

Research Article

RMTS: A Novel Approach to Transmission Scheduling in Ad Hoc Networks by Salvaging Unused Slot Transmission Assignments

Vikas Bollapragada Subrahmanya  and **Harlan B. Russell**

*Department of Electrical and Computer Engineering, 317 Fluor Daniel Engineering Innovation Building,
Clemson University, Clemson, SC 29634, USA*

Correspondence should be addressed to Vikas Bollapragada Subrahmanya; vbollap@clemson.edu

Received 30 June 2018; Revised 19 October 2018; Accepted 14 November 2018; Published 12 December 2018

Academic Editor: Giovanni Stea

Copyright © 2018 Vikas Bollapragada Subrahmanya and Harlan B. Russell. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In mobile ad hoc networks (MANETs) there is no centralized supervision over the network operations. In such networks, it is imperative to have protocols that are implemented locally in a distributed manner. To achieve high network performance utilizing only the wireless channels a distributed transmission-scheduling medium-access control (MAC) protocol is desirable. This type of protocol can guarantee a certain level of quality of service (QoS) and works well in congested networks. However, the transmission schedules are often decided in advance and if a node does not have a packet to transmit in its assigned slot then the slot is unused. We propose a protocol called recovering minislots transmission scheduling (RMTS) that salvages these unused transmission slots that would have otherwise been wasted. We show that the proposed protocol produces significant improvements in network performance over traditional transmission-scheduling approaches.

1. Introduction

A mobile ad hoc network (MANET) is a unique type of network in which there is no centralized supervision over the operation of the network. These type of networks can be easily deployed because little preplanning is required. This feature makes MANETs very desirable in situations where there is no existing infrastructure or the current infrastructure has failed. MANETs have been traditionally used during military operations and disaster management. With advent of smart devices almost every device today is equipped with a radio. A MANET provides an economical solution to setup a temporary network for these radios to communicate with each other.

A wireless channel is a shared medium. When multiple nodes transmit at the same time they create multiple-access interference. A receiver that is exposed to a strong multiple-access interference environment is likely to experience significant errors. Hence, medium access control (MAC) protocols have been of interest. MAC protocols control how nodes gain access to the channel. Traditionally, there have been two

types of MAC protocols: contention based and transmission-scheduling based.

In contention based MAC protocols the nodes compete with each other to gain access to the channel. The protocols described in [1–5] provide a few such examples. One distinct characteristic of these protocols is that there is always a possibility that multiple nodes might attempt to transmit at the same time resulting in a collision at the receiving node. These protocols work very well in low traffic scenarios but tend to underperform in higher traffic scenarios where there is a higher probability of a collision.

In transmission-scheduling based MAC protocols the nodes gain access to the channel by temporarily reserving the channel. There are different approaches to reserving the channel. The reservation is based on time in TDMA, on frequency in FDMA, and on code in CDMA. We focus on time based reservation schemes in this work, in which the time is divided into slots. These slots are then assigned to nodes in such a manner that the multiple-access interference caused by the concurrent transmissions remains at acceptable levels. In many implementations the slot assignments repeat

periodically and are grouped into frames. Typically, every node is guaranteed at least a single slot in a frame. These protocols usually schedule broadcast transmission opportunities so that a transmission can be received by any of the transmitter's immediate neighbors.

One major drawback of these protocols is that if a node does not have a packet to transmit in its assigned slot then the slot is wasted. This limits the performance of these protocols at lower traffic rates as the delay is quite high as compared to the contention-based MAC protocols. In this work we propose a protocol which salvages these unused slots. Our proposed scheme called *recovering minislot transmission scheduling* (RMTS) extends scheduling-based MAC protocols that employ periodic slot assignments (a preliminary investigation of RMTS is presented in [6]). The key idea of our scheme is that a node has the capability to detect an unused slot and capture the slot from the node that was scheduled to transmit but did not. One restriction on a substitute transmission is that it should cause similar interference compared to the node to which the slot was originally assigned. This is to ensure that the other transmissions that are scheduled in that slot are not disrupted.

RMTS is designed with a focus on multihop networks. Through simulation investigations we show tremendous improvement in performance over traditional scheduling algorithms for multihop networks with a relatively low density of nodes. However, to demonstrate the robustness and versatility of the design, investigations are also run for networks with medium and high node densities without changing any parameters of RMTS. We show that RMTS outperforms traditional scheduling algorithms by a considerable margin for a wide range of network densities.

Emerging MANETs are heterogeneous in nature. Any newly designed protocol must be robust enough to work well in such scenarios. Accordingly we also investigate scenarios where nodes do not generate traffic in a uniform manner. Even in such scenarios RMTS maintains an acceptable level of performance while the other protocols perform very poorly under nonuniform traffic. Finally, we demonstrate that by selecting the substitute transmissions carefully as opposed to randomly selecting them further improvement in the network performance can be achieved.

2. Related Work

The MAC protocol plays a pivotal role in the functioning of an ad hoc network and there are a wide variety of designs. The authors in [7] provide an extensive survey of the MAC protocols designed for ad hoc wireless networks. The article, however, predominantly focuses on contention based MAC protocols. In general, contention based protocols have received more consideration as compared to scheduling based protocols because of their relative ease of operation. In this section we focus on the work done on the latter. Many of these approaches are known to produce better network performance than traditional contention based approaches under heavy traffic loads and dense deployment of nodes.

In one of the earliest works [8], the authors show that optimally assigning interference free broadcasting schedules

in multihop networks is NP hard. Both centralized and distributed algorithms are developed that provide an approximate solution that runs in polynomial time. This approach however requires network wide coordination to agree up on a frame size. Hence, it does not adapt well to changes in connectivity because the schedules must be rebuilt each time there is a change in topology.

A popular approach to transmission scheduling is to assign color numbers to nodes and use them to determine the schedules. The author in [9] surveys the underlying methodology of this approach and proposes a centralized scheduling algorithm called RAND. A distributed extension to RAND is DRAND and it is presented in [10]. In DRAND there is a contention phase in which the slot schedules are decided and slot synchronization is not required. The performance of DRAND is shown to be similar to RAND. Lyui's protocol [11] is another approach in which a similar color assignment scheme is used but it employs a different algorithm to use these color numbers to assign the transmission slots. Investigations in [12] show that the performance of Lyui's protocol is similar to that of RAND. Lyui's protocol however does offer certain additional advantages by supporting mobility and maximizing the slot assignments.

In the approaches that we have discussed previously, the slot assignments are fixed and every node is guaranteed a single slot in the frame irrespective of the traffic at that node. In the approaches to follow the nodes can negotiate for multiple slots in a single frame depending upon their projected traffic levels. One such protocol is USAP [13] where a node is allowed to request multiple slots from a pool of unassigned slots depending on its traffic projections. The nodes are capable of detecting and resolving any conflicts that arise from their requested slot assignments. In [14] the authors propose a delay efficient scheme where the slots are assigned to a traffic flow. The idea behind this approach is that the nodes in a particular flow are assigned adjacent slots in a frame so that once a packet leaves the first node in the frame it does not undergo any further queuing delay. For deterministic packet arrivals, the proposed scheme works much better than traditional color based schemes. However, for non-deterministic packet arrivals, the delay is higher than for color based schemes.

Another approach to scheduling is to alternate between contention and reservation periods. During the contention phase, the nodes contend with each other for the slot assignments. These short contention periods are then repeated periodically to calculate new slot assignments. The five-phase reservation protocol (FPRP) [15] is an example of such an approach where a node uses a five-way exchange among the nodes that are within 2-hops to negotiate for slot assignments. In [16] an extension to the FRRP is proposed that reduces the frame length.

