High Spatial-Reuse Distributed Slot Assignment Protocol for Wireless Ad Hoc Networks

Muhammad Hafeez Chaudhary, and Bart Scheers
Royal Military Academy Belgium
{mh.chaudhary, bart.scheers}@rma.ac.be

***Abstract*—Application of ad hoc networks in mission-critical environments requires wireless connectivity that meets certain quality-of-service (QoS). In such networks mechanism to control access to the shared wireless channel is crucial to ensure efficient channel utilization and to provide the QoS. TDMA based MAC protocols are considered to be appropriate for this kind of applications; however, finding an efficient and distributed slot assignment protocol is crucial. In this paper, a distributed slot assignment protocol is developed which gives high spatial reuse of the channel. To assign slots, the protocol does not need global topology information: Each node assigns slots based on the local topology information. The protocol can find the conflict-free slot assignment with limited message overhead. We evaluate the performance of the proposed protocol and show that the protocol gives considerably better channel utilization efficiency than exiting distributed slot assignment protocols.**

I. Introduction

The last decade has seen an explosive growth in applications of wireless communications and networking technology bringing ubiquitous mobile service into the everyday realm. A recent forecast by CISCO suggests that during the current year there will be more wirelessly connected devices than the total human population. The increase in wireless devices is expected to continue with increase in personal/home gadgets and sensors/pervasive computing devises. This would further increase the density of the wireless devices. Moreover, the demand for high-speed wireless data transfer is increasing at astronomical rates. The phenomenal increase in the wirelessly connected devices and the applications running on them have put an extremely high premium on the communications spectrum, and thus placing great demand on designing spectrum efficient communication and networking protocols to meet the requirements of the current and emerging applications in wireless networking. Currently, a key area of research is *mobile ad hoc networking*, which is driven by the requirement of having a technology that enables a disparate set of mobile devices/nodes create a network on demand, as the need arises, to accomplish an assigned mission.

In a wireless network, simultaneous transmissions of two or more nodes in the same channel may not be successful if their intended receivers are in the radio interference range of more than one transmitter. A mechanism to control access to the shared wireless channel is crucial to ensure efficient channel utilization and to provide *quality-of-service* (QoS). Ad hoc networks for mission-critical applications and emergency response services require that the data be delivered to the destination node reliably and within certain time limits. To support such QoS, schedule based medium access control protocols using TDMA scheme are deemed suitable [1].

There are numerous algorithms dealing with TDMA slot scheduling in ad hoc networks, whereby nodes can find conflict-free slot assignment. In [1], a TDMA slot scheduling protocol named GinMAC is presented; the protocol requires the network be arranged in a tree-like structure. Building on the GinMAC, in [2] the BurstProbe slot assignment protocol is proposed. These slot scheduling protocols require global topology information which may not be available in ad hoc networks that are inherently bereft of any central coordinating and controlling infrastructure. In [3], [4] a slot assignment protocol called USAP is proposed which allows nodes to get conflict-free slot assignment in a distributed way using local topology information. Kanzaki and his colleagues proposed a protocol named ASAP in [5], [6] which can be viewed as an extension of the USAP, adding details on dynamic frame-size selection and more detailed procedures about the nodes joining/leaving the network. Another related protocol named DRAND is proposed in [7], and later on extended to Z-MAC [8] which combines artifacts of both TDMA and CSMA medium access schemes. The main focus of these protocols (and others like in [9], [10]) is to find conflict-free slot assignment to nodes in a distributed way. The maximization of the channel utilization efficiency (i.e., the spatial reuse of the slots) is not explicitly considered. Thus from channel utilization viewpoint, these protocols may give suboptimal performance.

In this paper, we propose a *high spatial-reuse distributed slot assignment protocol* (HUDSAP). In the protocol, the nodes find slot assignment using their local topology information. The protocol introduces a priority mechanism by which nodes having higher *number of one-hop neighbors* (NoNs) assign slots first. Each node computes its priority index independently using only the information form the nodes within its contention zone. We show that the protocol achieves substantially higher channel utilization efficiency than the DRAND and related protocols.

The remainder of the paper is organized as follows: Section II gives preliminaries on slot assignment and the problem formulation; Section III presents details of the proposed slot assignment protocol; Section IV outlines an adaptive frame length selection scheme; Section V illustrates the performance of the protocol by simulation examples; and finally Section VI gives some concluding remarks.

II. Preliminaries and Problem Formulation

For the slot assignment we represent the network by a graph $G = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of vertices that correspond to the nodes in the network and $\mathcal{E}$ is the set of edges representing the wireless links between the nodes. We assume that for any two distinct vertices $i, j \in \mathcal{N}$, an edge (i, j) exits in $\mathcal{E}$ if and only if i and j can hear each other—that is, all edges are bidirectional.

Given a graphical representation of the network, the TDMA slot assignment problem with the associated assignment constraints can be defined as an equivalent graph coloring optimization problem. The equivalence between the two problems is *one-to-one*; that is, two nodes receive different slots if and only if the corresponding vertices have received different colors. Moreover, the total number of slots is equal to the total number of colors used.

