensor networks.

Dr. Shen serves as the Technical Program Committee Chair for Qshine'05, Co-Chair for IEEE Broadnet'05, WirelessCom'05, International Federation for Information Processing (IFIP) Networking'05, 2004 International Symposium on Parallel Architectures, Algorithms and Networks, and IEEE Globecom'03 Symposium on Next Generation Networks and Internet. He is also an Associate Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the Association for Computing Machinery (ACM) *Wireless Network*, *Computer Networks*, *Dynamics of Continuous, Discrete and Impulsive Systems–Series B: Applications and Algorithms*, *Wireless Communications and Mobile Computing* (Wiley); and the *Computer Networks* (Elsevier). He has been Guest Editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC), *IEEE Wireless Communications*, and *IEEE Communications Magazine*. He received the Premier's Research Excellence Award (PREA) from the province of Ontario, Canada, for the demonstrated excellence of his scientific and academic contributions in 2003 and the Distinguished Performance Award from the Faculty of Engineering, University of Waterloo, for his outstanding contribution in teaching, scholarship, and service in 2002. He is a registered Professional Engineer in Ontario.

Yanda Li (SM'87) was born in 1936 in Guangdong, China. He received the B.S. degree from the Department of Automatic Control, Tsinghua University, Beijing, China, in 1959.

He became a Faculty Member in the Department of Electrical Engineering, Tsinghua University, in 1959. He joined the Department of Automation, Tsinghua University, in 1970. During the academic years 1979–1981, he was a Visiting Scholar in the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (MIT), Cambridge. His current research interests include digital signal processing, neural networks, intelligent control, and wireless communication.

Mr. Li has been a Fellow of the Chinese Academy of Sciences since 1991.