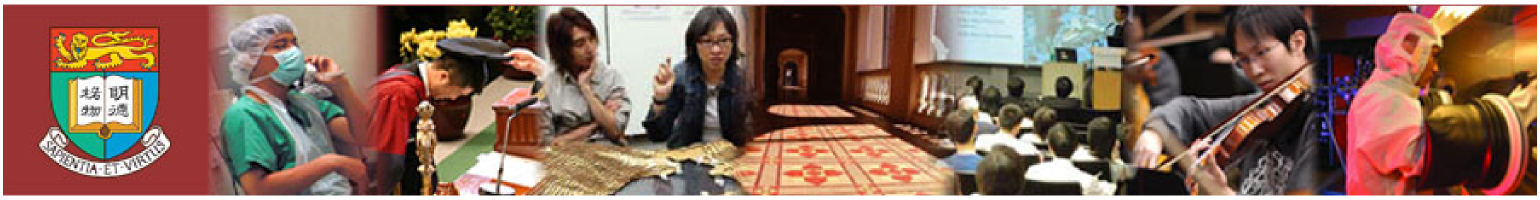


Title	End-to-End Delay Distribution Analysis for Stochastic Admission Control in Multi-hop Wireless Networks
Author(s)	Jiao, W; Sheng, M; Wong Lui, KS; Shi, Y
Citation	IEEE Transactions on Wireless Communications, 2014, v. 13 n. 3, p. 1308-1320
Issued Date	2014
URL	http://hdl.handle.net/10722/216925
Rights	Creative Commons: Attribution 3.0 Hong Kong License

End-to-End Delay Distribution Analysis for Stochastic Admission Control in Multi-hop Wireless Networks

Wanguo Jiao, Min Sheng, *Member, IEEE,* King-Shan Lui, *Senior Member, IEEE,* and Yan Shi, *Member, IEEE*

***Abstract*—Admission control is important in achieving QoS guarantees in multi-hop wireless networks. An efficient admission control algorithm requires an accurate estimation of the end-to-end delay distribution of the network. In this paper, we propose a method to estimate the end-to-end delay distribution under the general traffic arrival process and Nakagami-m channel model. We firstly propose a novel two-dimensional Markov Chain to model the node behaviors in a multi-hop multi-rate IEEE 802.11 network that is subject to interference and error prone channel. By combining the basic Probability theory and Network Calculus, we analyze the delay a packet experiences at each hop along a path. The per-hop delay result is used to develop the distribution of the end-to-end delay of a randomly chosen path. We then develop an admission control scheme for the traffic with stochastic QoS guarantees. Finally, through simulation results, we verify the accuracy of our analytical model and the effectiveness of the proposed algorithm.**

***Index Terms*—Stochastic admission control, end-to-end delay, IEEE 802.11 DCF, multi-hop wireless networks, effective bandwidth.**

I. Introduction

With growing demand for real-time applications over wireless networks, increasing attention has been paid to real time quality-of-service (QoS) guarantees in wireless networks. The QoS requirements, in terms of delay bounds, delay variations, and delay violation probabilities, of different applications are very diverse [1]. For example, some applications, such as control signaling of cellular networks, require *hard guarantees* that all packets must arrive the destination within a certain delay bound. On the other hand, most multimedia applications, including voice over internet phone (VoIP), multimedia streaming, and online game, can tolerant a certain probability of QoS violation [2]. The QoS guarantees of these applications are referred as *soft* QoS guarantees. The delay requirement of applications requiring soft QoS guarantees can be expressed as $\Pr(ETE\ DELAY > D_{\max}) \leq \varepsilon$, which means the probability that the delay of the traffic exceeding $D_{\max}$ should be smaller than ε. Because of the randomness property of wireless networks, such as wireless channel unreliability and sharing, providing hard guarantees is impossible. In this paper, we focus on soft QoS guarantees for real-time applications in multi-hop wireless networks.

TABLE I
Comparisons between the Analytical Results in Admission Control

	mean value delay jitter	acceptable delay	maximum delay violation probability
mean value variance	√	×	×
upper bound	×	√	×
CDF	√	√	√

Different protocols at different layers have been developed to provide QoS guarantees. In the medium access control (MAC) layer, the classical protocol IEEE 802.11e allocates different MAC parameters according to the priorities of the traffic. Nevertheless, it cannot guarantee the QoS required by the traffic [3]. There are a lot of QoS routing algorithms developed at the network layer. They attempt to find the best route in terms of largest available bandwidth or minimum end-to-end delay to provide hard guarantees [4], [5]. Unfortunately, these QoS routing algorithms may not be suitable for the traffic with soft QoS requirements. To provide QoS guarantees efficiently, we need an admission control algorithm, such that after understanding the QoS provided by the current network, we can reject infeasible requests before spending any effort in serving them. This paper aims at designing stochastic admission control (SAC) for real-time applications (Web browsing, and online game) that have stochastic end-to-end delay constraints. To achieve this goal, we first need to understand the probabilistic end-to-end delay performance of the network.

Numerous works [4]–[19] have been done to analyze the end-to-end delay of multi-hop wireless networks. Some works just focused on finding the mean value or the variance of end-to-end delay [4]–[8], [11], [12]. However, the mean value and the variance alone cannot provide enough information for soft QoS guarantees. On the other hand, the cumulative distribution function (CDF) can reflect the probabilistic property of delay, which is very useful for providing soft QoS guarantees. Several efforts have been done to provide probabilistic bounds of end-to-end delay of wireless multi-hop networks [9], [13]–[19]. Table I provides a comparison of several delay metrics in admission control.

Manuscript received December 26, 2012; revised July 4, 2013 and December 7, 2013; accepted December 31, 2013. The associate editor coordinating the review of this paper and approving it for publication was M. J. Hossain.

This paper was supported by National Natural Science Foundation of China (61172079, 91338114, 61231008, 61201141), 111 Project (B08038), National S&T Major Project (2012ZX03004002-003), Shaanxi Province Science and Technology Research and Development Program (2011KJXX-40).

W. Jiao, M. Sheng, and Y. Shi are with the State Key Lab of ISN, Xidian University, Xian, Shaanxi, China (e-mail: {wgjiao87, mshengxd}@gmail.com, yshi@xidian.edu.cn).

K.-S. Lui is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: kslui@eee.hku.hk).

Digital Object Identifier 10.1109/TWC.2013.013014.122055

We consider three types of QoS requirements: 1) mean value and delay jitter; 2) acceptable delay; and 3) maximum delay and violation probability. The QoS requirement in term of mean value ($E[T]$) and delay jitter (ξ) means the average end-to-end delay of the flow is not allowed to be larger than $E[T]$ and the variance of the end-to-end delay is smaller than ξ. Acceptable delay means the delay of the flow must not exceed the acceptable threshold at all time. The QoS requirement in term of maximum delay and violation probability means the delay should not exceed the maximum delay by a certain probability, which is a soft QoS requirement. There is a "$\surd$" in the box when a certain metric provides enough information for the admission control algorithm, such that it can make a decision to admit or reject the flow with a certain QoS requirement. Metric *mean value and variance* cannot admit flows with acceptable delay and soft QoS requirements appropriately because it does not tell the maximum delay. Similarly, metric *upper bound* is not good enough because it does not provide mean value or probability information. On the other hand, according to probability theory, the CDF reflects both mean value and delay upper bound. Therefore, the distribution of end-to-end delay is a more suitable metric for admission control.

Unfortunately, most of the existing models derived the delay distribution by assuming the channel is ideal in the sense that the channel has constant capacity and is error free [4]–[8], [11], [12], [16], [17]. However, a main feature of wireless networks is that the wireless channel is time-varying and unreliable. Ideal channel assumption ignores the retransmissions caused by channel errors or outage, leading to inaccurate analytical results. To capture the influence of non-ideal channels on delay performance, the effective capacity function has been proposed to model QoS capacity of wireless channel at the data link layer in [20]. The authors in [13], [14] extended the results in [20] to multi-hop mesh networks. Since these results are based on the assumption that the traffic arrival rate is constant, they cannot be applied to networks with dynamic traffic arrival rate. To the best of our knowledge, there is no existing model that can accurately estimate the end-to-end delay distribution of multi-hop wireless networks with bursty traffic over time-varying and unreliable channels.

In additions, there is an important factor that influences the delay performance of multi-hop wireless networks but has been usually ignored in previous analyses. Due to the transmission characteristic of electromagnetic signals, the interference range of signals is often larger than the transmission range. Therefore, in multi-hop wireless networks, a node locating outside the sensing range of an existing communication may interfere this transmission. We refer this problem as the *hidden interfering terminal problem*[1], which will be explained in Section II. Although this issue would prolong the delay a packet suffers, to the best of our knowledge, no existing work considers that. In this paper, we incorporate the delay introduced by the hidden interfering terminal problem when we calculate the packet service time.

Our main contributions in this paper can be concluded as follows. First, we are the first to investigate the influences of traffic burstiness and channel unreliability on end-to-end delay in a multi-hop 802.11 network. We achieve it by proposing an analytical model to analyze the end-to-end delay distribution of the multi-hop 802.11 network with the Markovian arrival process and slow fading channels. Second, we study the influence of the hidden interfering terminal issue on end-to-end delay and derive a formula for calculating the end-to-end delay distribution that incorporates this factor. Third, based on the estimation of the end-to-end delay distribution, an efficient stochastic admission control algorithm (SACA) is developed to provide soft guarantees for the traffic with stochastic QoS requirements.

The reminder of this paper is organized as follows. In Section II, we present an overview of related works. The network model is described in Section III. We analyze the distribution of node delay in Section IV and the distribution of the end-to-end delay in Section V. In Section VI, we verify the analytical results through simulations. In Section VII, the applications of the analytical results are introduced and the SACA is described. The simulation results of the SACA are also presented to demonstrate the efficiency of the algorithm. At last, we conclude the paper in Section VIII.

