

## **UC Santa Cruz**

### **UC Santa Cruz Previously Published Works**

#### **Title**

A State-Aware Persistence Strategy for Multiple Access Protocols with Carrier Sensing

#### **Permalink**

<https://escholarship.org/uc/item/35x0x4kz>

#### **Author**

Garcia-Luna-Aceves, J.J.

#### **Publication Date**

2018

Peer reviewed

# A State-Aware Persistence Strategy for Multiple Access Protocols with Carrier Sensing

J.J. Garcia-Luna-Aceves

Computer Engineering Department, UC Santa Cruz, Santa Cruz, CA 95064

PARC, Palo Alto, CA 94304

jj@soe.ucsc.edu

**Abstract**—All channel-access protocols designed to date based on carrier sensing (namely CSMA, CSMA/CD, CSMA/CA) and the standards based on them (e.g., IEEE 802.11 DCF) have used transmission strategies that are independent of the state of the protocol. In particular, most protocols assume a non-persistent transmission strategy in which a node with a packet to send that detects a busy channel simply backs off. We introduce the first state-aware persistence strategy for carrier-sense multiple access (CSMA) protocols. A node with a packet to send that detects a busy channel decides to transmit once the channel is free depending on whether the ongoing busy period is successful and how early its local packet is ready relative to the start of the ongoing busy period. We provide a simple unifying analysis for CSMA operating in a wireless network with non-persistence and state-aware persistence. Our analysis considers the use of acknowledgments (ACK) and takes into account the effect that receive-to-transmit turnaround times have on performance. The results show that state-based persistence can provide better throughput values relative to a non-persistent strategy, and that new persistence strategies are needed that tailor the amount of persistence to the channel traffic load.

## I. INTRODUCTION

The channel-access methods that have been developed over the years based on carrier sensing, including those that are path of such protocol standards as IEEE 802.11, assume the same basic transmission strategies first proposed as part of the ground-braking work by Kleinrock and Tobagi on Carrier-Sense Multiple Access (CSMA) [7]. In most channel-access protocols based on carrier sensing, a node with a packet to send that detects a busy channel simply backs off, which is called a non-persistent transmission strategy. The vast majority of contention-based channel-access methods use this strategy, and very few protocols have used a persistent strategy in which a node with a packet to send that detects a busy channel waits until the channel is free and transmits at that time.

Kleinrock and Tobagi introduced the notion of  $p$ -persistence [7] in their description of CSMA. In a nutshell, a node with a packet to transmit that finds the channel idle transmits with probability  $p$  and a node with a packet to transmit that finds the channel busy waits until the channel becomes idle again and the transmits with probability  $p$ .

The obvious limitations of non-persistent and persistent transmission strategies are well known. A non-persistent strategy over-reacts to busy-channel conditions when the channel traffic is light, and a persistent strategy becomes too aggressive when channel traffic increases. However, the few studies

addressing persistence in channel-access methods using carrier sensing have assumed the same approach to persistence defined by Kleinrock and Tobagi [9], [10], [11].

Only the limited-persistence approach [3], [5], which was proposed in the context of CSMA and CSMA with collision avoidance (CSMA/CA), has deviated from this norm. In a nutshell, nodes that find the channel idle transmit their packets, and a node with a packet to send that finds the channel busy waits for a limited amount of time proportional to the time it takes to send a request-to-send (RTS) packet. If the channel continues to be idle after that time, the node backs off; otherwise, the node transmits. The limitations with the limited-persistence approach is that the persistence time used by nodes that find the channel busy is a given constant independent of the operation of the protocol, and in [3] the nodes that receive packets to send closer to the end of the current transmission period are favored over those that receive packets to send earlier in the ongoing transmission period.

This paper introduces a new transmission strategy for channel-access protocols based on carrier sensing that takes into account the state of the protocol and limits the time during which a node is allowed to persist with its transmission when the channel is busy. We call this transmission strategy *State-Aware Persistence*, and is applicable to any channel-access method based on carrier sensing, including CSMA and many other approaches (e.g., [1], [2], [4], [6], [12]). Section II presents the state-aware persistence strategy in the context of CSMA with priority ACKs.

