Key-Activation Multiple Access (KAMA)

Dylan Cirimelli-Low
Computer Science and Engineering Department
University of California, Santa Cruz
Santa Cruz, CA, USA
dcirimel@ucsc.edu

J.J. Garcia-Luna-Aceves
Computer Science and Engineering Department
University of California, Santa Cruz
Santa Cruz, CA, USA
jj@soe.ucsc.edu

***Abstract*—KAMA is introduced and analyzed. KAMA organizes the channel into a sequence of equal time slots and uses a distributed election algorithm to determine which of the known nodes have the priority to transmit during each time slot. KAMA eliminates the need for special signaling packets or the use of special time slots dedicated for signaling packets by means of transmission keys. A node that is unknown to its neighbors can compete for transmission to become known in a subset of time slots in each frame. This subset is defined by a transmission key that maps its identifier to a pattern of on and off time slots in each frame. The performance of KAMA is analyzed analytically and by simulation. KAMA is shown to be more efficient than NAMA, TDMA, CSMA, and CSMA/CA.**

***Index Terms*—ad-hoc networks, channel access, CSMA, MAC protocols**

I. Introduction

Extensive research has been carried out since the introduction of the ALOHA protocol more than 50 years ago to enable efficient channel access in wireless networks by eliminating the negative effects of multiple access interference (MAI). The basic schemes used to orchestrate channel access in medium access control (MAC) protocols can be divided into contention-based schemes like ALOHA itself and Carrier-Sense Multiple Access (CSMA) and contention-free schemes. Section II provides a brief summary of channel-access schemes, given that many surveys already exist on the subject (e.g., [1]). A review of this prior work reveals that contention-based schemes are not suitable for multi-hop wireless networks because of the negative effects of hidden terminals and exposed terminals, which render carrier sensing and collision-avoidance handshakes ineffective. On the other hand, contention-free schemes have problems of their own, which stem from the need to introduce a well-defined channel-access structure in order to avoid or considerably reduce the negative impact of MAI. However, they have the potential of enabling efficient channel access in multi-hop wireless networks. This paper focuses on collision-free channel access of a single channel shared over a multi-hop wireless network.

The contribution of this paper is the introduction and analysis of a new method for contention-free channel access suitable for multi-hop wireless networks that is efficient and much simpler than prior contention-free schemes that support the concurrent use of transmission opportunities by multiple nodes without MAI.

Section III presents the Key-Activation Multiple Access (KAMA) protocol. The novelty of KAMA consists of using: (a) Distributed elections in much the same way as prior schedule-based schemes based on elections, but without the need for special signaling packets or time slots dedicated to the transmission of such packets; and (b) transmission keys defined by the encoding of node identifiers to determine the time slots of a frame in which a node that is entering the network and is unknown to others can compete for the channel. The use of transmission keys to allow unknown nodes to compete for the channel allows KAMA to organize the channel based simply on frames consisting of a large number of equal time slots, without the need to reserve some of them for the exchange of signaling packets. As a result, the throughput of KAMA can be close to the channel capacity. In addition, KAMA uses carrier sensing to expedite the addition of new nodes even further.

Section IV models the performance of KAMA with and without carrier sensing. Section V compares the performance of KAMA, CSMA, CSMA/CA, and TDMA based on analytical and simulation results. The results clearly show that KAMA is more efficient and fair than all the alternative MAC protocols. KAMA performs twice as efficiently as CSMA/CA in multi-hop networks, which is important because CSMA/CA has remained the most widely used MAC protocol for this setting because of its simplicity. Section VI presents future research directions that would improve the robustness of KAMA while preserving its simplicity.

II. Related Work

The channel-access mechanisms used in MAC protocols can be classified as contention-based or contention-free [1]. Contention-based schemes typically rely on carrier sensing to prevent collisions with ongoing transmissions, or collision avoidance to establish ephemeral one-packet reservations of the channel using control signaling handshakes. Contention-based approaches rely on random-backoff procedures to resolve collisions once they have occurred. Although collision avoidance approaches are attractive because of their simplicity, it is well known that such protocols are unfair and that their performance quickly degrades in the presence of hidden- and exposed-terminal problems. Considerable work has been done

ISBN 978-3-903176-42-3 ©2021 IFIP

over the years to eliminate the performance issues associated with contention-based schemes by providing collision-free channel access [1]. The MAC protocols based on collision-free channel access methods are based on dividing access to the channel among nodes in a way that prevents MAI. The most popular of these approaches consists of organizing the channel in time, typically by using transmission frames consisting of multiple time slots. Several approaches have been proposed to allocate time slots to transmitters, including reservations, elections, and codes that determine the time slots in which nodes may transmit.

Some collision-free MAC protocols are topology-independent and assign either a time slot to a single transmitter in the network (e.g., fixed assignment TDMA) or assign a subset of time slots to a transmitter in a way that with probability 1 at least one of the time slots is assigned to only one transmitter (e.g., TSMA [2]). The main limitation with topology-independent schemes is that the channel can be drastically underutilized in multi-hop networks, and TSMA and similar schemes based on coding methods result in channel capacities similar to that slotted ALOHA.

Because of the bandwidth re-use limitations of topology-independent transmission-scheduling schemes, many topology-dependent MAC protocols [1] have been developed that assign the same time slot in a frame to multiple transmitters in a way that multiple access interference is avoided. These protocols use different methods to establish the desired topology-dependent transmission schedules [1], including: reservations, distributed queues, and distributed elections.