II. Related Work

Before introducing our work, we first give an overview of end-to-end delay analysis in this section. Since the end-to-end delay performance plays an important role in the QoS guarantees of real-time applications, there are many works on end-to-end delay analysis of wireless networks. However, since the end-to-end delay of multi-hop wireless networks is affected by the routing protocol, the MAC protocol, the quality of the physical channels, and mutual interference including inter-flow and intra-flow interference, calculating end-to-end delay for multi-hop networks is very challenging.

In [10], the authors derived the big-O expression of the end-to-end transmission delay of a network with both static and mobile nodes under perfect scheduling and routing. Some research efforts [7], [12] have been devoted to obtain the lower bound of the average end-to-end delay for every flow in the networks with predefined routing paths and perfect link scheduling. Some other works focused on the mean value or variance of end-to-end delay [4]–[6], [8] by considering the effect of the MAC protocol alone. All these works did not analyze delay distribution.

There are some available works on the distribution of the end-to-end delay. In [11], the CDF of end-to-end delay in a network with classical random linear network coding and automatic repeat re-quest (ARQ) has been analyzed. Reference [15] only considered channel error of an ARQ network model in the analysis. The real-time queuing network theory was used to derive end-to-end delay distribution when the network is heavily loaded in [18]. Wang et al. in [17] provided the distribution of the end-to-end delay of wireless sensor networks with the Poisson arrival process and error-free channels. All these works assumed the channel capacity is a constant, which is not practical in wireless networks.

[1] Note that the hidden interfering terminal problem is different from the hidden terminal problem. While the hidden terminal problem has been solved by the RTS/CTS scheme of the IEEE 802.11 protocol, the scheme cannot resolve the hidden interfering terminal problem.

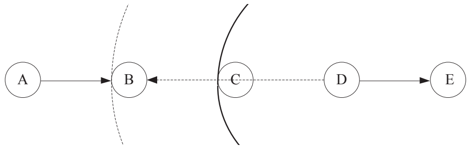


Fig. 1. Hidden interfering terminal problem.

Recently, network calculus has been extended to calculate probabilistic delay bounds of multi-hop wireless networks. In [9], [19], the authors derived the probabilistic end-to-end delay upper bound in the information-theoretic perspective. An important practical method characterizing delay distribution was proposed in [13], [14], which is based on the effective capacity theory [20]. The simulation results in [13], [14] verified that the effective capacity theory is a useful tool to model the wireless channel with time-varying capacity. However, there are two reasons why these works are not suitable for distributed multi-hop wireless networks. First, they assumed the traffic arrival rate at each node was constant, which does not agree with the bursty and diverse nature of the traffic. Second, these works did not consider the effects of channel sharing and transmission collisions caused by uncoordinated access policy in distributed networks.

The IEEE 802.11x standard system has been vigorously developed recently. At the same time, more and more devices support IEEE 802.11 series. These two facts prompt IEEE 802.11 wireless access to be more and more popular. Hence, analyzing the end-to-end delay so as to provide QoS guarantees for real-time applications in this kind of wireless networks is an important problem. Nevertheless, as discussed before, there is no accurate analytical model for analyzing the end-to-end delay distribution of such networks. In this paper, we focus on calculating the distribution of the end-to-end delay of multi-hop IEEE 802.11 distributed coordinated function (DCF) networks with general arrival process and error-prone multi-rate channels. The RTS/CTS scheme is used to coordinate node transmissions at the MAC layer. Authors in [4], [21], [22] have estimated the mean value and the variance of end-to-end delay under this scheme.

However, an important issue influencing node delay has not been considered. In [4], [21], [22], the authors assume a packet transmission must be successful when CTS is successfully received by the sender. Nevertheless, it may not be true if the interference range is larger than the transmission range. Fig. 1 illustrates the hidden interfering terminal problem. In the figure, the interference range is two hops while the communication range is one hop. Suppose that A wants to talk to B. A sends a RTS to B, and B replies a CTS. A then starts data transmission. Before the transmission completes, D wants to talk to E. Because node D could not hear the data transmission from node A to node B, the RTS will be sent by node D, and this RTS will destroy the data packet from node A to node B. There is no existing delay analytical model considers this problem.

To fill the research gap, we propose an analytical model to calculate the distribution of the end-to-end delay for a multi-hop wireless network in this paper. The model considers traffic burstiness, hidden interfering terminal problem, and multi-rate error-prone channels.

III. Network Model

Let N and F denote the set of the nodes and the set of traffic flows in the network, respectively. The nodes are uniformly and independently distributed over a torus area. The path of each flow $f \in F$ has been predetermined. Unlike [13], [14] which assumed the nodes can send and receive at the same time, we consider the scenario where each node is equipped with a single bidirectional half-duplex antenna. Since the node delay is affected by the traffic arrival process, data link layer protocol, and the signal modulation at the physical channel, we will introduce our network model for each factor below.

A. Traffic Arrival Process

Traditionally, to simplify the analytical model, it is assumed that the traffic arrival process at each node is either a deterministic process, a Poisson process, or always saturated. However, the measurements of real traffic suggest that the traffic arrival process is self-similar and long-range dependent, and no random process model can capture all the properties of the traffic arrival process [23]. The Markov modulated Poisson process (MMPP) is a doubly stochastic process with the arrival rate varying according to a multi-state ergodic continuous-time Markov chain [24]. Some works have proved that the QoS performance can be approximated well when the MMPP is used to model the traffic arrival process [24]–[28]. The more states an MMPP has, the more accurate the results will be, but, both computational and analytical complexities rise with the number of states. In [26], the authors demonstrate that the two-state MMPP (MMPP2) strikes a good balance of complexity and accuracy. In this paper, we assume the traffic arrival process of each node i is an MMPP2 with parameters $(r_1, r_2, \lambda_1, \lambda_2)$, where r_i and λ_i are the transition rate and arrival rate of state i, i =1,2, respectively.

According to the effective bandwidth theory [29], the effective bandwidth function $\alpha_B(u)$ characterizes the minimum bandwidth required when the required QoS level of the traffic is u. The effective bandwidth function of an MMPP2 with parameters $(r_1, r_2, \lambda_1, \lambda_2)$ has been given in [29].

B. Data Link Layer

The RTS/CTS scheme of 802.11 DCF protocol is used as the MAC protocol. Lack of space forbids describing the 802.11 DCF protocol in detail. Interesting readers please refer to [30], [31]. As mentioned earlier, because the transmission range is shorter than the interference range, the network suffers from the hidden interfering terminal problem. The backlogged packets are served in first-in-first-out order.

We assume the buffer size of each node is infinite, and then no packet will be dropped due to buffer overflow. However, in practice, the buffer size is finite. To verify this assumption, we simulate 15 different queue models, and the results are shown in Fig. 2. The average service rates and queue sizes of all queue models are the same, and are 10 packets/s and 100 packets, respectively. From Fig. 2, we can find the drop probabilities of all queue models are no more than 1.4% when the traffic arrival rate is 9 packets/s (heavy load). As we assume the network is under a stable situation, nodes are never in overload condition. Thus, the assumption of no packets are dropped due to buffer overflow is acceptable for most practical scenarios.

C. Physical Layer

Almost all IEEE 802.11x protocols support multiple transmission modes (corresponding to different channel rates): 802.11b protocol has four transmission modes, 802.11a protocol provides eight transmission modes, and 802.11g even supports twelve channel rates. In this paper, we assume an adaptive modulation and coding (AMC) scheme is used. The AMC scheme has K transmission modes.

Our work can be applied to any physical channel model which assumes the channel does not change within a single transmission. To illustrate the analysis, we choose the Nakagami-m fading channel model as the physical channel model in this paper because the Nakagami-m fading channel model can be applied to a large class of slow fading channels. For example, the Rayleigh channel model can be represented by putting m to be 1.

Let π_k denote the stationary probability that mode k is used. The calculation for π_k can be found in [32]. Denote the transmission error probability as e_k when mode k is used. According to [32], the packet error probability at the physical layer can be calculated as

$$p_{er} = \sum_{k=0}^{K} \pi_k e_k. \tag{1}$$

Assume that exactly one packet can be encapsulated in one PHY frame and all frames have the same size of L bits with the help of padding bits. Hence, when mode k is used, the time of transmitting one packet is $d_k = \frac{L}{r_k}$ (s), where r_k bits/s is the channel rate when mode k is used. The distribution of the packet transmission time at the physical layer (T_{p_t}) can be derived as

$$\Pr(T_{p\_t} = x) = \begin{cases} \pi_K, & x = d_K \\ \pi_{K-1}, & x = d_{K-1} \\ \vdots & \\ \pi_1, & x = d_1 \\ \pi_0, & x = 0 \end{cases}. \tag{2}$$

A wireless network with multi-rate channels can be analyzed by choosing the appropriate K. In a single-rate channel network, $K = 1$. The corresponding T_{p_t} and p_{er} can be calculated by applying $K = 1$ into Eq. (2) and Eq. (1).

IV. Per-hop Delay Analysis

The distribution of the end-to-end delay can be derived by using the distribution of per-hop delay because the end-to-end delay is the sum of per-hop delay along the path. In this section, we will derive the distribution of the delay a packet experiences in a random node in the network described in Section III. This node delay contains two parts: access delay and queuing delay. While queuing delay is the amount of time that a packet spends in the buffer before it becomes the head of the queue, access delay is the time duration from the moment the packet becomes the head of queue to it is successfully transmitted or dropped. Queuing delay is determined by the traffic arrival process and the distribution of access delay. Hence, we first analyze access delay.

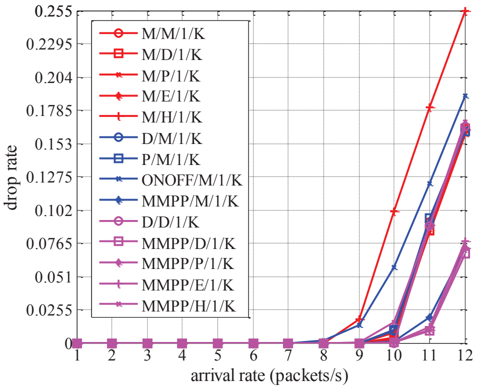


Fig. 2. Packets drop rates of different queue models under different arrival rates. "M" is Poisson process, "D" is deterministic process, "P" is Pareto distribution, "E" is Erlangian distribution, "H" is Hyper-exponential distribution, "ONOFF" is on-off model, and "MMPP" is MMPP2.

