KALOHA: ike i ke ALOHA

J.J. Garcia-Luna-Aceves
Computer Science and Engineering Department, University of California, Santa Cruz, CA 95064
PARC, Palo Alto, CA 94304
Email: jj@soe.ucsc.edu

***Abstract*—A new family of channel-access schemes called KALOHA (for "Knowledge in ALOHA") is introduced. KALOHA consists of modifying the pure ALOHA protocol by endowing nodes with knowledge regarding the local times when packets and acknowledgments are received, and sharing estimates of channel utilization at the medium access control (MAC) layer. The only physical-layer feedback needed in KALOHA is the reception of correct data packets and their ACKs. A simple Markov-chain model is used to compare the throughput of KALOHA with ALOHA and slotted ALOHA. The analysis takes into account the amount of knowledge that nodes have and the effect of acknowledgments and turnaround latencies. The results demonstrate the benefits derived from using and sharing knowledge of channel utilization at the MAC layer. KALOHA is more stable than ALOHA and attains more than double the throughput of ALOHA, without the need for carrier sensing, requiring time slotting at the physical layer, or using other physical-layer mechanisms.**

***Keywords*-Channel access; ALOHA; analytical modeling**

I. INTRODUCTION

The introduction of the ALOHA protocol by Abramson [1] resulted in a plethora of medium-access control (MAC) protocols for untethered networks being designed and adopted over more than 50 years. CSMA (carrier-sense multiple access) [13] extended ALOHA by endowing receivers with carrier sensing (ability to listen to the channel), and is arguably the most notable successor of ALOHA for wireless networks. The enormous success CSMA resulted in WiFi becoming a pervasive "last link" of the Internet, and there is considerable ongoing effort aimed at improving the efficiency of MAC protocols based on carrier sensing. However, as attractive as CSMA is, the focus of this paper is squarely on ALOHA, because its simplicity is extremely attractive for untethered networks in which carrier sensing is either not possible or becomes too onerous. Examples of such networks are: Untethered networks with long propagation delays (e.g., underwater sensor networks, satellite networks and space networks); wireless networks that operate in noisy environments or terrains in which hidden-terminal interference is prevalent; and IoT deployments consisting of very simple nodes, whether they operate on a peer-to-peer basis or communicate through gateways (e.g., class A devices in LoRaWAN [8]) or access points.

In the basic ALOHA design, a node transmits whenever it has a packet to send and applies a backoff strategy after detecting that its transmission was unsuccessful. The amount of work on ALOHA improvements has been rather limited compared to the amount of work that has been done in the context of MAC protocols using carrier sensing. As Section II summarizes, the most notable improvement over basic ALOHA without requiring carrier sensing is slotted ALOHA, which requires clock synchronization at the physical layer. Furthermore, most of the improvements on ALOHA assume slotted ALOHA, and all previous approaches based on slotted ALOHA that take advantage of channel-state information require receivers to distinguish among time slots that are idle, carry successful transmissions, or contain collided packets.

This paper explores using knowledge held by nodes accessing a common channel to improve the efficiency of ALOHA without any physical-layer assistance other than the ability of a node to decode successfully transmitted packets and their acknowledgments (ACK). The resulting family of channel-access schemes is called **KALOHA**, for "Knowledge in ALOHA," which motivates the second part of the title of this paper in Hawaiian: "ike i ke ALOHA."

Section III presents KALOHA and the use of increasing amounts of knowledge. In the simplest instantiation of KALOHA nodes remember the *local time* when the last acknowledgment was received and use that knowledge to establish virtual time slotting, such that all nodes tend to start their transmissions around the same time, differing from each other by at most one maximum propagation delay, without requiring any clock synchronization. Nodes implement transmission persistence strategies to access the channel with different probabilities depending on events that occur while they wait to transmit. Lastly, nodes can share at the MAC layer their knowledge of perceived channel utilization to adapt the persistence with which they attempt to transmit their packets.

In contrast to prior work on adapting transmission persistence in slotted ALOHA (see [4]) and CSMA (e.g., [9]), KALOHA does not require the use of physical-layer mechanisms to differentiate between idle periods and busy periods during which packet collisions occur.

Section IV presents a Markov-chain model for the computation of the throughput of KALOHA and its comparison with the throughput of ALOHA and slotted ALOHA. The model is based on the one introduced by Sohraby et al.

[18] for the analysis of one-persistent CSMA. The analysis addresses the performance impact of nodes having different amounts of knowledge, and considers the effect of acknowledgments (ACK) and turnaround latencies of radios.

Section V provides numerical results comparing the performance of KALOHA, ALOHA, and slotted ALOHA. The results show that adding knowledge in ALOHA renders substantial throughput gains without the need for time slotting, which would require clock synchronization at the physical layer and the design of transmission frames consisting of a fixed number of transmission slots. Section VI presents our conclusions and directions for future work.

II. Related Work

A number of approaches aimed at improving the efficiency of ALOHA without requiring carrier sensing have been reported over the years. This prior work can be categorized into: time slotting at the physical layer; stabilizing techniques based on ALOHA with time slotting; repetition strategies in which a node transmits the same packet multiple times; collision resolution; enabling the decoding of packets in the presence of collisions; dynamic selection of time slots within a transmission frame; and collision-avoidance handshakes and reservations or elections.

Roberts [16] introduced slotted ALOHA, which requires senders to transmit at the beginning of time slots established by clock synchronization at the physical layer. The key advantage of slotted ALOHA is that it doubles the maximum throughput attainable with pure ALOHA. This is the result of reducing the time during which a data packet is vulnerable to multiple-access interference (MAI) to one packet length, rather than two packet lengths as it is the case in ALOHA. Subsequently, framed slotted ALOHA [14] was proposed in which time slots are organized into frames and users select different time slots for their transmissions randomly. Most of the ALOHA improvements proposed over the years that do not involve carrier sensing have assumed slotted ALOHA or framed slotted ALOHA.

Several stabilizing techniques have been proposed and analyzed in the context of slotted ALOHA and framed ALOHA (see [4], [22]) that attempt to control the probabilities with which nodes transmit to result in at most one data packet being offered to the channel per time slot.

A number of collision-resolution approaches have been proposed to improve over the basic throughput of slotted ALOHA and framed ALOHA (e.g., [5], [10], [19]).

Several proposals consist of using repetition strategies with which each node transmits the same packet multiple times, and relying on physical-layer techniques like code division multiple access (CDMA) or successive interference cancellation (SIC) to attain multi-packet reception (MPR) [10], [11], [15], [17]. The use CDMA and MPR together with repetition strategies have also been applied to pure ALOHA (e.g., [12]).