Another class of scheduling protocols is called pseudo-random protocols whereby the states of the nodes are decided by pseudo-random number generators (PRNG). The nodes can then deterministically predict the states of nodes just by knowing the seeds of their respective PRNG. An example of such an approach is SEEDDEX [17] where nodes can essentially publish their schedules to their neighbors by

simply exchanging the seeds of PRNG with them. A major drawback of all the protocols described above is that a slot is wasted if a node to which a slot is assigned does not have a packet to transmit.

In this article we propose a scheme where every slot is extended by a small but fixed number of minislots. Every node chooses some of its neighbors as auxiliary transmitters. In the event that a node to which a slot is assigned does not transmit, at the end of the first minislot the first auxiliary node attempts to transmit and so on. In this manner we attempt to salvage the slots that would otherwise be wasted. The investigations that come closest to the work in this article are described in [18, 19].

In [18] a minislotted approach is considered. In a network with N nodes, every slot is extended by $N-1$ minislots. Nodes are then assigned priority based on some rule in each slot. At the beginning of a slot the node with the highest priority transmits a carrier signal for the duration of a minislot if it has a packet to transmit. If the node with the next highest priority does not detect the carrier by the end of first minislot it transmits a carrier signal. This process is repeated until a carrier signal is detected. At the end of the minislot $N-1$ the node that has transmitted the carrier transmits its packet. The only way a slot goes unutilized is when every node in the network does not have a packet to transmit. This approach yields good results when the size of the network is small but it does not scale well for a network with a large number of nodes. Furthermore, this approach assumes all nodes are within communication range so there can be only one transmission at a time. Our work extends the use of minislots for multiple-hop networks and allows more than one node to transmit in the same slot. We account for the hidden terminal problem and control the multiple-access interference when scheduling transmissions.

A protocol to permit more than one node the opportunity to utilize a slot is described in [19]. The approach, called CAMA, is an extension of USAP [13]. The network is divided into multiple cliques, and slots are assigned to the cliques instead of individual nodes using USAP. In the slot assigned to a particular clique the nodes constituting that clique contend for the slot via minislots at the beginning using a non-persistent CSMA protocol [20]. However, the authors of [19] point out that the implementation is quite complex, requires careful tuning based on connectivity details, and has higher overhead than required by USAP.

3. System Model

Time is divided into equal slots. The slot duration is equal to the time taken to transmit a packet. Nodes are assumed to be synchronized to the slot boundaries. See [15, 21] for examples of methods to achieve slot synchronization. The basic channel model is similar to the one from our previous work [22]. The mode of communication between the nodes is half duplex, that is, at any given instant nodes can either transmit or receive but not both. The nodes are assumed to radiate with equal strength in all directions, that is, the antenna gain is independent of the direction. The nodes use direct-sequence spread-spectrum (DSSS) modulation. We employ a

signal-to-noise-plus-interference ratio (SINR) model where a transmission is considered to be correctly decoded only if the SINR at the receiving node is greater than a threshold β . For a transmission from node x to node y to be successful the SINR at y , denoted by $\xi_{x,y}$, must satisfy

$$\xi_{x,y} = \frac{P_r(x,y) \times T_c \times N_s}{N_o + \sum_{z \neq x} P_r(z,y) \times T_c} \geq \beta \quad (1)$$

where $P_r(x,y)$ is the power received by node y from a transmission from node x , T_c is the chip duration, N_s is the spreading factor, and N_o is the measure of white noise at the receiver. The term $\sum_{z \neq x} P_r(z,y) \times T_c$ is the interference at node y . It represents the accumulation of energy at y from all transmissions other than from x .

The urban area cellular radio path loss model described in [23] is employed. According to the model the power received by a node y from a transmission from node x , denoted by $P_r(x,y)$, is given by

$$P_r(x,y) = P_t \times \left(\frac{c}{4 \times \pi \times \nu \times d_{x,y}} \right)^\alpha \quad (2)$$

where P_t is the transmit power, c is the speed of the light, ν is the frequency of the transmission, $d_{x,y}$ is the Euclidean distance between nodes x and y , and α is the path loss exponent.

The transmit power P_t is assumed to be constant throughout this work. We set the transmit power such that the signal to noise ratio at a distance of R from the transmitter is equal to β , where R is the transmission radius. The signal to noise ratio at which the bit error rate is equal to 10^{-5} is chosen as β .

Before we proceed further, we describe terminology employed throughout this document. For a given node its *1-neighbors* are the nodes with which communication is possible, that is, nodes that are one hop away. The *1-neighborhood* of a node includes the node and its 1-neighbors. Likewise, *2-neighbors* are nodes that are not 1-neighbors but share a common 1-neighbor. The *2-neighborhood* of a node comprises the node, its 1-neighbors, and its 2-neighbors.

4. Lyui's Algorithm

We develop RMTS as an extension of Lyui's algorithm, initially described in [11]. Lyui's algorithm is a distributed scheduling algorithm that assigns collision free schedules in MANETs. In this section, we briefly describe the algorithm. Refer to [12] for a detailed explanation and a discussion on the merits of the algorithm. The transmission schedule is determined by the colors of the nodes. A node picks the smallest possible positive number as its color while ensuring that this color is unique in its 2-neighborhood. The color assignment problem is well known, and there are numerous distributed coloring algorithms available [9]. Every node maintains a neighbor table to store information of the nodes in its 2-neighborhood. After a node picks a color it relays this information to every node in its 2-neighborhood. A node ν is a candidate to transmit in a slot t if its color number C_ν

C_v	$P(C_v)$	t																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
2	2		x		x			x				x			x		x	
3	4			x					x					x			x	
4	4				x					x				x				x
5	8					x									x			
6	8						x									x		
7	8							x									x	
8	8								x									x

FIGURE 1: Transmission slots as assigned by Lyui's algorithm.

Minislot #1	Minislot #2	...	Minislot # N_a	Original slot in Lyui's algorithm
-------------	-------------	-----	------------------	-----------------------------------

FIGURE 2: New slot in RMTS.

satisfies the following equation for some nonnegative integer n

$$t = C_v + n \times P(C_v) \quad (3)$$

where $P(C_v)$ is the smallest nonnegative power of 2 greater than or equal to C_v , that is, it is the smallest nonnegative integer k such that

$$P(C_v) = 2^k \geq C_v \quad (4)$$

In slot t each node prepares a candidate list. In this list a node includes all of its 2-neighbors that satisfy (3) for that slot. A node will be assigned to transmit in that slot if it is included in its candidate list and has the highest color amongst other members of the list. Hence, a node only requires the colors of its 2-neighbors in order to calculate its transmission assignments. For additional information on how initial schedules are formed and a protocol for updating color numbers in mobile scenarios refer to [24] and [25], respectively.

In Figure 1 we show the candidate color numbers in the first sixteen slots. A candidate in a slot is denoted by an x in the table. There can be multiple candidates in a slot. The node with the largest color number amongst those candidates is assigned to transmit in that slot. For example, in slot number one, the node with color number 1 is the only candidate and it is assigned to transmit in that slot. In slot number seven, the nodes with color numbers 1, 3, and 7 are candidates. A node with color number 3 defers if there is a node with color number 7 in its 2-neighborhood. Otherwise, it can transmit in this slot. Notice that in this case the frame size is equal to eight, that is, the transmission assignments repeat every eight slots. In general, the frame size for a node is equal to $P(C_m)$, where c_m is the maximum color number in that node's 2-neighborhood.