Reservation-based schemes (e.g., DAMA [3], MSAP [4]) either require a portion of the transmission frame to be dedicated to reservation-signaling packets, or a portion of each time slot to be used for handshakes between transmitters and receivers or the use of mini-slots for the transmission of reservations. In both cases, reservations are made using slotted-ALOHA type of contention. The limitations with approaches based on reservations include delays in establishing reservations and channel under-utilization resulting from dedicating parts of a transmission frame for reservation signaling. In addition, only a subset of the proposed schemes work correctly in multi-hop networks.

A number of schemes have been proposed based on the establishment and maintenance of distributed transmission queues (e.g., [5]). In a nutshell, a transmission queue consists of a sequence of transmission turns that grows and shrinks on demand, and a short time period is dedicated for requests to join the queue. The most recent examples of this family of approaches have been shown to be very efficient and fair. The main limitation of all protocols based on this approach is that they require a fully-connected network or centralized control to operate correctly.

Bao and Garcia-Luna-Aceves [6] introduced a family of protocols based on *neighborhood-aware contention resolution* (NCR), which is a distributed algorithm that grants collision-free transmission slots based on distributed hash-based elections. The most popular of the proposed schemes based on this approach is Node Activation Multiple Access (NAMA). NAMA guarantees that a transmission from a node is received by all its neighbors without MAI, provided that nodes have knowledge of all other nodes within two hops. NAMA and its variants require complex transmission frames with a portion of the frame used for time slots dedicated to data packets and a signaling portion of the frame used to allow nodes to become known.

A number of subsequent MAC protocols based on the NCR algorithm have been proposed to improve energy consumption at each node. For example, TRAMA [7] is an extension of NAMA that promotes energy saving by allowing nodes to hibernate in slots for which they are not intended receivers. In addition to exchanging neighbor information, nodes using TRAMA periodically transmit transmission schedule information that includes the indented receivers of their transmissions. TRAMA avoids contention in scheduled time slots by periodically entering a random access period during which arriving nodes may announce themselves. The length of the random access period is dynamic, based on the perceived congestion of the network, so that unused random access time is minimized. The main limitations with all existing MAC protocols based on distributed election schemes are that: (a) they may incur long delays allowing nodes to join the network; and (b) they require the use of transmission frames with complex structures in order to allocate portions of the time slots to signaling packets needed to run the elections.

III. Key-Activation Multiple Access

A. Preliminaries and Design Motivation

The design of KAMA attempts to attain the high throughput of topology-dependent scheduling schemes based on elections like NAMA, and the frame simplicity that, at least in theory, topology-independent scheduling schemes like TDMA and TSMA may have.

NCR [6] has been adopted in many previous contention-free channel-access schemes based on elections. NCR assumes that each node i in the network has knowledge of its two-hop neighborhood (also called its contention set), M_i, and that all nodes in the network agree upon a integer transmission context t. Typically, transmission contexts are derived from the ordering of time slots; however, a transmission context could represent a transmission channel or something else. Using the NCR algorithm, each node locally generates a priority using the following priority function:

$$p_k^t = \texttt{Rand}(k \oplus t) \oplus k, k \in M_i \cup \{i\} \quad (1)$$

where k is the unique integer identifier of a node, and $\oplus$ is the concatenation operator, and Rand(x) returns a uniformly distributed pseudo-random number generated from seed x.

The importance of NCR is that it has been proven [6] that, if $\forall j \in M_i, p_i^t > p_j^t$, i may transmit during transmission context t without interference from any $j \in M_i$. NCR can be thought of as a two-coloring graph algorithm that colors a node i with color P if this property holds and ensures that there are no other nodes with color P in M_i.

The limitation of prior approaches based on NCR is that they require special time slots and special signaling packets to operate. On the other hand, previous code-based scheduling schemes have been designed to provide nodes with a subset of time slots during which they can transmit, and ensure that at least one of them is assigned uniquely to a single node [2]. The problem with these schemes is that their maximum throughput is similar to slotted ALOHA, which stems from ensuring that at least one time slot is uniquely assigned to a node independently of the network topology. Their advantage is that they can operate correctly with simple transmission frame structures.

In a nutshell, KAMA endows nodes with two methods to access the channel. Known nodes use NCR to access the channel, and new nodes entering the network attempt to access the channel only during a subset of time slots determined by transmission keys that are unique to them because they are derived by mapping their node identifiers to the set of time slots that form a transmission frame. To expedite the addition of new nodes to the network, new nodes are given priority over known nodes in at least some of the time slots of a transmission frame.

Algorithm 1 Generate key set

Let $\texttt{Rand}(j)$ output a sequence of pseudo-random integers generated from seed j

procedure GENERATE KEY(Seed j)
  $S \leftarrow \{1, 2, \ldots, l\}$
  $bv \leftarrow 0$ initialized bit vector of length l
  $k \leftarrow 0$
  while $k < d$ **do**
    $r \leftarrow \texttt{Rand}(j) \texttt{ mod } |S|$
    $bv_r \leftarrow 1$
    $k \leftarrow k + 1$
    $S \leftarrow S \setminus S_r$
  end while
  return bv
end procedure

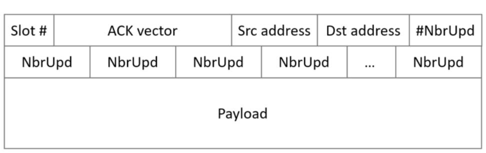


Fig. 1. KAMA packet structure

B. Keys and Channel Organization in KAMA

KAMA organizes the channel into a series of transmission frames, and each transmission frame consists of l transmission turns, where l is a large integer that is also the length of the transmission keys used to control access to the channel. A transmission turn can be any context for NCR. For simplicity, however, this paper assumes that a transmission turn is simply a time slot and that a single shared channel is used.