Section III presents a unifying model for the computation of the throughput of non-persistent CSMA and state-aware persistent CSMA taking into account the use of priority ACKs and the impact of receive-to-transmit turnaround delays that may be longer than propagation delays. The model defines a simple three-state embedded Markov chain for the channel-access protocol based on the state of the channel when a new arrival occurs. For completeness, the throughput of non-persistence CSMA with priority ACKs is also derived following the proof in [4].

Section IV provides numerical results using practical values of relevant parameters. These results show that using a state-aware persistence policy is a viable approach to improving the performance of contention-based channel-access protocols based on carrier sensing compared to a non-persistent policy. Section V presents our conclusions.

## II. STATE-AWARE PERSISTENCE IN CSMA WITH PRIORITY ACKS

### A. Basic Approach for State-Aware Persistence

A state-aware persistence strategy requires that nodes monitor the state of the protocol operation and the state of the channel for the presence of carrier continuously. Fortunately, this is also the case in any channel-access protocol based on carrier sensing and virtual carrier sensing, which is required for the use of priority ACKs. We denote by  $T_c$  the persistence time period a node allows when it has a packet to send and the channel is busy. The local time when the node detects carrier is denoted by  $T_c$  and the local time when the node receives a local packet to send is denoted by  $T_p$ . A node that has obtained a new local packet to send carries out the following steps as part of the channel-access protocol being used:

- 1) If the channel is idle, transition to a state where the node can transmit the packet
- 2) If the channel is busy (node is in REMOTE state) then:
  - (a) Compute  $TD = T_p - T_c$
  - (b) Enter BACK-OFF state if  $TD < 0$
  - (c) Enter PERSIST state if  $TD < 0$
  - (d) If an ACK to other node is heard in PERSIST state, transmit after no carrier or virtual carrier are present

The previous steps can be applied to different types of channel-access methods based on carrier and virtual-carrier sensing, including CSMA/CD [8], CSMA/CA [1], and receiver-initiated collision avoidance methods [2], and other protocols [12]. However, in this paper we focus on how CSMA with priority ACKs can be modified to account for state-aware persistence.

The variables  $T_p$  and  $T_c$  are maintained separately from the state machine of the channel-access protocol, and  $TD$  is computed as soon as a local packet is ready for transmission. Time  $T_c$  is reset when the channel becomes idle again, taking into account the fact that priority ACKs follow a successful data packet and a virtual carrier of  $\tau + \rho$  (turnaround time plus a maximum propagation delay) must be observed. Time  $T_p$  and  $TD$  are reset when the node transitions out of the REMOTE state.

### B. State-Aware Persistent CSMA with Priority ACKs

Fig. 1 illustrates the operation of state-aware persistent CSMA with priority ACKs using a state machine. State-aware persistent CSMA operates just like non-persistent CSMA when a node has to transmit a packet and the channel is idle. The main difference between the two variants is the addition of the PERSIST state. The LOCAL state is used to emphasize the need for a radio to transition from listen to transmit mode, which involves a latency of  $\tau$  seconds during which the node is unable to listen to the channel.

A node that receives a local packet to send and detects no carrier transitions to the LOCAL state and transmits its packet once its radio is in the transmit mode. The node transitions to the DATA state and waits for an ACK from the receiver. It transitions to the PASSIVE state if an ACK is received or to

the BACK-OFF state if the ACK is not received in order to schedule a retransmission.

A node that detects carrier while in the PASSIVE state transitions to the REMOTE state and remembers the value of  $T_c$  (the local time when carrier was detected). If the node has no local packet (LP) to send and carrier goes down without receiving a data packet intended for itself (indicated as “no DATA to self” in the state machine), the node simply goes back to the PASSIVE state silently. If the node receives a data packet for itself from a transmitter (shown as “DATA to self” in the state machine) and the node has no local packet to send, it sends an ACK and goes to the PASSIVE state. The node remembers the arrival of a local packet to send and the local time when that occurs ( $T_p$ ) while it waits to decode an ongoing transmission while in the REMOTE state.

A node in the REMOTE state with a local packet to send that decodes a remote packet sent to itself sends an ACK to the sender and transitions to the PERSIST state if  $TD < 0$ , or transitions to the BACK-OFF state if  $TD > 0$ . Similarly, a node in the REMOTE state that cannot decode a remote data packet for itself transitions to the PERSIST state if  $TD < 0$ , or to the BACK-OFF state if  $TD > 0$ . Once in the PERSIST state, the node waits until the end of carrier and virtual carrier is detected. If an ACK was received indicating a successful transmission period, the node transmits its data packet, and transitions to the DATA state. If no ACK is received when the end of carrier and virtual carrier occurs, the node transitions to the BACK-OFF state.