The hidden interfering terminal problem should be considered in access delay analysis. The access delay can be obtained through analyzing the MAC protocol. Hence, we first propose a Markov Chain to model node behaviors at the MAC layer which includes hidden interfering terminal phenomenon.

A. MAC Protocol Analysis

Denote the probability that a station attempts to transmit packets in a random slot by p_{tr}, and the failure probability of a transmission by p_{fail}. The hidden interfering terminal problem is statistically independent of RTS collision and transmission error. RTS collision and transmission error are independent of each other. Hence, p_{fail} can be expressed as

$$p_{fail} = 1 - (1 - p_{col})(1 - p_{cor})(1 - p_{er}), \tag{3}$$

where p_{er} is the packet error probability developed in Eq. (1), p_{col} is the probability that a collision occurs when a node is transmitting RTS, and p_{cor} is the probability that the data transmission is interrupted by hidden interfering terminals. Denote the number of neighbors and hidden interfering terminals of the tagged node by N_{ng} and N_{hd}, respectively. p_{col} and p_{cor} can be expressed as $p_{col} = 1 - (1 - p_{tr})^{N_{ng}}$ and $p_{cor} = 1 - (1 - p_{tr})^{N_{hd}}$, respectively.

All nodes in the network are identical, thus, it is enough to analyze the behavior of one node to predict the behavior of other nodes. Fig. 3 shows our novel two-dimensional Markov chain, which is used to characterize the tagged node with an MMPP2 arrival process and Nakagami-m fading channel.

In Fig. 3, state *Empty* represents there is no packet in the buffer. The traffic arrival process is an MMPP2 with parameters $(r_1, r_2, \lambda_1, \lambda_2)$. *Empty* state contains two sub-states, *Empty* 1 and *Empty* 2, representing the arrival process of node in *state* 1 and *state* 2, respectively. The transition probability from *Empty i* to *Empty j* is denoted by $p_{ij}, i, j = 1, 2$ and p_{ij}

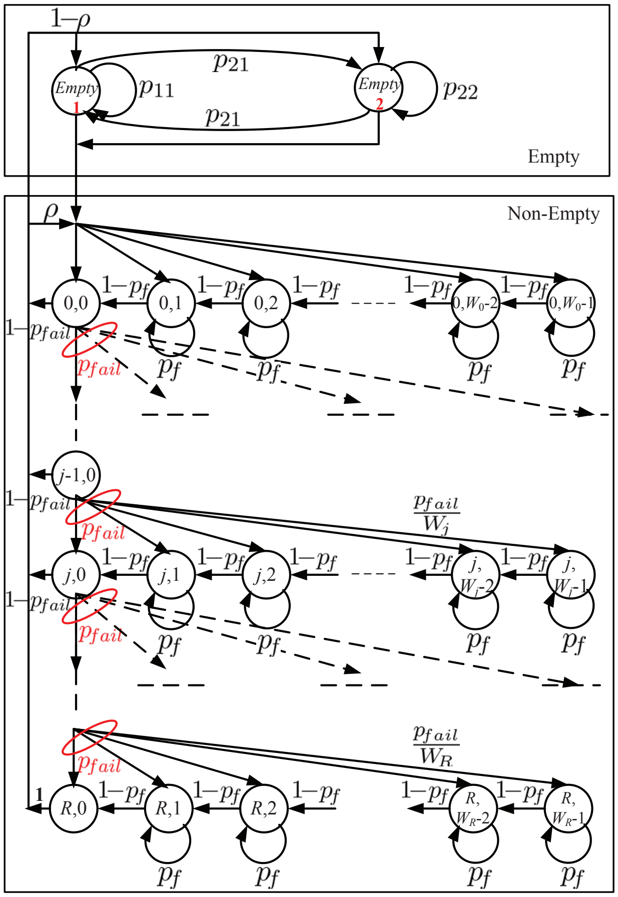


Fig. 3. The Markov Chain for a random node in multi-hop wireless networks.

satisfies

$$\begin{cases} p_{12} = 1 - \exp(-r_1 T_{slot}), & p_{11} = 1 - p_{12} \\ p_{21} = 1 - \exp(-r_2 T_{slot}), & p_{22} = 1 - p_{21} \end{cases}, \tag{4}$$

where T_{slot} is the slot length which is defined by the standard. In IEEE 802.11b [30], $T_{slot} = 20\mu s$, while in IEEE 802.11g, $T_{slot} = 9\mu s$.

Bi-dimensional state (*j*, *k*) in Fig. 3 represents the tagged node is at the *j*-th backoff stage and the backoff counter value is k at a randomly chosen slot. The backoff stage number j is incremented by 1 after each unsuccessful transmission as long as it is smaller than *R*. At the same time, the contention window size is doubled until it is larger than the maximum value $2^m W_0$, where W_0 is the initial contention window size defined in the protocol [30]. The value of *k* will be decremented by 1 if the channel is idle during a whole slot. When it reaches zero, the station transmits. The value of *k* is set according to *j* and the transmission result. If the transmission fails, *j* is changed to *j*+1 and the value of *k* is uniformly chosen in the range $[0, W_{j+1}]$; otherwise, *j* becomes 0 and the range of the value of *k* is $[0, W_0]$. In the Markov chain shown in Fig. 3, ρ represents the probability that after a packet transmission is completed, there is at least one packet in the buffer. ρ has the same meaning as the utilization in the queuing theory, which is the fraction of the time in which the server is busy. Thus, according to the queuing theory, $\rho = \bar{\lambda} * \bar{T}_{ser}$, where $\bar{\lambda}$ is the average arrival rate and $\bar{\lambda} = \frac{r_1 + r_2}{\lambda_1 r_2 + \lambda_2 r_1}$ for the MMPP2 case, and $\bar{T}_{ser}$ is the average packet service time[2]. We will discuss $\bar{T}_{ser}$ in Section IV-B. According to the previous description, the one-step transition probabilities satisfy

$$\begin{cases} q(j, k-1|j, k) = 1 - p_f, j \in (0, R), k \in (2, W_i - 1) \\ q(j, k|j, k) = p_f, j \in (0, R), k \in (1, W_i - 1) \\ q(j, k|j-1, 0) = p_{fail}/W_i, j \in (1, R), k \in (0, W_i - 1) \\ q(0, k|j, 0) = \rho p_{suc}/W_0, j \in (0, R-1), k \in (0, W_0 - 1) \\ q(Empty|j, 0) = (1-\rho) p_{suc}, j \in (0, R-1) \\ q(0, k|R, 0) = \rho/W_0, k \in (0, W_0 - 1) \\ q(Empty|R, 0) = 1 - \rho, \\ q(0, k|Empty) = p_{gen}/W_0, k \in (0, W_0 - 1) \end{cases}, \tag{5}$$

where p_f is the probability that the backoff node finds the channel is busy and freezes the backoff counter, p_{suc} is the probability that a frame is successfully transmitted, $p_{fail} = 1 - p_{suc}$, and p_{gen} is the probability that at least one packet comes to MAC layer during one slot. For an MMPP2 case, $p_{gen} = \frac{r_2}{r_1+r_2}(1 - e^{-\lambda_1 T_{slot}}) + \frac{r_1}{r_1+r_2}(1 - e^{-\lambda_2 T_{slot}})$. Let $q_{(j,k)}$ be the probability of the node in state (*j*, *k*). From Fig. 3 and the transition probabilities in Eq. (5), the transmission probability of the tagged node can be derived as

$$p_{tr} = \frac{(1 - p_{fail}^{R+1})}{(1 - p_{fail}) \left[\left(\sum_{j=0}^{R} (1 + \frac{W_j - 1}{2(1 - p_f)}) p_{fail}^j \right) + \frac{1-\rho}{p_{gen}} \right]}. \tag{6}$$

It is worth noting that in Eq. (6), there is still one unknown parameter: p_f, which is the probability that the channel is busy in a random slot when the tagged node is in backoff state. Referring to Fig. 3, a backoff node enters a backoff state from either a transmission state or from a previous backoff state. Note that a node enters a backoff state towards the end of a time slot. When the tagged node enters a backoff state, the channel may become either busy due to a transmission started by other nodes, or stay idle for the entire duration of a standard slot. Therefore, a two-state Markov Chain shown in Fig. 4 is presented to model channel state from the tagged node's perspective. Through solving this Markov Chain, the value of p_f can be derived.

In Fig. 4, states *Idle* and *Busy* represent the channel is in idle state and the channel is captured by other nodes from the tagged node's point of view, respectively. The probability that the tagged node enters a backoff state and senses the channel is idle equals the probability that none of its neighbors (N_{ng}) transmits packets. If the channel is busy, the tagged node finds the channel is idle in the next slot when none of involved nodes who make the channel busy chooses 0 as the backoff value. Note that to keep our model tractable, although different nodes have different backoff window sizes, we do not consider this difference. Instead, we approximate the probabilities of selecting zero as a new backoff counter value by using the average backoff window size ($\bar{W}$). So the

[2] In this paper, "average packet service time" is the same as "average access delay". In the rest of the paper, we using these two terms interchangeably.