Node i generates its *key set* for each node $k \in M_i \cup \{i\}$, where M_i is the contention set as in NAMA. A key set is a bit vector of length l with exactly d bits, where $d < l$ '1' bits, generated from the unique identifier of node k. A key set is generated using Algorithm 1. A node that has a '1' in an index n is said to have a key to n.

A node in KAMA runs elections based on NCR to determine which known node should transmit in a time slot n of a frame among those known nodes in $M_i \cup \{i\}$ with a key to n. However, before a node can participate in distributed elections based on NCR, it needs to learn its contention set by listening to the channel, and it also needs to make itself known to its own neighbors. The steps taken to accomplish this are described next.

C. Neighbor Discovery and Error Control

In contrast to prior election-based channel-access schemes, KAMA processes new node arrivals without the need for special signaling packets or dedicating time slots for such packets. After initializing, a node listens to the channel for at least one complete transmission frame to make a best-effort attempt to learn the identities of the nodes in its contention set, which allows the node to compute their transmission keys.

Given that only the known node with the highest priority in a context is allowed to transmit, explicit acknowledgements (ACK) from the receivers cannot be sent within the same transmission turn; furthermore, a packet intended to many receivers would induce the transmission of too many explicit ACK's. To account for this, KAMA nodes maintain an l-length bit vector of ACK's and each node includes the ACK vector in the header of each packet. When a node successfully hears a packet in a transmission turn of a frame, it sets to 1 the corresponding bit in the ACK vector; otherwise, it resets to 0 the corresponding bit. If a node transmits in time slot t and receives a packet with an ACK vector stating $\text{ACK}_t = 1$ from its intended receiver within one frame time, the sender considers its own transmission to be successful.

Because contention sets are generated from the identities of a node's two-hop neighborhood, changes in the state of the one-hop neighborhood of a node must be relayed to its own neighbors. The ACK vector helps to reveal inconsistencies in contention set information. If node i transmits in time slot t and receives *any* transmission stating $\text{ACK}_t = 0$, then node i infers that an inconsistency must exist between its own state and the state of at least one other node in its contention set. Accordingly, the transmitter backs off for a random amount of transmission frames, keeps listening to the channel and, upon return, includes its list of neighbors in its own transmission as an attempt to synchronize contention sets. For the same reason, nodes that overhear collisions or have a change in their neighbor sets should transmit lists of neighbors in their next transmissions.

A node considers itself unknown to the network when it is first initialized. Accordingly, it transmits only during time slots for which it has a key. If a node i transmits in any time slot t and one frame elapses in which every packet received by

i states $ACK_t = 1$, then node i can infer that all its neighbors know about its presence. At this point, node i considers itself known by its neighbors.

Every KAMA packet states the current slot number, which allows arriving nodes to synchronize their local state with the network's. If the network is partitioned, disagreements on the KAMA state may exist between cliques of nodes. When partitions in the network are merged, either due to mobility or the arrival of a node which bridges the partitions, nodes which hear conflicting slot numbers should adopt the smaller of the two, which facilitates the integration of one clique into the other.

D. KAMA Transmission Strategy

If a node has not been acknowledged by its neighbors using the mechanism described in Section III-C, it may access the channel at the start of a time slot only if has a key to the current slot. Otherwise, if $\forall j \in M_i, p_i^t > p_j^t$, then i may access the channel if no carrier has been sensed after one maximum propagation delay τ, which is sufficient for i to sense whether an unacknowledged node is transmitting in the current slot. This policy allows nodes which may not be known to the network to transmit and become known, without causing collisions.

Provided that $d << l$, nodes are able to join the network quickly in KAMA since a node i that has not been acknowledged by its contention set is still likely to be successful joining the network, even if the node itself has little knowledge of its own contention set. This follows from the fact that any neighbor of node i that has been acknowledged will sense the transmission from i within τ time and yield, and any $j \in M_i$ that is unacknowledged but does not have a key to the current time slot does not transmit.

There are different ways in which transmission priorities can be assigned to unknown nodes with respect to known nodes in order to reduce the delays incurred by nodes in joining the network. The simplest approach is to simply give priority to new nodes joining the network, which has been our implicit assumption so far. However, a trade-off exists between how quickly new nodes can join the network and the maximum throughput that can be attained when neighborhoods change. Section IV addresses this performance trade-off between the stability of the system and the delay incurred by arriving nodes. Granting arrivals priority in every slot would effectively squelch all existing nodes, and subsequently introduce high end-to-end delay until the arrivals are acknowledged. Conversely, if arrivals are never given priority, it may take many transmission frames before an arrival is acknowledged, especially in high-traffic networks.