The performance of ALOHA and slotted ALOHA can also be improved using collision-avoidance (CA) handshakes (e.g., [7]), reservations (e.g., [20]), or or distributed election algorithms (e.g., [3]). We do not address these approaches in detail, because they use ALOHA or slotted ALOHA as part of their signaling.

More recently, machine-learning approaches have been proposed for nodes to select time slots for their transmissions in framed slotted ALOHA in such a way that the utilization of the channel is optimized over time [6], [21].

It is clear that all prior ALOHA improvements have relied on physical-layer support. This, however, results in one or more constraints, namely: (a) requiring time slotting at the physical layer, (b) requiring predefined transmission frames consisting of a fixed number of time slots, (c) assuming that nodes can distinguish among time slots that are idle or have collisions, (d) assuming specific probability distributions for retransmission policies, and (e) using packet arrival rates to estimate the number of backlogged packets.

In slotted ALOHA or framed slotted ALOHA the duration of a time slot must be defined in advance. Consequently, either channel bandwidth is wasted when short packets are sent over longer time slots, or long data packets must be sent over multiple time slots, which increases delivery latencies substantially. If explicit acknowledgments (ACK) are used, either the time slots must be long enough to allow for a data packet and ACK in the same time slot, or ACK's are sent in separate time slots, which may result in much longer delivery latencies. An additional limitation of framed slotted ALOHA is that the number of time slots must be determined upfront and the success probability in a given tine slot depends on the size of the transmission frame. Furthermore, while time slotting at the physical layer can be implemented easily in networks in which all traffic goes to and from satellite transponders or base stations that can enforce clock synchronization, maintaining clock synchronization in multi-hop untethered networks faces many challenges, especially if propagation delays are long. In practice, transceivers may be unable to determine whether a time slot was empty of had collisions, node transmission policies are unknown, and it is not possible to guess arrival rates. CDMA and MPR techniques do attain higher throughput in ALOHA; however, they require more expensive hardware than simple single-antenna transceivers.

III. KALOHA

A. Overview

The objectives in KALOHA are to: (a) operate with simple half-duplex transceivers without carrier sensing, clock synchronization, MPR, or the ability to distinguish between time periods when the channel is idle or has collisions; (b) reduce the vulnerability period of a data packet transmitted into the shared channel as much as possible; (c) support the efficient transmission of variable-length data packets and

ACK's; and (d) allow nodes to reduce the rate at which they access the channel when congestion increases.

The simplest version of KALOHA consists of adding *virtual time slotting* and a *persistence transmission strategy* to ALOHA. These features only require knowledge of the local times when acknowledgments (ACK) are received and can be implemented using simple half-duplex transceivers that cannot distinguish between an idle virtual time slot and one involving collisions.

Virtual time slotting consists of implementing time slotting at the MAC layer without the need for clock synchronization. Each node uses the local time when it received the last successful ACK as the new time origin (t_o) from which it can establish the start and end of virtual time slots of equal length T relative to time t_o. The value of T must be longer than the time needed for a sender and a receiver to exchange the longest allowed packet and the corresponding ACK. This leads to the conservative approach of making T equal to the aggregate of a maximum packet length allowed, the time needed to send an ACK, one maximum round-trip time, and two turnaround times.

A node with a packet to send delays its transmission to the beginning of the next virtual time slot, and the time origin for virtual time slotting is reset to the current local time after the correct reception of an ACK. Virtual time slotting amounts to a simple modification of floor acquisition (virtual carrier sensing) and forces nodes to start transmitting their packets within one propagation delay of each other.

Nodes use a common transmission-persistence strategy by setting the values of *persistence probabilities* with which they transmit at the start of a virtual time slot depending on events that took place in the previous virtual time slot. This is similar to the persistence schemes proposed in the past in the context of CSMA [13].

Lastly, each node can learn about the ongoing utilization of the channel by independently keeping track of the average number of virtual time slots between two successful packet transmissions, or by sharing with others its perceived average of the offered traffic load per virtual time slot. Nodes then adapt their persistence probabilities based on that knowledge. As we show in Section IV, sharing knowledge of channel utilization is needed in order to attain channel utilization that is better than slotted ALOHA when nodes cannot distinguish between idle periods and collisions of data packets.

The rest of this paper describes the mechanisms used in KALOHA when all nodes communicate over a single broadcast radio channel and can hear one another. A collision-avoidance (CA) mechanism needs to be used in KALOHA when hidden terminals exist in order to attain better channel utilization than pure ALOHA, and additional spatiotemporal issues must be addressed when propagation delays are substantially different among nodes and are large relative to packet transmission times.

B. KALOHA State Machine

Fig. 1 illustrates the operation of KALOHA using a state machine. Each node keeps track of the last time when a successful data packet and its ACK were transmitted, and resets the value of t_o to equal its local current time when this occurs. Given the value of t_o, every node organizes its access to the common channel based on virtual time slots of duration T seconds each. KALOHA operates in a way similar to slotted ALOHA because each node with a packet to send delays its transmission to the start of the next virtual time slot. The values of the persistence probabilities are either constants or maintained separately from the state machine shown in the figure.

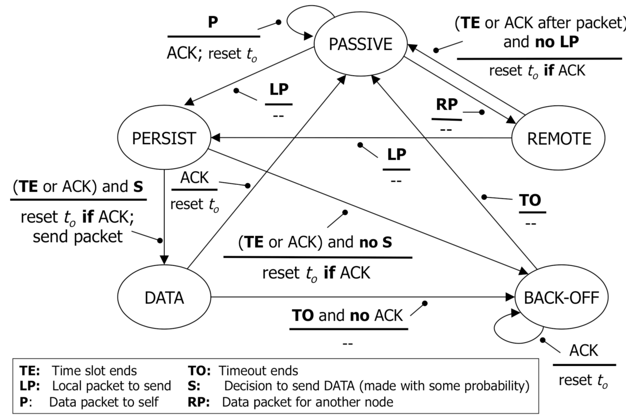

Figure 1. Operation of KALOHA

A node is initialized in the PASSIVE state and waits for a local data packet or a remote data packet. If a node in the PASSIVE state receives a data packet for itself correctly, it sends and ACK to the sender and resets t_o to equal the current local time. If the node receives a local packet to send, it transitions to the PERSIST state. A node in the PASSIVE state that decodes a data packet for another node transitions to the REMOTE state to allow enough time to take place for a complete handshake between a remote sender and a receiver.