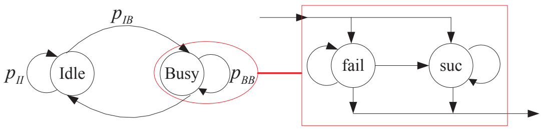


Fig. 4. State transition diagram of channel state.

transition probabilities in Fig. 4 satisfy

$$\begin{cases} P_{II} = (1-p_{tr})^{N_{ng}}, & P_{IB} = 1 - P_{II} \\ P_{BI} = \sum\limits_{n=1}^{N_{ng}} Inv(n)(1-\frac{1}{\bar{W}})^n, & P_{BB} = 1 - P_{BI} \end{cases}, \quad (7)$$

where $\bar{W}$ is the average backoff window size over all states, $Inv(n)$ is the probability that there are *n* neighbors transmit packets. $Inv(n)$ can be computed as $Inv(n) = C_{N_{ng}}^n p_{tr}^n (1-p_{tr})^{N_{ng}-n}$, where $C_a^r = \binom{a}{r} = \frac{a(a-1)\cdots(a-r+1)}{r!}$. When a node chooses W_j as the backoff window size, it has suffered *j* transmission failures. Hence, $\bar{W}$ can be calculated as $\bar{W} = \sum\limits_{j=0}^{R} p_{fail}^j W_j$. According to Markov chain theory, the probability of *Idle* state P_I and the probability of *Busy* state P_B satisfy

$$\begin{cases} P_I = P_I P_{II} + P_B P_{BI} \\ P_B = P_I P_{IB} + P_B P_{BB} \end{cases}. \quad (8)$$

Substituting Eq. (7) into Eq. (8), we can derive P_I and P_B by using the normalized condition: $P_I + P_B = 1$. When the tagged node senses the channel is busy, it freezes the backoff counter. Therefore, p_f is the same as P_B, and p_f can be expressed as

$$p_f = P_B = \frac{P_{IB}}{P_{IB} + P_{BI}}. \quad (9)$$

B. Access Delay Calculation

The effects of traffic burstiness, channel unreliability, and hidden interfering terminals have been analyzed in Section IV-A. We now use the analysis results to calculate the distribution of access delay.

According to the transmission process of a random packet shown in Fig. 3, the distribution of access delay T_{ser} can be expressed as

$$\Pr(T_{ser} = t) = \sum_{r=0}^{R} p_{fail}^r p_{suc} \Pr(T_{tr,r} = t), \quad (10)$$

where p_{suc} and p_{fail} have been obtained in Section IV-A, $T_{tr,r}$ is the time that a packet who suffers r, $0 \le r \le R$, retransmissions before it is transmitted successfully experiences. Thus, $T_{tr,r}$ includes the time caused by *r* unsuccessful transmissions, *r* backoffs, and the last successful transmission. $T_{tr,r}$ can be expressed as

$$T_{tr,r} = T_{suc} + rT_{fail} + \sum_{j=0}^{r} T_{bofj}, \quad (11)$$

where T_{suc}, T_{fail} and T_{bofj} are the times required by a successful transmission, an unsuccessful transmission, and the *j*-th backoff, respectively. According to the standard [30], T_{suc} can be expressed as

$$T_{suc} = DIFS + 3*SIFS + RTS + CTS + T_{p\_t} + ACK. \quad (12)$$

The values of *DIFS*, *SIFS*, *RTS*, *CTS* and *ACK* are constants defined in the standard [30]. By substituting Eq. (2) into Eq. (12), the distribution of T_{suc} can be derived.

Unlike existing works which only consider transmission failures caused by collisions, we also consider *non-collision* failures because of transmission errors and hidden interfering terminals. Note that the time spent on transmission failure caused by transmission error is equal to the time caused by hidden interfering terminals. Let T_{fail_c} and T_{fail_ei} represent the amounts of time caused by transmission collision and *non-collision* case, respectively. According to the standard [30], T_{fail_c} and T_{fail_ei} can be expressed as

$$T_{fail\_c} = DIFS + RTS + SIFS + CTS. \quad (13)$$

$$T_{fail\_ei} = DIFS + 2*SIFS + RTS + CTS + T_{p\_t} + EIFS. \quad (14)$$

As shown in Eq. (13), T_{fail_c} is a constant. Substituting Eq. (2) into Eq. (14), the distribution of T_{fail_ei} can be derived. When the transmission is unsuccessful, while the probability that the transmission fails due to transmission collision is $\frac{p_{col}}{p_{fail}}$, the probability of the data transmission failure caused by non-collision is $\frac{p_{fail}-p_{col}}{p_{fail}}$. As a result, T_{fail} can be calculated as

$$\Pr(T_{fail} = x) = \begin{cases} \frac{p_{col}}{p_{fail}}, & x = T_{fail\_c} \\ \frac{p_{fail}-p_{col}}{p_{fail}} \Pr(T_{fail\_ei} = x), & others \end{cases}. \quad (15)$$

When the transmission fails due to transmission collision, the failure time is T_{fail_c}. Therefore, $\Pr(T_{fail} = T_{fail\_c})$ is $\frac{p_{col}}{p_{fail}}$. In other case when the transmission failure is caused by *non-collision*, the failure time is T_{fail_ei}. That is, T_{fail} has the same distribution as T_{fail_ei} with the probability $\frac{p_{fail}-p_{col}}{p_{fail}}$. Therefore, $\Pr(T_{fail} = x)$ is $\frac{p_{fail}-p_{col}}{p_{fail}} \Pr(T_{fail\_ei} = x)$.

Denote the actual length of a backoff slot by T'_{slot}. If the backoff node finds the channel is busy, it will freeze the backoff counter. Then $T'_{slot} \neq T_{slot}$, and T'_{slot} is not a constant. Therefore, T_{bofj} cannot be expressed as $\Pr(T_{bofj} = kT_{slot}) = 1/W_j, k \in [0, W_j - 1]$, as in the previous work [31]. To obtain the distribution of T_{bofj}, we first derive the distribution of T'_{slot}.

When the backoff node senses the channel is idle, $T'_{slot} = T_{slot}$. Therefore, $\Pr(T'_{slot} = T_{slot})$ is $(1-p_{tr})^{N_{ng}}$. As shown in Fig. 4, when the channel is busy, it may be a collision, a non-collision, or a successful case. The time spent on a collision, a non-collision, and a successful case are T_{fail_c}, T_{fail_ei}, and T_{suc}, which have been analyzed before. When more than one neighbor sends RTS, the channel is captured by a collision and $T'_{slot} = T_{fail\_c} + T_{slot}$. Therefore, $\Pr(T'_{slot} = T_{fail\_c} + T_{slot})$ is $\sum_{i=2}^{N_{ng}} \binom{N_{ng}}{i} p_{tr}^i (1-p_{tr})^{N_{ng}-i}$. When the backoff node senses the channel is busy because of a successful case, a backoff slot contains a successful transmission and a standard slot. That is, T'_{slot} has the same distribution as T_{suc} with the probability $N_{ng}p_{tr}(1-p_{tr})^{N_{ng}-1}(1-p_{cor})(1-p_{er})$. Therefore, $\Pr(T'_{slot} = x)$ is $N_{ng}p_{tr}(1-p_{tr})^{N_{ng}-1}(1-$

$$\Pr(T'_{slot}=x)=\begin{cases}(1-p_{tr})^{N_{ng}}, & x=T_{slot}\\ \sum\limits_{i=2}^{N_{ng}}\binom{N_{ng}}{i}p_{tr}^{i}(1-p_{tr})^{N_{ng}-i}, & x=T_{fail\_c}+T_{slot}\\ N_{ng}p_{tr}(1-p_{tr})^{N_{ng}-1}(1-p_{er})(1-p_{cor})\Pr(T_{suc}=x-T_{slot}), & x=T_{suc}+T_{slot}\\ N_{ng}p_{tr}(1-p_{tr})^{N_{ng}-1}(1-(1-p_{cor})(1-p_{er}))\Pr(T_{fail\_ei}=x-T_{slot}), & x=T_{fail\_ei}+T_{slot}\end{cases}. \quad (16)$$

$p_{cor})(1-p_{er})\Pr(T_{suc}=x-T_{slot})$. Similarly, the probability that the channel is busy because of a non-collision case is $N_{ng}p_{tr}(1-p_{tr})^{N_{ng}-1}(1-(1-p_{cor})(1-p_{er}))$. Therefore, the distribution of T'_{slot} can be calculated as Eq. (16).

Using the distribution of T'_{slot} in Eq. (16), the distribution of T_{bofj} can be computed as

$$\Pr(T_{bofj}=x) \quad (17)$$
$$=\begin{cases}1/W_j, & x=0\\ \Pr(kT'_{slot}=x)/W_j, & k\in[1,W_j-1],x\neq 0\end{cases}.$$

By substituting $\Pr(T_{suc}=x)$, Eq. (15), and Eq. (17) into Eq. (11), the distribution of $T_{tr,r}$ can be derived as Eq. (18).

After substituting Eq. (18) into Eq. (10), the distribution of access delay T_{ser} can be calculated. From Eq. (18), we find that T_{ser} is not continuous but discrete, and it has finite values denoted by $T=\{T_1,T_2,\cdots,T_m,\cdots,T_M\}$. According to probability theory, the mean value of access delay which is used in calculating ρ can be calculated as

$$\bar{T}_{ser}=\sum_{t\in T}t\Pr(T_{ser}=t). \quad (19)$$

Equation (10) indicates that T_{ser} is a function of p_{fail} which depends on p_{tr} as shown in Eq. (3). Equation (6) of p_{tr} contains a parameter ρ whose calculation depends on the mean value of T_{ser}. Therefore, p_{tr}, p_{fail}, and T_{ser} depend on each other. To obtain the distribution of access delay, we must solve non-linear equations. We develop a simple iterative algorithm to solve these non-linear equations. Algorithm 1 gives the details. In the algorithm, ε_t is the tolerable calculation error, and $p_{tr\_1}\in(0,1)$ is the initial value of p_{tr}.