If carrier sensing is not available to nodes, priority access in time slots cannot be given to unacknowledged nodes. Hence, any node simply transmits if it has the highest priority within its known two-hop neighborhood. A simple approach to modify KAMA to account for the lack of carrier sensing consists of making every node operate the same whether or not it has been acknowledged by its neighbors, i.e., simply use NCR. This variant of KAMA can be viewed as the equivalent of NAMA in KAMA, but is much simpler to implement. We refer to it as KAMA-NCS for no-carrier-sensing.

E. Examples of KAMA Operation

Figure 2 illustrates the ability of a new node to quickly succeed in joining the network in KAMA. A new node whose arrival is denoted by a_1 finishes its observation frame during t_1 and has a key to time slot t_2. Accordingly, it transmits at the start of t_2. Another new node whose arrival is denoted by a_2 does not have a key to t_2 and therefore it does not transmit in that time slot. The known node with the highest priority in slot t_2, e, has a data packet arrival during t_1; however, it senses the carrier from the transmission from a_1 after τ time and yields the time slot. As a result, new node with arrival a_1 is the only node to transmit in t_2 and succeeds joining the network.

Figure 3 illustrates how collisions may occur in KAMA-NCS. Without the use of keys for the transmissions of new nodes, both arrivals a_1 and a_2 transmit in t_2. Furthermore, without the use of priorities for transmissions by new nodes joining the network, e transmits as well. The end result is a collision of three packets. Clearly, the use of carrier sensing reduces the maximum length of data packets by τ seconds; however, this is a good performance trade off in wireless networks in which propagation delays are much smaller than the length of a time slot.

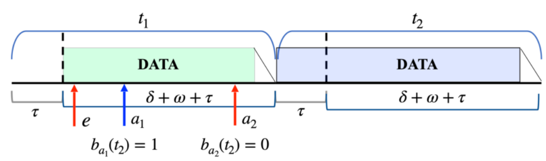


Fig. 2. KAMA priority transmission avoids collision

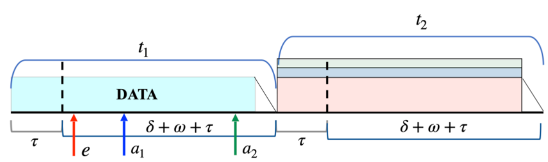


Fig. 3. KAMA-NCS collision between arriving nodes and existing node

IV. Performance Analysis of KAMA

We analyze the performance of KAMA assuming a Poisson traffic model with parameter λ such that there are λ arrivals of packet transmissions to the network per unit time. The arrival of packets at a node is assumed to be independent from the arrival of packets at any other node, and nodes that back off are assumed to do so for a sufficiently long amount of time such that any re-transmission takes place independently from an original transmission. Packet arrivals are also assumed to come from a very large population of nodes, and there is no queuing of backlogged packet arrivals.

All packets are assumed to be of length δ and a packet can be transmitted in a single time slot without fragmentation. Furthermore, all nodes in the system are assumed to be connected in a way that any two transmissions may cause multiple access inference (MAI) and any transmission is sensed by all other nodes after τ seconds. MAI is the only source of packet losses or errors and nodes are never able to decode a packet subjected to MAI. A turnaround time of ω seconds is incurred before a node can receive or transmit a packet after changing states. P_s is the probability that a time slot is utilized successfully. The average number of nodes in the system, N, is assumed to be independent of the node arrival rate. The following probabilities are used for KAMA:

$$P_a = P\{\text{new nodes have priority in a time slot}\}$$

$$P_e = P\{\text{known nodes have priority in a time slot}\} = 1 - P_a$$

We consider two independent Poisson sources, namely: λ_a, which represents the aggregate rate of packet arrivals from nodes that are unknown to the system; and λ_e, which is the aggregate arrival rate of packets from existing nodes.

For analysis purposes, it is assumed that there are N known nodes in the network. These nodes are assumed to have consistent knowledge of their existing contention sets, and no knowledge of new nodes arriving into the network, until they succeed advertising their presence.

The probability that a known node has a packet to transmit during a time slot for which it has the highest priority is denoted by μ. This is just the probability that the known node with the highest priority among N known nodes has at least one packet arrival in the preceding time slot; therefore,

$$\mu = 1 - e^{-(\lambda_e/N)T} \tag{2}$$

A. Throughput

The following theorems state the throughput of KAMA with and without carrier sensing when there are N known nodes in the system and a new node may become known.

Theorem 1: The throughput of KAMA is

$$S = \frac{(1-P_a)\mu + [(1-\mu)\lambda_a T\rho + (1+\lambda_a T\rho)\mu P_a]e^{-\lambda_a T\rho}}{1 + \omega/\delta + 2\tau/\delta} \tag{3}$$

Proof: The throughput of KAMA is the ratio of the time a slot is spent successfully transmitting data packets and the duration of a time slot. A time slot consists of a turnaround time τ to listen for a transmission from an unknown node, an receive-to-transmit turnaround time ω, a payload transmission of length δ, and a second turnaround time τ to allow for the receiver to hear the transmission. On average, δP_s seconds are spent transmitting without MAI in any given time slot. Therefore, the throughput of KAMA is simply

$$S = \frac{\delta P_s}{\delta + \omega + 2\tau} = \frac{P_s}{1 + \omega/\delta + 2\tau/\delta} \tag{4}$$

A time slot in KAMA is utilized successfully if either a single new node transmits successfully, or the known node with the highest priority among the N known nodes has a packet top transmit. How these events occur depends on the type of time slot. A new node is allowed to transmit in a given time slot only if it has a key to the current time slot, which occurs with probability ρ. On average, $\lambda_a T\rho$ new nodes may attempt to transmit in a time slot.