A node transitions from the REMOTE to the PERSIST state if it receives a local data packet to send. On the other hand, a node in the REMOTE state transitions to the PASSIVE state if it has no local packet to send when it receives an ACK or the current virtual time slot ends. The node resets the value of t_o before transitioning to the PASSIVE state if it receives an ACK.

A node in the PERSIST state waits until it either receives an ACK or the current virtual time slot ends. The node transmits its data packet at the beginning of the next virtual time slot with a persistence probability whose value depends on the type of event that occurred during the virtual time slot. The node transitions to the DATA state after if it decides to transmit its data packet, and transitions to the BACK-OFF state otherwise. In either case, the node resets the value of t_o if it received an ACK during the current virtual time slot.

A node in the DATA state that receives the ACK it needs transitions to the PASSIVE state after resetting the value of t_o. On the other hand, the node transitions to the BACK-OFF state if it does not receive an ACK during an ACK timeout period. A node in the BACK-OFF state computes a random back-off time after which it transitions to the PASSIVE state and attempts to transmit as needed, and resets t_o after receiving an ACK while in the BACK-OFF state.

C. Virtual Time Slotting

Starting with a time origin t_o, nodes determine the start of the next virtual time slot every T seconds unless a successful transmission occurs. The value of t_o is reset to the current local time after the correct reception of an ACK. As we have stated, the duration of T is the time needed to receive the longest-possible data packet and its ACK, i.e., $T = \delta + \alpha + 2(\omega + \tau)$, where δ is the maximum packet length, α is the length of an ACK, ω is the turnaround delay incurred by a transmitter, and τ is the maximum propagation delay. The length of a successful transmission period involving a data packet of length δ_i is $\delta_i + \alpha + 2(\omega + \tau)$.

Fig. 2 illustrates the way in which virtual time slotting works. The example shown in the figure assumes that the same propagation delay exists between any two nodes. The first transmission period in the example is a successful exchange during which five nodes receive local packets to send. As the figure shows, only two nodes among those five nodes decide transmit their packets at the beginning of the next virtual time slot, which starts when nodes receive the ACK of the successful exchange. No nodes have packets to send during the second virtual time slot, and three nodes have packets to send during the third virtual time slot, but only one node chooses to transmit and this results in a successful transmission period that starts at time $t_o + 2T$ according to the local clock of each node. All nodes reset t_o after receiving the ACK during the last virtual time slot shown in the figure.

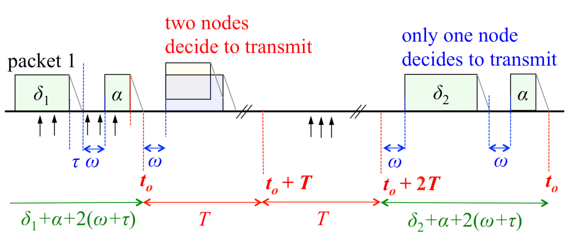

Figure 2. Example of virtual time slotting in KALOHA

In practice, the use of local times means that the start of virtual time slots at different nodes may be off by as much as one maximum propagation delay, depending on distances from a source to receivers. In a satellite-based network, propagation delays between any two ground nodes are very similar, given that all ground nodes communicate through the same satellite transponder and the ACK's can be implicit, i.e., a sender assumes that its transmission is successful if it receives from the downlink what it sent to the uplink. By contrast, propagation delays are small compared to the rest of the length of T in terrestrial ad-hoc wireless networks but explicit ACK's from receivers, gateways, or access points are needed. A maximum allowable propagation delay τ should be used to define the length of a virtual time slot, and this value depends on the physical-layer characteristics of the network (e.g., maximum distances among nodes, type of transmission medium).

D. Transmission Persistence Strategies

There are two possible persistence strategies to allow nodes with simple transceivers to be aggressive with their transmissions when the channel is lightly loaded and can lead to higher efficiency in stable conditions. One strategy consists of using the same persistence probability φ after each virtual time slot. The second strategy consists of using $\varphi = 1$ after a virtual time slot with a successful transmission and $\varphi \leq 1$ after a virtual time slot without a successful transmission, which may be an idle or busy virtual time slot. Choosing fixed values of φ is difficult, because nodes need to be aggressive during periods of light traffic loads and far less so during periods of high traffic loads, which motivates the need for learning.

E. Shared Learning of Channel Utilization at the MAC Layer

Channel throughput can be improved by adapting the persistence probability used by each node based on perceived channel utilization. The only physical-layer indicator of channel utilization available to nodes with simple transceivers is the perceived average number of virtual time slots elapsed between two successive successful transmissions, which we denote by $\tilde{v}$. Unfortunately, $\tilde{v}$ is a bad indicator of channel utilization, because large values of $\tilde{v}$ may be the result of either time slots that are empty or time slots containing collisions. Not surprisingly, all prior approaches focusing on stabilizing slotted ALOHA avoid this problem by assuming that nodes can identify good, bad, and idle time slots [4] using physical-layer mechanisms.

KALOHA utilizes *shared learning of channel utilization* to overcome the limitations of the physical layer by allowing nodes to collaborate with each other at the MAC layer. We describe a simple instantiation of MAC-layer shared learning of channel utilization based on the minimum amount of shared information. The study of more sophisticated approaches to such shared learning is the subject of future work.

A node includes a signal Δ (as succinct as one bit) in each transmitted packet stating a Boolean value that reflects its perceived state of channel utilization. The node sets $\Delta = 1$

if it perceives channel utilization above a threshold value γ, i.e., $\tilde{\upsilon} > \gamma$, and sets $\Delta = 0$ if $\tilde{\upsilon} \leq \gamma$. The following equation is an example of how the persistence probability in KALOHA is made a function of Δ, and consequently of values of $\tilde{\upsilon}$ that are qualified through shared learning.

$$\varphi(\Delta, \tilde{\upsilon}) = \begin{cases} 1 & \text{if } \Delta(\tilde{\upsilon}) = 0 \\ \rho & \text{if } \Delta(\tilde{\upsilon}) = 1, \;\; \rho \in \mathbf{R}^{+}, 0 < \rho < 1 \end{cases} \quad (1)$$

The intent in Eq. (1) is for node to be 1-persistent when channel utilization is light in order to attain good efficiency, and be ρ-persistent when channel utilization goes above some threshold γ in order to reduce congestion.

