UC Santa Cruz

UC Santa Cruz Previously Published Works

Title

Using Radio Connectivity To Define Transmission Schedules in Multihop Wireless Networks

Permalink

<https://escholarship.org/uc/item/9c97t23x>

Authors

Garcia-Luna-Aceves, J.J.
Masilamani, A.

Publication Date

2014-10-01

Peer reviewed

Using Radio Connectivity To Define Transmission Schedules in Multihop Wireless Networks

J.J. Garcia-Luna-Aceves and Ashok N. Masilamani

Department of Computer Engineering

University of California, Santa Cruz

Santa Cruz, CA 95064

Email: {jj, ashok}@soe.ucsc.edu

Abstract—Spatial Classification Multiple Access (SCMA) is introduced as an example of using the radio connectivity among nodes for the dynamic establishment of distributed transmission schedules in wireless multi hop wireless networks. The shared channel is organized into transmission frames whose length in number of time slots is defined solely by the need to avoid hidden-terminal interference, rather than some arbitrary number of time slots related to network size. SCMA is shown to attain feasible transmission schedules within a finite time; and is compared with representative examples of traditional approaches to medium access control (MAC) based on contention, transmission scheduling, and reservations. The results of the analysis show that SCMA attains higher packet delivery ratio, lower average end-to-end delays, and better useful throughput than traditional MAC protocols.

I. INTRODUCTION

The aim of all medium access control (MAC) protocols is to improve the throughput and transmission delays experienced by nodes in the presence of multiple access interference (MAI) resulting from other transmissions within the interference range of intended receivers. Interestingly, as Section II summarizes, all prior MAC protocols operate based on contention, scheduling, or reservation schemes that establish transmission times by reacting to the radio connectivity among nodes in the network. This independence between the selection of transmission times by the MAC protocol and the radio connectivity among nodes results in: (a) the inability of contention-based MAC protocols to cope adequately with MAI at high load; (b) the need for reservation and scheduling MAC protocols to rely on probabilistic methods to reserve or elect time slots; and (c) the need to organize the channel into transmission frames whose length in number of time slots has little to do with MAI, and must be large to enable nodes to win or reserve time slots using probabilistic methods.

This paper introduces a new approach, which we call SCMA (Spatial Classification Multiple Access), for the sharing of multiple access channels in wireless networks with or without hidden terminals. SCMA is described in Section III and its operation relies on three main components: (a) assigning connectivity labels to nodes using a distributed algorithm that classifies nodes into node pools, such that nodes in a given pool cannot cause MAI for nodes in different pools; (b) allocating each node pool to a different time slot in the

transmission frame used to organize the shared channel; and (c) using a deterministic algorithm to classify the nodes in the same node pool to establish a collision-free transmission schedule within each node pool. SCMA is the first approach for medium access control that uses a distributed algorithm to classify nodes based on their connectivity in order to derive collision-free transmission schedules dynamically.

Section IV proves that the spatial classification of nodes in SCMA enables the organization of the channel into fixed-length transmission frames whose number of time slots is defined solely by the need to avoid MAI. This number is independent of the network size and density, and renders small transmission frames that lead to smaller end-to-end delays.

Section V analyzes the performance of SCMA in terms of packet-delivery ratio, goodput, end-to-end delays, and network-joining times. The results of the simulation experiments show that SCMA is a far more efficient alternative to channel access than contention, reservations and topology-dependent transmission scheduling for all performance metrics. Furthermore, the connectivity labels used in SCMA enable the use of a deterministic approach for nodes to join the network and handle connectivity changes due to mobility (i.e., acquire and fix their transmission schedules). This allows very short network-joining times (e.g., shorter than 3 seconds for networks with 500 nodes) while allowing the network to use 90% of the bandwidth for scheduling and transmitting data packets.

II. RELATED WORK

Since the introduction of ALOHA and CSMA, many MAC protocols have been proposed to control access to a common wireless channel using contention schemes that attempt to eliminate collisions due to MAI (e.g., see [20]). While these protocols succeed to some extent, their performance degrades at high loads, because they are unaware of which nodes are attempting to transmit and must simply react to the effects of MAI perceived in the channel. A number of location-based MAC protocols have been proposed (e.g., [13], [19]) that increase channel reuse by assigning different channels to different geographical areas, and allowing nodes to know their own location. However, these protocols do not establish transmission schedules.

All the MAC protocols based on transmission-scheduling or reservation schemes designed to date assume that the channel is divided into transmission frames consisting of a number of time slots being somehow related to the number of nodes in the network or the density of the network [3].

Topology-independent transmission-scheduling protocols assign a unique code to each node, which defines the times when the node is allowed to transmit in a frame and ensures that at least one time slot exists during which no interfering node can also transmit (e.g., [7], [15]). Unfortunately, this independence from network topology comes at a very high performance cost; Kunz and Rentel [16] have shown that this approach has similar performance to that of slotted ALOHA.

MAC protocols based on reservations (e.g., HRMA [18], FPRP [22]) organize the channel into frames consisting of a fixed number of time slots, and each time slot is divided into several mini-slots dedicated for the contention and reservation of the time slots as well as the transmission of data in the time slot. By necessity, the number of time slots per transmission frame must be large.