5. Recovering Minislots Transmission Scheduling (RMTS)

In RMTS there are two types of transmitters in every slot. Nodes use Lyui's algorithm to calculate their slot assignments. In a given slot, the nodes that are assigned to transmit by

the virtue of Lyui's algorithm are called primary transmitters. The transmissions arising out of these assignments are called primary transmissions. For each slot in which a node is a primary transmitter, it chooses N_a of its 1-neighbors as auxiliary transmitters and forms an auxiliary set, ζ . At the beginning of a slot, the primary nodes in that slot have the first chance to transmit. For each primary node, the auxiliary transmitters chosen by that node monitor the channel to detect the primary transmission. If no transmission is detected, one of the auxiliary transmitters will attempt to transmit. The resulting transmission is called an auxiliary transmission.

In RMTS each slot is extended by N_a minislots as shown in Figure 2. The auxiliary transmitters chosen by a node are numbered 1 to N_a . This auxiliary number (A.No) determines the order in which the auxiliary transmitters attempt to transmit in the event that the primary transmitter does not. At the end of the first minislot if the auxiliary transmitter with A.No equal to 1 has not detected a transmission it will attempt to transmit. By the end of the second minislot if the auxiliary transmitter with A.No equal to 2 has not detected a transmission it will then attempt to transmit, and so on. In this manner the probability of the transmission slot being utilized is increased.

Special care has to be taken while choosing the auxiliary transmitters. The primary transmissions are scheduled such that the expected signal-to-noise ratio at the receiver is greater than the threshold β . If we choose just any 1-neighbor as an auxiliary transmitter it could disrupt the other scheduled transmissions in that slot. Hence, not every 1-neighbor of a node is eligible to be an auxiliary transmitter. An auxiliary transmission must create multiple-access interference similar to that which would have been created by a primary transmission, and an auxiliary node must be able to detect the transmissions of other nodes from the auxiliary set it belongs to.

For a node x to be eligible to be an auxiliary transmitter of node y , the signal to noise ratio of the link between the two nodes, denoted $\epsilon_{x,y}$, must satisfy

$$\epsilon_{x,y} \geq P \times \beta \quad (5)$$

TABLE 1: Typical entry from a neighbor table.

Node ID	Color	Type	SNR	A.No
u	2	1	80	1

where $P(\gg 1)$ is a multiplicative factor that controls the level of the multiple-access interference caused by an auxiliary transmitter. If the value of P is large then the interference environment created by an auxiliary transmission will be similar to that of a primary transmission. However, a large value of P limits the set of the 1-neighbors that are eligible to be auxiliary transmitters. A smaller value for P permits a larger set of 1-neighbors to be chosen as auxiliary nodes. However, the interference caused by these nodes could be very different to that of a primary transmission. For moderate values for P , the resulting increase in interference is negligible when compared to the improvement in the network performance, as shown later.

To ensure that the auxiliary transmitters of a primary transmitter are able to detect each other's transmissions they must form a clique. The details for selecting the auxiliary nodes are given next.

5.1. Neighbor Table. In this section we describe the neighbor table of a node. Each node maintains a neighbor table, \mathcal{N} , to store information about its 2-neighbors. This information is typically used by the nodes for building routing tables, calculating their transmission schedule, and selecting their auxiliary nodes. Table 1 depicts an entry from the neighbor table of node k .

The neighbor table for node k contains an entry for every node in its 2-neighborhood. The first column contains the identity of the neighbor, called u . The second column contains the color number of node u . The third column indicates if node u is a 1-neighbor or a 2-neighbor. The fourth column stores the SNR of the link to node u . The last column indicates if node k belongs to the auxiliary set of node u . The values in this column can be between 0 to N_a with 0 indicating that node u has not selected node k as an auxiliary transmitter. A non-zero A.No value indicates that node k belongs to the auxiliary set of node u and the value is the minislot to which it is assigned.

5.2. Selecting the Auxiliary Nodes. Each node maintains an estimate of the SNR from the transmissions it receives from each of its 1-neighbors. After a node builds its neighbor table it broadcasts it to all of its 1-neighbors so that each node has a copy of neighbor tables of all of its 1-neighbors.

Using the SNR estimates from its neighbor table and the condition (5) a node x forms a list of eligible nodes \mathbb{S}_x . Node x then creates a sub-graph, \mathbb{G}_x , using nodes from \mathbb{S}_x as vertices. An edge exists in \mathbb{G}_x between a pair of vertices if and only if they list each other as 1-neighbors in the copies of their respective neighbor tables (which the node has stored previously). From \mathbb{G}_x the node x then selects a maximal clique, \mathbb{A}_x . The node then orders elements of \mathbb{A}_x according to some priority. Because every subset of clique is also a clique, node x selects the first N_a nodes from the ordered \mathbb{A}_x to form

its auxiliary list, c_x . The position of a node in the auxiliary set c denotes its A.No value. The node then broadcasts this auxiliary set to all of its 1-neighbors. Specifically, after node y receives an auxiliary list c_x from node x it checks to see if it is included in the list. If it is included in c_x it updates the last column of the entry corresponding to node x in its neighbor table with its position in the list. Forming a maximal clique is an NP hard problem. However, there are algorithms for finding maximal cliques that have acceptable run times for graphs with hundreds of vertices (see [26, 27]).

5.3. Ordering the Auxiliary Nodes. Once the maximal clique of eligible nodes is formed they are ordered based on some priority. In the first method called *random-order* the nodes are ordered randomly. That is, the auxiliary nodes are selected at random from the list of eligible nodes at the beginning of the simulation. The selected nodes are then fixed for the duration of the simulation. However, the selected auxiliary nodes may not always be able to take advantage of unused slots as they themselves might not have packets to transmit. Hence, two other metrics are used to order the nodes in the maximal clique, namely *utilization* and *queue-length*. *Utilization* of a node is defined as the fraction of the available transmission opportunities to the node that it utilizes. This includes all the node's primary assignments as well as any auxiliary assignments that become available to the node because the slot was not utilized by another node. *Queue-length* of a node is defined as the number of packets queued up at that node. Whenever a node transmits a packet it includes the values of these metrics in the header. The 1-neighbors of the node decode the header and store this information in their neighbor tables. We use the feedback from the previous transmissions to order the auxiliary nodes.

Picking auxiliary nodes by using the last two metrics increases the probability of auxiliary transmissions but requires a small amount of additional overhead. The values of these metrics do not remain constant over time. Hence, every node has to recalculate its auxiliary list periodically. This list is calculated locally at every node and has to be shared with its 1-neighbors only. We assume that the nodes transmit this information along with their periodic control messages. In our simulation every node recalculates its auxiliary list after every 128 slots. Because the gains in performance for the utilization and queue-length metrics are modest for typical multiple-hop networks, we present our initial investigations using the simpler random ordering metric. However, we do show that in certain high density networks the additional overhead required for the other two metrics can result in substantial improvements in network performance. Unless otherwise mentioned it is assumed that the nodes are ordered randomly in all the presented results.

5.4. Features. One aspect in which RMTS varies from traditional dynamic scheduling protocols is fairness. In RMTS, every node is guaranteed a transmission opportunity at least once in a frame. Only when a node does not have a packet to transmit in its assigned slot do the other nodes attempt to utilize that slot. In this manner nodes with higher traffic

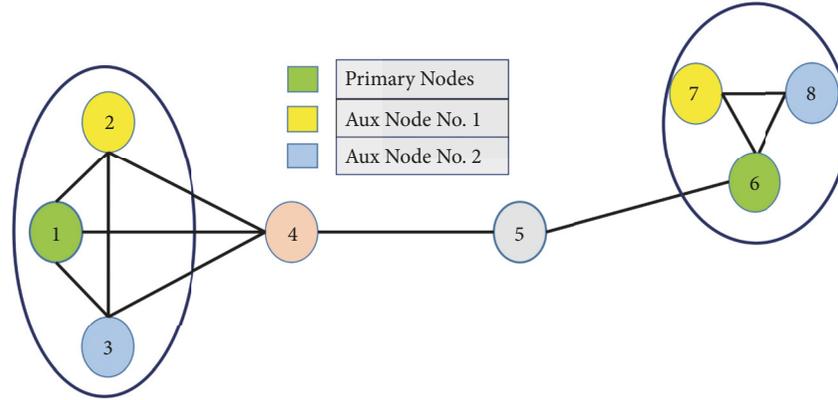


FIGURE 3: A Simple Network.

might be able to transmit multiple packets in a single frame. However, these additional transmission opportunities do not come at the expense of the nodes with lower traffic.