A node in the BACK-OFF state computes a random back-off time after which it transitions to the PASSIVE state and attempt to transmit as needed if there is no carrier detected.

It is important to observe that making  $\rho = 0$  results in the traditional non-persistent CSMA protocol.

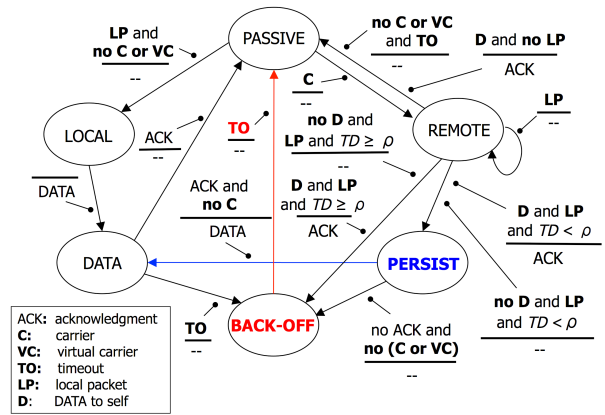


Fig. 1. Operation of state-aware persistent CSMA with priority ACKs

## III. THROUGHPUT ANALYSIS

### A. Model and Assumptions

We assume the same traffic model first introduced by Kleinrock and Tobagi [7] to analyze CSMA with priority ACKs. This model is only an approximation of the real case; however, our analysis provides a good baseline for the comparison of

the transmission strategies in the context of different channel-access methods.

According to the model, there is a large number of stations that constitute a Poisson source sending data packets to the channel with an aggregate mean generation rate of  $\lambda$  packets per unit time. We assume the use of priority acknowledgments (ACK) in all protocols, because they are needed in practice to account for transmission errors not due to multiple-access interference. Each node is assumed to have at most one data packet to sent at any time, which results from the MAC layer having to submit one packet for transmission before accepting the next packet.

The hardware is assumed to require a fixed turn-around time of  $\tau$  seconds to transition from receive to transmit or transmit to receive mode for any given transmission to the channel. According to the parameters assumed in IEEE 802.11 DCF, this value may be larger than the propagation delay  $\alpha$ . The transmission time of a data packet is  $T_0$  and the transmission time for an ACK is  $T_1$ .

We assume that, when a node has to retransmit a packet it does so after a random retransmission delay that, on the average, is much larger than the time needed for a successful transaction between a transmitter and a receiver and such that all transmissions of data packets can be assumed to be independent of one another. The channel is assumed to introduce no errors, so multiple-access interference (MAI) is the only source of errors. Nodes are assumed to detect carrier and collisions perfectly. To further simplify the problem, we assume that two or more transmissions that overlap in time in the channel must all be retransmitted (i.e., there is no power capture by any transmission), and that any packet propagates to all nodes in exactly  $\alpha$  seconds. Equally important, the channel-access protocol is assumed to operate in steady state, with no possibility of collapse.

Given our assumptions, the utilization of the channel can then be viewed as consisting of *transmission periods* that can be classified based on the number of transmissions at the beginning of a transmission period. According to the operation of state-aware persistent CSMA, the type of the next transmission period depends on the arrivals that take place during the persistent time  $\tau$  of the current transmission period.

We call an idle period a transmission period of type 0, or  $TP_0$ , because no transmissions take place at the beginning of the transmission period. Similarly, we call a transmission period that starts with a single transmission a transmission period of type 1, or  $TP_1$ , and call a transmission period that starts with two or more transmissions a transmission period of type 2, or  $TP_2$ .

The type of the next transmission period that occur in the channel depends on the type of the current transmission period and the number of arrivals during the current period if it is successful. Therefore, we can define an embedded Markov chain with three states, one for each type of transmission period that is possible under the time-persistence transmission policy we assume. This leads to the three-state Markov chain shown in Fig. 2. Sohraby et al. [9] also used a three-state

Markov chain formulation to analyze the throughput of 1-persistent CSMA and 1-persistent CSMA/CD with no ACKs.