A time slot that gives priority to new nodes is utilized successfully if either exactly one new node has a key to the time slot and hence transmits, or no new nodes transmit but the known node with the highest priority for the time slot has a packet to transmit, which is given by Eq. 2. The latter is possible due to carrier sensing by known nodes. With Poisson arrivals, the probability of exactly one new node having a key to the current time slot is $\lambda_a T\rho e^{-\lambda_a T\rho}$, and the probability that no new nodes have a key to the current time slot is $e^{-\lambda_a T\rho}$. Accordingly, the probability of success for a time slot that gives priority to new nodes is

$$\lambda_a T\rho e^{-\lambda_a T\rho} + \mu e^{-\lambda_a T\rho} \tag{5}$$

On the other hand, a time slot that gives priority to known nodes is used successfully if the known node with the highest priority has a packet to transmit, or it does not transmit but a single new node has a key to the time slot and transmits. The probability of success for this case is then

$$\mu + (1-\mu)\lambda_a T\rho e^{-\lambda_a T\rho} \tag{6}$$

Given Eqs. (5) and (6) and the fact that the two types of time-slot priorities are mutually exclusive, the probability of success for any given time slot is

$$\begin{aligned} P_s &= P_a(\mu e^{-\lambda_a T\rho} + \lambda_a T\rho e^{-\lambda_a T\rho}) \qquad (7)\\ &\quad + (1-P_a)(\mu + (1-\mu)\lambda_a T\rho e^{-\lambda_a T\rho}) \\ &= (1-P_a)\mu + [(1-\mu)\lambda_a T\rho + (1+\lambda_a T\rho)\mu P_a]e^{-\lambda_a T\rho} \end{aligned}$$

The result follows from substituting Eq. 7 into Eq. 4. ■

Theorem 2: The throughput of KAMA-NCS is

$$S = \frac{\delta}{\delta + \omega + \tau}\left(\mu + (1-\mu)\frac{\lambda_a T}{N+1}\right)e^{-\frac{\lambda_a T}{N+1}} \tag{8}$$

Proof: The throughput of KAMA-NCS is given by the ratio of the time a slot is spent successfully transmitting and the duration of a time slot. A time slot in KAMA-NCS consists of a turnaround time ω, a payload transmission of length δ, and a propagation delay τ needed to allow for the receiver to finish hearing the transmission. On average, δP_s seconds are spent transmitting successfully in a time slot. Hence, the throughput of KAMA-NCS is

$$S = \frac{\delta P_s}{\delta + \omega + \tau} \tag{9}$$

A new node in KAMA-NCS transmits in a time slot if it determines that it has the highest priority among the nodes it knows, which occurs with probability $\frac{1}{N+1}$. Therefore, given that no new node is known to any other node, $\frac{\lambda_a T}{N+1}$ new nodes transmit perceive themselves as having the highest priority in a time slot and transmit. With Poisson arrivals, the probability that no new nodes transmit in a slot is $e^{-\frac{\lambda_a T}{N+1}}$ and the probability that a single new node transmits is $\frac{\lambda_a T}{N+1}e^{-\frac{\lambda_a T}{N+1}}$.

A time slot is used successfully if either no new node attempts to transmit in the time slot and the known node with the highest priority transmits, or a single new node transmits in the time slot and the known node with the highest priority for the time slot does not transmit. Therefore, a time slot in KAMA-NCS is utilized successfully with probability

$$P_s = \mu e^{-\frac{\lambda_a T}{N+1}} + (1-\mu)\frac{\lambda_a T}{N+1} e^{-\frac{\lambda_a T}{N+1}} \tag{10}$$

The result follows from substituting Eq. 10 into Eq. 9. ■

B. Network-Joining Delay

To compare the variants of KAMA with NAMA in terms of delays incurred by nodes in joining the network, we make the simplifying assumptions that a new node arriving to the network has complete knowledge of known nodes (without the need to listen to the channel for one complete frame) and that a node does not back off after a failure. Given that this simplification applies to NAMA and all KAMA variants, the results provide an accurate picture of the relative differences among the various schemes.

Theorem 3: The average time a new node arriving to the network spends being unknown in KAMA is

$$D_{KAMA} = \frac{\delta + \omega + 2\tau}{\rho(1-(1-P_a)\mu)} e^{\lambda_a T \rho} \tag{11}$$

Proof: The probability that any new node transmits in a time slot follows a geometric distribution with parameter ρ. Hence, a node waits an average of $\frac{1-\rho}{\rho}$ time slots before transmitting. If a node fails its transmission it will re-attempt in the next time slot for which it has a key. The average number of time slots that elapse before a node is successful is

$$N_T = P_s\left(\frac{1-\rho}{\rho}+1\right) + (1-P_s)\left[\left(\frac{1-\rho}{\rho}+1\right) + N_T\right]$$

which reduces to $N_T = \frac{1}{\rho P_s}$. Each time slot lasts $\delta + \omega + \tau$, so the total time a new node spends before its first success is $N_T(\delta + \omega + \tau)$.