Another important aspect is that the design of RMTS is not specific to Lyui's algorithm. Our protocol will work well with most distributed scheduling algorithms as it does not depend on how the primary transmitters are assigned.

5.5. A Simple Example. Let us consider a simple network consisting of eight nodes as shown in the Figure 3. Nodes 1 and 6 are the primary nodes and can transmit in the same slot without interfering with each other. Node 1 selects nodes 2 and 3 as its auxiliary nodes. Similarly, node 6 selects nodes 7 and 8 as its auxiliary nodes.

In RMTS in any given slot there are multiple cliques of nodes and any one node from each of these cliques can transmit simultaneously whilst causing minimal interference to each other's transmissions. The circles in the Figure 3 represent the cliques for this example. At the beginning of the slot the primary nodes (i.e., nodes 1 and 6) have the first opportunity to transmit. By the end of the first minislot if a first auxiliary node (i.e., nodes 2 or 7) does not detect a transmission then it can transmit if it has a packet. Similarly, by the end of the second minislot if a second auxiliary node (i.e., nodes 3 or 8) still does not detect a transmission then it can transmit if it has a packet. In this manner we have roughly tripled the chance that an assigned slot is utilized.

6. Other Protocols

We compare the performance of RMTS with three other protocols. The first one is Lyui's algorithm. The other two protocols are described in this section.

6.1. SEEDEX. SEEDEX [17] is a unique type of distributed scheduling algorithm where the transmission schedules are pseudo-random. In every slot a node is in one of two possible modes. The node is in the possibly transmit (PT) mode with probability p_s or the listen (L) mode with probability $1 - p_s$. These modes are decided by a pseudo-random number generator that generates a sequence of 1's and 0's according to

an i.i.d Bernoulli sequence with 1 indicating that the node is in PT mode and a 0 indicating that the node is in L mode. The nodes then exchange the seeds of their pseudo-random number generators with their 2-neighbors. The seeds are the only information that nodes need in order to calculate their schedules as this enables a node to know what nodes their 2-neighbors are in at any given time.

Whenever a node has a packet to transmit it waits for a slot in which it is in PT mode and the receiving node is in L mode. The transmitting node then counts the number, n , of the 1-neighbors of the receiving node that are also in the PT mode in that slot. The node then transmits with probability $\min\{\alpha/(n+1), 1\}$, where α is a parameter that decides how aggressively a node attempts to transmit. Following [17], we set p_s equal to 0.21. There are two different values specified for α (1.5 and 2.5) in [17]. We set α equal to 2.5 as it maximizes the performance in our investigations.

6.2. Unlimited RMTS (u-RMTS). In RMTS once a node forms a maximal clique, \mathbb{A} , from the set of eligible nodes, \mathbb{S} , it chooses N_a nodes from it at random to form its auxiliary set, ζ . In the event that a node does not transmit in its assigned slot only the nodes from ζ can attempt to transmit in that slot. In u-RMTS the auxiliary set is calculated in the same manner but there is no limit on the number of auxiliary transmitters that a node can choose. However, we do not model the overhead incurred by the presence of a large number of minislots. The u-RMTS serves as an approximate upper bound on the performance of RMTS.

7. Simulation Setup

We developed a custom network simulator in C to study the properties of our proposed protocol. The simulations were run on the palmetto cluster, the supercomputing cluster at Clemson [28]. In the simulations the time is assumed to be divided into slots. The physical and link layers are modeled as described in the above sections. The network layer is modeled in such a manner that nodes relay packets based on fixed forwarding tables. The initial forwarding tables are formed at the beginning of each simulation. For simplicity a

TABLE 2: Network densities.

Name	Density	Avg # of Hops	Diameter
D-1	1 node per 25 sq.m	1.2	2
D-2	1 node per 50 sq.m	1.9	4
D-3	1 node per 75 sq.m	3.0	7

centralized algorithm is used to calculate these initial tables. The algorithm is based on min-hop routing with an imposed penalty for links that have poor SINR values. It is similar to the algorithm employed in [22]. For a detailed explanation of the algorithm and the cost metric used see [29].

7.1. Simulation Parameters. The network considered in our simulations consists of 100 nodes that are located according to a uniform distribution in a fixed area. Simulation results are presented for networks with three different densities. These densities are listed in Table 2 where the average number of hops denotes the average number of relays for a packet to reach its destination and the diameter denotes the maximum number of relays required by some packet to reach its destination.

All the nodes transmit at the same power. The transmission power is set such that at a distance of 200 m from the transmitter the SNR is equal to β . The queue at each node can hold up to 10 packets. When a packet arrives at the node that is full it is discarded. The time to live (TTL) counter is set equal to 200 slots. At the beginning of each slot a node checks its queue and discards the packets whose TTL counters have expired.

We do not implement link or network layer acknowledgments in this study. However, we do implement link level acknowledgments while simulating SEEDEx [17] as part of its MAC layer design. Upon successful reception of a packet, the receiver sends an acknowledgment and we assume it is perfectly received.

Three network metrics are reported: *end-to-end completion rate*, *delay*, and *end-to-end throughput*. End-to-end completion rate is the fraction of packets delivered to their destinations. Delay is the average number of slots that elapse between when a packet is generated to when it reaches its destination. End-to-end throughput is the average number of packets delivered to their destinations per slot. The metrics are obtained over 4000 slots for each network and are averaged over 100 random networks.

The values of N_a and P are chosen to get the best end-to-end completion rate performance for networks with low density and are accordingly set equal to five and ten respectively for all the simulations. A detailed explanation on how these values are set can be found in [29]. All of these above simulation parameters are listed in Table 3. All of the system parameters are listed in Table 4.

7.2. Packet Generation. At the beginning of each slot, every node in the network generates a single packet with some probability. This probability is fixed for that node throughout the simulation. Traditionally in network simulations all the nodes are assumed to have same probability of generating a

TABLE 3: Simulation parameters.

Parameter	Symbol	Value
Number of Nodes	N	100
Number of Slots	Slots	4000
Numbers of Simulations	Sims	100
Time to Live	TTL	200 slots
Queue Size	Qs	10 packets
Range	R	200 m
Number of Auxiliary Nodes	N_a	5
Multiplication Factor	P	10

TABLE 4: System parameters.

Parameter	Symbol	Value
Chip Duration	T_c	$2.9 * 10^{-7}$
Spreading Factor	N_s	128
Wavelength	λ	0.125
Path Loss Exponent	α	3.5
SINR Threshold	β	8
One-Sided Spectral Density	N_o	$4 * 10^{-21}$

packet in any given slot. In addition, we model heterogeneous networks in which the generation rate is not same at all nodes. Two different models are utilized: *uniform* and *gamma* packet generation models.

7.2.1. Uniform Packet Generation Model. In this model we assume that all the nodes have the same probability of generating a packet at the beginning of a slot. This probability is equal to γ/N , where γ is average number of packets generated per slot and N is the number of nodes in the network.

7.2.2. Gamma Packet Generation Model. In this model we assume that all the nodes have a different probability of generating a packet. These probability values are drawn from a gamma distribution with mean value equal to γ/N . The gamma distribution is chosen because we can have multiple distributions, each with the same mean but with a different variance. We use three different gamma distributions with parameters given in Table 5, where a is the shape parameter, and b is the scale parameter.