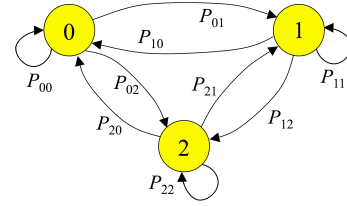


Fig. 2. Markov chain for time-persistence channel-access protocols

Because the channel-access protocol operates in steady state, we have a homogeneous Markov chain, and the channel must return to any given state within a finite amount of time. We denote by  $p_i$  ( $i = 0; 1; 2$ ) the stationary probability of being in state  $i$ , i.e., that the system is in a type- $i$  transmission period. The transition probability from state  $i$  to state  $j$  is denoted by  $P_{ij}$ . The average time spend in state  $i$  is denoted by  $T_i$ . We can then define the throughput of the network to be the percentage of time in an average cycle that the channel is used to transmit data successfully, which is

$$S = \frac{p_1 \bar{U}}{p_0 T_0 + p_1 T_1 + p_2 T_2} \quad (1)$$

We can use the facts that the channel must be in one state at every instant and the channel must transition from one state to another state including itself with probability 1, plus the balance equations for state 1 and state 2, to express the state probabilities as functions of the transition probabilities. From the Markov state diagram in Fig. 2 we have the following four equations

$$\begin{aligned} p_1(P_{12} + P_{10}) &= p_2 P_{21} + p_0 P_{01}; \\ p_0(P_{01} + P_{02}) &= p_1 P_{10} + p_2 P_{20}; \\ p_0 + p_1 + p_2 &= 1; \quad P_{10} + P_{11} + P_{12} = 1 \end{aligned} \quad (2)$$

Given that arrivals are Poisson, there can be no more than one arrival at any instant and hence  $P_{02} = 0$ . On the other hand, because the system is in equilibrium, there must be an arrival within a finite time once the channel is idle, and we have  $P_{01} = 1$  and  $P_{00} = 0$ . In addition, with state-aware persistence, a node that detects a busy channel must back off unless the current transmission period is successful; therefore,  $P_{22} = P_{21} = 0$  and  $P_{20} = 1$ .

Substituting the values of the state probabilities we have obtained in Eqs. (2), we have

$$\begin{aligned} p_0 &= \frac{P_{10} + P_{12}}{1 + P_{10} + 2P_{12}} = \frac{1 - P_{11}}{1 + P_{10} + 2P_{12}}; \\ p_1 &= \frac{1}{1 + P_{10} + 2P_{12}}; \quad p_2 = \frac{P_{12}}{1 + P_{10} + 2P_{12}} \end{aligned} \quad (3)$$

Substituting Eqs. (3) in Eq. (4) we obtain the following expression of  $S$  as a function of transition probabilities  $P_{10}$  and  $P_{12}$ ,  $\bar{U}$ , and the average times of each transmission period:

$$S = \frac{\bar{U}}{(1 - P_{11})T_0 + T_1 + P_{12}T_2} \quad (4)$$

The following sections obtain the values of the required transition probabilities, average times spent in each state, and  $\bar{U}$  for state-aware persistent CSMA and non-persistent CSMA.

### B. State-Aware Persistent CSMA with Priority ACKs

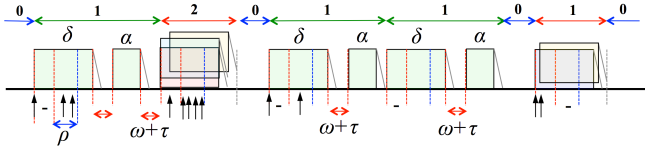


Fig. 3. Transmission periods in state-aware persistent CSMA with priority ACKs

Figure 3 illustrates the transmission periods that may occur with state-aware persistent CSMA with priority ACKs. The figure illustrates a sequence of transmission indicated by the numbers 0, 1, and 2. As the figure shows, only a  $TP_1$  can be successful, and nodes are allowed to persist for seconds only during successful transmission periods. The following theorem provides the throughput of the protocol. The proof takes advantage of the fact that, because arrivals are Poisson distributed, any event involving packet arrivals in a given time interval is independent from another event involving packet arrivals in a non-overlapping time interval.