Algorithm 1 Obtain $\Pr(T_{ser}=t)$

```
Input: $N_{ng},N_{hd},T_{p\_t},p_{er},r_1,r_2,\lambda_1,\lambda_2,R,W_0,m,\varepsilon_t$;
Initialization: $p_{tr\_1}=1,p_{tr}=0.01,\overline{T}_{ser}=\overline{T}_{p\_t}$;
while $|p_{tr}-p_{tr\_1}|>\varepsilon_t$ do
    $p_{tr\_1}=p_{tr}$;
    $p_{fail}=1-(1-p_{tr\_1}^{N_{ng}})(1-p_{tr\_1}^{N_{hd}})(1-p_{er})$;
    Substitute $p_{fail}$, $p_{tr\_1}$ and $T_{p\_t}$ into Eq. (10-18) to derive Pr($T_{ser}=t$);
    Use Eq. (19) to calculate $\overline{T}_{ser}$;
    Substitute $p_{fail}$ and $\overline{T}_{ser}$ into Eq. (6) to get a new value for $p_{tr}$.
end while
Output: $\Pr(T_{ser}=t)$ and the set $T$.
```

C. Queuing Delay Analysis

As mentioned before, queuing delay is a function of access delay and traffic arrival process. Given the traffic arrival process, the queuing delay can be calculated based on the distribution of access delay. Queuing theory provides an accurate queuing delay estimation of systems with memoryless arrival or memoryless service process. However, queuing theory no longer works well in estimating delay distribution when neither the traffic arrival process nor the service process is memoryless [33]. The distribution of access delay shown in Eq. (10) indicates that it has many discrete values which do not obey a geometric distribution. We can conclude that both the arrival process and the service process are not memoryless, and queuing theory cannot be applied to analyze the distribution of queuing delay. Recently, effective bandwidth theory has been widely used to obtain the QoS performance of computer networks, such as queuing delay of a system with general arrival process [29]. In this section, we apply effective bandwidth theory to obtain queuing delay.

Let the effective bandwidth function of a certain traffic arrival process be $\alpha_B(u)$ (u is the QoS exponent) and the channel rate be C. We further let θ^* be the solution of $\alpha_B(u)=C$. According to the effective bandwidth theory [29], the distribution of queuing delay (W) can be calculated as $\Pr(W>x)=\gamma\exp(-xC\theta^*)$, where γ is the probability that the buffer is nonempty.

The formula cannot be applied directly when the channel rate is not a constant, which is the situation considered in this paper. Fortunately, with the help of probability theory, effective bandwidth theory can be extended to calculate the distribution of queuing delay of the system with multiple service rates.

The access delay of the tagged node in Eq. (10) has finite values, as a result, we can extend effective bandwidth theory [29] to analyze queuing delay of the tagged node. We first derive the distribution of queuing delay when the service rate is one element of T, and then calculate the distribution of queuing delay when consider all values of service time using probability theory. As shown in Eq. (10), the probability that packet service time T_{ser} equals T_m, $T_m\in T$, is $\Pr(T_{ser}=T_m)$. $T_{ser}=T_m$ means that the tagged node can serve $1/T_m$ packets in one second. Given effective bandwidth function $\alpha_B(u)$, the distribution of queuing delay W_{T_m} when the channel rate is $1/T_m$ satisfies

$$\Pr(W_{T_m}\leq x)=1-\gamma_{T_m}\exp(-x\theta^*_{T_m}/T_m),$$

where $\theta^*_{T_m}$ can be obtained through solving the equation $\alpha_B(u)=1/T_m$.

The distributions of queuing delay when the service time equals other elements in T can be calculated in the same way. The queuing delay (T_{que}) has the same distribution as W_{T_m} with the probability $\Pr(T_{ser}=T_m)$. According to

$$\Pr(T_{tr,r}=t)=\sum_{x=0}^{t}\sum_{y=0}^{t-x}\Pr(T_{suc}=x)\Pr(\sum\nolimits_{j=0}^{r}T_{bofj}=y)\Pr(T_{fail}=t-x-y). \quad (18)$$

probability theory, the distribution of queuing delay (T_{que}) can be calculated as

$$\Pr(T_{que}\le x)=\sum_{T_m\in T}\Pr(T_{ser}=T_m)\Pr(W_{T_m}\le x). \quad (20)$$

So far, access delay and queuing delay have been obtained. The node delay is composed of access delay and queuing delay, and node delay (T_{sig}) satisfies $T_{sig}=T_{ser}+T_{que}$. According to Eq. (10) and Eq. (20), by applying probability theory, the distribution of node delay ($\Pr(T_{sig}\le x)$) can be derived as

$$\Pr(T_{sig}\le x)=\sum_{T_m\in T}\Pr(T_{ser}=T_m)\Pr(T_{que}\le(x-T_m)). \quad (21)$$

V. End-to-End Delay Distribution Calculation

Using the distribution of node delay given in Eq. (21), we calculate the end-to-end delay distribution of an arbitrary communication pair in the network in this section.

A. End-to-End Delay Derivation

Let $rp_f=(h_0,\cdots h_{H_f})$ denote the routing path of flow *f*, $f\in F$, where $H_f\ge 1$ is the number of hops of flow *f*, $h_i\in N, 0\le i\le H_f$ is the *i*-th hop node. Since end-to-end delay can be calculated through summing up the per-hop delay along the path, the end-to-end delay of rp_f can be expressed as $T_{ete,rp_f}=\sum_{i=0}^{H_f-1}T_{h_i,sig}$, where $T_{h_i,sig}$ is the node delay of the *i*-th hop (h_i).

The distribution of node delay $\Pr(T_{h_i,sig}\le x)$ of the *i*-th hop h_i is shown in Eq. (21). According to probability theory, the probability density function (PDF) of $T_{h_i,sig}$ can be computed as

$$f_{h_i,sig}(t)=\frac{d\Pr(T_{h_i,sig}\le t)}{dt}. \quad (22)$$

As $T_{ete,rp_f}=\sum_{i=0}^{H_f-1}T_{h_i,sig}$, the probability that T_{ete,rp_f} is smaller than *x* can be expressed as

$$\Pr(T_{ete,rp_f}\le x)=\Pr(\sum_{i=0}^{H_f-1}T_{h_i,sig}\le x). \quad (23)$$

Substituting Eq. (22) into Eq. (23), the distribution of T_{ete,rp_f} can be calculated as

$$\begin{aligned}&\Pr(T_{ete,rp_f}\le x) \qquad (24)\\ &=\int_0^x(f_{h_0,sig}*f_{h_1,sig}*\cdots*f_{h_{H_f-1},sig})(t)dt,\end{aligned}$$

where $(f*g)(t)$ is the convolution of $f(t)$ and $g(t)$.

We have investigated the computational complexity when only one transmission mode is used, and found that T_{ser} has more than ten thousand values. It implies that the computational complexity of calculating T_{ser}, which is solving nonlinear equations, is very high. When the number of transmission modes at the physical layer increases, the complexity increases quickly. From the end-to-end delay formula (Eq. (24)), we notice that the calculation of end-to-end delay depends on node delay distribution, which just contains three parameters: $\Pr(T_{ser}=T_m)$, γ_{T_m} and $\theta^*_{T_m}$. To reduce the complexity, we propose a simple sampling algorithm to approximate these three parameters and calculate the end-to-end delay distribution based on these sample results.

B. A Sampling Algorithm

As discussed in Section IV-C, given the effective bandwidth function $\alpha_B(u)$ and the distribution of service time $\Pr(T_{ser}=T_m)$, γ_{T_m} and $\theta^*_{T_m}$, the end-to-end delay distribution can be calculated. We propose a sample algorithm in this sub-section. In this algorithm, we first obtain the arrival process parameters.

Authors in [26] provided a method to calculate the parameters of MMPP2. To obtain the effective bandwidth function of the traffic arrival process, we first employ the method proposed in [26] to estimate the parameters of MMPP2, and then substitute the results into the formula of effective bandwidth function in [29].

Given $\alpha_B(u)$ and T_m, $\theta^*_{T_m}$ can be calculated. Since γ_{T_m} is the probability that the buffer is nonempty, γ_{T_m} can be calculated as $\bar{\lambda}T_m$ according to queuing theory. Assume the distribution of the service time is known, then, T is known, and the corresponding $\theta^*_{T_m}$ and γ_{T_m} can be calculated. Then, substituting these $\theta^*_{T_m}$ and γ_{T_m} into Eq. (24), the distribution of the end-to-end delay can be obtained. Therefore, a sampling algorithm is proposed to obtain $\Pr(T_{ser}=T_m)$.

According to the analysis in Section IV, during sampling, T_{ser} should be recorded from a packet becomes the head of the queue, instead of from the moment it is being transmitted. First, we take S samples over T_{itvl} and record the latest packet service time (T_{sn}) at each sampling epoch. We use the sample values of T_{sn} to approximate the distribution of packet service time T_{ser}. Let $T_{s\min}=\min\limits_{1\le n\le S}\{T_{sn}\}$, $T_{s\max}=\max\limits_{1\le n\le S}\{T_{sn}\}$. Then, $[T_{s\min},T_{s\max}]$ is split into $P, P\ge 100$, none-overlapping intervals. The number of sample values falling in $[T_{s,p},T_{s,p+1}], 0\le p\le P$, is recorded, and then these numbers are used to compute the frequency of the sample values falling in this interval. According to the law of large numbers, the frequency can be used to approximate the distribution of service time. Hence, we use the frequency of the sample values falling in $[T_{s,p},T_{s,p+1}], 0\le p\le P$, to estimate the probability that the service time is $(T_{s,p}+T_{s,p+1})/2$. Though the more the sample values, the more accurate the result will be, more memory will be needed to keep the samples. According to [20], we set $S=2000$ and $T_{itvl}=2$s in this paper.

γ_{T_m} and $\theta^*_{T_m}$ are calculated using the sample results. By substituting γ_{T_m}, $\theta^*_{T_m}$ and $\Pr(T_{ser}=T_m)$ into Eq. (24), the end-to-end delay can be computed.

TABLE II
AMC PARAMETERS

Model	1	2	3	4	5	6
channel rate (Mbps)	2	4	6	9	12	18

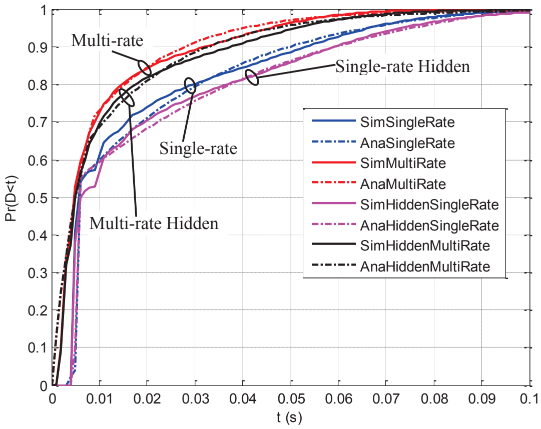


Fig. 5. Access delay distribution for different channel rates.

