

A Method for Estimating Access Delay Distribution in IEEE 802.11 Networks

Ehsan Haghani, Michael N. Krishnan, and Avideh Zakhor
Video and Image Processing Lab
Department of Electrical Engineering and Computer Science
University of California, Berkeley
Email: {Haghani, Mkrishna, Avz}@eecs.berkeley.edu

Abstract—The volume of multimedia traffic over wireless networks has been steadily increasing over the past decade. Unlike web browsing applications, multimedia data needs to satisfy stringent delay requirements since late packets are as good as lost packets. In this paper, we present a framework for the nodes in 802.11 networks to estimate the distribution of uplink access delay in Distributed Coordination Function (DCF) MAC mechanism using locally available information. The access delay for a packet is defined as the time between the packet arriving at the head of line of MAC queue, and its ACK being received. In our proposed framework, each node periodically records channel occupancy information to estimate the distribution of access delay. We use NS-2 simulations to verify the accuracy of our proposed approach.

I. INTRODUCTION

Wireless Local Area Networks (WLANs) have been pervasively implemented as “WiFi Hotspots” by carriers in order to provide users with broadband wireless connectivity, and to reduce the cost of service and traffic load in wide area wireless networks. Increased number of WiFi networks interfering with each other, in addition to the increased number of users can significantly deteriorate the Quality of Experience (QoE) of multimedia applications. This is particularly alarming since the volume of multimedia traffic over wireless networks has been skyrocketing over the past few years, and is likely to grow at an extremely rapid rate in the coming years [1].

Many multimedia applications such as VoIP or video conferencing require the nodes to send delay sensitive traffic into the network. Stringent delay requirements, together with large volume of data typically associated with multimedia applications involving video, pose unique challenges to providing acceptable QoE for multimedia applications. To provide users with acceptable QoE, it is beneficial for the application layer at the nodes to be aware of the delay characteristics of their WLAN connections. In 802.11 networks, nodes compete to access the channel and may experience delays due to the activities of the surrounding nodes. Further, a transmission attempt may fail due to either collision or wireless channel loss. Either way, the transmitted packet waits for the next transmission attempt increasing the overall delay of the packet.

In IEEE 802.11 networks, Distributed Coordination Function (DCF) or CSMA/CA uses carrier sensing to determine the availability of the shared medium prior to transmission. Two

different approaches to carrier sensing are supported by DCF: Virtual Carrier Sensing (VCS) and Physical Carrier Sensing (PCS). In VCS, each node uses the request to send/clear to send (RTS/CTS) messages to reserve the access to channel prior to transmission. It is assumed that nodes surrounding the destination receive the CTS message, thus avoiding collision. However, this approach is rarely used in practice due to its large overhead. Moreover, there are several scenarios where RTS/CTS results in lower throughput [2]. In PCS, each node examines the status of the channel prior to transmission by comparing the measured received energy in the wireless channel with the Carrier Sense Threshold (CST). The node attempts channel access only if the measured energy level of the channel is less than the CST indicating that the channel is idle; otherwise, the node backs off to wait for a random period of time.

Since Access Points (AP) do not have enough information about the delay characteristics of the nodes, it is potentially more accurate and computationally efficient for each node to individually estimate its own delay. In this paper, we are primarily concerned with estimating the *access delay* at each node which is defined as the time between the packet arriving at the head of line of MAC queue and its ACK being received.

The delay characteristics of 802.11 networks have been extensively studied in recent years. A large number of papers focus on the average and variance of packet delay [3], [4]; however, these moments are typically inadequate for delay sensitive applications where percentile delay distribution is needed. Existing approaches to estimating access delay in 802.11 networks make non-realistic assumptions such as network being under saturated traffic, no hidden node being present in the network, number of competing nodes being known, or no loss happening due to physical layer errors [5]–[9]. As mentioned earlier, inefficiencies of RTS/CTS signaling for mitigating hidden node problem preclude VCS as a feasible solution. In addition, since practical implementations of 802.11 networks include multiple APs with overlapping coverage, hidden node collisions are unavoidable. Furthermore, it is possible for neither the nodes nor the APs to know the exact number of competing nodes due to collision with nodes in neighboring networks.