Topology-dependent transmission scheduling protocols establish transmission schedules taking into account the MAI caused by neighboring nodes and in some cases the traffic at each node. The assignment of time slots is based either on the election of entities competing for the data time slots (nodes or links), or the selection of reservation requests for data time slots according to a set of predefined rules. Some schemes require an initial topology-independent schedule, followed by some negotiation among network nodes used to obtain a final schedule (e.g., [5], [10], [21]).

Many topology-dependent scheduling schemes are based on distributed election algorithms [1], [2]. To elect transmission schedules, each node knows the identities of all other nodes one and two hops away from itself, and the present time in the network. Depending on the protocol, nodes use a contention-based approach like slotted ALOHA during a control section defined for either the entire frame or each time slot to communicate to their neighbors either the identifiers of their own neighbors and themselves, or the identifiers of the links to their own neighbors. Each node builds and maintains a list of contending entities (nodes or links) and uses this list to determine which node should be given access to the channel during each time slot of the data section of the frame. To accomplish this task, the node applies a permutation function on the list of contending entities to select a winning node from the list of nodes for each time slot of the transmission frame. Some protocols also allow for nodes that win the election of a time slot to reserve the time slot. The main limitation of these MAC protocols is that the time taken for all nodes to access the channel at least once or the jitter of consecutive channel accesses by the same node may become very large as the number of nodes in the two-hop neighborhood increases.

Considerable work has been reported on the establishment of efficient transmission schedules in a distributed manner taking into account the nodal traffic demands and attempting to limit the overhead incurred in the establishment of

schedules that approach the optimum [4], [14]. However, these approaches are not practical because of the signaling overhead they incur in updating the state of traffic demands.

Our summary indicates that prior MAC approaches attempt to assign channel access times to nodes based on either fixed assignments, contention, reservations, transmission scheduling, or geographical locations. None of these prior schemes take advantage of the inherent ordering among nodes derived by their network connectivity. Because prior MAC protocols do not use information related to the spatial ordering among nodes to grant access to the shared channel, probabilistic schemes or brute-force contention are needed to establish transmission schedules, which leads to the coupon collector's problem [11]. Just as important, the lack of spatial ordering information requires schedule-based schemes to define the size of the transmission frames they use according to the network size or density of the network. This is a problem, because transmission frames that are too small are unable to accommodate all nodes requiring time slots, and transmission frames that are too large can induce large transmission delays. Furthermore, a large number of time slots dedicated to signaling provides faster convergence in the establishment of schedules, but at the expense of longer delays experienced by data packets using time slots dedicated to data transmission. Conversely, a small number of time slots dedicated for signaling exacerbates the coupon collector's problem.

III. SPATIAL CLASSIFICATION MULTIPLE ACCESS (SCMA)

The essence of SCMA is the classification of nodes based on their network connectivity to determine the transmission times assigned to nodes in a way that avoids MAI. Collision-avoidance MAC protocols address MAI reactively on a packet-by-packet basis using RTS-CTS handshakes (request to send, clear to send). By contrast, SCMA handles MAI proactively by having nodes share a *Coordinated Transmission Schedule* (CTS) defined by *connectivity labels* assigned to nodes that reflect their radio connectivity. The components of SCMA are: (a) a distributed algorithm that assigns each node to a pool of nodes based on its distance in hops to a *network beacon* node elected for the entire network and its distance in hops to another *local beacon* node with the same distance to the network beacon; (b) a transmission frame with a time slot for each node pool that may cause MAI with other pools; (c) a deterministic algorithm that classifies the nodes in the same node pool to establish a collision-free transmission schedule within each node pool; and (d) signaling based on Hellos to convey CTS information.

We make three assumptions for the operation of SCMA, which are the same as those used in most prior MAC protocols based on transmission scheduling. The radios used in the network are half-duplex and can tune to only one channel at a time. Each node in the network is assigned a unique node identifier and radio links are bidirectional. Time is slotted with time slots having a fixed duration, and any pair of nodes can be synchronized at the time-slot level. The time slotting needed

in SCMA can be attained in practice using such distributed clock synchronization schemes as those demonstrated in the past (e.g., [17], [9]) or by taking advantage of GPS at each node.

The neighborhood of a node consists of those nodes whose transmissions the node can decode, which we call one-hop neighbors, and the one-hop neighbors of those nodes, which we call two-hop neighbors. However, SCMA does not assume that MAI occurs only among one and two-hop neighbors. The minimum number of hops h needed for two nodes to transmit concurrently without causing MAI is taken into account in the transmission algorithm (see Section III-E).

A. Labels and Channel Structure

The channel is organized in transmission frames consisting of a constant number of time slots. Each time slot in SCMA is used for the transmission of a *Hello* and zero or more data packets. Each *Hello* contains an entry for each one-hop neighbor of the transmitting node and the node itself. Each entry consists of the node identifier of the node and its *connectivity labels*, which describe the inherent spatial ordering among nodes resulting from their radio connectivity, and are used to harmonize the spatial ordering of nodes with their scheduled transmission times.

Fixed TDMA can be viewed as one extreme of the harmonization of spatial connectivity with transmission times in which a pool consisting of a single node is assigned to each time slot by means of a one-to-one mapping between node identifiers and time slots. Clearly, this approach attains the fastest assignment of a time slot to a label, because it is done in zero time; however, the utilization of the channel can be very poor, unless the network is fully connected. At the other extreme, we can view such reservation and election schemes as FPRP and NAMA as the assignment of all node identifiers to each time slot of a pre-defined transmission frame. Ideally, the utilization of the channel can be very high, but unfortunately the MAC protocol is forced to spend considerable time in selecting a specific label (i.e., a node identifier) as the winner from the large pool of nodes assigned to each time slot.