*Theorem 1:* The throughput of state-aware persistent CSMA with priority ACKs is

$$S_{SP} = \frac{P}{K + (P - e^{-\lambda\alpha}) + e^{-\lambda(\alpha+\delta)} + 1 + (P+1)J} \quad (5)$$

where  $P = P_{12} = 1 - (1 + \lambda\alpha)e^{-\lambda\alpha}$ ;  $J = \alpha + \delta + 2\alpha + 1$ ; and  $K = 1 + \lambda\alpha + \lambda\delta$ .

*Proof:* We observe that new arrivals can occur in the first  $\alpha + \delta$  seconds of a  $TP_1$  or  $TP_2$  because it takes  $\alpha$  seconds for the start of the first transmissions to propagate to all nodes, and a given node that perceives the channel being idle incurs  $\delta$  seconds transitioning from receive to transmit mode and is deaf during that time.

The transition probability  $P_{11}$  equals the probability that no packets arrive during the vulnerability period of the ongoing  $TP_1$  and only one packet arrives during the persistence interval of the transmission period. Therefore,

$$P_{11} = e^{-\lambda(\alpha+\delta)}(1 + \lambda\alpha)e^{-\lambda\alpha} = e^{-\lambda(\alpha+\delta)}e^{-\lambda\alpha} \quad (6)$$

The transition probability  $P_{12}$  equals the probability that two or more packets arrive during the persistence interval of the transmission period, regardless of the number of arrivals during the vulnerability period of the current transmission period. In turn, this probability can be expressed in terms of the complement of that event. Accordingly,

$$P_{12} = 1 - (e^{-\lambda(\alpha+\delta)} + (1 + \lambda\alpha)e^{-\lambda(\alpha+\delta)}e^{-\lambda\alpha}) = 1 - (1 + \lambda\alpha)e^{-\lambda(\alpha+\delta)} \quad (7)$$

The value of  $\bar{U}$  is simply the average time in a  $TP_1$  dedicated to successful data. Given that a successful  $TP_1$  occurs with probability  $e^{-\lambda(\alpha+\delta)}$ , we have  $\bar{U} = e^{-\lambda(\alpha+\delta)}$ .

On the other hand, the average value of an idle period ( $T_0$ ) is simply the average interarrival time of packets, and given that arrivals are Poisson distributed with parameter  $\lambda$  we have  $T_0 = 1/\lambda$ .

The actual length of a  $TP_1$  or  $TP_2$  is a function of the time between the first and the last transmission in the transmission period, which is a random variable  $Y$  that can assume values between 0 and  $\alpha + \delta$ . If the time period between the start of the first and the last data packets in a collision interval equals  $y$  seconds, then there are no more packet arrivals in the remaining time of the vulnerability period of the first packet of the collision interval, i.e.,  $\alpha + \delta - y$  seconds. Accordingly,  $P(Y > y) = F_Y(y) = e^{-\lambda(\alpha + \delta - y)}$ . Therefore, given that  $Y$  assumes only non-negative values, the average value of  $Y$  equals

$$\begin{aligned} \bar{Y} &= \int_0^{\alpha + \delta} (1 - F_Y(t)) dt = \int_0^{\alpha + \delta} 1 - e^{-\lambda(\alpha + \delta - t)} dt \\ &= \alpha + \delta + \frac{1 - e^{-\lambda(\alpha + \delta)}}{\lambda} \end{aligned} \quad (8)$$

Given that a  $TP_2$  starts with two or more transmissions, no success can occur in it, and hence it consists of overlapping packets that cannot be decoded by the intended receivers. Accordingly, the average length of a transmission period of type 2 equals  $\bar{Y} + \alpha + \delta$ , and substituting the value of  $\bar{Y}$  we have

$$T_2 = \alpha + \delta + 2 \frac{1 - e^{-\lambda(\alpha + \delta)}}{\lambda} \quad (9)$$