P_s is the probability that an arrival is successful given it transmits, so $P_s = (P_a + P_e(1-\mu))e^{-\lambda_a T \rho}$. Therefore the total incurred delay is $(\delta+\omega+2\tau)[\rho(P_a+P_e(1-\mu))]^{-1}e^{\lambda_a T \rho}$. ■

Theorem 4: The average time a new node arriving to the network spends being unknown to the system in KAMA-NCS is

$$D_{KAMA-NCS} = \frac{(N+1)(\delta+\omega+\tau)}{(1-\mu)} e^{\frac{1}{N+1}\lambda_a T} \tag{12}$$

Proof: The probability that any new node that arrives in the network will transmit in a time slot follows a geometric distribution with parameter $\frac{1}{N+1}$, so a node will transmit, on average, after N time slots. The average number of time slots that elapse before a node is successful is

$$N_T = P_s(N+1) + (1-P_s)((N+1) + N_T)$$

which reduces to $N_T = \frac{N+1}{P_s}$.

A time slot is $\delta + \omega + \tau$ long, so the total time elapsed is $\frac{(N+1)(\delta+\omega+\tau)}{P_s}$. A node will be successful if no other arriving node transmits and no known node transmits, hence $P_s = (1-\mu)e^{-\frac{\lambda_a T}{N+1}}$ and the result on the total delay incurred follows. ■

V. Performance Comparison

A. Results from Analytical Model

We compare the results derived for KAMA and KAMA-NCS in Section IV with NAMA as proposed in [6], as well as CSMA and CSMA/CA, and TDMA with static time-slot assignments. We assume that all data packets have 1500 bytes transmitted at 10 Mbps, which renders a packet time of $\delta = 0.0012s$. We also assume that transmit-to-recieve and reciev-to-transmit turnaround times are $\omega = 1\mu s$, the propagation delay between any two nodes is $\tau = 1\mu s$, and the average network size is $N = 20$, unless stated otherwise. For Figure 5, the parameter α is used to correlate the two Poisson traffic generators λ_a and λ_e such that $\lambda = \alpha\lambda_a + (1-\alpha)\lambda_e$.

1) Throughput Results: Fig. 4 compares the throughput in steady state of KAMA, NAMA, CSMA/CA (Eq. 12 in [8]), CSMA with and without hidden terminals (Eq. 8 in [8], Eq. 20 in [9]) and TDMA. Like KAMA-NCS, a TDMA time slot is $\delta + \omega + \tau$ long. A TDMA time slot is successful whenever the current owner of the channel has a packet to send, so $S_{TDMA} = \delta\mu/(\delta + \omega + \tau)$.

No nodes are arriving or leaving the network and let $\lambda = \lambda_a$. For CSMA and CSMA/CA we consider acknowledgements of 40 bytes, and RTS and CTS packets of 14 bytes. All time values are normalized such that $\delta = 1$, allowing the offered load to be $G = \lambda$.

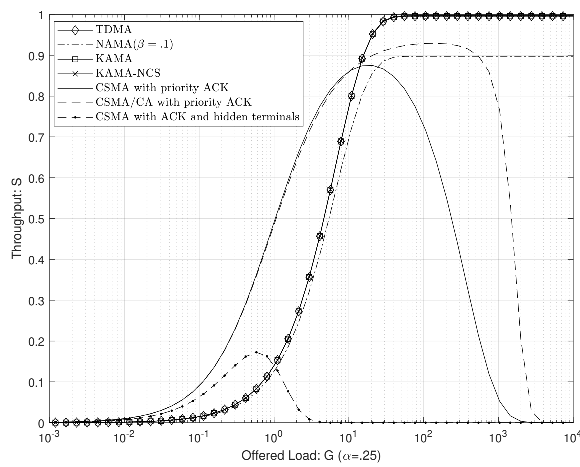


Fig. 4. Throughput results in steady state

Both KAMA and KAMA-NCS achieve optimal channel utilization in steady state because every time slot is used by a known node without MAI. However, at light loads, CSMA may perform better because a time slot is used in TDMA, NAMA or KAMA only if the winner for the time slot has data to send. In contrast, any node may access the channel successfully in a contention-based protocol. In a multi-hop network, the performance of CSMA and CSMA/CA rapidly degrade due to the hidden-terminal problem.

2) Network-Joining Delay Results: Figure 5 illustrates the average delays incurred by new nodes in joining the network for NAMA, KAMA-NCS, and KAMA with different values of P_a and ρ. To calculate the delays in NAMA, we assume that a portion β of the channel time is reserved for signaling. The probability that a new node succeeds transmitting in a mini-slot is the probability that no other arrivals transmit in the mini-slot. With the same assumptions introduced in Section 5.1.1, this probability is $e^{-\lambda_a T/5}$. Therefore, the delay in NAMA is

$$D_{NAMA} = \frac{\delta + \tau + \omega}{5\beta} e^{\lambda_a (T/5)} \tag{13}$$

For any value of ρ, the delays incurred with smaller values of P_a are higher because arriving nodes have priority in fewer time slots. In networks with very small arrival rates, a small ρ is preferable because arrivals will transmit more aggressively; however, under high load a large ρ is useful in reducing contention among priority arrival transmissions. The delay joining the network in NAMA is lower than in KAMA-NCS for $N = 20$ because the dedicated mini-slots provides new nodes more opportunities to transmit when they arrive. KAMA results in lower delays joining the network than in NAMA or KAMA-NCS at high loads because new nodes have more opportunities to join without MAI from known nodes. Given the simplicity KAMA, these results indicate that KAMA is a better alternative than NAMA.