It is clear that a trade-off exists between the efficiency of channel utilization and the time required to harmonize spatial ordering with transmission times. Furthermore, the size of the pool of nodes competing for each time slot of the transmission frame plays a key role on the speed with which nodes can attain transmission schedules. SCMA attempts to establish a sensible trade-off between efficiency and speed of the channel scheduling task by first classifying nodes into pools allocated to different time slots, and then establishing a deterministic selection algorithm to make nodes take turns accessing the time slot associated with the pool to which they belong.

SCMA assigns to each node a connectivity label consisting of two parts, as shown in Fig. 1. The first part of the label denotes the *pool* of nodes to which a node belongs, and consists of a vector of l sub-labels, where $1 \leq l \leq N$ and N is the number of nodes in the network. The second part of the label denotes the *turn* assigned to a node among the nodes

Fig. 1. Connectivity labels in SCMA

that belong to its same pool, and consists of a *slot-number* and a *rank*.

Section IV shows that, if h is the minimum number of hops needed for two nodes to transmit concurrently without causing MAI and l is the number of labels used per node, then a transmission frame of $m \geq h^l$ time slots suffices to ensure that no two transmissions from nodes from different pools of nodes assigned the same time to transmit can collide with each other. The minimum value of h is three in order to avoid the hidden terminal problem. We describe SCMA assuming that network nodes are classified into h^2 different *pools*, with each pool being assigned a separate time slot in the transmission frame, and nodes in the same pool are one- or two-hops away from each other.

We focus on an SCMA implementation based on a vector of two sub-labels that classify nodes into pools of nodes (see Fig. 1(b)), and denote the labels by *LabelA* and *LabelB*. Pools are assigned time slots in a way that nodes in two different pools and assigned the same time slot number cannot interfere with one another. The turn component of their labels determines how nodes in the same pool take turns with one another to avoid MAI in the same pool. The turn of a node is denoted in its connectivity label by stating the turn given to the node within the pool, and the length of the turn-taking schedule built for the time slot assigned to the pool to which the node belongs. The following two sections describe how the pool and the turn components of connectivity labels are computed dynamically.

B. Information Stored and Exchanged

The information maintained and exchanged at each node allows the node to establish its own connectivity label and establish collision-free transmission schedules dynamically. Each node transmits a *Hello* message to its immediate neighbors every *HelloInterval* seconds, and updates its connectivity label whenever it receives *Hello*s from its one-hop neighbors or after failing to receive *Hello*s from a given neighbor for a multiple of the *HelloInterval*.

A *Hello* from node i consists of a list of tuples for node i itself and each of its known one-hop neighbors. The tuple regarding node j in the *Hello* from i is $\{nid_j^i, RA_j^i, LA_j^i, RB_j^i, LB_j^i, SN_j^i, R_j^i\}$, where the entries in the tuple state

Fig. 2. Connectivity labels and corresponding pools for example network

the node identifier, Root-A, Label-A, Root-B, Label-B, slot number, and rank for node j . Node i stores the tuples reported in the most recent Hello from each neighbor. Label-A consists of the distance from node j to a *network beacon* node called Root-A ($d_j^i(RA)$) and the sequence number originated by the network beacon ($s_j^i(RA)$). Label-B consists of the distance from node j to a *local beacon* node called Root-B ($d_j^i(RB)$) and the sequence number originated by the local beacon ($s_j^i(RB)$). The local beacon for node j has the same distance to the network beacon as node j .

The tuple stored at node i with the information sent by neighbor k regarding node j is denoted by $\{nid_{kj}^i, RA_{kj}^i, LA_{kj}^i, RB_{kj}^i, LB_{kj}^i, SN_{kj}^i, R_{kj}^i\}$. The connectivity label computed by node i is denoted by $\{RA^i, LA^i, RB^i, LB^i, SN^i, R^i\}$, with $LA^i = [d^i(RA), s^i(RA)]$ and $LB^i = [d^i(RB), s^i(RB)]$.

Each node also maintains a *Coordinated Transmission Schedule* (CTS) that defines the transmission schedule assumed by the node. The CTS maintained by node k in pool n is denoted by CTS_{kn} , and consists of a row for each node in its one-hop neighborhood, including those that are not in the same pool as the node, and a column for each time slot required in the schedule. Fig. 1(c) shows the CTS maintained at nodes d , e , and g of the example network.

C. Pool Division of Nodes

A node creates the pool-division portion of its connectivity label based on minimum-hop distances to a network beacon (called *Root-A*) and a local beacon (called *Root-B*). The classification of nodes into pools is based on radio connectivity, and is accomplished by establishing a network-wide directed acyclic graph (DAG) rooted at the elected network beacon, and multiple DAGs among nodes with the same distance to the network beacon and rooted at local beacons.

The signaling in SCMA takes place over multiple hops. A single node is elected as the network beacon for the entire network using a distributed election based on *Hellos*. The network beacon is called *Root-A* and node i selects its *Root-A* and its *Label-A* based on the identifier of the proposed *Root-A*, the root sequence number updated only by the selected *Root-A*, and the distance to *Root-A*. When a node is initialized, it

sets *Root-A* and *Root-B* to equal its own node identifier and assigns zero to *Label-A* and *Label-B*.