A  $TP_1$  succeeds if no arrivals occur during its vulnerability period, which occurs with probability  $e^{-\lambda(\alpha+\delta)}$ . If successful, the transmission period includes an ACK, and otherwise it consists of overlapping data packets as in a  $TP_2$ . Given that arrivals are assumed to be Poisson distributed, there can be no more than one arrival at any instant, which means that the case when  $Y = 0$  occurs when the transmission period succeeds, because the first and the last transmission in the period are the same. Therefore,

$$T_1 = T_2 + e^{-\lambda(\alpha+\delta)}(\alpha + \delta) \quad (10)$$

Substituting Eq. (9) in Eq. (10) we obtain

$$T_1 = \alpha + \delta + 2 \frac{1 - e^{-\lambda(\alpha + \delta)}}{\lambda} + \alpha + \delta + \frac{1 - e^{-\lambda(\alpha + \delta)}}{\lambda} \quad (11)$$

Substituting the values of  $\bar{U}$  and  $T_0$  and Eqs. (6), and (10) in Eq. (4) we have

$$S = \frac{e^{-\lambda(\alpha+\delta)}}{1 - e^{-\lambda(\alpha+\delta)} - e^{-\lambda(\alpha+\delta)} + T_2 + e^{-\lambda(\alpha+\delta)}(\alpha + \delta) + T_2(P_{12})} \quad (12)$$

where  $\nu = \alpha + \delta$ .

Substituting Eqs. (7) and (9) in Eq. (12) and simplifying we obtain Eq. (5).

### C. Non-Persistent CSMA with Priority ACKs

The original throughput results for non-persistent CSMA by Kleinrock and Tobagi [7] assume an ideal secondary channel over which ACKs are sent in 0 time. We consider the throughput of non-persistent CSMA with priority ACKs. Figure 4 illustrates the transmission periods in non-persistent CSMA with priority ACKs.

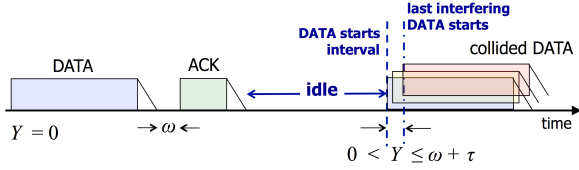


Fig. 4. Transmission periods in non-persistent CSMA with priority ACKs

**Theorem 2:** The throughput of non-persistent CSMA with priority ACKs is

$$S_{NP} = \frac{\bar{U}}{\bar{B} + \bar{T}} \quad (13)$$

*Proof:* The proof is presented in [4] and is included here for completeness.

The protocol is assumed to operate in steady state, with no possibility of collapse. Accordingly, the average channel utilization is given by [7]

$$S = \frac{\bar{U}}{\bar{B} + \bar{T}} \quad (14)$$

where  $\bar{B}$  is the expected duration of a busy period, defined to be a period of time during which the channel is being utilized;  $\bar{T}$  is the expected duration of an idle period, defined as the time interval between two consecutive busy periods; and  $\bar{U}$  is the time during a busy period that the channel is used for transmitting user data successfully.

The vulnerability period of a packet is  $\tau + \omega$ , and hence the probability that it is sent without MAI equals  $P_S = e^{-(\tau + \omega)}$ . The average length of an idle period  $\bar{T}$  is  $1/\lambda$ , because packet arrivals are Poisson distributed with parameter  $\lambda$ , and the average time period used to transmit useful data  $\bar{U}$  is simply  $P_S = e^{-(\tau + \omega)}$ .

An ACK is sent free of MAI if the data packet sent by a transmitter does not collide with any other transmission, and this data-ACK handshake takes  $\tau + \omega + 2\omega$  seconds. The probability that this occurs equals the probability that no arrivals of other data packets take place within  $\tau + \omega$  seconds from the start of the successful packet, i.e.,  $P_S$ .

If a data packet collides with other packets, then no receiver is able to decode any of them. The length of a collision interval is given by  $Y + \tau + \omega$ , where  $Y$  is a random variable representing the time between the arrival of the data packet that starts the collision interval and the arrival of the *last* data packet that creates a collision. A collision interval occurs with probability  $1 - e^{-(\tau + \omega)}$ .