Using a constant value of ρ with $0 < \rho < 1$ is the simplest approach, but does not result in the best performance, and the analysis of more sophisticated approaches to defining ρ as a function of channel utilization needs to be investigated. For example, if the value of Δ were a fairly accurate estimate of the average number of arrivals per virtual time slot, then ρ could be set to $1/\Delta$. The end result would be very similar to what prior approaches advocate for stabilizing slotted ALOHA when nodes are able to detect empty, good, and bad time slots [4]. However, in contrast to that prior work, nodes in KALOHA learn about channel utilization by sharing knowledge at the MAC layer rather than by relying on physical-layer mechanisms to make independent estimates.

The value of $\tilde{\upsilon}$ is updated after the reception of a successful packet as follows:

$$\tilde{\upsilon} = \alpha \upsilon_s + (1 - \alpha)\tilde{\upsilon} \quad \text{with } \alpha \in \mathbf{R}^{+} \text{ and } 0 < \alpha < 1 \quad (2)$$

where υ_s is the latest sample of the number of virtual time slots elapsed from the prior successful transmission to the latest successful transmission. The initial value of $\tilde{\upsilon}$ can be set initially to 0 or an arbitrary positive value, and α is a parameter used to assign more or less weight to the latest sample. The threshold value of γ indicates that congestion may start taking place as the average gap between successful transmissions becomes smaller.

A node keeps two counters to set the value of Δ, and their values are reset when either a node resets $\Delta = 0$ from $\Delta = 1$, or sets $\Delta = 1$ from $\Delta = 0$. Counter C_0 states the number of packets received stating $\Delta = 0$ and counter C_1 states the number of packets received stating $\Delta = 1$.

If a node currently has $\Delta = 0$, the node sets $\Delta = 1$ when $\tilde{\upsilon} < \gamma$ and either the node's last own packet was not successful or $C_1 > C_0 + \epsilon$, where $1 \leq \epsilon \in \mathbf{Z}^{+}$. Stating $\Delta = 1$ in the node's packets inform other nodes about the onset of channel congestion. Conversely, if a node currently has $\Delta = 1$, the node sets $\Delta = 0$ when $\tilde{\upsilon} \geq \gamma$ and both the node's last own packet was successful and $C_1 \leq C_0 + \epsilon$. Packets stating $\Delta = 0$ inform other nodes that channel congestion is subsiding.

We assume $\epsilon = 1$ for simplicity, but the value of ϵ can be set dynamically.

IV. Throughput Analysis

A. Model and Assumptions

We assume the same traffic model first introduced by Kleinrock and Tobagi [13] to analyze CSMA and ALOHA with priority ACKs. This model is only an approximation of the real case; however, our analysis provides a good baseline for the comparison of KALOHA with ALOHA and slotted ALOHA. According to the model, there is a large number of stations that constitute a Poisson source sending data packets to the the channel with an aggregate mean generation rate of λ packets per unit time. Each MAC protocol is assumed to operate in steady state, with no possibility of collapse.

The channel is assumed to introduce no errors, so multiple access interference (MAI) is the only source of errors. Two or more transmissions that overlap in time in the channel must all be retransmitted and any packet propagates to all nodes in exactly τ seconds. The hardware is assumed to require a fixed turn-around time of ω seconds to transition from receive to transmit or transmit to receive mode for any given transmission to the channel. The transmission time of a data packet is δ and the transmission time for an ACK is α. A node retransmits a packet after a random retransmission delay that is much larger than the time needed for a successful transaction between a transmitter and a receiver on the average, and such that all transmissions of data packets can be assumed to be independent of one another.

With our assumptions, the utilization of the channel can be viewed as consisting of a sequence of virtual time slots that can be classified based on the number of transmissions at the beginning of the time slot. A virtual time slot that is idle, i.e., has no transmissions taking place, is called a time slot of type 0 or TS_0, because no transmissions take place at the beginning of the transmission period. Similarly, we call a time slot that starts with a single transmission a time slot of type 1 (TS_1) and call a time slot that starts with two or more transmissions a time slot of type 2 (or TS_2).

The number of nodes that have packets to send during the current time slot and how they choose to persist with their transmissions determines the type of the next time slot that occurs. A three-state Markov chain can thus be used to characterize the performance of KALOHA. Sohraby et al. [18] also used a three-state Markov chain formulation to analyze the throughput of one-persistent CSMA and one-persistent CSMA/CD with no ACKs. To the best of our knowledge, no similar treatment of ALOHA or its variants has been done in the past.

Given our assumption of steady-state operation, we obtain a homogeneous Markov chain, and the channel must return to any given state within a finite amount of time. We denote by π_i ($i = 0, 1, 2$) the stationary probability of being in state i, i.e., that the system is in a type-i virtual time slot. The transition probability from state i to state j is denoted by P_{ij}. The average time spent 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

$$S = \frac{\pi_1 \overline{U}}{\pi_0 T_0 + \pi_1 T_1 + \pi_2 T_2} \tag{3}$$

where $\overline{U}$ is the average length of time during which the channel is used to transmit a data packet successfully.

To obtain the state probabilities needed in Eq. (3) we use the assumption that the system is in equilibrium, which means that the channel must be in one state at every instant, the channel must transition from one state to another state including itself with probability 1, and the probability of leaving any given state must equal the probability of moving into the same state. Therefore,

$$\pi_0(P_{01} + P_{02}) = \pi_1 P_{10} + \pi_2 P_{20}; \tag{4}$$
$$\pi_1(P_{10} + P_{12}) = \pi_0 P_{01} + \pi_2 P_{21}; \tag{5}$$
$$\pi_2(P_{20} + P_{21}) = \pi_0 P_{02} + \pi_1 P_{12}; \tag{6}$$
$$\pi_0 + \pi_1 + \pi_2 = 1; \tag{7}$$
$$P_{j0} + P_{j1} + P_{j2} = 1 \quad \text{where } j = 0,\ 1,\ 2. \tag{8}$$

The values of the transition probabilities needed in Eqs. (4) to (8) depend on the transmission persistence strategy being used. To simplify the analysis we assume that all virtual time slots have the same length. This means that $T_0 = T_1 = T_2 = T$ and $T = \delta + \alpha + 2(\omega + \tau)$, where δ is the maximum packet length allowed.

B. KALOHA with Implicit ACK's

To provide a direct comparison of KALOHA with known results for ALOHA and slotted ALOHA [1], [16], we assume that turnaround delays are nil, propagation delays are not relevant, and ACK's are implicit with a transmitter learning the fate of its data packet from the physical layer without the need for the intended receiver to send an explicit ACK. An example of this scenario is a satellite-based network in which all nodes have essentially the same propagation delay to a transponder that simply transmits in the downlink what it receives in the uplink from transmitters.