The objective of the slot assignment optimization problem is to minimize the number of slots used to find the transmission schedule for all nodes in the network under the constraint that a node can only assign a slot that has not been used within the contention zone of the node. The contention zone of a node is assumed to be limited to its two-hop neighbors. As is often assumed in designing MAC protocols, such a definition of the contention zone is imposed to remove the *hidden-terminal* problem. The hidden-terminal problem arises when two nodes cannot hear transmissions of each other but a third node can hear transmissions of both of them.

An optimal solution for slot scheduling problem is known to be *NP-complete* [11], [12]: That is, the computational complexity of finding the solution increases exponentially as the number of nodes increases; which means it becomes prohibitively time consuming to find the optimal slot schedule. That is why to solve the slot assignment problem , in the literature, heuristic-based suboptimal solutions are proposed that vary in their schedule length, the convergence time, and the message overhead. To this end, in [13], three *greedy* heuristic-based slot assignment procedures are proposed: namely, the RAND (random), the MNF (minimum neighbor first), and the PMNF (progressive minimum neighbor first), listed in increasing order of complexity. The basic principle underlying each of these schemes is essentially the following: first, give a unique label to each node, and then assign slots to nodes in decreasing order of their labels. In RAND, the nodes are labeled in a random way; in MNF, the node with minimum number of neighbors is labeled first; and in PMNF, the nodes are labeled as in MNF with a difference that after labeling a node, the node and its edges are removed. Effectively that means, at each step among the nodes that have not been assigned slot yet, the RAND takes a node at random and allots time slot to it; the MNF takes the node with maximum number of neighbors and allots slot to it; and the PMNF first removes the nodes and the associated edges that have already been assigned slots, then within the updated network assign slot to the node with maximum number of neighbors.

It has been shown in [13] that the *schedule length* (SL) achieved with the three schemes is in the order $SL_{\text{PMNF}} \leq SL_{\text{MNF}} \leq SL_{\text{RAND}}$. That is, from the spatial reuse point of view, the PMNF is the most efficient and the RAND the least efficient among the three labeling schemes. However, the problem with these schemes is that they require knowledge of the global network topology; that is, they are centralized schemes. For ad hoc networks, distributed slot assignment schemes are sought because such networks are devoid of any central coordinating and controlling infrastructure and the global topology knowledge is hard to come by at individual nodes.

Recently a distributed implementation of RAND, known as DRAND, is proposed in [7] which can achieve the same channel utilization efficiency as the RAND. The RAND ordering can be viewed as archetype of the channel scheduling in wireless networks. The roots of many heuristics for slot scheduling algorithms in the literature can be found to be equivalent to the RAND [3], [5], [7], [8], [14]. The appeal of RAND lies in its simplicity and the ease with which its distributed version can be implemented. However, as shown in [13], the performance (in terms of SL) of the RAND can be significantly inferior to the MNF and PMNF. In this work, we present a distributed implementation of MNF, which we call as the HUDSAP, that can achieve same channel utilization efficiency as the MNF by using only the local topology knowledge at individual nodes. Towards this end, the ensuing section gives details of the proposed protocol.

III. Slot Assignment Protocol

We assume that the time is divided into slots and the nodes are synchronized on the slot boundaries. The protocol operates in two main phases: *neighborhood discovery phase* and *slot assignment phase*. Fig. 1 shows the state diagram of the protocol, where the slot assignments phase is divided into four states: *node classification*, *waiting slot assignment*, *active slot assignment*, and *completed slot assignment*. In the neighborhood discovery phase, a node collects information about the nodes within its two-hop neighborhood, that is, one-hop neighbors (ONs) and two-hop neighbors (TNs)[1], the NoNs of these nodes, and their assigned slots. Based on that, the node constructs *neighbor information table* (NiT), an example of which is shown in Fig. 2. In the slot assignment phase, the nodes assign slots in a distributed way as explained next.

Each node compares its NoNs with the NoNs of the nodes, in its NiT, which have not yet assigned slots, and based on that, the node classifies itself in one of the following three groups:

1) In *node group I* (NG-I) if the NoNs of the node is greater than the NoNs of all nodes in its NiT that have not yet assigned slots; for instance, in Fig. 2 initially nodes e and p will place themselves in this group.
2) In NG-II if the NoNs of the node is equal to the NoNs of some (one or more) nodes in its NiT that have not yet assigned slots; for example, in Fig. 2, node b and l will initially place themselves in this node group.
3) In NG-III if the NoNs of the node is less than the NoNs of some (one or more) nodes in its NiT that have not

[1]The two-hop neighbors are strict two-hop neighbors, that is, it excludes the one-hop neighbors that can also be reached by another one-hop node.

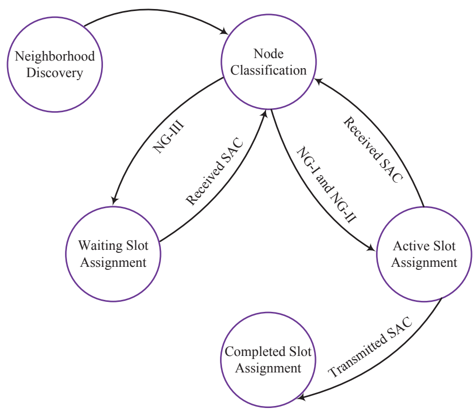
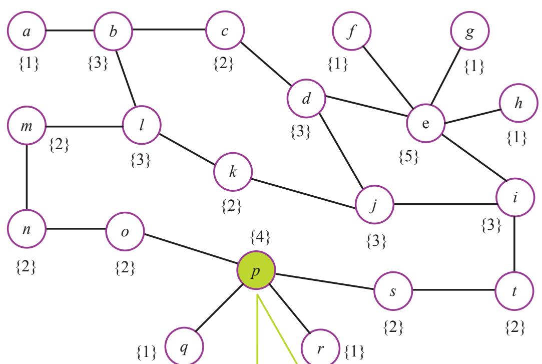


Fig. 1: State transition diagram of the HUDSAP.