In this paper, we propose a framework to estimate the

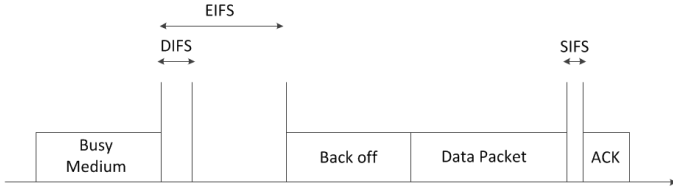


Fig. 1. DCF process.

access delay for the uplink (UL) at each node of an 802.11 network. Specifically, we develop a method by which each node estimates the access delay distribution using available local information such as channel occupancy. Our solution is applicable to realistic topologies with hidden nodes and unknown number of competing nodes. Further, we make no assumption about traffic saturation. Since each node can estimate the distribution of its access delay individually, our solution is scalable, and allows the nodes to readjust to the dynamics of the network by re-estimating delay distribution as needed.

The outline of the paper is as follows. Section II provides an overview of the channel access mechanism in 802.11 networks. In Section III, we describe our proposed framework for estimating the distribution of access delay. We present NS-2 simulation results in Section IV, and concluding remarks in Section V.

II. CHANNEL ACCESS MECHANISM IN 802.11 NETWORKS

In 802.11 networks, each node uses DCF, a CSMA/CA protocol, to determine the availability of the shared medium prior to transmission. Fig. 1 presents the DCF process for a successful packet transmission. If the medium is sensed idle for a period of Distributed Inter Frame Spacing (DIFS), the node proceeds with the transmission of the packet; otherwise, it waits till the medium remains idle for a period of DIFS. It then initiates a back off process whereby it randomly chooses a value for the back off counter. The back off counter is uniformly distributed in the interval $[0, CW - 1]$, where CW is the contention window size. At the first transmission attempt of a packet, CW is set to a minimum contention window size, CW_{min} and after each failed retransmission of the packet its value is doubled. The maximum possible value for contention window is $CW_{max} = 2^z CW_{min}$, where z is the maximum number of times that CW can be doubled. The value of CW remains at CW_{max} if further retransmissions are failed. The values of CW_{max} and CW_{min} are defined in the IEEE 802.11 standard.

During the back off process, the node initiates the back off counter with a random value and decrements the counter by one at every idle time slot. The node freezes counting at every busy time slot, and proceeds decrementing the counter after the channel becomes idle for an inter frame spacing time. This inter frame spacing can have one of two durations; DIFS if the node could correctly decode the header of the packets received during the busy period, and Extended IFS (EIFS) which is longer than DIFS, otherwise.

The receiver sends an ACK frame after a Short IFS (SIFS) period from the time it successfully receives the packet. The transmitter assumes that the packet is lost if it does not receive the ACK packet within the ACK timeout period and initiates the retransmission process for the packet. The packet is finally dropped if maximum number of retransmissions is reached. The node sets $CW = CW_{min}$ and starts a new back off process if it receives the ACK from the destination, or if the packet is discarded.

III. PROPOSED FRAMEWORK

In this section, we propose an analytical method to estimate the distribution of access delay for DCF protocol. This is the delay from the time a packet reaches head of the queue for transmission in the MAC layer of the transmitter till its ACK is received, including the waiting and transmission time consumed for retransmissions. We propose a distributed method through which each node locally estimates its access delay distribution in uplink, i.e., from the node to the AP.

Our approach is for each node to sense its local channel occupancy by recording the busy or idle status of its channel at each time slot in order to generate a binary busy/idle (BI) vector [10]. This can be used to generate the histograms representing the distributions of the length of the busy and idle periods in the unit of time slot. Specifically, the length of a busy/idle period is the number of consecutive time slots that the medium stays busy/idle. Fig. 2 shows samples of derived distributions of busy and idle periods at a node obtained by recording the BI vector for 3 seconds in an NS-2 simulation comprising of 7 APs and 50 randomly placed nodes. These distributions are generated under saturated network conditions and fixed number of stationary nodes.