8. Results

We developed RMTS for multihop networks and hence we focus on the results for the simulation studies with a low node density. We tabulate the important metrics for other scenarios. Additionally, the graphs for all the scenarios can

TABLE 5: Parameters for the gamma packet generation model.

Dist Name	a	b	Mean $a \times b$	Variance $a \times b^2$
Gamma1	γ/N	1	γ/N	γ/N
Gamma2	$\gamma/5N$	5	γ/N	$5\gamma/N$
Gamma3	$\gamma/10N$	10	γ/N	$10\gamma/N$

TABLE 6: Values of Γ .

Density	L	S	R	% imp in Γ	
				L	S
D-1	.10	.25	.66	520%	160%
D-2	.11	.31	.51	360%	63%
D-3	.17	.43	.61	250%	40%

be found in [29]. First, we present the results utilizing the uniform packet generation model. In Section 8.3 the results with gamma packet generation model are given. In all of the following figures the x-axis represents the average number of packets generated per slot by all the nodes in the network (γ).

8.1. Results with Uniform Packet Generation Model. Figure 4 shows the end-to-end completion rate for networks with the low node density model D-3. Note there is little difference in the network performance for RMTS and u-RMTS protocols even though in u-RMTS there is no limit on the number of auxiliary nodes. This is because in scenarios with a low density of nodes there are a limited number of potential auxiliary nodes. Utilizing RMTS results in significant improvement in the value of the end-to-end completion rate compared to SEEDEX and Lyui's protocols. In order to quantify the improvement we define the end-to-end completion rate threshold, Γ , as the value of γ beyond which the end-to-end completion rate is less than 90%. The RMTS protocol increases Γ by approximately 250% compared to Lyui's protocol and by approximately 40% compared to SEEDEX. The values for Γ for other node density scenarios are listed in Table 6, where L refers to Lyui's protocol, S refers to SEEDEX, and R refers to RMTS. The last two columns in the table show the percentage improvement in Γ of RMTS over Lyui's protocol and SEEDEX, respectively.

Even in scenarios with the uniform packet generation model certain nodes in the network can have more packets queued up than the others. For example a node could be a bottleneck node as a lot of traffic is routed through it. Hence, it always has more packets queued up than its neighbors. If equal opportunities are assigned to nodes the nodes with more packets queued up will eventually drop some packets leading to poor end-to-end completion rate. RMTS automatically takes care of this disparity by allowing a node to be selected as an auxiliary transmitter by any number of its 1-neighbors. So in such scenarios the node that has more packets queued up can transmit multiple times in a single frame resulting in better end-to-end completion rate.

Figure 5 shows the end-to-end delay for scenarios with low node density. Once again RMTS outperforms the other protocols significantly. For example, the maximum delay for

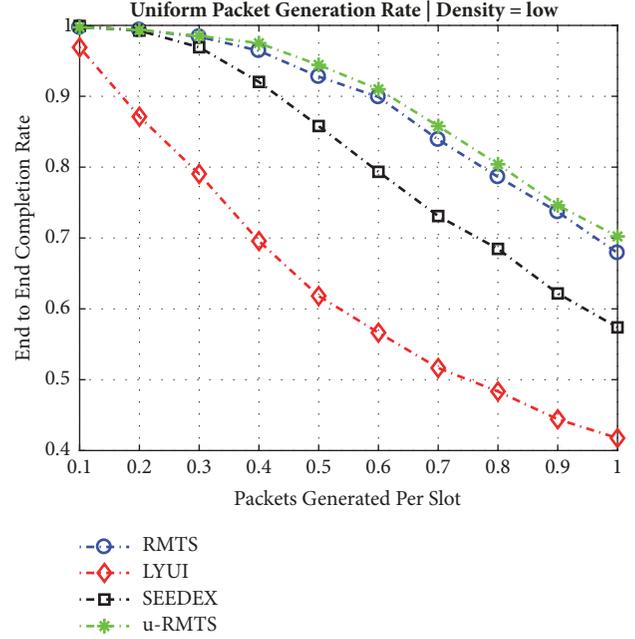


FIGURE 4: End-to-End completion rate for low density.

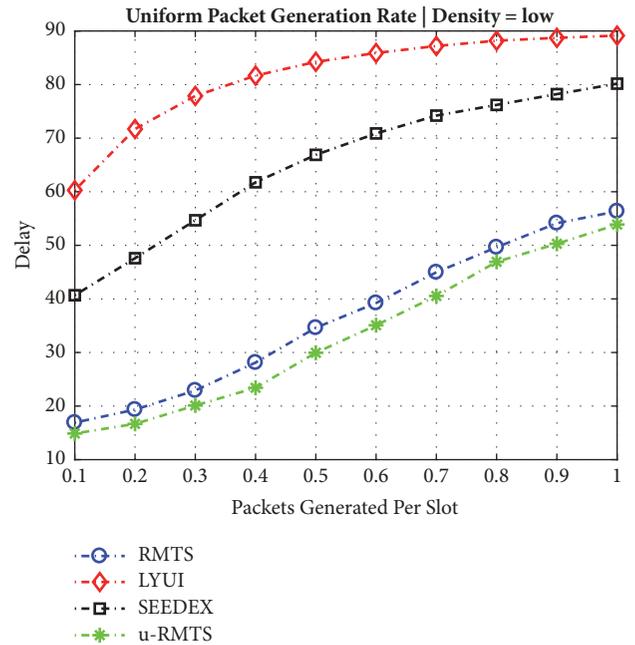


FIGURE 5: Delay for low density.

RMTS is less than minimum delay for Lyui's protocol. As Lyui's protocol and SEEDEX have Γ values less than RMTS we compare the delay values of RMTS with them at their respective Γ values. For Lyui's protocol Γ is 0.17 and at this packet generation rate there is an increase of about 267% in delay compared to RMTS. Similarly, for SEEDEX Γ is 0.43 and at this packet generation rate there is an increase of about 107% in delay compared to RMTS. Table 7 lists the values of percentage increase in the delay in other node density scenarios and similar trends are observed. Some of the nodes

TABLE 7: Values of percentage increase in delay over RMTS.

Density	% Inc in delay over RMTS	
	Lyui's	Seedex
D-1	221%	132%
D-2	270%	108%
D-3	267%	107%

 TABLE 8: Values of percentage increase in σ of delay over RMTS.

Density	% Inc in std of delay over RMTS	
	Lyui's	Seedex
D-1	107%	81%
D-2	186%	74%
D-3	206%	71%

in the network have an opportunity to transmit before their assigned slot in the frame as auxiliary nodes. This results in reduced delay for those particular transmissions which results in an reduced average delay in the network.

Figure 6 shows the standard deviation, σ , of the delay. We use σ to measure the jitter in the delay. Once again we compare σ at the Γ values for Lyui's protocol and SEEDEX, respectively. Accordingly, Lyui's protocol has an increase of approximately 206% in σ compared to RMTS, and SEEDEX has an increase of about 71% in σ compared to RMTS. We list the values for other scenarios in Table 8. The improvement in delay results in a better throughput performance and is shown in Figure 7.