$Y$  varies from 0 to  $\tau + \omega$ , and  $Y = 0$  occurs when a packet is successful, because arrivals are Poisson distributed. Accordingly, the average length of a busy period equals

$$\bar{B} = \bar{Y} + \tau + \omega + e^{-(\tau + \omega)}(\tau + \omega + 2\omega) \quad (15)$$

If the time period between the start of the the first and the last data packets in a collision interval equals  $y$  seconds, then there are no more packet arrivals in the remaining time of the vulnerability period of the first packet of the collision interval, i.e.,  $\tau + \omega - y$  seconds. Accordingly,  $P(Y = y) = F_Y(y) = e^{-(\tau + \omega - y)}$ . Therefore, given that  $Y$  assumes only non-negative values, the average value of  $Y$  equals

$$\begin{aligned} \bar{Y} &= \int_0^{\tau + \omega} (1 - F_Y(t)) dt = \int_0^{\tau + \omega} 1 - e^{-(\tau + \omega - t)} dt \quad (16) \\ &= \tau + \omega - \frac{1 - e^{-(\tau + \omega)}}{\lambda} \end{aligned}$$

Substituting Eq. (16) in Eq. (15) we have

$$\bar{B} = \tau + \omega + 2\omega - \frac{1 - e^{-(\tau + \omega)}}{\lambda} + \tau + \omega + \frac{1 - e^{-(\tau + \omega)}}{\lambda} \quad (17)$$

Substituting the values of  $\bar{U}$ ,  $\bar{B}$ , and  $\bar{T}$  into Eq. (14) we obtain Eq. (13).

Making  $\omega = 0$  in the state machine of state-persistent CSMA renders the non-persistent version of CSMA in which a node with a packet to send that detects carrier or virtual carrier dimly backs off. Hence, it should not be surprising that making  $\omega = 0$  in Eq. (5) results in Eq. (13), which helps validate our main result and the Markov-chain formulation we have presented.

## IV. PERFORMANCE AS A FUNCTION OF PERSISTENCE

### A. Assumptions

The throughput attained by a channel-access protocol is a function of the physical layer and medium-access control (MAC) layer. However, for the channel-access protocols we consider, the physical-layer overhead is roughly the same for all the MAC protocols. For simplicity, we do not consider the PHY-level overhead in our comparison, which means that the actual throughput attained by any of the protocols we consider would be reduced by roughly the same amount.

We assume a channel data rate of 1 Mbps even though higher data rates are common today; this is done just for simplicity. We assume MAC-level lengths of signaling packets similar to those used in IEEE 802.11 DCF. For simplicity, however, we assume that an ACK is 40 bytes. We assume that  $\tau$  is an order of magnitude longer than the propagation delay, which results in lower throughput for all values of  $\lambda$ , because the vulnerability period of a data packet is that much longer.

We normalize the results to the length of a data packet by making  $\tau = 1$ ,  $G = \lambda$ , and  $a = \omega$ ; and by using the normalized value of each other variable, which equals its ratio with  $\tau$  (e.g., the normalized RTS length is  $\tau = \omega$ ).

### B. Numerical Results

We compare the throughput of state-aware persistent CSMA with priority ACKs with its non-persistent counterpart. We use Eq. (5) for state-aware persistent CSMA with priority ACKs to present the throughput ( $S$ ) versus the offered load ( $G$ ) attained for different values of  $\omega$ .

Figure 5 shows the results for a local-area scenario that highlights the performance of the protocols when latencies are very short and signaling overhead is small relative to the time needed to transmit data packets. Physical distances are around 500 meters, and the duration of a data packet is 1500 bytes, which is an average-length IP packet and takes 0.012s to transmit at 1 Mbps. We use a normalized propagation delay of  $a = 1 \cdot 10^{-4}$ .

The curves shown in Fig. 5 correspond to values of  $\alpha$  equal to zero,  $\alpha = 1$ ,  $\alpha = 2$ ,  $\alpha = 4$ ,  $\alpha = 8$ , the length of an ACK ( $k$  in the figure),  $k = 100$ , and  $\beta = 1$ .

Fig. 5.  $S$  vs.  $G$  for time-based persistent CSMA with priority ACKs

As we have pointed out, the case in which  $\alpha = 0$  corresponds to non-persistent CSMA with priority ACKs, and Eq. (5) renders the same results as Eq. (13).

It is clear that smaller values of  $\alpha$  render higher values of throughput at higher loads. This is a consequence of nodes backing off more aggressively when the channel is busy as the value of  $\alpha$  decreases, and the extreme case is when  $\alpha = 0$ .