In the remainder of this section, we propose a method to estimate the access delay a packet encounters within each retransmit attempt.

A. Access Delay at a Retransmission Attempt

We define d_H^m to be the access delay distribution of a packet at its m^{th} retransmit attempt, which is the time between the packet arriving at the head of MAC queue after $m - 1$ failed transmissions till it gains access to the channel. As shown in Fig. 3, access delay time is comprised of consecutive busy and idle periods. Therefore,

$$d_H^m = I_1^m + B_1^m + \dots + B_{N_m-1}^m + I_{N_m}^m \quad (1)$$

where B_i^m and I_i^m represent lengths of the i^{th} busy and idle periods respectively in units of time slots during the m^{th} retransmission attempt, and N_m is a random variable indicating the number of idle periods during the back off in the m^{th} retransmission attempt. It is assumed here that d_H^m begins with an idle period; however, this may not be the case for the first transmission attempt, i.e., $m = 1$, as discussed later.

As described in Sec. II, a node attempts its m^{th} retransmission of a packet when the back off counter counts W_m idle

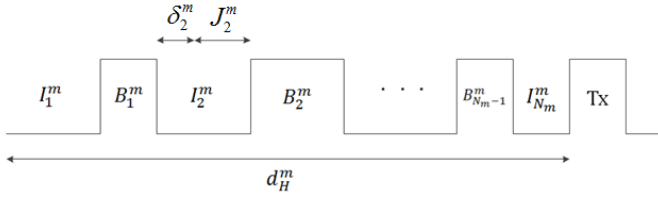


Fig. 3. Access delay at the m^{th} retransmission.

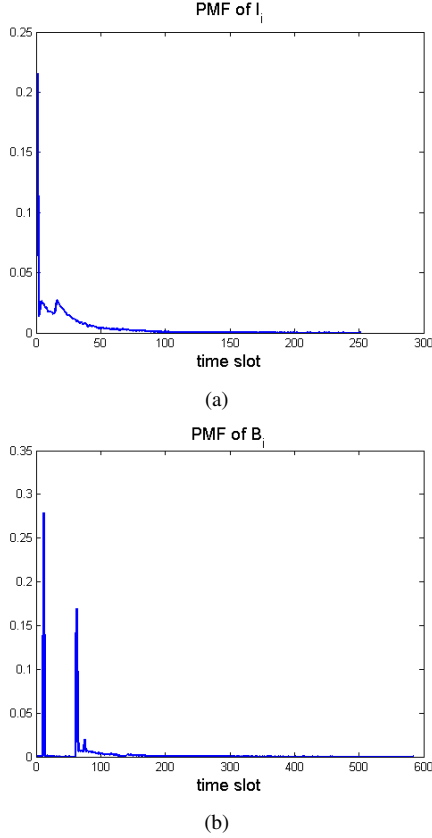


Fig. 2. Derived distributions for a sample node: (a) idle (b) busy.

slots, where W_m is the random variable denoting the value of the back off counter at the m^{th} retransmission attempt. The back off counter pauses after each busy period for the length of DIFS or EIFS. Define δ_i^m as the length of i^{th} period a node pauses in its m^{th} retransmission attempt. Since we assume each node to record the number of times it pauses for DIFS or EIFS, it is possible to empirically estimate the distribution of δ_i^m at each node via

$$P_g = Pr\{\delta_i^m = DIFS\} = 1 - Pr\{\delta_i^m = EIFS\}. \quad (2)$$

For any value of i and m , B_i^m , I_i^m , and δ_i^m are independent identically distributed random variables. As such, for the remainder of the paper, we omit the index m for simplicity unless there is a need to emphasis the m^{th} retransmission attempt.