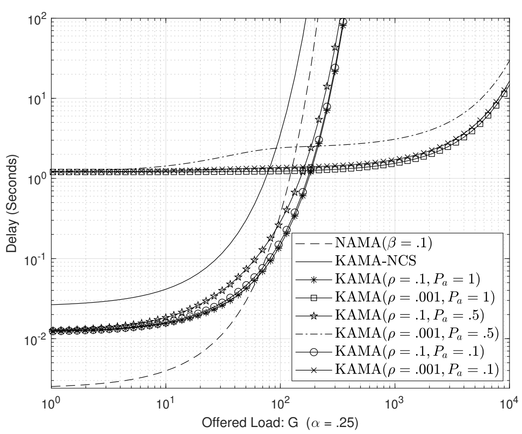


Fig. 5. Delay incurred by arriving node

B. Simulation Results

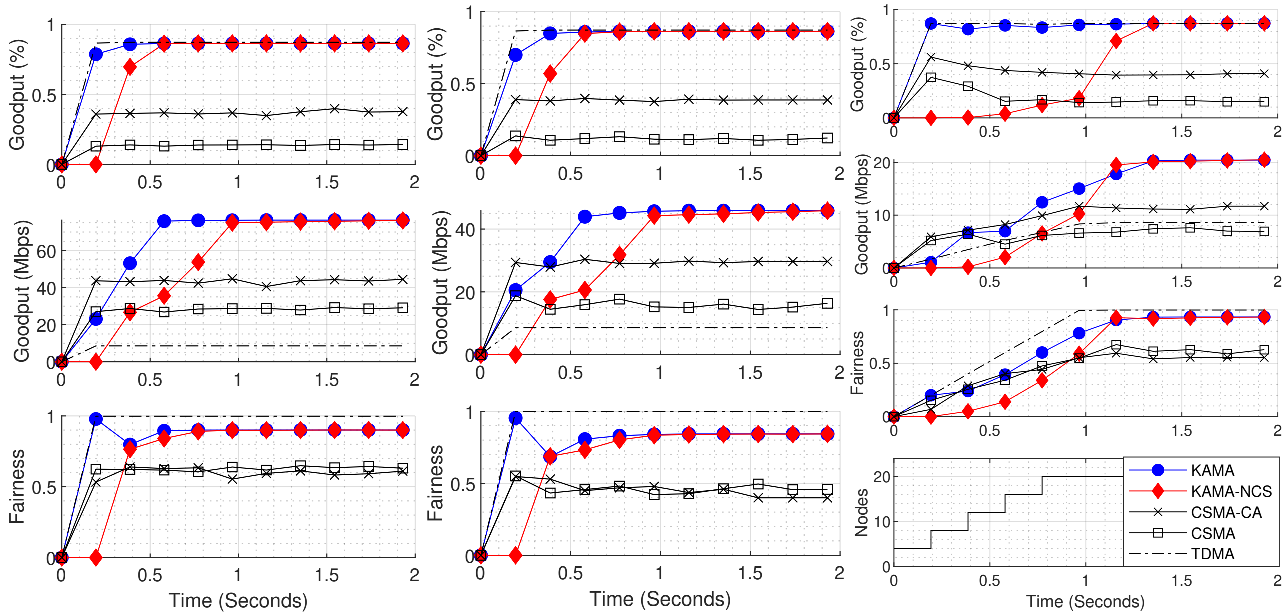
1) Simulation Setup: We implemented KAMA, CSMA with priority ACKs, CSMA/CA with priority ACKs, and fixed-share TDMA in the ns-3 network simulator, and compared its performance in multi-hop topologies. Given that the analytical results show that KAMA-NCS outperforms NAMA, which is much more complex, NAMA was not simulated. In each trial, nodes were assigned random 48-bit MAC addresses to scramble their key sets. Each time slot is sufficiently long to transmit 1500 bytes of payload data and signaling for up to 20 neighbors. Any unused portion of the signaling space is used to transmit additional payload data. All data is transmitted at 10 Mbps and all transmissions include a Physical Layer Convergence Procedure (PLCP) sublayer header of 24 bytes, which is transmitted at 1 Mbps. We assume that no channel capture or errors occur and the only form of interference is that due to multiple access interference at receivers. All experiments assume a fully saturated network such that every node always has data to transmit. KAMA uses a binary exponential back-off with a minimum exponent of 2 and maximum exponent of 5. In all experiments, KAMA uses a frame length of 128 slots and 4 key slots per frame. CSMA and CSMA/CA use a minimum backoff exponent of 4 and maximum exponent of 10. CSMA/CA implements a SIFS of $10\mu s$ and a DIFS of $50\mu s$, one SIFS + two $20\mu s$ backoff slots. RTS, CTS and ACK packets are all assumed to be 14 bytes.

Several metrics are used to evaluate the performance of the protocols: Goodput (%) is defined to be the ratio of payload bytes received by the network and the total number of bytes transmitted while Goodput (Mbps) refers to the raw number of payload bits received at the MAC layer normalized for time. To evaluate fairness, we use Jain's Fairness Index, which yields $\frac{1}{n}$ when a single node monopolizes the channel and 1 when each node uses $\frac{1}{n}$ of the channel bandwidth. When calculating fairness we only consider bytes which are successfully received. Each data point represents the time equivalent to a single KAMA transmission frame in all simulation results and each data point represents the mean of 10 trials.