8.2. Slot Utilization in the Network. Slot utilization of a node x , denoted by U_x , is defined as the fraction of node x 's assigned slots that are utilized, and while measuring this metric both the primary and auxiliary transmissions are considered. Let the probability that node x does not have a packet to transmit in its assigned slot be P_x . Assume that all the auxiliary nodes have the same probability of not having a packet to transmit in the slot. The probability that the slot will not be utilized is equal to $P_x^{N_a+1}$. The average slot utilization for node x is approximated as

$$U_x \approx 1 - P_x^{N_a+1} \quad (6)$$

The total utilization, U_{avg} , of the network averaged over all the nodes is given below:

$$U_{avg} = 1 - \frac{\sum_{x=1}^N P_x^{N_a+1}}{N} \quad (7)$$

The values of P_x are estimated from the simulations. In Figure 8 we plot the slot utilization predicted by (7) along with the slot utilization obtained from the simulations for scenarios with low density.

8.3. Results with Gamma Packet Generation Model. For the remaining results we focus on end-to-end completion rate only. The results for other metrics follow the same trends

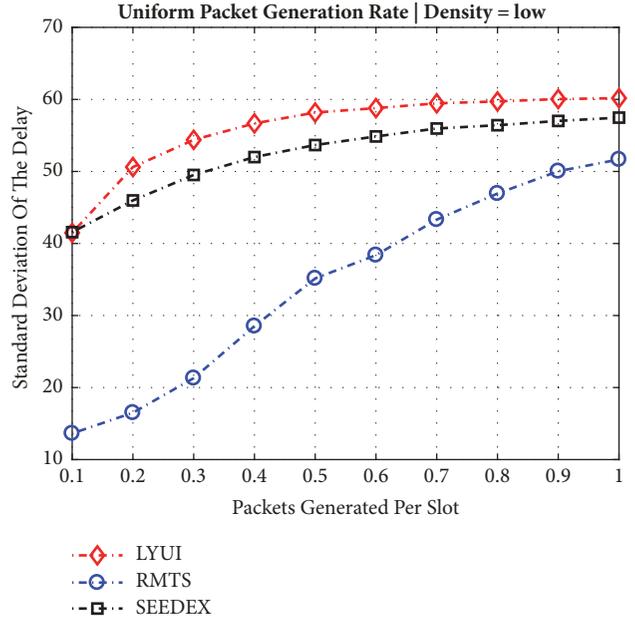


FIGURE 6: Standard deviation of delay for low density.

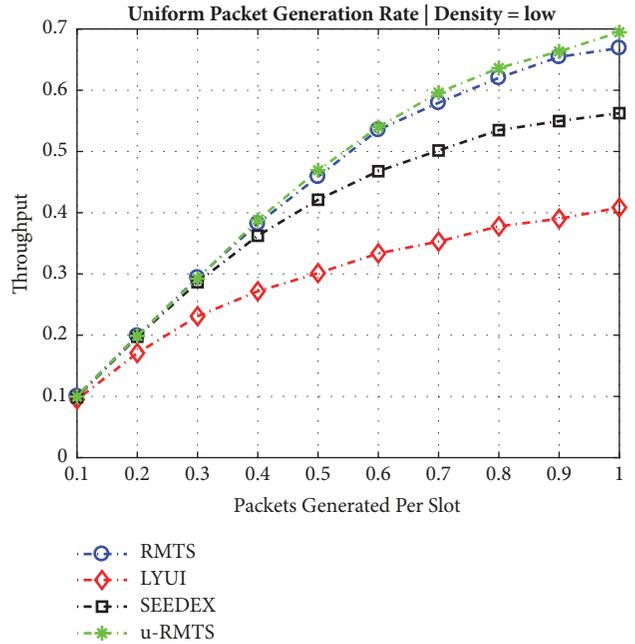


FIGURE 7: Throughput for low density.

and are excluded. Figures 9, 10, and 11 show the end-to-end completion rates for scenarios with the gamma packet generation model and the low node density. In these figures the solid lines represent the results with the uniform packet generation model and the dotted lines represent the results with the gamma packet generation model. We include the results with uniform packet generation model to show the drop off in the performance. Figure 9 shows the end-to-end completion rate for the scenario with the gammal packet generation model. The gammal model has the least variance

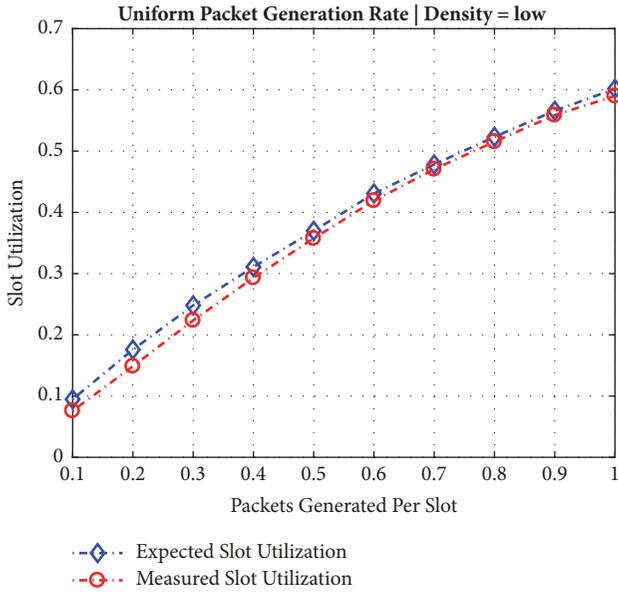


FIGURE 8: Slot utilization for low density.

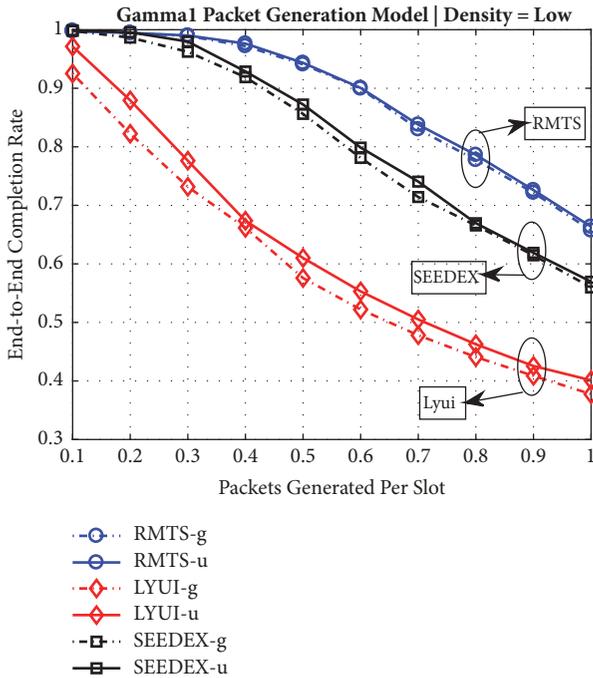


FIGURE 9: End-to-end completion rate for Gamma1 model.

and is very similar to the uniform model. Hence, we do not see any significant difference in the performance between the scenarios with those models.

Figure 10 shows the end-to-end completion rate for the scenario with the gamma2 packet generation model. It is evident that RMTS has the least drop off in the completion rate while SEEDEX and Lyui's protocol have significant reductions in completion rate. The RMTS protocol has about 144% increase in the value of Γ compared to SEEDEX. Lyui's

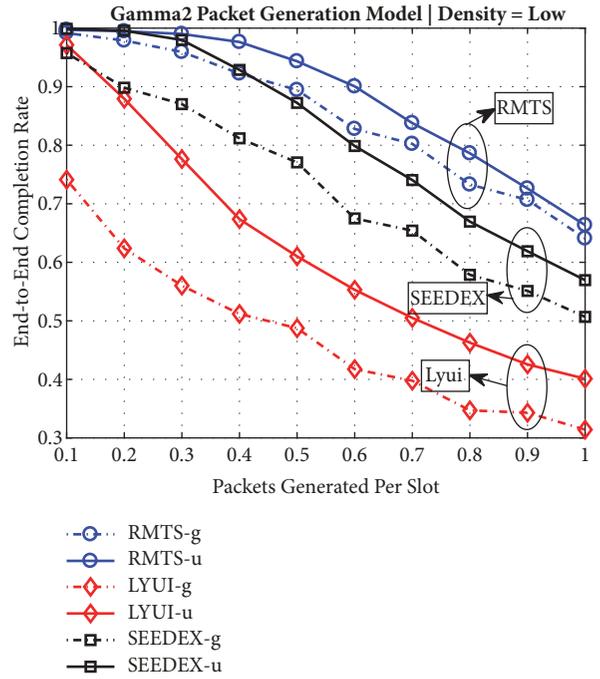


FIGURE 10: End-to-end completion rate for Gamma2 model.