To compute the distribution of d_H^m , we need to find the

distribution of N_m . To proceed, define

$$J_i \triangleq \max(0, I_i - \delta_i). \quad (3)$$

Intuitively, J_i is the amount that the back off counter is decremented during the i^{th} idle period, as shown in Fig. 3. When $I_i < \delta_i$, a new busy period begins before the pause period has ended; in this case, the back off counter does not get a chance to be decremented and hence $J_i = 0$. Otherwise, the back off counter is decremented by $I_i - \delta_i$. To find the distribution of J_i , we define

$$U \triangleq f_{I_i}(I_i) \otimes f_{\delta_i}(-\delta_i) \quad (4)$$

where $f_{I_i}(I_i)$ and $f_{\delta_i}(\delta_i)$ are the PMFs of I_i and δ_i respectively and \otimes denotes convolution. Since $J_i(j) \geq 0$ for all j , the distribution of J_i , denoted by $f_{J_i}(j)$, is zero for $j < 0$. When $I_i \leq \delta_i$, from (3) we conclude $J_i = 0$. Thus,

$$Pr\{J_i = 0\} = Pr\{I_i \leq \delta_i\} = \sum_{h=-\infty}^0 U(h).$$

Therefore, the distribution of J_i can be written as

$$f_{J_i}(j) = \begin{cases} U(j) & j > 0 \\ \sum_{h=-\infty}^0 U(h) & j = 0 \\ 0 & j < 0 \end{cases} \quad (5)$$

Since J_i is the actual amount of back off decrement at i^{th} period, by definition, we obtain:

$$\sum_{i=1}^{N_m} J_i = W_m. \quad (6)$$

Define non-decreasing sequence $S_n \triangleq \sum_{i=1}^n J_i$ to represent the total number of time slots the back off timer has decremented by the end of n^{th} idle period. Denoting the realization of random variable W_m by w_m , N_m is the first passage time of the sequence S_n above the threshold w_m and as such, can be written as:

$$N_m = \arg \min_n S_n \geq w_m. \quad (7)$$

The distribution of S_n is found by convolving the distribution of J_i with itself for n times. However, for sufficiently large values of n , the distribution of S_n can be modeled with a normally distributed random variable with known mean and variance according to the central limit theorem.

Since N_m is the first passage time of sequence S_n above threshold w_m , we have

$$\begin{aligned} P\{N_m = n\} &= P\{S_n \geq w_m, S_{n-1} < w_m\} \\ &= P\{S_n \geq w_m | S_{n-1} < w_m\} P\{S_{n-1} < w_m\} \\ &= (1 - P\{S_n < w_m | S_{n-1} < w_m\}) P\{S_{n-1} < w_m\} \\ &= \left(1 - \frac{P\{S_n < w_m, S_{n-1} < w_m\}}{P\{S_{n-1} < w_m\}}\right) P\{S_{n-1} < w_m\} \\ &= P\{S_{n-1} < w_m\} - P\{S_n < w_m, S_{n-1} < w_m\} \\ &= P\{S_{n-1} < w_m\} - P\{S_n < w_m\}. \end{aligned} \quad (8)$$

In last of equality of (8), we use the fact that $S_n \geq S_{n-1}$.

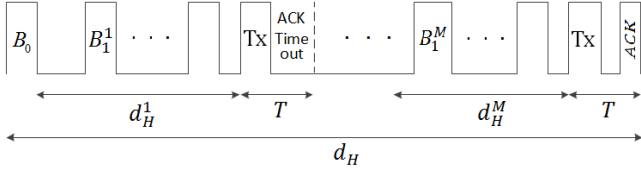


Fig. 4. Total access delay.

