



# Simple and Efficient Collision-Free Channel Access in Multi-Hop Wireless Networks

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

Key-Activation Multiple Access (KAMA) is introduced. KAMA organizes the channel into a sequence of equal time slots, uses a distributed election algorithm to determine which of the known nodes have the priority to transmit during each time slot, and uses transmission keys to eliminate the need for special signaling packets or the use of special time slots dedicated for signaling packets. Simulation results in multi-hop networks shown that KAMA is more efficient than TDMA, CSMA, and CSMA/CA.

## CCS CONCEPTS

• **Networks** → **Network protocols**; **Link-layer protocols**; **Network performance modeling**.

## KEYWORDS

ad-hoc networks; channel access; CSMA; MAC protocols

### ACM Reference Format:

Dylan Cirimelli-Low and J.J. Garcia-Luna-Aceves. 2021. Simple and Efficient Collision-Free Channel Access in Multi-Hop Wireless Networks. In *Proceedings of the 24th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '21), November 22–26, 2021, Alicante, Spain*. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3479239.3485674>

## 1 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.

Contention-based channel-access 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, existing contention-free channel-access 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. Contention-free channel-access schemes organize the shared channel(s) into a sequence of periodic transmission opportunities in time. These schemes range from the simple time-division of the channel into a fixed number of time slots assigned statically to different nodes, which is highly inefficient in multi-hop networks or in the presence of on-off transmission patterns, to schemes that use complex structures that divide the channel using different types of transmission turns for data traffic than for signaling traffic [1–5]. The more complex schemes attain much better channel utilization in multi-hop networks.

Section 2 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 3 compares the performance of KAMA, CSMA, CSMA/CA, and TDMA based on simulations. 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 4 presents our conclusions.

## 2 KEY-ACTIVATION MULTIPLE ACCESS

### 2.1 Design Motivation

Bao and Garcia-Luna-Aceves [1, 2] 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



This work is licensed under a Creative Commons Attribution International 4.0 License.

MSWiM '21, November 22–26, 2021, Alicante, Spain.

© 2021 Copyright is held by the owner/author(s).

ACM ISBN 978-1-4503-9077-4/21/11.

<https://doi.org/10.1145/3479239.3485674>

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

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:

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

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

The importance of NCR is that it has been proven [1] 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 [4]. 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.

## 2.2 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 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.

## 2.3 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  $\text{ACK}_t = 1$ , 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.

## 2.4 KAMA Transmission Strategy

If a node has not been acknowledged by its neighbors using the mechanism described in Section 2.3, it may access the channel at the start of a time slot only if it 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  $\tau$ , 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  $d \ll l$ , nodes are able to join the network quickly in KAMA. The reason for this is that 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  $\tau$  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.

## 3 PERFORMANCE COMPARISON

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 fully-connected and multi-hop topologies.

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. KAMA uses a binary exponential back-off with a minimum exponent of 2 and maximum exponent of 5, which results in a maximum possible back off of 32 frames for a given time slot. 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 Mbps refers to the raw number of payload bytes 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.

### 3.1 Fully-Connected Experiments

The fully-connected experiments use a population of 20 half-duplex transceivers and assume that nodes always have data available to transmit. In each trial, nodes are placed randomly in a circular area with a radius of 300 m, which yields a maximum propagation delay of 1  $\mu$ s. In the multi-hop experiments, nodes were placed uniformly in a 10x10 grid. In each trial, nodes were assigned random 48-bit MAC addresses to scramble their key sets.

In the cold start experiment shown in Figure 1a, 20 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, only one of a node's priority transmissions must be successful for the node to be acknowledged. 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 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.

CSMA performs better than CSMA/CA due to the signaling overhead introduced by DCF. Although the RTS and CTS packets are only 14 bytes, they must carry physical layer headers of 240 bytes, normalized for the data rate. Due to the intrinsic unfairness of the binary exponentially back-off, a single node is able to monopolize the channel, which results in very poor fairness. As expected, TDMA is perfectly fair and uses the channel optimally for the entire experiment. Once every KAMA node is known, nodes stop transmitting neighborhood updates and KAMA performs nearly identically to TDMA.

In the two-node ramp experiment, two nodes are introduced to the network 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 two nodes and the KAMA key density is 4, KAMA only utilizes 8 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 two seconds of the experiment, where the goodput is optimal but the channel is not fully utilized.

Figure 1c illustrates that nodes arriving in larger bulks is comparatively beneficial for KAMA. Despite the increased concentration of node arrivals, the goodput of KAMA remains nearly unchanged while the channel utilization increases at a quicker rate. This illustrates the benefits derived from the priorities enabled by carrier sensing in KAMA.