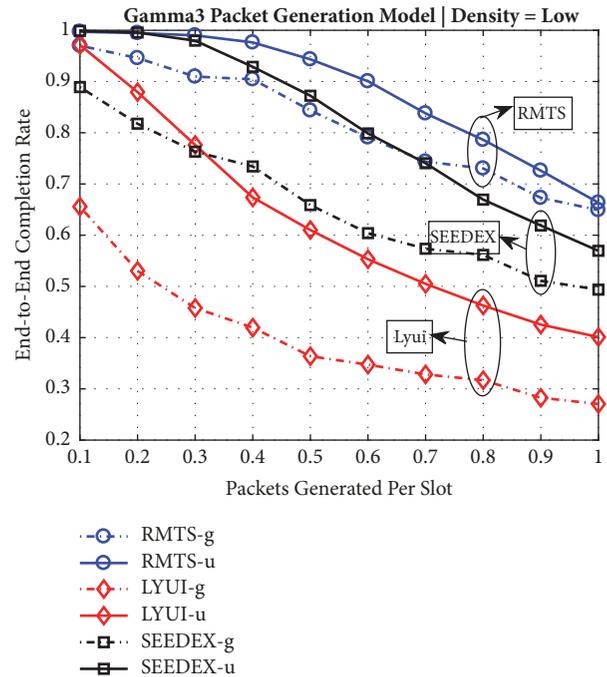


FIGURE 11: End-to-end completion rate for Gamma3 model.

protocol does not achieve an end-to-end completion rate of 90% for any of the packet generation rates used in the study.

Figure 11 shows the end-to-end completion rate for the scenario with the gamma3 packet generation model. It is again evident that RMTS has the least reduction in the completion rate. In fact RMTS is able to deliver at least 90% of the packets for packet generation rates up to 0.4 packets

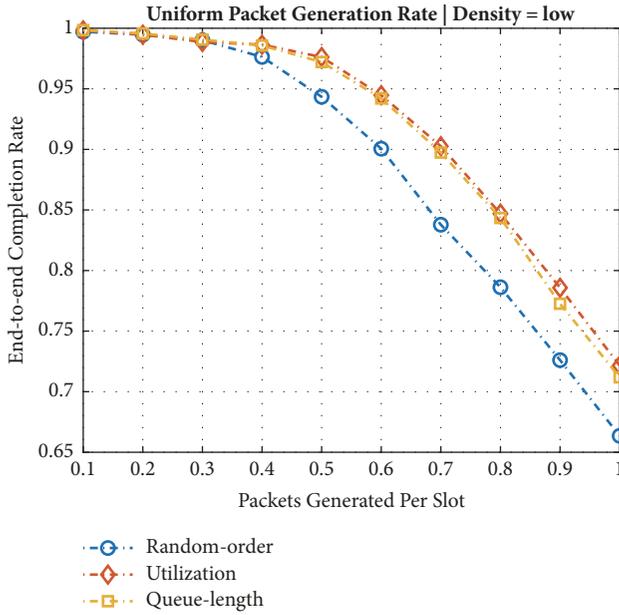


FIGURE 12: Effect of ordering the aux nodes for low density.

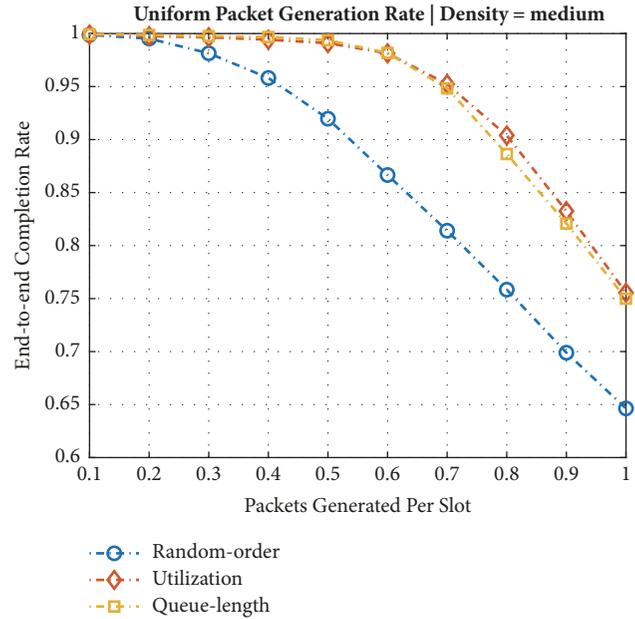


FIGURE 13: Effect of ordering the aux nodes for medium density.

per slot. Both SEEDX and Lyui’s protocols fail to achieve an end-to-end completion rate of 90% for any packet generation rates simulated.

8.4. Ordering the Auxiliary Nodes. In the section we look at the result of ordering the auxiliary nodes according to different priorities. Once again the focus is on the end-to-end completion rate. Figure 12 focuses on the end-to-end completion rates for networks with low density that utilize the uniform packet generation model. It shows the end-to-end completion rates for RMTS with different auxiliary node orderings. Both the queue-length and the utilization metrics result in a similar performance and we can achieve a maximum of about 17% improvement in the value of Γ over random-order. In networks with low density the number of eligible 1-neighbors from which a node can pick its auxiliary nodes is usually less than N_a . Hence, reordering them does not produce a significant improvement in the performance.

Figure 13 focuses on the end-to-end completion rates for networks with medium density. Once again both queue-length and utilization metrics results in a similar performance and we can achieve a maximum of about 50% improvement in the value of Γ over random-order. Unlike the previous case there is a significant improvement in the performance. In this scenario every node in the network has plenty of eligible nodes from which it can choose the auxiliary nodes. By choosing these nodes based on the metrics we can further increase the slot utilization resulting in better network performance.

Figure 14 focuses on the end-to-end completion rates for networks with high density. In this scenario queue size once again results in better network performance. Using utilization on the other hand leads to poor performance. In the networks with dense scenarios the frame size is greater than or equal to the number of nodes in the network. A large frame size

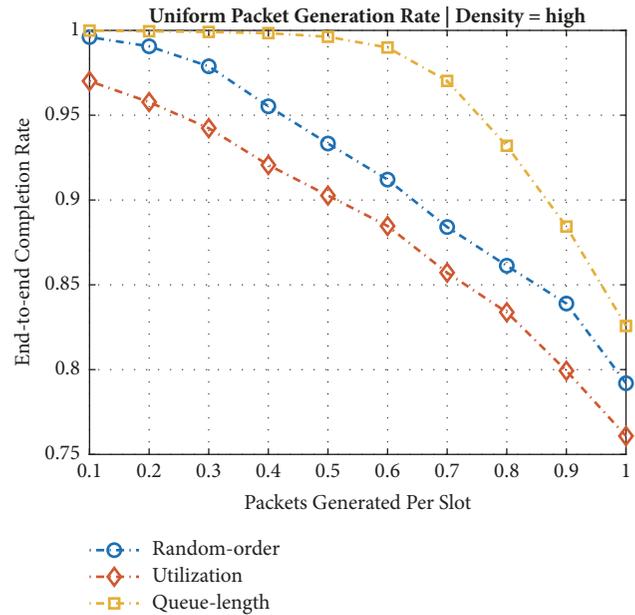


FIGURE 14: Effect of ordering the aux nodes for high density.

leads to all the nodes having a similar utilization and hence ordering the nodes based on this is not effective. We can achieve a maximum of about 35% improvement in the value of Γ over random-order.