In the scenario we have assumed, the propagation delays become irrelevant and we can set $\delta = T$ and $T_0 = T_1 = T_2 = T$. Given that a TS_1 is always successful we also have $\overline{U} = T$. Substituting these results in Eq. (3) and using Eq. (9) we obtain

$$S = \pi_1 \tag{9}$$

The following theorems state the throughput of KALOHA as a function of the length of virtual time slots and the persistence strategies used by nodes.

Theorem 1: The throughput of KALOHA with implicit ACK's and equal persistence probability after every time slot is

$$S = \varphi\lambda T e^{-\varphi\lambda T} \tag{10}$$

Proof: Fig. 3 illustrates the types of time slots resulting from these assumptions. As the figure indicates, the probability that a node with a packet to send transmits after any type of virtual time slot is φ.

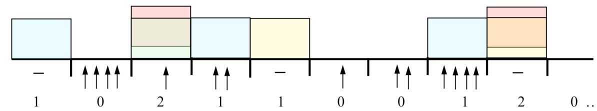

Figure 3. KALOHA with the same persistence probability in all time slots

The type of the next virtual time slot is only a function of the number of arrivals during the current virtual time slot that persist , because the same persistence probability is used during any type of virtual time slot. Accordingly,

$$P_{0j} = P_{1j} = P_{2j} \quad \text{where } \ j = 0,\ 1,\ 2 \tag{11}$$

From Eq. (9) we have $\pi_0 + \pi_2 = 1 - \pi_1$. Using this result and Eq. (11) with $j = 1$ in Eq. (7) we have that

$$\pi_1(P_{10} + P_{12}) = \pi_1(1 - P_{11}) = (1 - \pi_1)P_{11} \tag{12}$$

Therefore,
$$\pi_1 = P_{11} \tag{13}$$

A transition from a TS_1 to another TS_1 requires that either one arrival occurs during the current virtual time slot or that some arrivals did occur in the current time slot but only one of those arrivals persists. We denote by $(K = i)$ the event that i nodes with packets to send persist at the end of the current time slot, and by $(N = n)$ the event that n nodes receive packets to send during the current time slot. Clearly, no node can persist if no packet arrivals occur during the persistence interval of the current transmission period. Therefore,

$$P\{(N = 0)\} = P\{(N = 0) \cap (K = i)\} \quad \text{and}$$
$$P\{(K = i) \mid (N = 0)\} = 0 \quad \text{for } i > 0$$

For any nonnegative value of n, we also have that

$$P\{(N = n) \cap (K = i)\} = P\{(K = i) \mid (N = n)\} P\{(N = n)\}$$

Accordingly, the transition probability P_{11} can be expressed as the sum of the probabilities of mutually exclusive events as follows

$$\begin{aligned} P_{11} &= \sum_{n=0}^{\infty} P\{(K = 1) \mid (N = n)\} P\{(N = n)\} \\ &= 0 + \sum_{n=1}^{\infty} P\{(K = 1) \mid (N = n)\} P\{(N = n)\} \end{aligned} \tag{14}$$

Because each node with a packet to send decides to persist with probability φ independently of any other node, we have for all $k \le n$ that

$$P\{(K = k) \mid (N = n)\} = \binom{n}{k} \varphi^k (1 - \varphi)^{n-k} \tag{15}$$

Using the fact that arrivals during a virtual time slot of length T are Poisson with parameter λ and substituting Eq. (15) with $k = 1$ in Eq. (14) we obtain

$$\begin{aligned} P_{11} = & \sum_{n=1}^{\infty} \binom{n}{1} \varphi (1-\varphi)^{n-1} \frac{(\lambda T)^n}{n!} e^{-\lambda T} \\ = & \sum_{m=0}^{\infty} \frac{\varphi \lambda T e^{-\lambda T} (\lambda T)^m (1-\varphi)^m}{m!} \\ = & \varphi \lambda T e^{-\lambda T} e^{\lambda T (1-\varphi)} = \varphi \lambda T e^{-\varphi \lambda T} \end{aligned} \quad (16)$$

The result for P_{11} in Eq. (16) is intuitive given the Poisson-arrival assumption of our model. Each arrival that takes place during a virtual time slot of T seconds is "selected" with probability φ to persist independently of other arrivals, which amounts to decomposing the Poisson source into two independent streams defined by φ and $1-\varphi$. Eq. (16) can be viewed as a consequence of this, because it is well known that decomposing a Poisson process with parameter λ into two or more independent streams results in each stream randomly selected with probability p being a Poisson process with parameter $p\lambda$. The result of the theorem follows from Eqs. (9), (13), and (16). ■

We note that making $\varphi = 1$ in Eq. (10) results in $S = Ge^{-G}$ with $G = \lambda T$, which is the well-known throughput result for slotted ALOHA.

Theorem 2: The throughput of KALOHA with implicit ACK's, a persistence probability of 1 after a successful transmission, and a persistence probability of φ after a virtual time slot without a successful transmission is

$$S = \frac{\varphi \lambda T e^{-\varphi \lambda T}}{1 + \lambda T \left(\varphi e^{-\varphi \lambda T} - e^{-\lambda T} \right)} \quad (17)$$

Proof: The probability that a node with a packet to send transmits after a TS_1 is 1, and the probability is φ after a TS_0 or TS_2. The transition from a virtual time slot that is either empty or has a successful transmission (either a TS_0 or TS_2) to the next time slot is the same function of the number of arrivals during the virtual time slot and and the value of φ. Therefore, we have

$$P_{0j} = P_{2j} \qquad \text{for} \qquad j = 0,\ 1,\ 2 \quad (18)$$

Substituting Eqs. (18) and (10) for $j = 1$ in Eq. (7) we obtain

$$\pi_1 = \frac{P_{01}}{1 - P_{11} + P_{01}} \quad (19)$$

We can follow the same approach used in Theorem 1 to show that $P_{01} = \varphi \lambda T e^{-\varphi \lambda T}$. Alternatively, we can simply argue that the stated value of P_{01} follows from the fact that, because arrivals are Poisson, the rate of persisting arrivals in a virtual time slot becomes $\varphi\lambda$ and only one persisting arrival can occur for a transition from an ST_0 to an ST_1. Given that all arrivals persist after a ST_1, $P_{11} = \lambda T e^{-\lambda T}$. The result of the theorem follows from Eqs. (9) and (19) by substituting the values of P_{01}, P_{11} in Eq. (19). ■

Once again, making $\varphi = 1$ results in the same throughput as slotted ALOHA.