Defining $\bar{\delta}_i$ to be actual length of i^{th} pause period in time slot unit,

$$\bar{\delta}_i \triangleq J_i - I_i = \min(\delta_i, I_i). \quad (9)$$

We obtain

$$\sum_{i=1}^{N_m} I_i = W_m + \sum_{i=1}^{N_m} \bar{\delta}_i. \quad (10)$$

Define $f_{d_H^m}(d)$ as the PMF of d_H^m . From (1), we obtain:

$$f_{d_H^m}(d) = \Pr\{d_H^m = d\} = \Pr\left\{\sum_{i=1}^{N_m} I_i + \sum_{i=1}^{N_m-1} B_i = d\right\}. \quad (11)$$

Inserting (10) into the above, we obtain:

$$\begin{aligned} f_{d_H^m}(d) &= \sum_{n=1}^{\infty} \Pr\left\{W_m + \sum_{i=1}^{N_m} \bar{\delta}_i + \sum_{i=1}^{N_m-1} B_i = d \mid N_m = n\right\} \\ &\times \Pr\{N_m = n\} = \sum_{n=1}^{\infty} \Pr\left\{\sum_{i=1}^{n-1} (\bar{\delta}_{i+1} + B_i) + \bar{\delta}_1 = d - W_m\right\} \times \Pr\{N_m = n\}. \end{aligned} \quad (12)$$

W_m is uniformly distributed over $[0, CW_m - 1]$ with CW_m being the size of contention window at the m^{th} retransmission attempt. Assuming random variable W_m takes on the value w_m , and for simplicity, assuming $\bar{\delta}_1 = \text{DIFS}$, we apply the chain rule again on (12) to obtain:

$$\begin{aligned} f_{d_H^m}(d) &= \frac{1}{CW_m} \sum_{w_m=0}^{CW_m-1} \sum_{n=1}^{\infty} \Pr\left\{\sum_{i=1}^{n-1} (\bar{\delta}_{i+1} + B_i) = d - w_m - \text{DIFS}\right\} \\ &\times \Pr\{N_m = n\} \end{aligned} \quad (13)$$

In this paper, we assume $CW_{min} = 32$ and $CW_{max} = 1024$. Thus,

$$CW_m = \min(1024, 2^{(m+4)}).$$

To obtain $f_{d_H^m}(d)$ in (13), we need to find the distribution of the summations of $B_i + \bar{\delta}_i$ defined as

$$Z_n \triangleq \sum_{i=1}^n (B_i + \bar{\delta}_i). \quad (14)$$

Assuming B_i and $\bar{\delta}_i$ to be independent, the distribution of the summation function, Z_n , can be found by convolving the empirical distribution of $B_i + \bar{\delta}_i$ for $n - 1$ times. Using the

central limit theorem, we model the distribution of Z_n , for $n > 20$, with a normally distributed random variable whose mean and variance are known from the empirical distributions of B_i and $\bar{\delta}_i$.

To summarize, (13) and (8) together with node measurements of distributions of $B_i + \bar{\delta}_i$ can be used to estimate the distribution of d_H^m .

B. Total Access Delay

Having estimated the distribution of the access delay at each retransmission attempt, we now estimate the distribution of the total access delay. Fig. 4 shows the total access delay, d_H . We assume that the maximum retransmit limit is γ , and the air time for transmission of a packet is T_1 . After transmitting each packet, the node waits for a fixed ACK timeout period, T_2 , and initiates the retransmission process if ACK is not received by then. Letting $T = T_1 + T_2$, the total access delay, d_H can be written as

$$d_H = B_0 + (d_H^1 + T) + \dots + (d_H^M + T) \quad 1 \leq M \leq \gamma \quad (15)$$

where M is a random variable representing the total number of transmissions of a packet till it is successfully received and B_0 is the length of the first busy period that a packet may encounter at the very first transmission attempt of a packet. It represents the case in which a packet arrives at the head of the MAC queue when the medium is busy due to the transmission of another node. Note that all retransmission attempts, i.e., $m > 1$, start with an idle period as none of the neighboring nodes can initiate a transmission within DIFS from the end of the last busy period.