Nodes elect the node with the smallest identifier as *Root-A* based on the *Hellos* they exchange with one another. The node elected as *Root-A* increases its network-beacon sequence number with every *Hello* it sends, and no other node can change the sequence number for *Root-A*. The *Label-A* of node i is a tuple consisting of the most recent sequence number available from *Root-A* and the smallest distance attainable through any neighbor that has reported the most recent network-beacon sequence number in a *Hello*. Node i uses a similar scheme to compute its *Label-B*, which consists of the minimum-hop distance to the local beacon node with the smallest node identifier that has the same *Root-A* and *Label-A* values as node i itself. Clearly, this is akin to maintaining loop-free minimum-hop routes to *Root-A* and *Root-B*, and Section IV shows that the use of root sequence numbers in connectivity labels ensures that the ordering they induce on nodes is correct.

Figure 2(a) shows the pool-division part of the connectivity labels of nodes in an example multi-hop network. Figure 2(b) shows the time slots assigned to pools, the labels associated with the pools, and the nodes associated with each pool for $h=3$. In the example, node a is elected as the network beacon (*Root-A*), and there is at least one *Root-B* elected for each minimum-hop distance value to *Root-A*. For example, nodes w, x, y, i, j and k have the same distance to node a ; and node w is elected as the local beacon (*Root-B*) for nodes x, y, i, j and k . Similarly, nodes m, l, n, o , and u are all three hops away from node a ; and node l is the *Root-B* for n, o , and u .

D. Turn Taking within Pools

Once a node computes the pool to which it belongs, the node schedules a time slot belonging to that particular pool to use for its own transmissions. This is done by constructing a *Coordinated Transmission Schedule* (CTS) at each node, which defines the time slots assigned to the nodes in the neighborhood of the node. It is computed from the *Hellos* received from one-hop neighbors. Algorithm 1 is used to build and maintain the CTS at node k . Unlike scheduling protocols based on long fixed-frame sizes, the frame size in SCMA is defined to be only long enough to accommodate MAI.

Hellos are used to set the *slotnumber* and the *rank* fields in the label of the node. The *rank* value determines the rate at which the schedule repeats for that node in its *CTS* and is equal to the smallest power of 2 that is greater than or equal to the maximum occupied slot number in the *CTS* of the node. The objective of using powers of two is to ensure collision-free channel access for nodes, as shown in Section IV, and to make the number of time slots of the *CTS* of the node proportional to the number of nodes in the two-hop neighborhood that belong to the same family. For example, let two nodes be in the same two-hop neighborhood, with one node having two nodes and the other node having three nodes in the two-hop neighborhood belonging to the same pool. In such a case, the node with three neighbors will have a *CTS* frame of length four and the node with one neighbor will have *CTS* frame of length two, thereby allowing for the cyclic repetition of the schedules of both the nodes.

Consider again the network of Fig. 2(a). Nodes that belong to Pool 5 transmit in the time slots that have the slot-number value of $5 + z \cdot h^2$, where z is a natural number. Nodes *c*, *d* and *e* use the turn-taking part of their connectivity labels to determine which of the three nodes transmit in each of these slots, i.e., the value of z for the particular node that is determined by the column value in the *CTS* table. Ideally, the nodes would take turns fairly to transmit in the time slots, which is what the *slot-number* and *rank* fields of the connectivity label attempt to accomplish. It can be seen from the *CTS*s shown in Fig. 2(c) that, while nodes *d* and *e* accesses the channel every 4 slots, node *g* accesses the channel every two slots. The rate at which a node accesses the channel depends on the number of nodes that belong to the same pool in its neighborhood.

Algorithm 1 Computing the *CTS* and slot-number

```

1:  $CTS_k[l, t]$  = CTS of node  $k$  where row ' $l$ ' = one-hop node; column ' $t$ ' = slot position.
2:  $max\_slot$  = maximum slot number occupied by a two-hop neighbor node of  $k$ .
3:  $OHN_k$  = one-hop neighborhood of  $k$ .
4: for all  $l$  in  $OHN_k$  do
5:   while  $t \leq max\_slot$  do
6:      $CTS_k[k, t] = CTS_k[k, t] \oplus CTS_k[l, t]$ 
7:   end while
8: end for
9: Let  $t_\phi$  be the first empty slot position and  $t_k$  be the slot position occupied by node  $k$  in  $CTS_k[k]$ .
10: if  $k \in CTS_k[k]$  OR  $t_\phi < t_k$  then
11:    $t_k = t_\phi$ 
12: end if
```

E. Transmission Algorithm

Algorithm 2 is used at each node to decide whether it should transmit on a given time slot. $(Time.slot.number \% h^2) + 1$ is calculated at the start of each slot to determine the *pool* to which the particular time slot belongs. Let OHN_k be the one-hop neighborhood of node k . $CTS_k[n, t]$ is the node identifier in the *CTS* of node k in the row corresponding to the one-hop neighbor n and in slot-number t . For each time slot, the node calculates $t_n = SlotID \% rank$ value, where *SlotID* is the slot number for the pool to which each slot belongs in the

network. For example, for $h = 3$, *SlotID* for slots 2, 11, 20... that belong to the same pool are 1, 2, 3..., respectively. It goes to *Transmit* state only if $CTS_k[n, t_n]$ is k and there is no other one-hop neighbor with a node ID other than k at t_n in its *CTS*.