C. Impact of ACK's and Packet Lengths

Fig. 2 illustrates the operation of KALOHA when explicit ACK's are used. As we have stated, our analysis assumes that all virtual time slots have the same length, which results in a slightly pessimistic view of the performance of KALOHA given that time slots are longer than needed (by $\alpha + \omega + \tau$) when they are idle or involve collisions. The following theorem states the throughput of KALOHA as a function of the persistence probability taking into account turnaround latencies and assuming that ACK's are explicit.

Theorem 3: Let $T = \delta + \alpha + 2(\omega + \tau)$, the throughput of KALOHA with explicit ACK's equals $(\delta/T)S^*$, where S^* is the throughput of KALOHA with implicit ACK's.

Proof: The proof follows almost directly from the proof of Theorems 1 and 2. Given that each TS_1 is successful, the value of $\overline{U}$ is simply δ, which is the portion of a successful virtual time slot used to transmit a data packet. On the other hand, because all virtual time slots have the same length T, it follows from Eq. (6) that the denominator in Eq. (3) becomes $T(\pi_0 + \pi_1 + \pi_2) = T$. Therefore, the throughput of KALOHA is simply $S = \pi_1 \delta / T$.

The value of π_1 depends on the transmission persistence strategy, and the result of the theorem follows by substituting the value of π_1 derived in Theorems 1 and 2. ■

The result from the previous theorem is fairly obvious under the assumption that all virtual time slots have equal length. The efficiency of KALOHA is reduced by the ratio of the average length of packets sent successfully over the length of a virtual time slot used for transmission. This reduction is actually smaller, because the length of a successful transmission period is proportional to the length of the transmitted packet.

The following theorem provides the throughput of ALOHA using the same conditions assumed for KALOHA. To the best of our knowledge, this is the first result that considers ALOHA with priority ACK's and turnaround latencies.

Theorem 4: The throughput of ALOHA with explicit ACK's is

$$S = \frac{\lambda \delta e^{-2\lambda\delta}}{1 + \lambda e^{-\lambda\delta} \left(\tau + \lambda e^{-\lambda\delta} [\alpha + \omega + \tau] \right)} \quad (20)$$

Proof: Figure 4 illustrates the types of transmission periods that occur in ALOHA when priority ACK's are used. The throughput of ALOHA is the percentage of time in an average cycle that the channel is used to transmit data successfully, which is

$$S = \overline{U} / \left(\overline{I} + \overline{B} \right) \quad (21)$$

where $\overline{U}$ is the average time of a successful data packet, $\overline{I}$ is the average length of time that the channel is idle, and $\overline{B}$ is the average length of time that the channel is busy.

A data packet is sent without MAI if there is no other packet arrival while the packet is being transmitted. Therefore, the probability that a packet succeeds is $P_S = e^{-\lambda\delta}$ and $\overline{U} = \delta P_S = \delta e^{-\lambda\delta}$. The value of $\overline{I}$ is simply $1/\lambda$, because arrivals are Poisson with parameter λ.

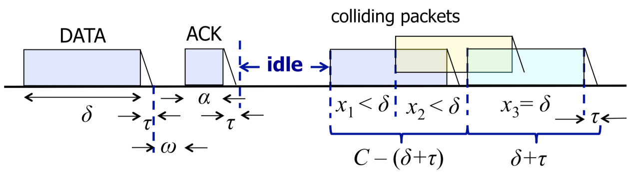

Figure 4. Transmission periods in ALOHA

As Fig. 4 illustrates, the channel is busy with either a successful transmission period of length T_S that includes a single data packet and an ACK, or a collision interval (CI) of length C involving two or more data packets. If the period is successful, an ACK follows the transmitted packet. If the period is a CI, a sequence of colliding packets arrive before the last packet transmitted in the CI.

If $\overline{N}$ is the average number of colliding packets in a CI and $\overline{X}$ is the average inter-arrival time of colliding packets in a CI we have that $\overline{C} = (\overline{N})(\overline{X}) + \tau$. For a CI to have k colliding packets there must be a packet arrival during the transmission time of each of the first $k-1$ packets and no arrival during the transmission time of the last packet of the CI. This corresponds to the geometric random variable in which the probability of successfully ending the CI is the probability that no arrivals occur during the δ seconds of the last packet transmission in the CI, i.e., $e^{-\lambda\delta}$. Hence, the average number of packets in a CI is $\overline{N} = e^{\lambda\delta}$. The inter-arrival times of colliding packets in a CI are exponentially distributed, and each must take some value in the open interval $(0,\ \delta)$. Therefore, the average of such times is

$$\overline{X} = \int_0^\infty (1 - F_X(t))dt = \int_0^\delta e^{-\lambda t}dt = \frac{1}{\lambda}\left(1 - e^{-\lambda\delta}\right) \tag{22}$$

Therefore, we have that

$$\overline{C} = \frac{e^{\lambda\delta}}{\lambda}\left(1 - e^{-\lambda\delta}\right) + \tau = \frac{e^{\lambda\delta} - 1}{\lambda} + \tau \tag{23}$$

To compute $\overline{B}$ we observe that a CI with a single data packet is actually a successful transmission period, which occurs with probability $e^{-\lambda\delta}$ and includes the transmission of an ACK. Therefore,

$$\overline{B} = \overline{C} + e^{-\lambda\delta}[\alpha + \omega + \tau] \tag{24}$$

Substituting Eq. (23) in Eq. (24) we obtain

$$\overline{B} = \frac{e^{\lambda\delta} - 1}{\lambda} + \tau + e^{-\lambda\delta}[\alpha + \omega + \tau] \tag{25}$$

The theorem follows by substituting the values of $\overline{U}$, $\overline{I}$, and $\overline{B}$ in Eq. (21) after some simplification. ■

We observe that making $\tau = \omega = \alpha = 0$ in Eq. (20) to assume implicit ACK's and that propagation delays and turnaround latencies are irrelevant results in the well-known result for the throughput of ALOHA with $G = \lambda\delta$.