Own

Node ID	Assigned Slots / Frame Length	NoNs
p	- / -	4

One-Hop Neighbors

Node ID	Assigned Slots / Frame Length	NoNs
o	- / -	2
q	- / -	1
r	- / -	1
s	- / -	2

Two-hop Neighbors

Node ID	Assigned Slots / Frame Length	NoNs
t	- / -	2
n	- / -	2

Fig. 2: Network topology with NoNs of each node within $\{.\}$. An example of NiT of node p is also shown.

yet assigned slots; for instance, in Fig. 2, all nodes will initially place themselves in this node group except nodes b, e, l, and p.

The classification at each node is done independently solely based on the local topology information available at the node in the form of NiT. After the classification, the node assigns slot to itself and the procedure of which depends on its node group. The slot assignment is managed differently in the three groups, as shall be explained in the ensuing sections. During the slot assignment phase, following control messages are exchanged between the nodes: *slot assignment request* (SAR), *slot assignment grant* (SAG), *slot assignment confirmation* (SAC), *slot assignment denial* (SAD), and *slot assignment failure* (SAF). The SAR, SAF, and SAC messages are transmitted by the node that attempts to assign a slot, and the SAG and SAD are response messages to the SAR message. These response messages are transmitted by the nodes within the contention area of the node that has sent out the SAR message.

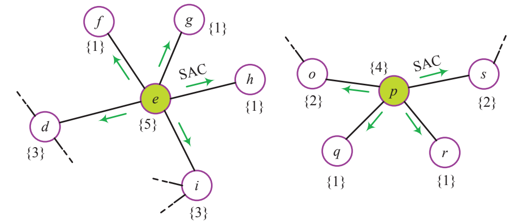


Fig. 3: Successful slot assignment to node e and p in NG-I.

A. Slot Assignment in NG-I

All nodes in this group assign the first free slot that has not yet been assigned to any node within their two-hop neighborhood. For the network of Fig. 2, initially the nodes e and p are in this group and there is no slot assigned to any node in the network; so both of these nodes assign slot 0 to themselves. After assigning the slot, the nodes announce their slot assignment to their neighbors by sending the SAC message as shown in Fig. 3. In this case the nodes do not have to wait for any confirmation from their one-hop and two-hop neighbors about their slot assignment, because the nodes in this group are sure that there is no other node within their contention area currently assigning slot until their slot assignment is complete; the other nodes in the contention area are prohibited from initializing slot assignment by the slot assignment procedures for NG-II and NG-III, as we shall see in the ensuing sections. Each one-hop neighbor after receiving the SAC announcement does the following things: updates its slot assignment information, forwards the SAC to its ONs, removes the node which transmitted the announcement from its NiT for further consideration during the classification step, and changes its state accordingly as shown in Fig. 1. It is interesting to note that all nodes in this group can complete slot assignment in one time-slot, as no slot assignment permission is required from neighbors.

B. Slot Assignment in NG-II

For slot assignment in NG-II, we propose two procedures: one based on an elaborate exchange of control messages, and the other based on a prioritization mechanism using node IDs.

1) Message Exchange Based Slot Assignment: Each node in this group has at least one or more nodes (that have not assigned slots yet) in its NiT with the same number of NoNs as

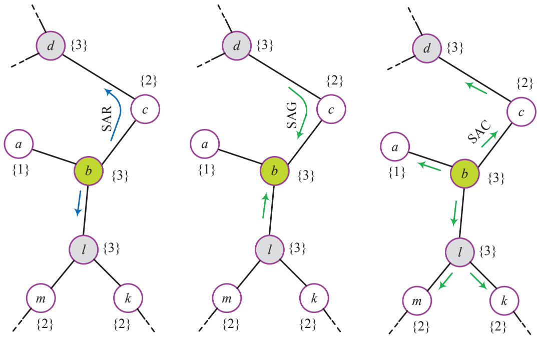


Fig. 4: Successful slot assignment to node b in NG-II.

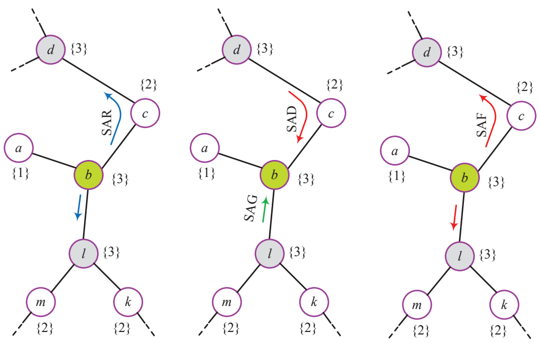


Fig. 5: Failed slot assignment to node b in NG-II.

that node itself. For node i, in this group, let N_i be the number of nodes with the same NoNs[2] as the node i. To assign slot, the node runs a lottery for which the probability of success is chosen as $P_a^{(i)} = 1/(N_i + 1)$. In case a node does not win the lottery, it will wait for certain time slots chosen at random before trying the lottery again. If the node wins the lottery, it assigns the minimum possible slot that has not been assigned to any node within its contention-zone and sends the SAR message to its neighbors. For the given network topology in Fig. 2, initially nodes b and l are in this group and the probability of winning the lottery for each of them is 1/3. The probability that only one of them wins the lottery in a given slot and thus avoid collision of their announcement messages is 2/9.