Algorithm 2 Transmission Algorithm

```

1: Let  $thn_i$  denote two-hop-neighborhood of node  $i$ .
2: Let  $E$  denote the rate of empty slots for the network ( $E \neq h^2$ ).
3: if  $Time.slot.num \% E == 0$  then
4:    $State_k \leftarrow Receive$ .
5:   return
6: end if
7: Let  $SlotID = (Time.slot.num - Time.slot.num \% h^2) / h^2 + 1$ 
8: Let  $t_n = SlotID \% rank(n)$ 
9: Pool  $p = Time.slot.num \% h^2 + 1$ .
10: Calculate  $LabelA_i \% h$  and  $LabelB_i \% h$ .
11: if Node  $i$  belongs to pool  $p$  then
12:   if  $CTS_k[k, t_n] == k$  then
13:     for all  $n$  in  $OHN_k$  do
14:       if  $CTS_k[n, t_n] \neq k$  OR  $CTS_k[n, t_n] \neq \Phi$  then
15:          $State_k \leftarrow Receive$ .
16:       return
17:     end for
18:   end if
19:    $State_k \leftarrow Transmit$ .
20: else
21:    $State_k \leftarrow Receive$ .
22: end if
23: else
24:    $State = Receive$ 
25: end if
```

F. Network Joining and Mobility Handling

The method used to handle mobility and joining of new nodes is similar in principle to the topology-transparent scheduling algorithm based on the evaluation of polynomials over a Galois field as first proposed by Chlamtac and Farago [6].

We define the *empty-slot set* as the set of predefined time slots that are made available for new nodes in the neighborhood and for nodes that choose a new schedule to transmit their labels. The average rate of empty slots E determines the time taken by nodes to converge to a schedule when it chooses a new schedule. The aggregate of empty-slot sets forms the *empty-slot metaframe* F_e . A node that must select a new transmission slot for its data transmissions or that joins the network selects a subset of time slots in the *empty-slot set* to transmit its *Hellos* in these slots. Let $GF(l)$ be a Galois field of order l , where $l = s^m$, s is a prime and $m \geq 1 \in \mathbb{Z}^+$ is an arbitrary positive integer. Every element in $GF(l)$ is labeled with the integers $0, 1, \dots, l - 1$. Every node in the network is assigned vector-identifier polynomials $VID_v[x]$ of degree k with coefficients in $GF(l)$ such that the polynomial assigned is unique for the node.

Let the empty-slot metaframe size be $l' = l^2$ slots. The set of time slots $S_{ev} \in F_e$ constitutes the time slots during which node v transmits *Hellos* before it is allowed to execute the transmission algorithm and start transmitting data packets.

In Algorithm 3, each polynomial is evaluated for every value between 0 and $l - 1$ and assigned to set S_{ev} at each node v . The frame of size l^2 can be visualized as composed of l subframes with l time slots each and there is one *Hello*

Fig. 3. Frame structure

Algorithm 3 New Node Algorithm

```

1:  $S_{ev}(0 : q - 1) = 0$ ; //Initializing  $q$  slots
2: for ( $i = 0 : q - 1$ ) do
3:    $S_{ev}(i) = iq + VID_v[i]$ 
4: end for

```

transmission slot within each of the subframes. To guarantee that there is at least one slot in S_{ev} in the metaframe of l^2 slots that is uniquely assigned to node v , two constraints must be satisfied: (a) $l^{k+1} \geq N$; and (b) $l \geq kD_{max} + 1$, where D_{max} is the maximum degree of a node, and N is the number of node in the network. The first constraint makes sure that every node in the network has a unique code. The second constraint guarantees that there is at least one transmission slot within F_e that is not shared by any two nodes (see [6] for the proof), thereby allowing nodes to transmit their *Hello*s without collision at least once in the metaframe.

Let $l_1, l_2, \dots$ be the sequence of increasing powers of primes (i.e, 2, 3, 4, 5, 7...), then Algorithm 4 is used to select the l value for a network.

Algorithm 4 Algorithm to set values of l

```

1:  $i=0$ ;  $k=0$ ;
2: while  $k < 1$  and  $l^{k+1} < N$  do
3:    $i = i + 1$ ;  $l = l_i$ ;
4:    $k = \lfloor \frac{(l-1)}{D_{max}} \rfloor$ 
5: end while

```

Figure 3 shows the empty slot super frame of size $l^2 = 25$ slots. When two nodes a and b join the neighborhood, they transmit their *Hello*s in one slot every five slots of the frame. It is guaranteed that there is at least one empty slot uniquely assigned to each node during which a and b transmit their *Hello*s without collision.

IV. CORRECTNESS OF SCMA

The following theorems prove that, once the nodes of a network have been spatially classified and assigned their corresponding connectivity labels, SCMA attains collision-free transmission schedules using a transmission frame with a length in number of time slots defined solely by the need to avoid MAI. The number of time slots per transmission frame in the network is denoted by n , the number of labels identifying the pools of nodes is denoted by l , and the minimum number of hops at which two nodes can transmit concurrently without causing interference with one another is denoted by h .

Theorem 4.1: SCMA attains correct ordering of nodes at every instant based on their *Label-A* and *Label-B* sub-labels.

Proof: The proof is by contradiction by assuming incorrect ordering of nodes with respect to any beacon node, which occurs only when routing loops to the beacon node are formed when nodes update their connectivity labels. The proof is the same for *Label-A* and *Label-B* sub-labels, and without loss of generality we focus on *Label-A* sub-labels. Assume that a routing loop of h hops L_c for *Root-A* is created when nodes in L_c update *Label-A* of their connectivity labels. Denote the *Root-A* node by c .