Using the chain rule, the PMF of d_H can be written as

$$\Pr\{d_H = \tau\} = \sum_{m=1}^{\gamma} \Pr\{d_H = \tau \mid M = m\} \times \Pr\{M = m\}. \quad (16)$$

We can estimate $\Pr\{d_H = \tau \mid M = m\}$ via

$$\begin{aligned} \Pr\{d_H = \tau \mid M = m\} &= \Pr\{B_0 + d_H^1 + \dots + d_H^m = \tau - (m-1) \cdot T\} \\ &= f_{B_0} \otimes f_{d_H^1} \otimes \dots \otimes f_{d_H^m} \end{aligned} \quad (17)$$

which is convolution of the PMFs of the access delays of all possible retransmit attempts $1 \leq m \leq \gamma$ with the PMF of B_0 to be discussed next.

As mentioned earlier, B_0 is the number of time slots between a packet arriving at the head of line of the MAC queue and the medium becoming idle. Thus,

$$\Pr\{B_0 = 0\} = \Pr\{idle\} = \frac{m_I}{m_I + m_B} \quad (18)$$

where m_I and m_B are averages of I_i and B_i respectively. For $b > 0$, the packet arrives at the head of line of MAC queue during a busy period of length L . In this case, $\Pr\{B_0 = b\}$ is the probability of packet arriving in a busy period and the medium stays busy for another b time slots. Therefore, for

TABLE I
SIMULATION PARAMETERS

Parameter	Value
No. Nodes	50
No. AP	7
Carrier Freq.	2.4 GHz
Bitrate	11 Mbps
DIFS	50 μ sec
EIFS	292 μ sec
T	1352 μ sec
Δ	3 sec
time slot	20 μ sec
Packet Size	1500 Bytes
γ	1 and 8

$b > 0$, we can write

$$\begin{aligned}
\Pr\{B_0 = b\} &= \Pr\{B_0 = b|busy\} \Pr\{busy\} \\
&= \sum_{l=b}^{B_{\max}} \Pr\{B_0 = b|L = l\} \Pr\{L = l|busy\} \Pr\{busy\} \\
&= \sum_{l=b}^{B_{\max}} \left(\frac{1}{l}\right) \left(\frac{f_{B_i}(l)l}{\sum_{r=1}^{B_{\max}} f_{B_i}(r)r}\right) \left(\frac{m_B}{m_B+m_I}\right) \\
&= \frac{1}{m_I+m_B} \sum_{l=b}^{B_{\max}} f_{B_i}(l)
\end{aligned} \tag{19}$$

where $f_{B_i}(l) = \Pr\{B_i = l\}$ and B_{max} is the maximum length of a busy period. Note that $\Pr\{B_0 = b|L = l\} = \frac{1}{l}$ because the arrival time of a packet in a busy period is uniformly distributed over l . Furthermore, since the probability of the packet arriving during a busy period of length l is proportional to $f_{B_i}(l)l$, we have $\Pr\{L = l|busy\} = \frac{f_{B_i}(l)l}{\sum_{r=1}^{B_{\max}} f_{B_i}(r)r}$ where the denominator is the average duration of busy periods, namely m_B . Thus, (18) and (19) result in the complete expression for distribution of B_0 which is then used in (17). We can now find the distribution of the total access delay, d_H , by inserting (17) in (16).

$\Pr\{M = m\}$ is the probability of success at exactly the m^{th} transmission attempt given all previous $m - 1$ attempts have failed. Even though each node can measure $\Pr\{M = m\}$, we have empirically found that it can be reasonably approximated via a geometric distribution with parameter P_L , i.e., the probability of loss at first attempt:

$$\Pr\{M = m\} = \begin{cases} P_L^{m-1}(1 - P_L) & 1 \leq m < \gamma \\ 1 - \sum_{m=1}^{\gamma-1} \Pr\{M = m\} & m = \gamma \end{cases} \tag{20}$$

We can now combine (20) and (17) to estimate $\Pr\{d_H = \tau\}$ in (16).