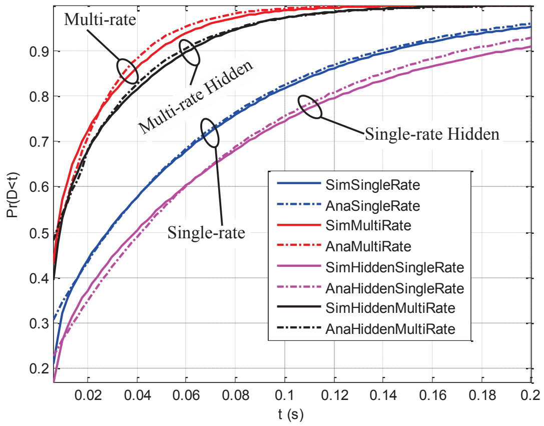


Fig. 6. End-to-end delay distribution for different channel rates.

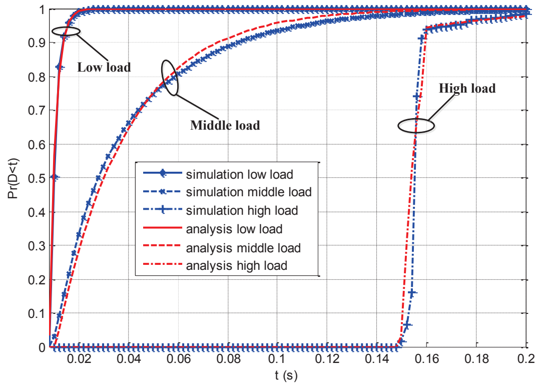


Fig. 7. End-to-end delay distribution under different traffic loads.

VI. SIMULATION RESULTS

To validate the accuracy of our analytical model, we compare the analytical results with simulation results in OPNET simulator. The MAC protocol parameters, such as the durations of *DIFS*, *SIFS*, as well as *RTS*, *CTS*, and *ACK* control frames, are the same as defined in 802.11 standards [30]. The channel is Rayleigh fading channel by setting Nakagami-m fading parameter *m*=1. Without loss of generality, we assume the interference range is twice the transmission range. We first demonstrate our methodology on the topology shown in Fig. 1 where each node uniformly chooses one of its neighbors as the destination of its packets. We set $\lambda_1 = \lambda_2$, and the MMPP2 is regressed to a Poisson process. The traffic arrival rate of each node is 25packets/s, and the packet size is 1024Bytes. While the "Single-rate" corresponds to set *K*=1 and the channel rate is 2Mbps, "Multi-rate" corresponds to $K = 6$ and the channel rates are given in Table II. Other AMC parameters are the same as Table II in [32]. The average SNR is 2.0. The results of access delay and end-to-end delay are shown in Fig.5 and Fig.6, respectively.

To observe the impact of hidden interfering terminals, we compare the results of the scenarios without ("Single-rate" and "Multi-rate") and with ("Single-rate Hidden" and "Multi-rate Hidden") hidden interfering terminals. From Figs. 5-6, it can be found that hidden interfering terminals degrade the delay performance and AMC can improve delay performance through enhancing the quality of transmission at the physical layer. The results in Fig. 5 indicates that our method can provide good estimations for access delay of both multiple channel rates and single channel rate with and without hidden interfering terminals. Note that all communications in Fig. 1 are single hop. The results in Fig. 6 verify the accuracy of our method in estimating node delay.

We now study the performance in a uniform topology which has 64 nodes that are uniformly and randomly distributed over a 1km×1km area. We choose a 6-hop path in a network to observe the delay. Each node in the network, except the source and the destination of the observed path, randomly chooses another node as the destination. We observe the end-to-end delay of the path under different traffic loads. The results are shown in Fig. 7.

In Fig. 7, the x-coordinates are delay values a packet experiences, and the y-coordinates are the CDF of the end-to-end delay of the path. While the blue lines with symbols are the simulation results, the red lines without symbols are analytical results. The analytical results are calculated using the sampling results which are collected by the sampling algorithm. We verify our model under different traffic load situations: a) low load = 1 packets/s, b) moderate load = 4 packets/s, and c) high load = 7 packets/s. The size of the packets is 1024Bytes. From Fig. 7, we can conclude that our method provides a good estimation for end-to-end delay distribution. Note that the end-to-end delay under "high load" looks very steep. In the simulation, the buffer size is 32 packets. When the network is in high load, some packets will be dropped due to overflow and the buffer is full with high probability. As a result, the queuing delay is a dominant part of the node delay. Meanwhile, when the buffer is often full, MAC layer is in saturation and access delay fluctuates less. Therefore, the node delay performance becomes more stable, and the end-to-end delay CDF becomes flat after a certain value.

To investigate the effect of path length on end-to-end delay, we also investigate the end-to-end delay performance of four flows in a sparse topology with 25 nodes and each node only has two neighbors. The four flows have different path lengths: two shorter paths (flow2 and flow3) with 3 hops

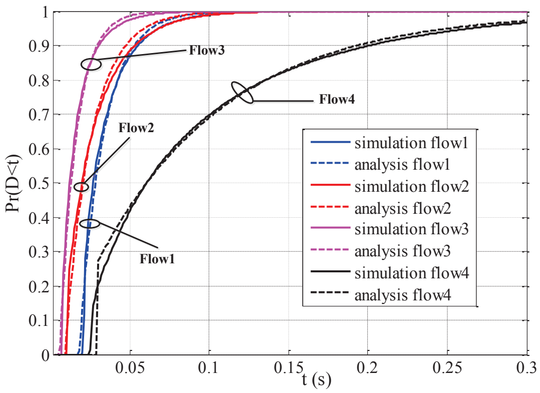


Fig. 8. End-to-end delay distribution for different flows on a sparse topology.

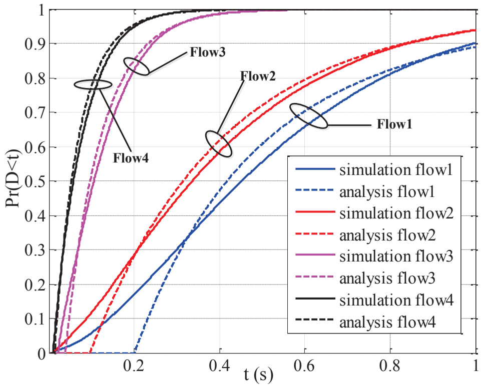


Fig. 9. End-to-end delay distribution for different flows on random topology.

and two longer paths (flow1 and flow4) with 8 hops. The arrival rates of flow1 and flow3 are 20 packets/s, and each packet is 256 bits. Flow2 and flow4 generate 25 packets of size 512 bits per second. The results are shown in Fig. 8. Further, we observe the influence of the node density on delay performance. We generate a topology in which nodes are not uniformly distributed. That is, the node densities in some areas are larger than in others. The node density of flow1 and flow2 are larger than flow3 and flow4. Flow1 and flow4 inject 20 packets of size 256 bits per second in the network while flow2 and flow3 inject 25 packets of size 512 bits per second. The results are shown in Fig. 9.

In Fig. 8 and Fig. 9, while the solid lines are the simulation results, dashed lines are the analytical results. Comparing the results of flow2 and flow4 or flow1 and flow3 in Fig. 8, we can find that the end-to-end delay grows with path length. From the results shown in Fig. 8, the end-to-end delay of flow4 is larger than flow1, we can conclude the higher the packet arrival rate of the flow, the larger the end-to-end delay. Similarly, in Fig. 9, the comparison of flow1 and flow2 or flow3 and flow4 indicates that the higher node density results in larger delay.

From Fig.8 and Fig. 9, we can conclude that our method is more suitable for smaller delay. For larger delay, our results are not precise due to the range of access delay is large. In the sampling algorithm, P is constant, and larger $[T_{s\min}, T_{s\max}]$ range results in a wider range in $[T_{s,p}, T_{s,p+1}]$. The large interval leads to inaccurate estimation for access delay. Though increasing P can improve the accuracy, it will immensely slow down calculation speed and require more memory. From the results shown in Figs. 7-9, $P = 200$ can provide an acceptable estimation for most cases.

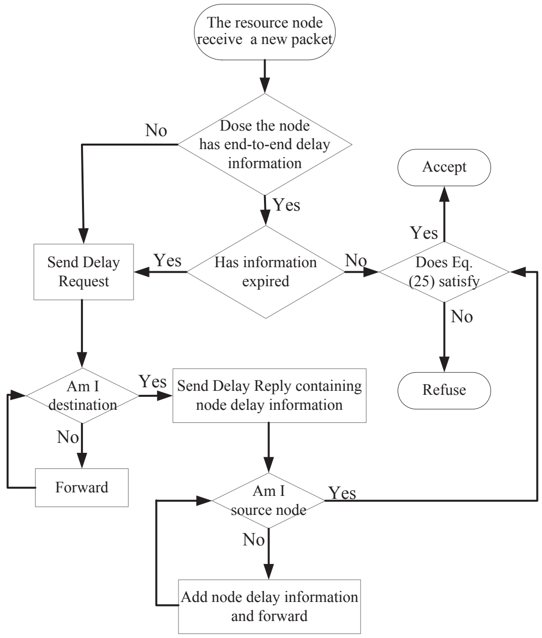

The resource node receive a new packet
Dose the node has end-to-end delay information
No
Yes
Send Delay Request
Yes
Has information expired
No
Does Eq. (25) satisfy
Yes
Accept
No
Refuse
Am I destination
Yes
Send Delay Reply containing node delay information
No
Forward
Am I source node
Yes
No
Add node delay information and forward

Fig. 10. The flow diagram for stochastic admission control algorithm.