9. Conclusion and Future Work

We present a new approach to transmission scheduling which is built upon salvaging the unutilized slot assignments. In most of the traditional scheduling algorithms the slot assignments are fixed and are independent of the traffic levels

at those nodes resulting in unutilized slots. In the proposed scheme we do have fixed slot assignments. However, in order to remedy the shortcomings of the fixed slot assignments auxiliary slot assignments are introduced. When a node to which a slot is assigned does not transmit one of its close 1-neighbors may transmit instead if it will produce a similar level of multiple-access interference. Essentially a slot is assigned to a group of nodes and the slot is not utilized only when all of those nodes do not have a packet to transmit. This results in a significant increase in slot utilization in many scenarios.

The extensive simulation results show that a significant improvement in the network performance can be achieved with RMTS. Also, RMTS retains the benefits of traditional scheduling algorithms such as those described in [8, 9, 12, 13, 15]. All these schemes are fair and robust and can guarantee some QoS across varying network topologies, densities, and traffic models.

With ever-increasing number of radios and emerging technologies like *internet of things* there is an increasing interest in dense networks. RMTS in its current form has good performance in dense networks. However, it was designed for multihop networks. Future work will consider variations of RMTS that would maximize reuse opportunities for very dense networks.

Data Availability

The underlying data generated from the simulations can be made available upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the Army Research Laboratory [Award no. W911NF-17-1-0244]. Additional support was provided by a gift from ARRIS.

References

- [1] N. Abramson, "The aloha system: another alternative for computer communications," in *Proceedings of the November 17-19, 1970, fall joint computer conference*, pp. 281–285, 1970.
- [2] J. Zander, "Slotted ALOHA multihop packet radio networks with directional antennas," *IEEE Electronics Letters*, vol. 26, no. 25, pp. 2098–2100, 1990.
- [3] P. Karn, "MACA—a new channel access method for packet radio," in *Proceedings of the ARRL/CRRL Amateur radio 9th computer networking conference*, vol. 140, pp. 134–140, 1990.
- [4] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: a media access protocol for wireless LAN's," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4, pp. 212–225, 1994.
- [5] IEEE 802.11 Working Group, *Wireless lan medium access control (mac) and physical layer (phy) specifications*, 1997.
- [6] V. B. Subrahmanya and H. B. Russell, "Recovering the reserved transmission time slots in an ad hoc network with scheduled channel access," in *Proceedings of the 35th IEEE Military Communications Conference, MILCOM 2016*, pp. 67–72, November 2016.
- [7] S. Kumar, V. S. Raghavan, and J. Deng, "Medium access control protocols for ad hoc wireless networks: a survey," *Ad Hoc Networks*, vol. 4, no. 3, pp. 326–358, 2006.
- [8] A. Ephremides and T. V. Truong, "Scheduling Broadcasts in Multihop Radio Networks," *IEEE Transactions on Communications*, vol. 38, no. 4, pp. 456–460, 1990.
- [9] S. Ramanathan, "A unified framework and algorithm for channel assignment in wireless networks," *Wireless Networks*, vol. 5, no. 2, pp. 81–94, 1999.
- [10] I. Rhee, A. Warriar, J. Min, and L. Xu, "Drand: distributed randomized tdma scheduling for wireless ad-hoc networks," in *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pp. 190–201, 2006.
- [11] W. P. Lyui, *Design of a new operational structure for mobile radio networks [Ph.D. dissertation]*, Clemson University, 1991.
- [12] J. L. Hammond and H. B. Russell, "Properties of a transmission assignment algorithm for multiple-hop packet radio networks," *IEEE Transactions on Wireless Communications*, vol. 3, no. 4, pp. 1048–1052, 2004.
- [13] C. D. Young, "USAP: a unifying dynamic distributed multi-channel TDMA slot assignment protocol," in *Proceedings of the Military Communications Conference (MILCOM '96)*, pp. 235–239, Mclean, Va, USA, October 1996.
- [14] J. Kim, J. Cha, and H. Park, "New delay-efficient TDMA-based distributed schedule in wireless mesh networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, 2012.
- [15] C. Zhu and M. S. Corson, "A five-phase reservation protocol (fprp) for mobile ad hoc networks," *Wireless Networks*, vol. 7, no. 4, pp. 371–384, 2001.
- [16] A. Bhatia and R. Hansdah, "DSLRL: A Distributed Schedule Length Reduction Algorithm for WSNs," in *Proceedings of the 2015 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 365–374, Hyderabad, India, May 2015.
- [17] R. Rozovsky and P. R. Kumar, "SEDEX: A MAC Protocol for Ad Hoc Networks," in *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, vol. 7, pp. 67–75, 2001.
- [18] L. Kleinrock and M. O. Scholl, "Packet Switching in Radio Channels: New Conflict-Free Multiple Access Schemes," *IEEE Transactions on Communications*, vol. 28, no. 7, pp. 1015–1029, 1980.
- [19] C. Young and J. Stevens, "Clique activation multiple access (CAMA): a distributed heuristic for building wireless datagram networks," in *Proceedings of the IEEE Military Communications Conference. Proceedings. MILCOM 98*, pp. 228–232, Boston, MA, USA.
- [20] R. Sunlin, "A hybrid distributed slot assignment TDMA channel access protocol," in *Proceedings of the IEEE Conference on Military Communications*, pp. 934–938, Monterey, CA, USA.
- [21] A. Ebner, H. Rohling, M. Lott, and R. Halfmann, "Decentralized slot synchronization in highly dynamic ad hoc networks," in *Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communications, WPMC 2002*, pp. 494–498, October 2002.

- [22] B. J. Wolf and H. B. Russell, "Immediate neighbor scheduling (INS): An adaptive protocol for mobile ad hoc networks using direct-sequence spread-spectrum modulation," *Ad Hoc Networks*, vol. 9, no. 3, pp. 453–467, 2011.
- [23] T. S. Rappaport, *Wireless Communications: Principles And Practice*, vol. 2, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1996.
- [24] B. Wolf, J. Hammond, D. Noneaker, and H. Russell, "Distributed formation of broadcast transmission schedules for mobile ad hoc networks," in *Proceedings of the IEEE MILCOM 2004. Military Communications Conference, 2004.*, pp. 1268–1274, Monterey, CA, USA.
- [25] P. K. Appani, J. L. Hammond, D. L. Noneaker, and H. B. Russell, "An adaptive transmission-scheduling protocol for mobile ad hoc networks," *Ad Hoc Networks*, vol. 5, no. 2, pp. 254–271, 2007.
- [26] P. R. Ostergard, "A fast algorithm for the maximum clique problem," *Discrete Applied Mathematics*, vol. 120, no. 1–3, pp. 197–207, 2002.
- [27] R. Carraghan and P. M. Pardalos, "An exact algorithm for the maximum clique problem," *Operations Research Letters*, vol. 9, no. 6, pp. 375–382, 1990.
- [28] Palmetto, "Clemson university's high-performance computing (HPC)," https://www.palmetto.clemson.edu/palmetto/userguide_palmetto_overview.html.
- [29] V. B. Subrahmanya, *A protocol to recover the unused time slot assignments in transmission scheduling protocols for channel access in ad hoc networks [Master's thesis]*, Clemson University, 2016.



Hindawi

Submit your manuscripts at
www.hindawi.com