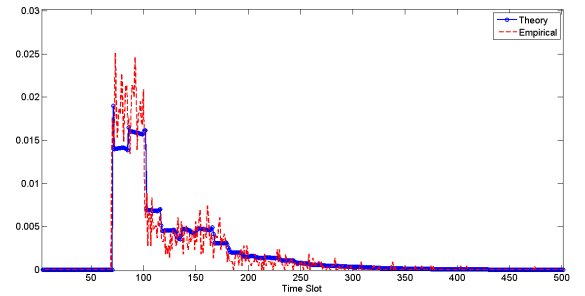
IV. SIMULATION RESULTS

In this section, we investigate the accuracy of the method proposed in Section III for estimating the access delay dis-

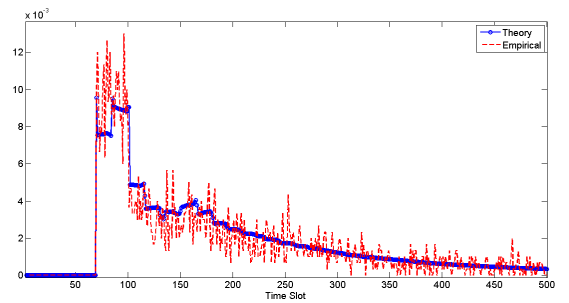
tribution in 802.11 networks using DCF. We use NS-2 for simulating a 802.11b network comprising of 7 APs covering the entire $450^m \times 500^m$ simulation area, and 50 stationary randomly distributed nodes. The network is simulated under saturated traffic condition; even though, our proposed method is applicable to any traffic condition. The network and simulation parameters are shown in Table I.

As explained earlier, each node records its channel occupancy to generate the BI data at the granularity of a time slot for 3 second long periods; this requires less than $7KB$ memory at each node. Using the BI data, nodes can estimate the distribution of the both busy and idle periods. In addition to BI data, each node estimates the distribution of δ_i by counting the number of times it successfully decodes the headers of the overheard packets. By estimating the distributions of B_i , I_i , and δ_i , each node can estimate of the distributions of J_i and Z_n functions using (5) and (14) respectively.

Fig. 5 compares the Probability Distribution Function (PDF) of the total access delay of our method to that of the empirical access delay resulting from the NS-2 simulations for a sample node for $\gamma = 1$ and $\gamma = 8$. The total simulation time is 200 seconds. As seen, the proposed method can accurately estimate the access delay.



(a) $\gamma = 1$



(b) $\gamma = 8$

Fig. 5. Theoretical and empirical comparison of access delay PDF for (a) $\gamma = 1$; (b) $\gamma = 8$.

A. Computational Complexity

We now describe the computational complexity of the proposed framework. The bulk of the computational complexity of our approach comes from estimating of the distribution of

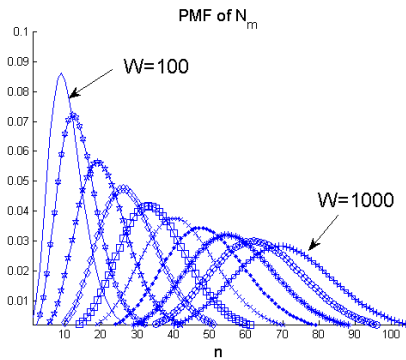


Fig. 6. Distribution of N_m for $W_m = 100, 200, \dots, 1000$.

access delay at each transmission attempt in (13). In Equation (13), we need to find the distribution of N_m via (8). Fig. 6 shows the distribution of N_m for a range of W_m values. As shown, the distribution of N_m has non-zero values only over a small range of n for all values of W_m . As such, finding the distribution of N_m is not computationally burdensome and only needs to be performed once. Furthermore, it is possible to find the small range of n which needs to be used in the inner summation of (13) in order to reduce its computational cost.

Another simplifying idea relies on the fact that distribution of the Z_n is modeled by a normally distributed random variable for sufficiently large values of n , which we have empirically observed to be $n > 20$. Therefore, the distribution of Z_n is smooth and the value of $\Pr\{Z_{n-1} = d - W_m - DIFS\}$ need not be computed for every value of W_m ; rather, it can be approximated by subsampling m for $n > 20$.