Let $h^i(c)$ be a neighbor of node i that can serve as next hop towards c because it sent a *Hello* reporting the most recent root sequence number from c and the shortest distance to c . Let $L_c = (n_1, n_2, \dots, n_h)$, with $n_i + 1 = h^{n_i}(c)$ for $1 \leq i \leq h-1$ and $n_1 = h^{n_h}(c)$.

According to SCMA, for each hop $n_i \in L_c (1 \leq i \leq h)$, a node selects a next hop that has reported the most recent root sequence number from c . Therefore, $s^{n_i}(c) \leq s^{n_{i-1}}(c) + 1$ for $1 \leq i \leq h-1$ and $s^{n_h}(c) \leq s^{n_1}(c)$. This implies that $s^{n_i}(c) = s^{n_{i+1}}(c)$ for $1 \leq i \leq h-1$ and $s^{n_h}(c) = s^{n_1}(c)$. Accordingly, $d^{n_i}(c) < d^{n_{i+1}}(c)$ for $1 \leq i \leq h-1$ and $d^{n_h}(c) < d^{n_1}(c)$, which means that $d^{n_i}(c) < d^{n_i}(c)$ for $1 \leq i \leq h$. This is a contradiction and hence the theorem is true. ■

Theorem 4.2: No packet transmitted by a node in a given pool as computed in SCMA can collide with any other transmissions from nodes assigned to different pools, provided that $n \geq h^l$.

Proof: Without loss of generality, assume that a specific node from each pool of nodes computed by SCMA is selected as the only node allowed to transmit from its own pool. The proof is by induction on the number of labels l .

Basis case: Assume that $l = 1$ and let the label be *Label*₁. According to SCMA, when a given network node a transmits a packet, the next node that can transmit a packet in the same time slot as node a is h hops away. The total number of nodes between the two transmitting nodes is $h-1$. Therefore, including node a , we need h distinct channels (slots) for each node in the domain where channel reuse is not possible. It follows that the minimum number of slots per frame is h^1 .

Inductive step: Assume that the result is true for $1 \leq l \leq k$ and let the labels used for each node be *Label*₁, *Label*₂, ..., *Label* _{k} . The number of slots required per frame with k labels is $n = h^k$. Let $l = k+1$ and let the labels used by nodes be *Label*₁, *Label*₂, ..., *Label* _{k} , *Label* _{$k+1$} . The

new total number of nodes within h hops is given by

$$n = h^k \times \sum_{Label_{k+1}=0}^{h-1} 1 = h^k \times h = h^{k+1}$$

Therefore, $n = h^l$. ■

Theorem 4.3: SCMA ensures that no packet transmitted by any node collides with any other transmissions, provided that $n \geq h^l$.

Proof: It follows from Theorem 4.2 that collisions can occur only due to concurrent transmissions from nodes in the same pool. Assume that two nodes a and b incur a collision. Nodes that belong to the same pool, and are within two hops of each other, choose *slot-number* and *rank* values to set the turn-taking part of the connectivity label.

Let a_1 be the time-slot number that is chosen by node a for transmission. Let r_a be the rank of node a . Node a transmits every r_a slot after a_1 . Similarly, let b_1 and r_b be the slot number and rank of node b . Without loss of generality, assume that $b_1 > a_1$. By definition, the rank of any node must be greater than or equal to the slot numbers chosen by the nodes in its two-hop neighborhood. Hence, it must be true that

$$1 \leq b_1 \leq r_a, \text{ and } 1 \leq b_1 \leq r_b. \quad (1)$$

$$1 \leq a_1 \leq r_a, \text{ and } 1 \leq a_1 \leq r_b. \quad (2)$$

From Eqs. (1) and (2), it must be true that

$$1 \leq a_1 \leq \min(r_a, r_b), \text{ and } 1 \leq b_1 \leq \min(r_a, r_b). \quad (3)$$

The ranks r_a and r_b are powers of two, and hence there are two integers k and l such that $r_a = 2^k$ and $r_b = 2^l$. Given that $b_1 > a_1$, it follows from Eq. (3). that

$$(b_1 - a_1) \min(2^l, 2^k) \quad (4)$$

For a collision to occur between nodes a and b , there must be integers $m \geq 0$ and $n \geq 0$ such that two nodes transmit in the same time slot, i.e.,

$$a_1 + n \cdot 2^k = b_1 + m \cdot 2^l \text{ and } b_1 - a_1 = n \cdot 2^k - m \cdot 2^l \quad (5)$$

Given that $b_1 > a_1$, it follows from Eq. 5 that

$$n \cdot 2^k - m \cdot 2^l > 0. \quad (6)$$

There are two cases to consider, either $2^l = 2^k$ or $2^l \neq 2^k$.

Case 1: Assume that $2^l = 2^k$. Then $b_1 - a_1 = 2^k(n - m)$. From Eq. (6) we must have $n > m$; therefore, $b_1 - a_1 \geq 2^k = \min(2^l, 2^k)$, which is a contradiction to Eq. (4).