For the slot assignment to be complete, the node has to wait for the SAG messages from nodes within its contention area. After the slot assignment is granted by the nodes in the contention area, the node i sends out a SAC message; an example of which is shown in Fig. 4. Similar to the NG-I, after receiving the confirmation message, the one- and two-hop nodes update their slot assignment information, and remove node i from their NiTs for further consideration in the node classification step. Interestingly, the node i does not have to wait for response messages from nodes that have NoNs different than the node itself as well as from nodes having same NoNs that have completed slot assignment. For example, if node i has only one node with the same NoNs, then a confirmation from only that node is needed: if that node is a one-hop neighbor then the slot assignment can be completed in three slots—one to send SAR, second to receive SAG, and third to send SAC; if that node is a two-hop neighbor, then can be done in five slots—two additional slots are used by the one-hop node to relay SAR/SAG messages; and in either case, a unicast addressing can be employed. This can reduce the traffic overhead substantially, and thereby reduces the chances of collision of the slot assignment control messages in the network.

[2]Throughout this section, when we refer to 'node(s) with same NoNs' it means node(s) having same NoNs, as the reference node, that has(have) not yet completed slot assignment.

When a node receives a SAR message of a one-hop node, the receiving node performs one of the following tasks:

1) If the receiving node and all of its ONs have NoNs which are different than the NoNs of the SAR transmitter, then the receiving node neither forwards the SAR to its ONs nor sends a reply message (i.e., SAG/SAD). The receiving node simply waits for the SAC/SAF message form the transmitter of SAR.
2) In case the receiving node has same NoNs as the transmitter of SAR but the NoNs of all of its ONs are different, the receiver node sends a SAG message to the transmitter. In this case also, the receiver does not forward the SAR to its ONs.
3) If the NoNs of the receiving node are different than the NoNs of the transmitter of the SAR message, but one (or more) of its ON(s) has (have) same NoNs, the receiver forwards the SAR to those ONs. In this case, the receiver has to relay back any reply message (i.e., SAG/SAD) from its ONs: The receiver node sends SAG message if all of its ONs with same NoNs reply with SAG, and otherwise sends SAD message.
4) If the receiving node and one (or more) of its on-hop nodes have same NoNs as the transmitter of the SAR message, the receiver node sends a SAD message to the transmitter if it has already sent a SAG for this particular slot to one of its ONs other than the transmitter of recently received SAR. Otherwise, the receiver first forwards the SAR to its ONs with the same NoNs as the transmitter of SAR and waits for their reply. Once the receiver receives responses of those nodes, it fuses them and replies with a SAG message if all those nodes send SAG, else it replies with a SAD message.

When a node receives a forwarded SAR message of a two-hop node, the receiving node performs one of the following tasks:

1) If the NoNs of the receiver are different than the NoNs of the originator of the SAR message, the receiver discards the SAR message and does nothing else.
2) In case the NoNs of the receiver are same as the NoNs of the originator of the SAR message, the receiver replies with a SAD message if it has already sent its own SAR message to its neighbors for the same slot, otherwise it replies with a SAG message.

When a node receives a SAF message originated from a one-hop node it executes one of the following tasks:

1) If the NoNs of the receiver and all of its ONs are different

than the transmitter of the SAF message, the receiver discards the message.
2) In case the NoNs of the receiver are different than the NoNs of the transmitter but some (one or more) of its ONs have same NoNs as the transmitter, then the receiver forwards the SAF to those ONs.
3) If the NoNs of the receiver are same as the transmitter, but the NoNs of all of its ONs are different than the transmitter, then the receiver frees the slot for which it has earlier sent the SAG message to the transmitter of SAF. After which, the receiver discards the SAF.
4) In case the receiver and some (one or more) of its ONs have same NoNs as the transmitter of SAF, the receiver releases the slot and forwards the SAF to those ONs having same NoNs.

When a node receives SAF originated from a two-hop node, the receiver discards the SAF if its NoNs are different than the transmitter of the SAF, else it releases the slot related to the SAF.

When a node receives a SAC message it updates its NiT; if the SAC is originated from a one-hop node, then the receiver also forwards the SAC to its ONs.

Remark 1: The slot assignment in NG-II by the preceding dialog based mechanism though requires message exchange between a subset of nodes in the contention area of a node, however, despite this the control message overhead could be high. For instance, if multiple nodes in a contention area try to assign slots at the same time, then none of them will succeed and they have to try again after a random back-off time, which would slow down the convergence of the algorithm. In this regard, next we present an alternative slot assignment procedure for NG-II which avoids the preceding detailed message exchange routine when assigning slots to the nodes.

