

RESEARCH

Open Access

New delay-efficient TDMA-based distributed schedule in wireless mesh networks

Jae-Hyun Kim*, Jae-Ryong Cha and Han-Joon Park

Abstract

Time division multiple access (TDMA)-based medium access control (MAC) protocols can guarantee quality of service (QoS) in wireless environments. However, in an environment where multihop packet transmissions are necessary for real-time communications, each node may experience the well-known queuing delay. This queuing delay increases multihop packet transmission delay, resulting in not meeting the delay bound of real-time applications in multihop wireless networks. This article first introduces two kinds of queuing delays that can occur in multihop wireless networks. Then, this article proposes a new delay-efficient TDMA-based distributed scheduling scheme to eliminate the secondary queuing delay. For the performance analysis of the proposed scheme, the scheduling overhead is first evaluated in terms of power consumption. Next, the multihop packet transmission delay of the proposed scheduling scheme is derived and validated through a simulation, before comparing the result with that of the conventional minimum length scheduling scheme which employs distance-2 graph coloring. According to the simulation and analysis results, for a deterministic packet arrival, the proposed scheme works well irrespective of the packet interarrival rate and outperforms the conventional graph coloring. However, in case of a non-deterministic packet arrival, the multihop packet transmission delay of the proposed scheme is slightly higher than that of the conventional graph coloring because the probability that each node has more than two packets increases at the beginning of the frame. However, the multihop packet transmission delay of the conventional graph coloring is intolerable when the packet interarrival rate is high.

Keywords: TDMA, Scheduling, Quality of service, Realtime services, Allocation order

1 Introduction

Wireless mesh networks (WMNs) are emerging communication networks consisting of nodes that automatically establish an ad-hoc network and maintain mesh connectivity. Because of their advantages over other wireless networks, WMNs are progressing rapidly and are inspiring numerous applications in commercial and tactical environments. With the popularity of WMNs, supporting quality of service (QoS) over multihop radio links is becoming an issue because multihop packet transmission delay increases quickly with the increase in the number of hops [1]. Previous studies on WMNs have mostly been studied based on 802.11 wireless local area networks (WLANs). One of the major drawbacks of such networks is that it is difficult for them to support QoS, which is

essential for supporting real-time applications, particularly in multihop wireless networks. This is because packet transmission delay is accumulated at each hop on a path. Meanwhile, scheduling schemes based on time division multiple access (TDMA) have been proposed for WMNs. Most TDMA scheduling schemes [2-13] for WMNs have been proposed for determining the minimum length schedules. However, although such schemes reduce the frame length, they may bring about queuing delay, which can increase the multihop packet transmission delay in WMNs.

Recently, TDMA-based QoS-aware scheduling schemes [14-19] have been proposed for supporting various applications such as voice and video calls in WMNs. However, these schemes necessarily need a centralized base station for achieving their goals, such as a minimum length schedule considering the scheduling delay, delay-constrained schedule of flows, and link activation schedule to bound end-to-end delay [1]. Brief descriptions of these schemes

*Correspondence: jkim@ajou.ac.kr
Won-Cheon Hall 208, San5 Wonchun-Dong, Suwon, Korea

are presented in the following section. Therefore, this article proposes a new delay efficient TDMA-based distributed scheduling scheme for eliminating secondary queuing delay, which is defined in subsequent sections, and for ultimately reducing the multihop packet transmission delay in WMNs.

2 Related study

This section introduces conventional studies related to the TDMA-based QoS-aware scheduling scheme. Recently, various TDMA-based QoS aware-scheduling schemes have been introduced for WMNs [1].

In [14], the authors schedule different types of flows for satisfying bandwidth and delay requirements. In the first phase, the algorithm attempts to allocate interference-free slots (using multiple channels) to a flow based on the maximum bandwidth requirement of the flow. The second phase is invoked for a flow if the first phase fails to allocate sufficient slots to satisfy even the minimum bandwidth requirement.

In the mechanism proposed in [15], first, a given network is transformed into a conflict graph whose vertices represent links and there is an edge between two vertices if two links conflict (interfere) with each other. Determining the order of transmission, in such a conflict graph, for a conflict-free TDMA schedule with minimum scheduling delay is NP-complete. Therefore, the authors formulate this problem as a linear programming optimization problem. Given the minimum scheduling interval and relative activation times, the authors show how to determine the minimum length TDMA schedule (actual assignment of links to slots) in polynomial time.