Case 2: Without loss of generality, assume that $2^l > 2^k$. We then have $l = k + p$ and $2^l = 2^p \cdot 2^k$, with p being a positive integer. Substituting in Eq. (6), we have $n \cdot 2^k - m \cdot (2^p \cdot 2^k) = 2^k(n - m \cdot 2^p) > 0$ and $(n - m \cdot 2^p) > 0$, because $2^k > 0$. It follows from this result that $b_1 = a_1 \geq 2^k = \min(2^l, 2^k)$, which is a contradiction to Eq. (4). It then follows that no two transmissions can collide with each other, as long as the transmitters have consistent neighborhood information. ■

V. PERFORMANCE OF SCMA

A. Network Joining Times

The average rate of empty slots and the maximum node degree D_{max} determine the time taken by nodes to converge to a schedule when new nodes join the network and when nodes choose new connectivity labels. Therefore, a tradeoff exists between the average bandwidth utility at each node and the schedule convergence time.

Figure 4 shows the value of l for different values of N and D_{max} using Algorithm 4. The maximum number of empty slots for nodes to transmit collision-free *Hello*s containing their connectivity labels is l^2 . Therefore, the convergence time is directly proportional to the percentage of slots allotted as empty slots and inversely proportional to the bandwidth utility. Figure 4 shows the average convergence times for different network sizes as a function of bandwidth utility when a time slot lasts 0.5ms. The results show that SCMA incurs small convergence times of just a few seconds, while supporting a large bandwidth utility even for high values of maximum node degree and network size.

B. Comparison with Other Protocols

We compare SCMA with IEEE 802.11 DCF, NAMA (Node Activation Multiple Access) [1], and FPRP (Five Phase Reservation Protocol) [22], which are good examples of contention-based channel access, transmission scheduling based on elections of time slots, and transmission scheduling based on reservations of time slots, respectively.

We used the discrete event simulator Qualnet [23] version 4.5, which provides a realistic simulation of the physical layer, and a well-tuned version of IEEE802.11 DCF. Each simulation was run for randomly distributed 100-node networks for ten different seed values. The time-slot duration for NAMA, FPRP and SCMA was set to 1ms, with the protocols capable of transmitting multiple data and control packets during a single time slot. The signaling packets needed for SCMA, NAMA and FPRP are sent at intervals of 500 milliseconds.

We use the 802.11b physical layer with a data rate of 11Mbps, given that the Qualnet simulator currently does not support the 802.11n physical layer. AODV is used as the common routing protocol running on top of all MAC protocols. We use packet delivery ratio, average end-to-end delay and the application goodput (useful throughput) as our performance metrics.

The simulation was done for two different terrain dimensions in order to study the performance of the protocols for different node densities, neighborhood sizes, and number of collision domains. The nodes have a transmission range of around 250m. We use a combination of random waypoint and group mobility models as our mobility model. The members of a group move following the group mobility model, whereas nodes inside the group move according to the random waypoint mobility model within the group area. The pause time is set to 10s and the minimum and maximum velocities are set to 1 and 5 m/s with a total of 5 groups. This mobility model

Fig. 4. Convergence time for networks of size $N = 100, 300$ and 500 nodes

attempts to depict common situations in which a few members of the same team tend to move together.

Packet-Delivery Ratio: To study packet-delivery ratio, the simulation was run for an increasing number of flows for 150s, with no packets being generated after 100s in order to allow for the maximum packets to get delivered with each protocol. Each flow generates 10 packets per second, with each packet consisting of 512B, and each flow has an average of four hops to destination in the 2000m x 2000m network and eight hops in the 2800 x 2800 network. It can be seen from Fig. 5(a) and 5(b) that SCMA performs much better than all the other protocols as the number of flows increases.

End-to-End Delay: Fig. 5(c) and 5(d) show the average end-to-end delay for the different protocols. SCMA attains much smaller end-to-end delays than NAMA and FPRP, because of the deterministic way in which time slots are assigned to nodes using their connectivity labels. It can also be observed that 802.11b offers less delay than SCMA when there are fewer flows, which is the result of relay nodes having to wait for their transmission turn and the lack of scheduling coordination among relays. However, the delays in SCMA are always below 300 ms and 802.11 incurs higher delays than SCMA as flow load increases.

Goodput Ratio: The two goodputs shown in Fig. 5(e)-(h) for the two terrain dimensions are: (a) the ratio of data packets received over the data packets sent; and (b) the ratio of data packets received over the total number of packets sent, which consists of all data packets sent, the routing control packets sent, and the MAC control packets sent. The first measure of goodput (Fig. 5(e) and (f)) is used to show the number of data packets lost for each protocol due to interference. It is clear from the results that SCMA and NAMA avoid collisions, and that 802.11 and FPRP do experience packet collisions, which can be attributed to the interaction between signaling packets and data packets. The second measure of goodput (Fig.

5(g) and (h)) shows the goodput achieved when all the control overhead is taken into account. When a single flow is present, 802.11b has better goodput than SCMA, because the control packets generated are proportional to the traffic load, while all nodes generate MAC control packets in SCMA. For more than one flow, SCMA outperforms 802.11 and FPRP. Though NAMA shows comparable goodput metrics, the delay incurred for the packets delivered is much higher. The importance of collision-free scheduling is very clear for 10 or more flows.

VI. CONCLUSION

We introduced SCMA, the first approach to medium access control (MAC) in wireless networks based on the classification of nodes according to their connectivity, rather than fixed assignments, contention, reservations, geo-location, or transmission scheduling based on elections or topology-independent codes. SCMA adopts a multi-hop signaling approach to assign connectivity labels to nodes, and uses these labels to establish transmission schedules in a distributed manner.