2) Alternative Priority Based Slot Assignment Procedure: When all nodes have unique IDs, the slot assignment in NG-II can be handled in a much simpler way, very much like in the NG-I. Let there be a one-to-one function Ψ_i which could map a node ID i to a unique numeric number μ_i, that is,

$$\Psi_i : i \mapsto \mu_i, \qquad \mu_i \neq \mu_j, \qquad \forall i \neq j, \qquad i, j \in \mathcal{N}. \quad (1)$$

Now any node i in NG-II, instead of randomly deciding about when to initiate slot assignment, compares its ID μ_i with the IDs of the nodes (that have not yet assigned slots) having same NoNs in its NiT. If its ID is greater than these nodes, it assigns the minimum possible slot(s) to itself which is(are) not yet taken by the nodes in its contention area; otherwise, the node does not try to assign slot(s) unless the preceding condition is true. After assigning the slot(s), the node sends out the SAC message to neighboring nodes. Note that, like the nodes in NG-I, for slot assignment by this procedure the nodes in NG-II are not required to exchange messages SAR/SAG/SAD/SAF.

Remark 2: The control message overhead of the alternative priority based slot assignment procedure is quite low compared to the message exchange based procedure. However, under the alternative slot assignment procedure, in certain cases, some nodes in NG-II may have to wait inordinate amount

Fig. 6: For linear network topology all nodes except the two nodes at the extremities are in NG-II. When the number of nodes is large and the nodes in the network are in increasing (or decreasing) order of their IDs, for slot assignment, the nodes on the right (left) edge have to wait until all nodes on their left (right) side have completed slot assignment.

of time for slot assignment, for example, as shown in Fig. 6, the nodes in NG-II on the right (left) hand side have to wait until all other nodes in the group have completed their slot assignment. In such scenarios where a node in NG-II has to wait excessively long time to initiate slot assignment, the message exchange based slot assignment mechanism could be employed. To be more specific, if a node in NG-II does not receive new slot assignment information, during a predefined time duration, about a node within its contention area having the same NoNs as the given node then the node initiates slot assignment according to the message exchange based procedure. In this regard, to avoid more than two nodes to initiate slot assignment at the same time after expiration of the predefined time duration, the nodes employ random back-off mechanism whereby a node waits for randomly chosen time-slots before starting slot allocation. If during this time, the node receives new slot assignment information, it will abort the message exchange based slot assignment procedure and revert to the priority based slot assignment procedure.

C. Slot Assignment in NG-III

The nodes in this group do not try to assign slots to themselves. They listen to the messages from their neighbors, assist in slot assignment of their neighbors by forwarding slot assignment control messages as discussed in the preceding sections, and update their slot assignment information. When a node, in this group, receives a SAC message about any node in its contention area, it removes that node from its NiT for further consideration in the node classification step.

Each node, in NG-II and NG-III, that has not yet assigned a slot to itself, whenever updates its slot assignment information and removes any node from consideration in its NiT, it reruns the node classification test shown in Fig. 1. Note that after the classification test, a node in NG-II that has not yet assigned a slot to itself may find itself in NG-I, and a node in NG-III may find itself in either NG-II or NG-I (cannot be in both because the groups are mutually exclusive). For example, when nodes e and p (which were in NG-I) finish slot assignment, their neighbors d, j, i, and o (which were in NG-III) reclassify themselves in NG-II. Each node handles the slot assignment according to the procedure specific to its current node group. It is interesting to note that the nodes can only upgrade their groups and thus the slot assignment protocol is bound to converge within limited time; that is, the slot assignment to all nodes will be completed within a bounded time. We will analyze the convergence time and the message complexity of the protocol in more details in our future work.

Proposition 1: The execution of the HUDSAP produces conflict-free slot assignment schedule.

Proof: To show that the slot assignment schedule produced by the HUDSAP is conflict-free, it suffices to note the following: 1) At any given time only nodes in NG-I and NG-II are assigning slots and the two groups are mutually exclusive; 2) By definition, all neighboring nodes of each node in NG-I do not assign slots until slot assignment is completed for the nodes in NG-I; and 3) Each node in NG-II assign slot which is not assigned to any other node in its neighborhood by exchanging the control messages, or by the prioritization mechanism. □

IV. Frame Length Selection

With uniform frame size across the network, although the design and implementation of the MAC protocol will be simplified, however, the channel resources will remain under-utilized. In the conventional TDMA slot assignment protocols, frame length is fixed based on the maximum expected number of nodes in the network, for instance, to ensure that each node gets at least one slot [3]. These protocols show poor channel utilization as they must leave enough unused slots for new coming nodes. Another possibility is to dynamically set the frame length for each node which is equal to the *maximum slot number* (MSN) assigned within the network. However, this would require each node to know the MSN. The propagation of the MSN within the entire network would not be adaptive to the local slot assignment changes—any change in the slot assignment may change the MSN and the new value has to be propagated throughout the network. Although by setting the network-wide same frame length based on the MSN effectively removes the requirement of a priori fixing the number of slots in a frame, however, it would still give lower channel utilization efficiency.