VII. Stochastic Admission Control Algorithm

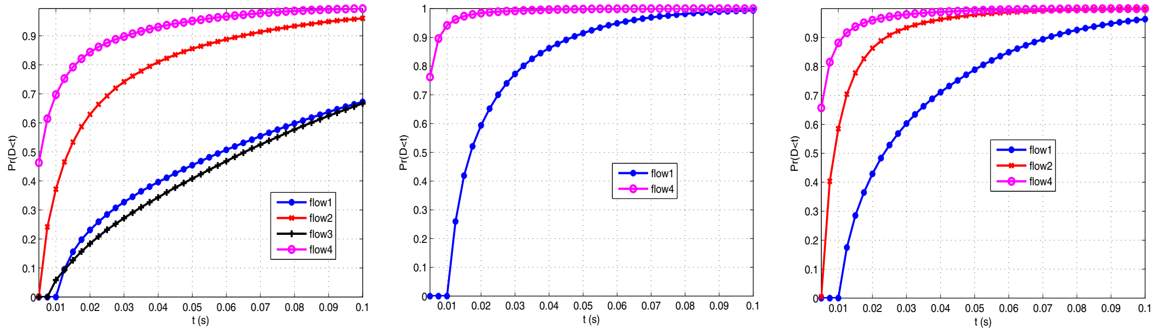
As an important property of wireless channel is unreliability, hard QoS guarantees described in [20] are impossible in wireless networks. As discussed in Section I, soft QoS guarantees require the distribution of delay which has not been well studied in available literature. In traditional admission control, the average delay is used as the admission control metric. If the average end-to-end delay is smaller than $D_{\max}$, which is the maximum delay the flow can tolerate, this flow is admitted into the network; otherwise, it is rejected. There are many applications in wireless networks requiring soft QoS guarantees. To guarantee soft QoS requirements of flows in wireless networks, a stochastic admission control algorithm (SACA) based on our end-to-end delay results is proposed.

The soft QoS requirement in term of end-to-end delay performance is often represented as

$$\Pr(T_{ete} > D_{\max}) \le \varepsilon, \tag{25}$$

where T_{ete} is the end-to-end delay of the path, $D_{\max}$ is the maximum tolerable delay, and ε is the tolerant violation probability. According to the QoS information ($D_{\max}$ and ε) provided by the request, the source node determines whether the current network delay can fulfill this requirement.

The flowchart of the SACA is illustrated in Fig. 10. Since many routing protocols, such as Dynamic Source Routing (DSR) protocol, can carry path information in route requests and route replies, the delay parameters can be stored in these messages. It is not necessary to define a new type of packets for collecting delay information. When a node receives a new flow request, it first checks whether it has the end-to-end delay information for this flow. If there is no delay information or the information has expired, the source node sends a request for obtaining end-to-end delay information.



(a) NO AC (b) Hard AC (c) Soft AC

Fig. 11. The end-to-end delay distribution of different admission control algorithms.

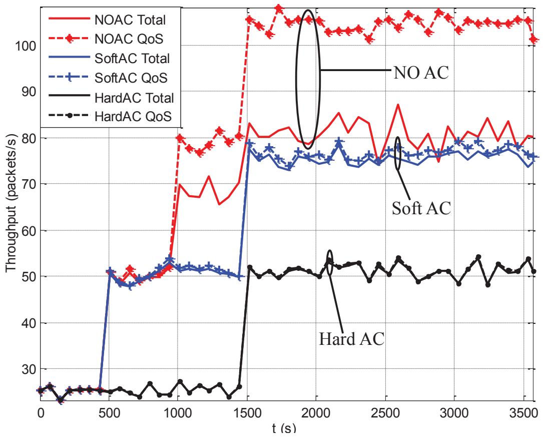


Fig. 12. The throughput comparison of different AC algorithms.

If up-to-date information is available, the source node admits the flow coming in if the end-to-end delay satisfies Eq. (25); otherwise, rejects.

To demonstrate the advantage of SACA, we compare the performance of the network using SACA with networks without QoS guarantees and with the hard QoS guarantees. When there is no admission control algorithm (NO AC), all flow requests are admitted. The hard guarantee (hard AC) ensures there is no packet experiences the end-to-end delay larger than $D_{\max}$, and the soft admission control algorithm (soft AC) guarantees the probability that the end-to-end delay is more than $D_{\max}$ is less than ε. The difference between hard AC algorithm and SACA is that hard AC uses the delay upper bound as the metric. In hard AC algorithm, when the maximum delay is larger than $D_{\max}$, this flow is rejected. In the simulation, $D_{\max} = 0.1$ and $\varepsilon = 5\%$. We investigate four flows imposed on the network shown in Fig. 1. The arrival rate of each flow is 25packets/s and the packet size is 512Bytes. The start times of flow1 (from A to E), flow2 (from C to E), flow3 (from B to E) and flow4 (from D to E) are 1s, 500s, 1000s and 1500s, respectively. The end-to-end delay performance is shown in Fig. 11 and the throughput of the network is illustrated in Fig. 12. Note that "Total" and "QoS" in Fig. 12 are the total packets and the packets satisfying the delay requirement received by the destination, respectively.

From Fig. 11, we can find hard AC algorithm (Hard AC) obtains better delay performance at the expense of rejecting more flows (only 2 flows are admitted). Though when there is no admission control (NO AC), more flows (4 flows) are admitted, the stochastic delay requirement cannot be guaranteed. Based on guaranteeing soft QoS requirement, SACA (Soft AC) admits more flows than hard AC algorithm. Throughput results shown in Fig. 12 demonstrate that hard AC can ensure the end-to-end delay of packets is not larger than $D_{\max}$. In "NO AC", more packets are admitted in the network, and more packets satisfy the delay requirement when compared to SACA. However, after 1000s, the packets satisfying the delay requirement account for no more than 87% of total admitted packets. It indicates that the QoS requirement of admitted flows is not guaranteed because more than 13% packets are dropped due to delay unsatisfied. In our SACA, the ratio of the packets satisfying delay requirement to the total packets is larger than 97.5%.

TABLE III
FLOW PARAMETERS

flows	start time	arrival rate (packets/s)	packet size (bits)	QoS requirement
flow1	1s	40	256	$D_{max} = 0.1$s, $\varepsilon = 5\%$
flow2	501s	50	512	$D_{max} = 0.2$s, $\varepsilon = 5\%$
flow3	1001s	50	512	$D_{max} = 0.1$s, $\varepsilon = 5\%$
flow4	1501s	40	256	$D_{max} = 0.2$s, $\varepsilon = 5\%$

To further observe the efficiency of SACA, we compare the throughput of SACA with traditional admission control algorithm on the sparse topology introduced in Section VI. There are four flows in the network and the main parameters are given in Table III. The throughput of the four flows and the whole network are given in Figs. 13-17.

In these figures, the x-coordinates are simulation time, and the y-coordinates are throughput of the flows. We compare the throughput under three cases: no admission control (NO AC), a traditional admission control algorithm uses average end-to-end delay (Hard AC[3]) as the metric and SACA (Soft AC). When the destination node receives a packet, it checks if the packet satisfies the QoS requirement. If not, the packet is dropped. The throughput satisfying the QoS requirements of flow1, flow2, flow3, flow4 and the whole network are shown in Fig. 13, Fig. 14, Fig. 15, Fig. 16 and Fig. 17, respectively.

[3] "Hard AC" in this simulation scenario is different from Figs. 11-12, which represents traditional admission control algorithm.

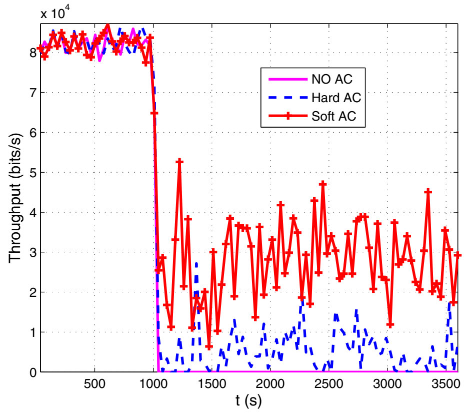


Fig. 13. The throughput of flow1.

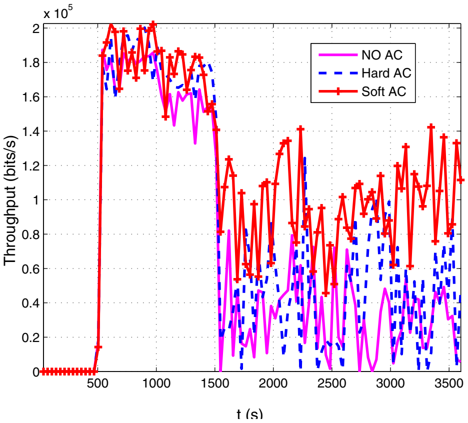


Fig. 14. The throughput of flow2.

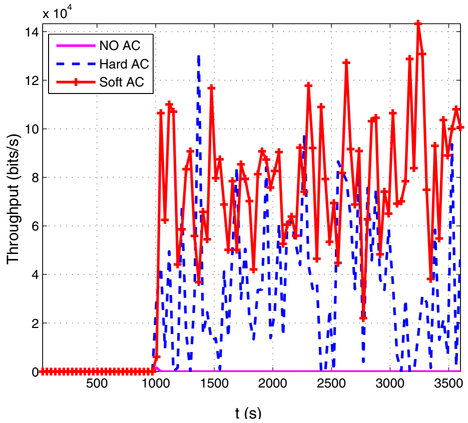


Fig. 15. The throughput of flow3.

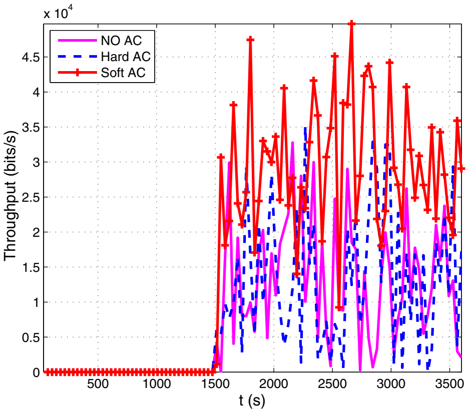


Fig. 16. The throughput of flow4.

In these figures, the solid lines are the results of NO AC, the dash lines are Hard AC and the dotted lines are SACA. From the figures, we can find the throughput employing our method is obviously larger than the others. Therefore, we can conclude that the SACA is more efficient than the traditional method which uses average end-to-end delay as metric (Hard AC).