An added advantage of SCMA over traditional transmission-scheduling schemes is that the channel can be organized into fixed-length transmission frames of the minimum length needed to combat MAI, independently of the network size, shape, density, or node degree. This eliminates having to make design decisions that may impact network performance negatively. SCMA uses deterministic schemes to classify nodes into non-interfering pools of nodes, and to classify nodes within pools so that nodes in the same pool do not interfere with one another. The deterministic nature of SCMA eliminates the coupon-collector problem that is common in many prior MAC protocols. Simulation experiments were used to illustrate that SCMA outperforms representative examples (IEEE 802.11, FPRP, and NAMA) of traditional MAC protocols based on contention, reservations, and elections in terms of packet delivery, goodput and end-to-end delays.

Fig. 5. Simulation results

VII. ACKNOWLEDGEMENTS

This work was supported in part by the Baskin Chair of Computer Engineering at UCSC.

REFERENCES

- [1] L. Bao and J.J. Garcia-Luna-Aceves, "A New Approach to Channel Access Scheduling for Ad Hoc Networks," *Proc. IEEE/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] L. Bao and J.J. Garcia-Luna-Aceves, "Distributed Channel Access Scheduling for Ad Hoc Networks," *Handbook of Algorithms for Wireless Networking and Mobile Computing* (A. Boukerche, Ed.), Chapter 4, Chapman & Hall/CRC, 2006.
- [4] L. Bui, S. Sanghavi, and R. Shrikant, "Distributed Link Scheduling with Constant Overhead," *IEEE/ACM Trans. Networking*, Vol. 17, No. 5, 2009.
- [5] I. Chlamtac, "Fair Algorithms for Maximal Link Activation in Multihop Radio Networks," *IEEE Trans. Commun.*, Vol. 35, No. 7, 1987.
- [6] I. Chlamtac and A. Farago, "Making Transmission Schedules Immune to Topology Changes in Multi-Hop Packet Radio Networks," *IEEE/ACM Trans. Networking*, Vol. 2, No. 1, 1994.
- [7] I. Chlamtac, A. Farago, and H. Zhang, "Time-spread Multiple-access (TSMA) Protocols for Multihop Mobile Radio Networks," *IEEE/ACM Trans. Networking*, Vol. 6 No. 5, pp. 804-12, 1997.
- [8] B.P. Crow et al., "IEEE 802.11 Wireless Local Area Networks," *IEEE Communications Magazine*, Vol. 35, No. 9, 1997.
- [9] P. Djukic and P. Mohapatra, "Soft-TDMAC: A software-based 802.11 overlay TDMA MAC protocol with microsecond synchronization," *IEEE Trans. Mobile Computing*, Vol. 11, No. 3, 2011.
- [10] A. Ephremides, T. Truong, "Scheduling Broadcasts in Multihop Radio Networks," *IEEE Trans. Commun.*, Vol. 38, No. 4, 1990.
- [11] W. Feller, *An Introduction to Probability Theory and Its Applications*, Vol. 1, John Wiley, 2nd Edition, 1957.
- [12] J.J. Garcia-Luna-Aceves, and A. Tzamaloukas, "Reversing The Collision-Avoidance Handshake in Wireless Networks," *Proc. IEEE/ACM MobiCom '99*, Aug. 1999.
- [13] S.Katragadda et al., "A Decentralized Location-Based Channel Access Protocol for Inter-vehicle Communication," *Proc. 57th IEEE Vehicular Technology Conf.*, April 2003.
- [14] C. Joo and N. Shroff, "Performance of Random Access Scheduling Schemes in Multi-hop Wireless Networks," *IEEE/ACM Trans. Networking*, Vol.17, No. 5, 2009.
- [15] J. Ju and V. Li, "An Optimal Topology-Transparent Scheduling Method in Multihop Packet Radio Networks," *IEEE/ACM Trans. Networking*, Vol. 6, No. 3, 1998.
- [16] C.H. Rentel, and T. Kunz, "On the Average-Throughput Performance of Code-based Scheduling Protocols for Wireless Ad Hoc Networks," *Proc. ACM MobiHoc 05*, May 2005.
- [17] C. H. Rentel and T. Kunz, "A mutual network synchronization method for wireless ad hoc and sensor networks," *IEEE Trans. Mobile Computing*, Vol. 7, No. 5, 2008.
- [18] Z. Tang and J. J. Garcia-Luna-Aceves, "Hop Reservation Multiple Access for Multichannel Packet Radio Networks," *Computer Communications*, Vol. 23, No. 10, 2000.
- [19] N. Wen and R. Berry, "Information Propagation for Location-based MAC Protocols in Vehicular Networks," *Proc. 40th Annual Conf. on Information Sciences and Systems (CISS)*, March 2006.
- [20] Y. Wang and J.J. Garcia-Luna-Aceves, "Collision Avoidance in Multihop Ad Hoc Networks," *Proc. IEEE MASCOTS '02*, Oct. 2002.
- [21] C. D. Young, "USAP: A Unifying Dynamic Distributed Multichannel TDMA Slot Assignment Protocol," *Proc. IEEE MILCOM '96*, Oct. 1996.
- [22] C. Zhu and M.S. Corson, "A Five-Phase Reservation Protocol (FPRP) for Mobile Ad Hoc Networks," *Wireless Networks*, Vol. 7, No. 4, 2001.
- [23] Qualnet4.5 "Scalable Network Technologies: <http://www.scalablenetworks.com>"