The channel utilization can be improved by variable frame length for each node depending on the slot assignment in its neighborhood, that is, to change the frame length dynamically according to the slot assignment to the nodes within its contention area. If the contention area of a node is limited to its two-hop neighborhood and reuse of slots is allowed outside this area, then each node can set its frame length which is a function of the MSN assigned within the contention area (instead of the entire network). Note that, the MSN allocated within the contention area cannot exceed the two-hop neighborhood size of the node. To have conflict-free transmission among nodes with different frame lengths, usually the lengths of frames are chosen as multiple of two [4]–[8],

To set the frame length L_i of node i according to the local MSN, the Z-MAC protocol in [8] proposed the following rule:

$$L_i = 2^{\kappa}, \tag{2}$$

where κ is a non-negative integer. The value of κ is selected such that the following holds

$$2^{\kappa-1} \leq F_i \leq 2^{\kappa} - 1, \tag{3}$$

where F_i is the MSN within the two-hope contention area of the node i. This scheme although could achieve better channel reuse than the uniform frame length rule across the entire network, however this is not optimal from the point of view of channel utilization efficiency. In this regard, we propose an alternative scheme by which local framing rule varies depending on the node connectivity. Specifically, we classify nodes in two groups: leaf nodes and non-leaf nodes. For leaf nodes[3] the frame length is selected as in the Z-MAC. For non-leaf nodes, the framing rule is modified as follows. Let $\tilde{F}_i$ be the MSN within the one-hop neighborhood of the non-leaf node i. The frame length is set as in (2) and (3) with F_i replaced by $\tilde{F}_i$. An example of the heterogeneous frame length (Z-MAC and proposed) and the uniform frame length across the network is given in Fig. 7. From the figure, we can observe increase in the channel reuse due to the variable length frame size. The increase in channel reuse directly translates into higher channel utilization efficiency as well decrease in the data transfer delays.

Once the nodes have decided their frame lengths, the information are exchanged with nodes within their respective contention areas. After which the nodes can start data transmission within their assigned time-slots. The Z-MAC framing

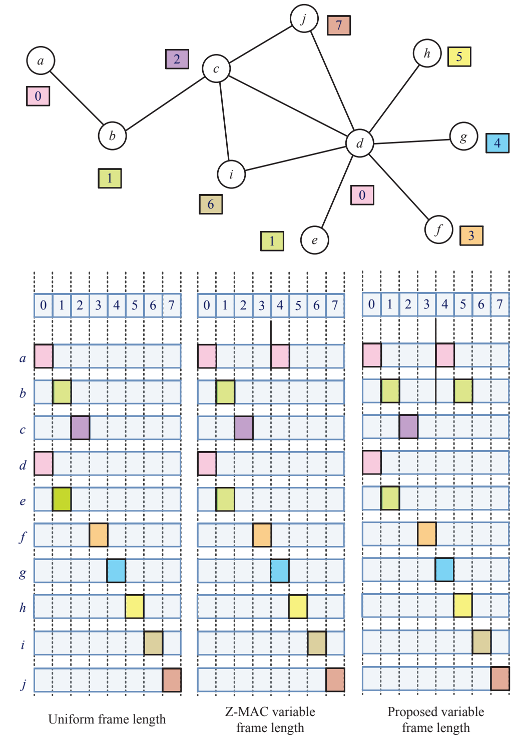


Fig. 7: An example of variable and fixed length TDMA frames for a given conflict-free slot assignment: In the case of uniform length frame, all nodes have 8-slot frame length; in the Z-MAC variable length frame strategy, node a has 4-slot frame length whereas all other nodes have 8-slot frame length; and in the HUDSAP variable length frame strategy, nodes a and b have 4-slot frame length whereas all other nodes have 8-slot frame length.

[3]A leaf node is a node which has only single one-hop neighbor; otherwise the node is a non-leaf node.

rule produces slot assignment which is always conflict-free. However, in the proposed framing scheme, there could be occasional conflicts in assigned slots. Such conflicts can be detected by the nodes once frame-length information is available. When a node detects a slot conflict due to frame size selection, it compares its assigned slot with the slot assigned to the node causing the conflict. If the slot of the given node is less than the conflicting node, then the given node selects its frame length according to the MSN within its two-hop area (instead of one-hop area); otherwise, it leaves the conflict resolution to the conflicting node which would use the same procedure to resolve the conflict.

V. Simulation Examples and Discussion

In this section, with some numerical experiments, we evaluate channel utilization efficiency (i.e., the spatial reuse of slots) of the proposed HUDSAP and compare it with the centralized schemes RAND and MNF of [13]. Note that the DRAND is proven to achieve the same channel utilization efficiency as RAND [7]; so the comparison of HUDSAP with DRAND is not performed here. For a given network, assuming uniform frame length, the channel utilization efficiency is measured in terms of the minimum number of slots used by the protocol to find the conflict-free slot assignment schedule. We deploy the nodes in a 400-by-400 planar region. We conduct two numerical experiments: In the first, we fix the transmission range of nodes to 40 and vary the number of nodes from 50 to 400; in the second, we fix the number of nodes to 300 and vary the transmission range from 10 to 50. Each point in the numerical results is obtained by averaging over 10^4 random deployments (according to uniform distribution) of the nodes in the region.