D. *Modeling Persistence as a Function of Channel Utilization*

To analyze the impact of $\varphi(\Delta, \tilde{\upsilon})$, we assume that Δ is changed from 0 to 1 when $\tilde{\upsilon} < \gamma$ with $\gamma > 2$. This is because the minimum gap between two successful packets while $\Delta = 0$ would be one empty time slot, which means $\tilde{\upsilon} = 2$. With nodes using $\varphi = 1$ while $\Delta = 0$, assuming $\gamma = 2.25 > 2$ corresponds to an average idle period on length 1.25. Because arrivals are Poisson in our model, it follows that $\overline{I} = 1/(1 - e^{-\lambda T})$. We thus have that $\lambda T \approx 1.6$ when $\overline{I} = 1.25$, and nodes must set $\Delta = 1$ if λT continues to increase beyond that value. With these considerations, we use Eq. (26) below to approximate the values of φ in Eq. (1) for different values of λ, and use the resulting approximated values to obtain numerical results for comparative purposes.

$$\varphi(\lambda) = \begin{cases} 1 & \text{if } \lambda T \leq 1.6 \\ \rho & \text{if } \lambda T > 1.6 \end{cases} \tag{26}$$

V. Numerical Results

We normalize the numerical results to the length of a data packet by assuming that all data packets have the same length δ and making $\delta = 1$ when explicit ACK's are considered, or $T = 1$ when implicit ACK's are assumed. We use $G = \lambda \times \delta$ or $\lambda \times T$, depending on the scenario being discussed. We also use the normalized value of each other variable, which equals its ratio with δ or T as needed. We assume a data-packet length of 1500 bytes and a normalized propagation delay $a = 1 \times 10^{-4}$. When explicit ACK's are considered they are assumed to consist of 40 bytes.

A. *Impact of Signaling Overhead*

Fig. 5 shows the throughput of KALOHA and ALOHA when explicit ACK's are used and turnaround latencies are taken into account. We assume that $\varphi = 1$ in KALOHA, which renders the same throughput results for the two persistence strategies we have discussed for KALOHA. We do this to focus on the signaling overhead and use Eq. (20) for ALOHA with ACK's. The results show that the signaling overhead is very small in both protocols when propagation delays are small. It is clear that the performance of KALOHA with variable-length data packets would be worse than the result shown in Fig. 5. Given that the throughput of KALOHA is proportional to the ratio δ/T, its performance with variable-length data packets would be reduced by as much as $\delta - \delta_{min}$, where δ_{min} is the length of smallest data packets. However, this is a major improvement over the corresponding degradation in ALOHA resulting from variable-length packets. Abramson [2] showed that the throughput of ALOHA with variable-length packets is far below Ge^{-2G}.

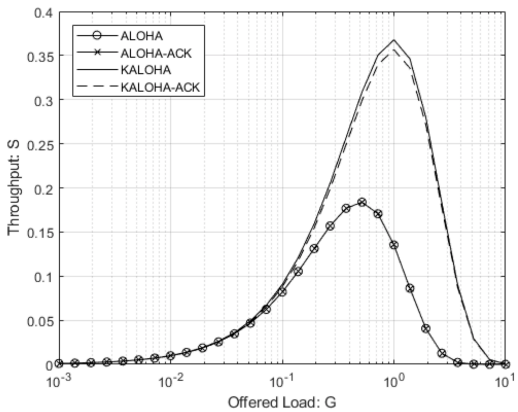

Figure 5. Signaling overhead in ALOHA and KALOHA

The performance of KALOHA and ALOHA with explicit ACK's and long propagation delays would degrade in roughly the same way, given that a packet-ACK exchange takes $T = \delta + \alpha + 2(\omega + \tau)$ seconds. The degradation in slotted ALOHA would be more pronounced than in KALOHA, because either time slots should last T seconds, or ACK's should be sent in the next time slot after the time slot with a successful transmission, with each time slot lasting $\delta + \omega + \tau$ seconds in that case.

B. KALOHA with Implicit ACK's vs. ALOHA

We compare the throughput of KALOHA and ALOHA when implicit ACK's are used and fixed values of φ are used in KALOHA. This is done to focus on the effect that simple amounts of knowledge have on performance independently of the impact of propagation delays.

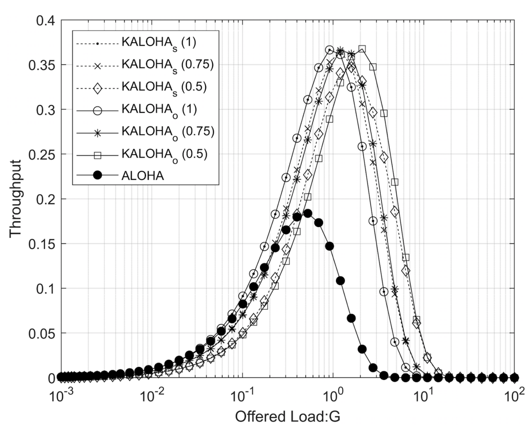

Figure 6. Throughput of ALOHA and KALOHA with fixed values of φ

Fig. 6 shows the throughput versus the offered load for KALOHA and ALOHA. In the figure, KALOHA$_o(p)$ refers to KALOHA with the same $\varphi = p$ being used after every virtual time slot, and KALOHA$_s(p)$ refers to KALOHA with $\varphi = 1$ after a successful packet is received and $\varphi = p$ when a virtual time slot occurs without a successful packet. The well-known throughput result of Ge^{-2G} is used for ALOHA [1]. The results show that using $\varphi = 1$ is the best policy for KALOHA if φ is a constant, and results in the same performance of slotted ALOHA, which can be viewed as the 1-persistent version of KALOHA. We also note that KALOHA performs much better than ALOHA for different values of φ. However, small values of φ lead to channel underutilization at light loads compared to ALOHA.

C. Adapting Transmission Persistence to Traffic Load

Figs. 7 and 8 show the throughput of ALOHA and KALOHA with implicit ACK's to focus on the effect that adapting φ can have on performance. The results for KALOHA in these figures approximate the values of φ given by Eq. (1) with the values of φ given by Eq. (26). KALOHA(1) corresponds to setting $\varphi = 1$. Both figures show results for different constant values of ρ, the persistence probability used by nodes after they perceive channel congestion.