2) Results: In the cold start experiments shown in Figs. 6a and 6b, a 10x10 grid of nodes are initialized at the same without any knowledge of each other. During the first frame, KAMA has a goodput of 40% due to collisions between priority transmissions. However, a node is acknowledged after a single priority transmission. Since none of the nodes have been acknowledged at the start of the experiment, they only transmit in the 4 slots for which they have keys, which results in poor channel utilization. All nodes have become known by all other nodes by the start of the second frame; however, the second frame is not fully utilized because nodes may not yet have confirmed they are known by their neighbors, which occurs after one full frame without receiving a NACK in the ACK vector bit corresponding to an arrival's priority transmission. By the start of the third frame, the channel is utilized successfully in every slot.

In KAMA-NCS, every node transmits in all the time slots of the first frame since, without any knowledge of other nodes, each node believes that it has the highest priority. Afterwards, all nodes randomly back-off from every time slot and contend only in time slots for which they elect themselves to be the winners. After about two seconds, nodes have correct neighborhood information and are out of back-off; this results in the channel being utilized optimally.

In the ramp-up experiment shown in Figures 6c, a four-node row is introduced to the network grid every 200 ms, until the population reaches 20 nodes. Nodes may still listen to channel while they are in stand-by; therefore, a node that is activated may immediately start transmitting. Given that the initial population of the network is four nodes and the KAMA key density is 4, KAMA only utilizes 16 of the 128 time slots in the first frame. Time slots may be unused if the arriving node is the winner, because there is a lapse of time between when a node is known by its neighbors and when it will start transmitting as a low-priority winner. This can be observed in the first second of the experiment, where the goodput is



(a) Cold start, Node degree 4 (b) Cold start, Node degree 8 (c) Ramp, Node degree 4

Fig. 6. Results from simulation experiments

optimal but the channel is not fully utilized.

Given that there are no priority transmissions in KAMA-NCS, an arrival will likely first collide with another transmission before becoming known, impacting the goodput.

Nodes transmit to each neighbor in a round-robin fashion, changing destinations each time an ACK is received. For KAMA and KAMA-NCS, the actual choice of neighbor is not important because KAMA requires that all neighbors acknowledge a transmission. KAMA and KAMA-NCS are able to quickly attain collision-free transmission schedules. When nodes have 8 neighbors and are acting with incomplete information, the goodput of KAMA-NCS is slightly lower due to an increased collision rate. In both cases; KAMA is able to converge on a schedule quicker than KAMA-NCS. Even though CSMA and CSMA/CA senders rotate their intended receivers, the end result is still unfair due to the use of binary exponential back-offs. As expected, TDMA is perfectly fair and collision-free, but has the lowest channel utilization because it has no spatial reuse of the channel.

VI. Conclusions

We introduced KAMA, the first protocol for collision-free channel access scheduling that does not require bandwidth to be dedicated to the exchange of signaling packets or the use of mini-slots. We showed through analytical modeling and simulations that KAMA quickly attains collision-free scheduling and outperforms NAMA, TDMA, CSMA, and CSMA/CA. Most importantly, we have shown that KAMA improves on prior work on distributed transmission scheduling by achieving near-optimal channel utilization in steady state.

Future work should focus on exploring novel ways of using the transmission keys introduced in KAMA. For example, keys could be used to determine when nodes are allowed to transmit data after they have joined the network. This approach would be a modification of NCR, and could be extended to multiple data channels. Keys could be used in the context of NCR to define on-off schedules for known nodes, so that nodes can preserve energy.

Acknowledgments

This material is based upon work sponsored by the National Science Foundation (NSF) under Grant CCF-1733884. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF or the U.S. government.

References

[1] A. Boukersche, et al., *Handbook of Algorithms for Wireless Networking and Mobile Computing*, CRC Press, 2005.

[2] I. Chlamtac and A. Farago, "Making transmission schedules immune to topology changes in multi-hop packet radio networks," *IEEE/ACM Trans. on Networking*, Feb. 1994.

[3] P. Feldman, "An Overview and Comparison of Demand Assignment Multiple Access (DAMA) Concepts for Satellite Communications Network

[4] L. Kleinrock and M. Scholl, "Packet Switching in Radio Channels: New Conflict-Free Multiple Access Schemes," *IEEE Trans. on Communications*, July 1980.

[5] J. J. Garcia-Luna-Aceves and D. Cirimelli-Low, "Queue-Sharing Multiple Access," *Proc. ACM MSWIM '20*, Nov. 2020.

[6] L. Bao and J.J. Garcia-Luna-Aceves, "A New Approach to Channel Access Scheduling for Ad Hoc Net-works," *Proc. ACM MobiCom '01*, July 2001.

[7] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient Collision-Free Medium Access Control for Wireless Sensor Networks," *Proc. ACM SenSys '03*, 2003.

[8] J.J. Garcia-Luna-Aceves, "Carrier-Sense Multiple Access with Collision Avoidance and Detection," *Proc. ACM MSWiM '17*, 2017.

[9] J. J. Garcia-Luna-Aceves, "Implementing Correct and Efficient Collision Avoidance in Multi-Hop Ad-Hoc Networks," *Proc. IEEE IPCCC '18*, 2018.