The results are plotted in Fig. 8 and Fig. 9. The figures show that the HUDSAP gives conflict-free slot assignment schedule that requires substantially less number of slots than the RAND (and consequently of the DRAND). That means, the spatial reuse of the slots, and consequently the channel utilization efficiency, is higher in HUDSAP. It should be noted that we deployed the node in a planar region of fixed area. Therefore, when either the number of nodes is small or the transmission range is small, or both, the performance gap is negligibly small. This is because, in such scenarios, the network is partitioned into small size sub-networks (each comprising a few nodes) that are disconnected from each other[4]. The performance of slot allocation protocols from spatial reuse point of view in such small networks does not differ much. However, if we impose the condition that the network is always connected, that is, any node can be reached from any other node (via multi-hops), then there would be a noticeable performance gap between the HUDSAP and the RAND even for a network comprising 50 sensors or less, which we observed in simulations that are not included here due to space constraints. Regarding the topological information, in DRAND each node requires knowledge about the identities of the nodes within its two-hop contention area and their slot assignment. Compared to that, in HUDSAP, each node also requires knowledge about the NoNs of the two-hop neighbors. Note that, although the NoNs of the one-hop neighbors is available in DRAND but that knowledge is not employed in slot scheduling.

[4]Here in simulations we do not impose the condition that the network is connected.

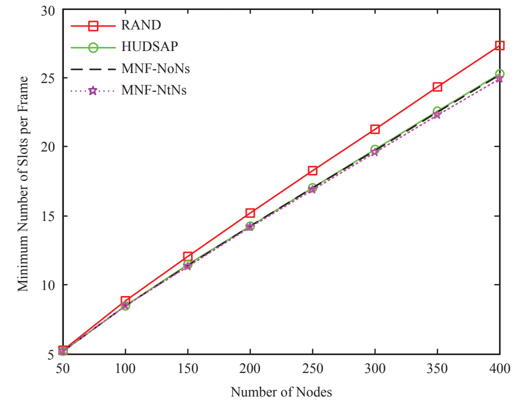


Fig. 8: Channel utilization efficiency comparison for fixed transmission range.

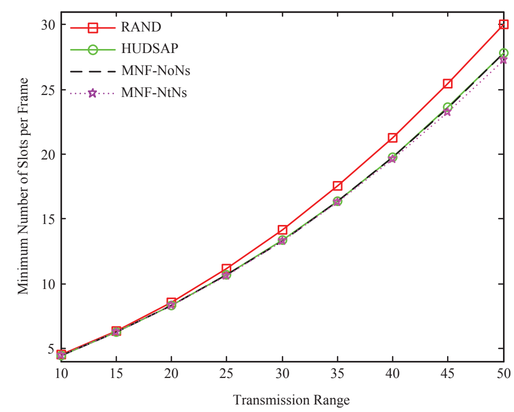


Fig. 9: Channel utilization efficiency comparison for fixed number of nodes.

For MNF two labeling schemes are considered, one based on the the NoNs and the other based on the *number of two-hop neighbors* (NtN) which are, respectively, denoted as MNF-NoNs and MNF-NtNs. The figures show that there is no appreciable gap between the schedule length of the HUDSAP and the MNF-NoNs. However, there is a marginal gap between the performance of the HUDSAP and the MNF-NtNs. The reason for this performance gap is that the HUDSAP is designed for ordering based on NoNs whereas the MNF-NtNs is based on the NtNs. Even though the performance gap is not substantial, it is straightforward to extend the HUDSAP where prioritization of slot assignment is based on the NtNs, in which case it is expected that the performance of the HUDSAP and the MNF-NtNs will converge. However, that will require each node to know the NtNs of all nodes within its contention area, which would entail additional protocol overhead. Given that there is a marginal gain in channel utilization efficiency, the additional overhead may not justify the gain. Besides, it should be noted that the HUDSAP is a distributed slot assignment protocol which relies on local topology knowledge at each node, whereas the MNF is a centralized protocol which requires global topology information.

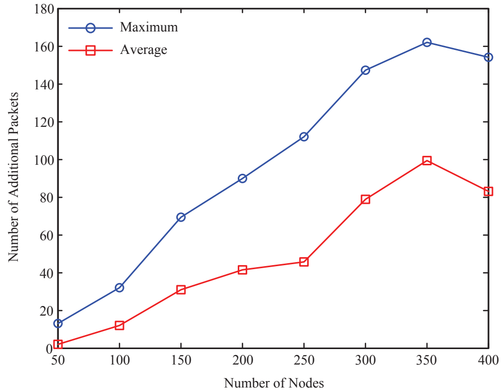


Fig. 10: Number of additional packets (maximum and average over deployments of the nodes) that can be transmitted within the frame length of Z-MAC under the proposed adaptive frame size selection scheme. Comparison is for fixed transmission range.

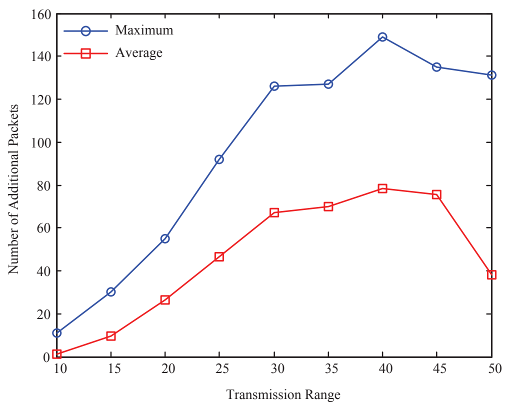


Fig. 11: Number of additional packets (maximum and average over deployments of the nodes) that can be transmitted within the frame length of Z-MAC under the proposed adaptive frame size selection scheme. Comparison is for fixed number of nodes.