Most delay sensitive applications require an end-to-end delay of less than $200ms$ requiring, (13) to be computed for $d < 200ms$. In addition, we usually do not need to find the distribution of the access delay for every value of d since in practical applications, it is sufficient to estimate it with the granularity of about $2ms$. Hence, (13) can be estimated for only about 100 values of d . Computing the distribution of access delay at each transmission attempt and estimating the total access delay from (16) is not complex as they require only γ convolutions which can be efficiently implemented.

Applying the above ideas to simplify computational complexity, we have computed the distribution of access delay for a node in less than 10 seconds by using MATLAB to process the data obtained from NS-2 on a machine with Intel Core i7 (1.7 GHz) processor.

V. CONCLUSIONS

In this paper, we have proposed a framework to estimate the access delay for the uplink at each node in IEEE 802.11

networks. In our approach, each node uses the locally available channel occupancy information to estimate the access delay distribution in a distributed manner. The proposed framework is applicable to general network topologies comprising of hidden nodes, or non-saturated traffic. We have used NS-2 simulations to show that our approach can accurately estimate the access delay distribution.

A competing approach for a node to empirically estimate the distribution of access delay for its 802.11 link is to empirically observe the delay of its packet. A major drawback of such an approach as compared to ours is the relatively long observation time needed in order to obtain a sufficiently accurate estimate. Further, this approach is not sufficiently fast to capture the dynamics of the wireless network. In addition, nodes can only find an estimate of the access delay distribution corresponding to their current traffic conditions.

A key contribution of this work has been to provide the nodes with a tool to efficiently estimate the access delay distribution as well as the impact of parameters such as maximum retransmit limit on the delay performance. We plan to extend, our contribution to estimating the end-to-end delay analysis that also includes the MAC queuing delay assuming packets' arrival pattern is known. With the end-to-end delay estimate, nodes are able to shape their delay sensitive traffic to satisfy the delay constraints at the application layer.

REFERENCES

- [1] "Cisco visual networking index: Global mobile data traffic forecast update, 2009-2014," White Paper, Cisco, Feb. 2010.
- [2] G. Anastasi, E. Borgia, M. Conti, and E. Gregori, "WiFi in ad hoc mode: a measurement study," in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications, PerCom 2004*, March 2004, pp. 145–154.
- [3] P. Chatzimisios, A. Boucouvalas, and V. Vitsas, "IEEE 802.11 packet delay - A finite retry limit analysis," in *Proceedings of the IEEE GLOBECOM'03*, vol. 2, Dec. 2003, pp. 950 – 954.
- [4] M. Carvalho and J. Garcia-Luna-Aceves, "Delay analysis of IEEE 802.11 in single-hop networks," in *Proceedings IEEE International Conference on Network Protocols (ICNP)*, Nov. 2003, pp. 146 – 155.
- [5] T. Issariyakul, D. Niyato, E. Hossain, and A. Alfa, "Exact distribution of access delay in IEEE 802.11 DCF MAC," in *Proceedings of the IEEE GLOBECOM'05*, Dec. 2005.
- [6] A. Banchs, P. Serrano, and A. Azcorra, "End-to-end delay analysis and admission control in 802.11 DCF WLANs," *Computer Communications*, vol. 29, no. 7, pp. 842 – 854, 2006.
- [7] T. Sakurai and H. L. Vu, "MAC access delay of IEEE 802.11 DCF," *IEEE Transactions on Wireless Communications*, vol. 6, no. 5, pp. 1702 – 1710, May 2007.
- [8] O. Tickoo and B. Sikdar, "Modeling queueing and channel access delay in unsaturated IEEE 802.11 random access MAC based wireless networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 4, pp. 878 – 891, Aug. 2008.
- [9] P. Raptis, V. Vitsas, and K. Paparrizos, "Packet delay metrics for IEEE 802.11 distributed coordination function," *Mobile Networks and Applications*, vol. 14, pp. 772–781, December 2009.
- [10] M. N. Krishnan, S. Pollin, and A. Zakhor, "Local estimation of probabilities of direct and staggered collisions in 802.11 WLANs," in *Proceedings of the IEEE GLOBECOM 2009*, Dec. 2009.