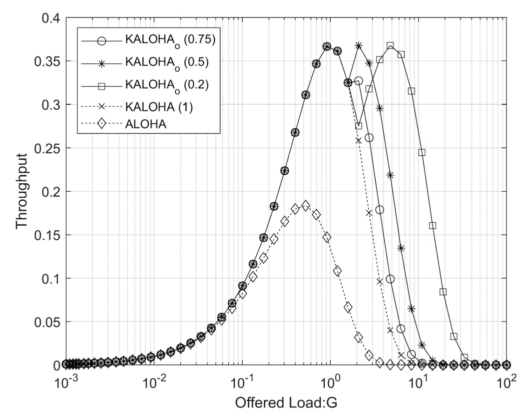

Figure 7. Throughput of KALOHA with same adaptive persistence after every virtual time slot

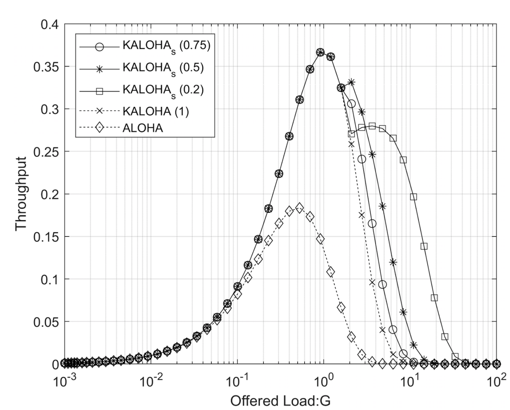

Figure 8. Throughput of KALOHA with persistence of 1 after successful packets and adaptive persistence after virtual time slots without success

Fig. 7 shows the results when the same adaptive persistence probability (i.e., 1 or ρ, depending on channel congestion) is used after any virtual time slot, and Fig. 8 shows the results when a persistence probability of 1 is always used after a successful packet and $\varphi = \rho$ is used after a virtual time slot without a success. It is apparent from the results that using a constant value for $\rho < 1$ improves throughput at high loads because it reduces the rate at which nodes submit packets when the channel is congested. However, it is clearly suboptimal and defining ρ as a function whose value decreases as the level of channel congestion increases is desirable. Doing this can increase throughput and make KALOHA stable.

VI. Conclusions and Future Work

We introduced **KALOHA**, which consists of using additional knowledge in ALOHA to improve its performance without the need to rely on physical-layer mechanisms as has been done in the past. The three innovations in KALOHA are virtual time slotting, the use of transmission persistence strategies, and shared learning of channel utilization at the MAC layer attained by including congestion information in transmitted packets.

We discussed the effect of using increasing amounts of knowledge in ALOHA, and showed that, even when a single bit is added to a packet to indicate whether or not congestion is perceived by the sending node, sharing knowledge of channel utilization at the MAC layer renders substantial performance benefits. We used a simple Markov-chain model inspired by the work reported by Sohraby et al. [18] on the analysis of 1-persistent CSMA to compare the performance of KALOHA with ALOHA. A more accurate analytical model of KALOHA's performance is a needed next step to take into account the actual values of φ. Carrying out simulations with different scenarios is equally important to investigate the impact of the parameters used in KALOHA and different functions used for φ. Another important area of future research is the use of CA techniques in KALOHA for networks in which not all nodes are connected to each other or to a central node.

Acknowledgments

This work was supported in part by the Jack Baskin Chair of Computer Engineering at UC Santa Cruz. Dylan Cirimelli-Low and Najmeh Mashhadi provided valuable help in the presentation of numerical results.

References

[1] N. Abramson, "The ALOHA System–Another Alternative for Computer Communications," *Proc. Fall Joint Computer Conference '70*, 1970.

[2] N. Abramson, "The Throughput of Packet Broadcasting Channels," *IEEE Transactions on Communication*, Jan. 1977.

[3] L. Bao and J.J. Garcia-Luna-Aceves, "Hybrid Channel Access Scheduling in Ad Hoc Networks," *Proc. IEEE ICNP '02*, Nov. 2002.

[4] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, 1992.

[5] F. Clazzer et al., "Enhancing Contention Resolution ALOHA Using Combining Techniques," *IEEE Trans. Commun.*, 2018.

[6] Y. Chu et al., "ALOHA and Q-Learning based Medium Access Control for Wireless Sensor Networks," *Proc. IEEE ISWCS '12*, 2012.

[7] C. Fullmer and J. J. Garcia-Luna-Aceves, "Solutions to Hidden Terminal Problems in Wireless Networks," *Proc. ACM SIGCOMM '97*, Sept. 1997.

[8] J. Haxhibeqiri et al., "A Survey of LoRaWAN for IoT: From Technology to Application," *Sensors*, Nov. 2018.

[9] M. Heusse et al., "Idle Sense: An Optimal Access Method for High Throughput and Fairness in Rate Diverse Wireless LANs," *Proc. ACM SIGCOMM '05*, 2005.

[10] G. Liva, "Graph-based Analysis and Optimization of Contention Resolution Diversity Slotted ALOHA," *IEEE Trans. Commun.*, 2011.

[11] E. Khaleghi et al., "Near-Far Effect on Coded Slotted ALOHA," *Proc. IEEE PIMRC 2017 Workshop on The Internet of Things (IoT)*, Oct. 2017.

[12] C. Kissling, "Performance Enhancements for Asynchronous Random Access Protocols over Satellite," *Proc. IEEE ICC '11*, June 2011.

[13] 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.

[14] H. Okada et al., "Analysis and Application of Framed ALOHA Channel in Satellite Packet Switching Networks - FADRA Method," *Electron. Commun. in Japan*, 1977

[15] E. Paolini, G. Liva, and M. Chiani, "Coded Slotted ALOHA: A Graph- Based Method for Uncoordinated Multiple Access," *IEEE Trans. Information Theory*, Dec. 2015.

[16] L.G. Roberts, "ALOHA Packet System with and without Slots and Capture," ACM SIGCOMM CCR, April 1975.

[17] F. C. Schoute, "Dynamic Frame Length ALOHA," *IEEE Transactions on Communications*, 1983.

[18] K. Sohraby et al., "Comments on Throughput Analysis for Persistent CSMA systems, *IEEE Trans. Commun.*, 1987.

[19] C. Stefanovic et al., "Frameless ALOHA Protocol for Wireless Networks," *IEEE Commun. Letters*, Dec. 2012.

[20] Z. Tang and J.J. Garcia-Luna-Aceves, "Hop Reservation Multiple Access (HRMA) for Ad-Hoc Networks," *Proc. IEEE INFOCOM '99*, 1999.

[21] Y. Yan et al., "Adaptation of the ALOHA-Q Protocol to Multi-Hop Wireless Sensor Networks," *Proc. IEEE European Wireless '14*, May 2014.

[22] J. Yu and L. Chen, 'Stability Analysis of Frame Slotted Aloha Protocol," *IEEE Trans. Mobile Computing*, May 2017.