Next we evaluate the impact of variable frame-length selection on the channel utilization efficiency. In this regard, for the preceding two network deployment scenarios, we compare the number of additional packets that can be transmitted within the frame length of Z-MAC when the frame-length is selected according to the proposed framing scheme. In Fig. 10 and Fig. 11 we plot the additional packets: average over 10^4 random deployments of the nodes and the maximum over the deployments. The figures show that with the proposed framing scheme, we could transmit more packets, which when seen for the network use over a considerably longer time window could translate into substantially higher throughput.

VI. Concluding Remarks

n this work, we proposed a distributed TDMA slot assignment protocol to schedule access of the wireless nodes to the shared wireless channel. The nodes, by this protocol, can find conflict-free slot assignment using only the local topology information. The proposed medium access scheduling protocol is suitable for wireless ad hoc networks for mission-critical applications and emergency response services which require wireless connectivity be provided that meet certain QoS requirements. We showed that the protocol can achieve higher channel utilization efficiency compared to the existing protocols like RAND and DRAND. We also proposed an adaptive frame size selection scheme which gives higher channel utilization than the framing scheme proposed in Z-MAC protocol.

In this work we give a qualitative description of our protocol and substantiate it with performance evaluation with a few numerical examples. In our planned ongoing work, to better understand the behavior of the protocol, we would do further analysis based on analytical modeling and event driven simulations. We plan to compare the convergence time and the control message overhead of the protocol. We also plan to study the behavior of the algorithm for dynamic topology changes due to the nodes joining and leaving the network, and the movement of the nodes within the network.

References

[1] P. Suriyachai, J. Brown, and U. Roedig, "Time-critical data delivery in wireless sensor networks," in *Proc. of 6th IEEE Int. Conf. on Distributed Computing in Sensor Systems (DOCSS'10)*, Santa Barbara, CA (USA), June 2010, pp. 216–229.

[2] J. Brown, B. McCarthy, U. Roedig, T. Voigt, and C. J. Sreenan, "Burstprobe: debugging time-critical data delivery in wireless sensor networks," in *Proc. of European Conference on Wireless Sensor Networks (EWSN'11)*, vol. 2, Bonn (Germany), Feb. 2011, pp. 195–210.

[3] C. D. Young, "USAP: A unifying dynamic distributed multichannel TDMA slot assignment protocol," in *Proc. of IEEE Military Commun. Conf. (MILCOM'96)*, vol. 1, McLean, VA (USA), Oct. 1996, pp. 235–239.

[4] ——, "USAP multiple access: Dynamic resource allocation for mobile multihop multichannel wireless networking," in *Proc. of IEEE Military Commun. Conf. (MILCOM'99)*, vol. 1, Atlantic City, NJ (USA), Nov. 1999, pp. 271–275.

[5] A. Kanzaki, T. Uemukai, T. Hara, and S. Nishio, "Dynamic TDMA slot assignment in ad hoc networks," in *Proc. of 17th Int. Conf. on Advanced Information Networking and Applications (AINA'03)*, Mar. 2003, pp. 330–335.

[6] A. Kanzaki, T. Hara, and S. Nishio, "An efficient TDMA slot assignment protocol in mobile ad hoc networks," in *Proc. of ACM Symposium on Applied Computing (SAC'07)*, Mar. 2007, pp. 891–895.

[7] I. Rhee, A. Warrier, J. Min, and L. Xu, "DRAND: Distributed randomized tdma scheduling for wireless ad hoc networks," *IEEE Trans. on Mobile Computing*, vol. 8, no. 10, pp. 1384–1396, Oct. 2009.

[8] I. Rhee, A. Warrier, M. Aia, , J. Min, and M. L. Sichitiu, "Z-MAC: A hybrid mac for wireless sensor networks," *IEEE/ACM Trans. on Networking*, vol. 16, no. 3, pp. 511–524, Jun. 2008.

[9] L. Bao and J. J. Garcia-Luna-Aceves, "A new approach to channel access scheduling for ad hoc networks," in *Proc. of 7th ACM Int. Conf. on Mobile Computing and Networking (SIGMOBILE'01)*, Rome (Italy), Jul. 2001, pp. 210–221.

[10] A. Rao and I. Stoica, "An overlay MAC layer for 802.11 networks," in *Proc. of ACM Int. Conf. on Mobile Systems, Applications, and Services (MobiSys'05)*, Seattle, Washington (USA), Jun. 2005, pp. 135–148.

[11] E. L. Lloyd and S. Ramanathan, "On the complexity of link scheduling in multi-hop radio networks," in *Proc. of 26th Conf. on Inform. Science and System (ISS'92)*, 1992.

[12] S. Even, O. Goldreich, S. Moran, and P. Tong, "On the NP-completeness of certain network testing problems," *Jr. on Networks*, vol. 14, no. 1, pp. 1–24, 1984.

[13] S. Ramanathan, "A unified framework and algorithm for (T/F/C)DMA channel assignment in wireless networks," in *Proc. of 16th IEEE Int. Conf. on Computer Commun. (INFOCOM'97)*, vol. 2, Kobe (Japan), Apr. 1997, pp. 900–907.

[14] A. Ephremedis and T. Truong, "Scheduling broadcasts in multihop radio networks," *IEEE Trans. on Commun.*, vol. COM-38, pp. 456–460, Apr. 1990.