VIII. Conclusion

In this paper, we proposed an analytical model for node delay distribution in a multi-hop wireless network and extended this work to calculate the end-to-end delay distribution of a random path. We then developed a stochastic admission control algorithm to provide soft QoS guarantees. To validate the accuracy of our analytical model and the efficiency of the proposed algorithm, we tested the estimation performance of the proposed model in several scenarios and investigated the performance of the algorithm at both packet-level and flow-level. In the future, we would like to apply our results to the joint admission control and QoS routing problem.

References

[1] End-user multimedia QoS categories, ITU-T Recommendation G.1010 Std., 2002.

[2] D. Wu, "Providing quality-of-service guarantees in wireless networks," Ph.D. dissertation, the Department of Electrical and Computer Engineering, Carnegie Mellon University, 2003.

[3] X. Chen, H. Zhai, X. Tian, and Y. Fang, "Supporting QoS in IEEE 802.11e wireless LANs," *IEEE Trans. Wireless Commun.*, vol. 5, no. 8, pp. 2217–2227, Aug. 2006.

[4] Y. Sun, E. M. Belding-royer, X. Gao, and J. Kempf, "A priority-based distributed call admission protocol for multi-hop wireless ad hoc networks," Univ. California Santa Barbara, UCSB, CA, Tech. Rep. 2004-20, Aug. 2004.

[5] H. Li, Y. Cheng, C. Zhou, and W. Zhuang, "Minimizing end-to-end delay: a novel routing metric for multi-radio wireless mesh networks," in *Proc. 2009 IEEE INFOCOM*, pp. 46–54.

[6] X. Min and H. Martin, "Towards an end-to-end delay analysis of wireless multihop networks," *Ad Hoc Netw.*, vol. 7, no. 5, pp. 849–861, July 2009.

[7] G. R. Gupta and N. Shroff, "Delay analysis for multi-hop wireless networks," in *Proc. 2009 IEEE INFOCOM*, pp. 2356–2364.

[8] N. Bisnik and A. Abouzeid, "Queuing network models for delay analysis of multihop wireless ad hoc networks," *Ad Hoc Netw.*, vol. 7, no. 1, pp. 79–97, Jan. 2009.

[9] A. Burchard, J. Liebeherr, and S. D. Patek, "A min-plus calculus for end-to-end statistical service guarantees," *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 4105–4114, Sept. 2006.

[10] S. M. Yu and S.-L. Kim, "End-to-end delay in wireless random networks," *IEEE Commun. Lett.*, vol. 14, no. 2, pp. 109–111, Feb. 2010.

[11] F. Chiti, R. Fantacci, R. A. Johnson, V. Crnojevć, and D. Vukobratovic, "End-to-end delay analysis for reliable communications over lossy channels: integrating network coding and ARQ schemes," in *Proc. 2009 IEEE GLOBECOM*, pp. 1–5.

[12] G. R. Gupta and N. B. Shroff, "Delay analysis and optimality of scheduling policies for multihop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 1, pp. 129–141, Feb. 2011.

[13] Y. Chen, J. Chen, and Y. Yang, "Multi-hop delay performance in wireless mesh networks," *Mob. Netw. Appl.*, vol. 13, no. 1-2, pp. 160–168, Apr. 2008.

[14] Y. Chen, Y. Yang, and I. Darwazeh, "A cross-layer analytical model of end-to-end delay performance for wireless multi-hop environments," in *Proc. 2010 IEEE GLOBECOM*, pp. 1–6.

[15] L. Le and E. Hossain, "An analytical model for ARQ cooperative diversity in multi-hop wireless networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 5, pp. 1786–1791, May 2008.

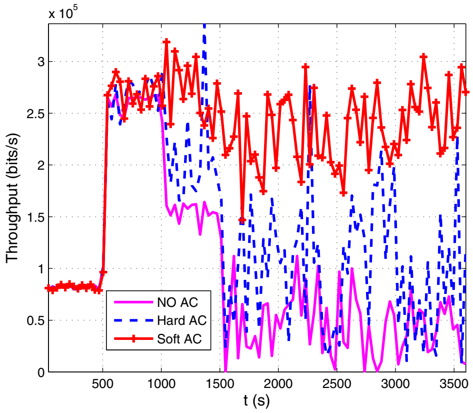


Fig. 17. The throughput of the network.

[16] J. Liebeherr, Y. Ghiassi-Farrokhfal, and A. Burchard, “On the impact of link scheduling on end-to-end delays in large networks,” *IEEE J. Sel. Areas Commun.*, vol. 29, no. 5, pp. 1009–1020, May 2011.

[17] Y. Wang, M. C. Vuran, and S. Goddard, “Cross-layer analysis of the end-to-end delay distribution in wireless sensor networks,” *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 305–318, Feb. 2012.

[18] S.-N. Yeung and J. Lehoczky, “End-to-end delay analysis for real-time networks,” in *Proc. 2001 IEEE RTSS*, pp. 299–309.

[19] J. Liebeherr, A. Burchard, and F. Ciucu, “Delay bounds in communication networks with heavy-tailed and self-similar traffic,” *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 1010–1024, Feb. 2012.

[20] D. Wu and R. Negi, “Effective capacity: a wireless link model for support of quality of service,” *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 630–643, July 2003.

[21] E. Felemban and E. Ekici, “Single hop IEEE 802.11 DCF analysis revisited: accurate modeling of channel access delay and throughput for saturated and unsaturated traffic cases,” *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3256–3266, Oct. 2011.

[22] F. Daneshgaran, M. Laddomada, F. Mesiti, and M. Mondin, “Unsaturated throughput analysis of IEEE 802.11 in presence of non ideal transmission channel and capture effects,” *IEEE Trans. Wireless Commun.*, vol. 7, no. 4, pp. 1276–1286, Apr. 2008.

[23] D. Willkomm, S. Machiraju, J. Bolot, and A. Wolisz, “Primary user behavior in cellular networks and implications for dynamic spectrum access,” *IEEE Commun. Mag.*, vol. 47, no. 3, pp. 88–95, Mar. 2009.

[24] A. Rajabi and J. W. Wong, “MMPP characterization of web application traffic,” in *Proc. 2012 IEEE MASCOTS*, pp. 107–114.

[25] R. Nagarajan, J. F. Kurose, and D. Towsley, “Approximation techniques for computing packet loss in finite-buffered voice multiplexers,” *IEEE J. Sel. Areas Commun.*, vol. 9, no. 3, pp. 368–377, Apr. 1991.

[26] H. Heffes and D. Lucantoni, “A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance,” *IEEE J. Sel. Areas Commun.*, vol. 4, no. 6, pp. 856–868, Sept. 1986.

[27] A. T. Andersen and B. F. Nielsen, “A Markovian approach for modeling packet traffic with long-range dependence,” *IEEE J. Sel. Areas Commun.*, vol. 16, no. 5, pp. 719–732, June 1998.

[28] Y. Wu, G. Min, and L. T. Yang, “Performance analysis of hybrid wireless networks under bursty and correlated traffic,” *IEEE Trans. Veh. Technol.*, vol. 62, no. 1, pp. 449–454, Jan. 2013.

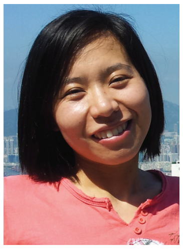
[29] C.-S. Chang and J. A. Thomas, “Effective bandwidth in high-speed digital networks,” *IEEE J. Sel. Areas Commun.*, vol. 13, no. 6, pp. 1091–1100, Aug. 1995.

[30] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE 802.11 WG Std., Aug. 1999.

[31] G. Bianchi, “Performance analysis of the IEEE 802.11 distributed coordination function,” *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

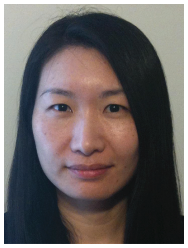
[32] Q. Liu, S. Zhou, and G. Giannakis, “Cross-layer combining of adaptive modulation and coding with truncated ARQ over wireless links,” *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1746–1755, Sept. 2004.

[33] G. Bolch, S. Greiner, H. d. Meer, and K. S. Trivedi, *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. John Wiley and Sons, 2006.



Wanguo Jiao received the B.S. degree in Network Engineering from Shijiazhuang Tiedao University, Shijiazhuang, China, in 2005. She is currently pursuing her Ph.D. degree in the State Key Laboratory of ISN, the School of Telecommunication Engineering, Xidian University. Her currently research interests include performance analysis of multi-hop wireless networks and cognitive radio networks and QoS protocol design.

Min Sheng (M'03), received the M. Eng and Ph.D. degrees in Communication and Information Systems from Xidian University, Shaanxi, China, in 1997 and 2000, respectively. She is currently a Full Professor at the Broadband Wireless Communications Laboratory, the School of Telecommunication Engineering, Xidian University. Her general research interests include mobile ad hoc networks, wireless sensor networks, wireless mesh networks, third generation (3G)/4th generation (4G) mobile communication systems, dynamic radio resource management (RRM) for integrated services, cross-layer algorithm design and performance evaluation, cognitive radio and networks, cooperative communications, and medium access control (MAC) protocols. She is a member of the IEEE. She has published 2 books and over 50 papers in refereed journals and conference proceedings. She was the New Century Excellent Talents in University by the Ministry of Education of China, and obtained the Young Teachers Award by the Fok Ying-Tong Education Foundation, China, in 2008.



King-Shan Lui received her Ph.D. degree in Computer Science from the University of Illinois at Urbana-Champaign, and she is now an Associate Professor in the Department of Electrical and Electronic Engineering in the University of Hong Kong. Her research interests include network protocols design and analysis, wireless networks, smart grids, and Quality-of-Service issues. She is a Senior Member of IEEE.

Yan Shi (M'10) received his Ph.D. degree from Xidian University in 2005. Now, he is an associate professor of State Key Laboratory of ISN in Xidian University. His present research interests include cognitive networks, modern switching technologies, and distributed wireless networking. He is a member of the IEEE.