At the other end of the spectrum for values of  $\alpha$ , with  $\alpha = \infty$ , the performance of state-aware persistent CSMA shows features corresponding to both non-persistent CSMA and the original 1-persistent version of CSMA discussed in the past by Kleinrock and Tobagi [7] and Sohraby et al. [9]. At light loads, state-aware persistent CSMA performs just like 1-persistent CSMA, which reduces channel-access delays. On the other hand, at higher loads the likelihood of successful busy periods decreases, which results in fewer occasions in which nodes persist with their transmissions and makes state-aware persistence behave more and more like non-persistence.

These results clearly show the need to use additional information about the state of the channel as part of the persistence strategy. More specifically, it would be desirable for nodes to be aggressive with their transmissions when the channel load is light and to back off quickly when the channel load is high. For example, the value of  $\alpha$  could approach  $\infty$  or the length of a successful busy period for small values of  $\alpha$  and approach 0 for large values of  $\alpha$ .

## V. CONCLUSIONS AND FUTURE WORK

We introduced state-aware persistence in the context of channel-access protocols based on carrier sensing. With this type of persistence strategy, a node with a packet to send that finds the channel busy persists only if the busy period is successful and its local packet arrived no later than  $\alpha$  seconds (the length of the persistence interval) from the time when the node started to detect carrier. A persisting node transmits its packet as soon as the channel becomes idle again.

We focused on CSMA with priority ACKs and provided an analytical model based on an embedded Markov chain that generalizes the approach first described by Sohraby et al. [9]. In contrast to previous analysis of persistent versions of CSMA, our analysis takes into account the use of ACKs and the effect that receive-to-transmit turnaround times have on performance. An interesting insight from our analysis is that persistence should be based on the state of the protocol *and* the state of the channel. More specifically, a node should persist aggressively after detecting carrier during a successful busy period if the traffic load is light, and should back-off quickly if the channel load is high.

Our future work focuses on the design of persistent strategies that take into account the perceived traffic load and applying the approach to a variety of channel-access protocols (e.g., [1], [2], [4], [6], [12]).

## ACKNOWLEDGMENTS

This work was supported in part by the Jack Baskin Chair of Computer Engineering at UC Santa Cruz.

## REFERENCES

- [1] C. L. Fullmer and J.J. Garcia-Luna-Aceves, "Solutions to Hidden Terminal Problems in Wireless Networks," *Proc. ACM SIGCOMM '97*, 1997.
- [2] J.J. Garcia-Luna-Aceves and A. Tzamaloukas, "Receiver-Initiated Collision Avoidance in Wireless Networks," *Wireless Networks*, 2002.
- [3] J.J. Garcia-Luna-Aceves and A. Tzamaloukas, "The Effect of Exerting Adequate Persistence in Collision Avoidance Protocols," *Proc. IEEE MoMuC '99*, Nov. 1999.
- [4] J.J. Garcia-Luna-Aceves, "Carrier-Resolution Multiple Access," *Proc. ACM PE-WASUN '17*, Nov. 2017.
- [5] J.J. Garcia-Luna-Aceves, S. Thompson, and J. Stern, "Time-Based Persistence in Channel-Access Protocols with Carrier Sensing," *Proc. IEEE IPCCC 2017*, Dec. 2017.
- [6] J.J. Garcia-Luna-Aceves, "Carrier-Tone Multiple Access with Collision Avoidance and Detection," *Proc. IEEE WCNC '18*, April 2018.
- [7] L. Kleinrock and F. A. Tobagi, "Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics," *IEEE Trans. Commun.*, 1975.
- [8] R. M. Metcalfe and D. R. Boggs, "ETHERNET: Distributed packet switching for local computer networks," *CACM*, vol. 19, no. 7, pp. 395 - 403, 1976.
- [9] K. Sohraby et al., "Comments on Throughput Analysis for Persistent CSMA systems," *IEEE Trans. Commun.* January 1987.
- [10] H. Takagi and L. Kleinrock, "Throughput Analysis for Persistent CSMA Systems," *IEEE Trans. Commun.*, July 1985.
- [11] F. A. Tobagi and V.B. Hunt, "Performance Analysis of Carrier Sense Multiple-Access with Collision Detection," *Comput. Networks*, Oct./Nov.1980.
- [12] A. Tzamaloukas and J.J. Garcia-Luna-Aceves, "Channel Hopping Multiple Access with Packet Trains," *Proc. IEEE MoMuC '00*, Nov. 2000.