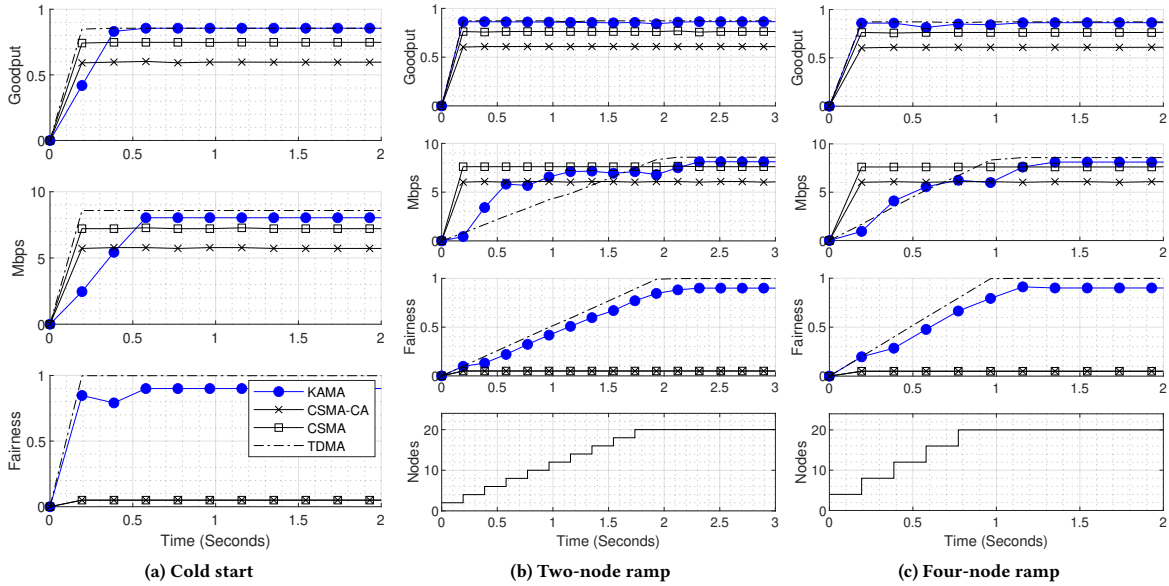


Figure 1: Fully-connected experiments

### 3.2 Multi-Hop Experiments

In both multi-hop experiments, KAMA senders choose random neighbors as receivers. The actual choice of neighbor is not important because KAMA requires that all neighbors acknowledge a transmission. CSMA and CSMA/CA nodes transmit to each neighbor in a round-robin fashion, changing destinations each time an ACK is received. KAMA is able to quickly attain collision-free transmission schedules. In this multi-hop setting, CSMA/CA outperforms CSMA; however, the goodputs of CSMA and CSMA/CA are much lower than in the fully-connected case due to MAI from hidden terminals. 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. TDMA is collision-free, but has the lowest channel utilization because it has no spatial reuse of the channel.

## 4 CONCLUSIONS

We introduced KAMA, a simple and efficient approach to collision-free channel access scheduling which, unlike all prior collision-free schemes based on distributed elections or reservations, does not require bandwidth to be dedicated to the exchange of signaling packets or the use of mini-slots. We have shown through analytical modeling and simulation that KAMA can quickly attain collision-free scheduling and is robust in practical network conditions. KAMA was shown to outperform NAMA, TDMA, and CSMA/CA. Most importantly, we have shown that KAMA improves on existing work by achieving near-optimal channel utilization in steady state.

## 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.

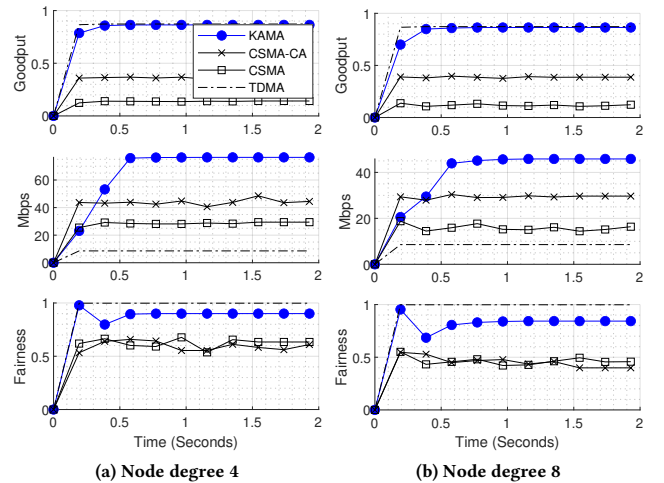


Figure 2: Multi-hop experiments

## REFERENCES

- [1] L. Bao and J.J. Garcia-Luna-Aceves, "A New Approach to Channel Access Scheduling for Ad Hoc Networks," *Proc. ACM MobiCom '01*, July 2001.
- [2] L. Bao and J.J. Garcia-Luna-Aceves, "Hybrid Channel Access Scheduling in Ad Hoc Networks," *Proc. IEEE ICNP '02*, Nov. 2002.
- [3] A. Boukersche, et al., *Handbook of Algorithms for Wireless Networking and Mobile Computing*, CRC Press, 2005.
- [4] 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.
- [5] Z. Tang and J.J. Garcia-Luna-Aceves, "A Protocol for Topology-Dependent Transmission Scheduling," *Proc. IEEE WCNC '99*, Sept. 1999.