To prevent computational complexity of the optimum solution, the authors in [16] have proposed a 'bottleneck first scheduling' (BFS) scheme, where scheduling decisions at stations having higher traffic loads are made before those having lower traffic loads. At each station, scheduling decisions for constant bit rate (CBR) packets with more hops to their destinations are made first. Through simulations, the authors show that the delay of BFS is better than 'earlier deadline first' (EDF) and 'first come first serve' (FCFS) scheduling.

In the mechanism proposed in [17], the nodes (and thus links) in the network are first labeled either even or odd (two-slot scheduling). Then, while determining paths, only those paths that go through nodes having alternate labeling are considered. Using sub-channelization of OFDMA, secondary interference between two links in the same slot is prevented by assigning different channels to the links. Once the slot requirements and routing paths are determined, each node employs a local (wireline-type) scheduling policy. The scheduling policy determines the order in which packets leave the buffer at each node, and the authors show that such a mechanism provides

two-approximation bounds for the end-to-end delay. The problem of finding feasible routes (to and from gateway to network nodes) in an even-odd labeled (tree) network is formulated as a linear program and heuristics using Dijkstra's shortest path algorithm have been proposed.

The authors of [18] have proposed a 'load-balanced weighted shortest path with a retry' routing heuristic. In this heuristic, first, the shortest-hop algorithm is used to determine a path. If one or more edges on the path are blocked, those are removed from the graph and the heuristic is applied again to find a suitable path. Next, a call admission control (CAC) algorithm is applied which considers the unsolicited grant service/real-time polling service in WiMAX. To manage the jitter value of a connection, the path from a source node to a destination node is partitioned into two segments; one segment is from the source to the penultimate node (the node just before the destination), and the other segment is the link between the penultimate node and the destination. The delay requirement is that the total delay on both the segments should be less than the delay constraint. Because of the split, to ensure the jitter constraint of the path, the scheduler only needs to look at the second segment between penultimate node and destination. This is achieved by fixing an offset value and scheduling the packets within the limit of the jitter grant interval.

In [19], an online algorithm that works in three phases has been proposed. In the first phase, the algorithm constructs an auxiliary graph from the given topology graph. A vertex in the auxiliary graph is a four-parameter tuple of the form (node, slot, channel, hop). An edge is formed between two vertices if a set of rules is satisfied. The rules model the interference and delay constraints. In the second phase, Dijkstra's shortest path algorithm is run to output the delay-constrained schedule while finding a routing path, assigning a channel, and scheduling links in the process. In the third phase, an unfavorable schedule is filtered to incorporate an arbitrary interference graph. The authors compare this algorithm with an offline optimal solution and show that it accepts around 90% of the calls with respect to the optimal solution.

3 System model

In this article, we model WMNs with a topology graph connecting the nodes that are present in each other's wireless range. The network can be represented with a directed connectivity graph $G(B, E)$, where $B = \{b_1, \dots, b_m\}$ is a set of nodes and $E = \{e_1, \dots, e_g\}$ is a set of directed links, and two nodes (u and v) are neighbors if $(u, v) \in E$. In the network, F is a set of flows, and a flow $f \in F$ is specified by a node set $R(f) = \{p_1, \dots, p_q\}$, where p_k is the k th node in a flow ($2 \leq q$; $k = 1$: the source node, $1 < k < q$: the intermediate node(s), and $k = q$: the destination node).

4 Queuing delay in multihop wireless networks

In this section, we describe two main factors that cause an increase in the multihop packet transmission delay in multihop wireless networks. In previous studies [2-13], TDMA scheduling is generally used to determine the minimum length schedules. However, such schedules may cause additional queuing delays, which have a negative influence on delay-sensitive networks. The QoS of real-time applications in multihop wireless networks may not be guaranteed if additional queuing delays occur.

Figure 1 shows a simple multihop wireless network in which queuing delay occurs. In single-hop wireless networks, queuing delay occurs when the packet arrival rate in a source node is higher than the packet transmission (service) rate. In this article, such a queuing delay is referred to as the primary queuing delay and primary queuing points (QPs) in Figure 1 represent the examples of such a case. On the other hand, in multihop wireless networks, an additional queuing delay may occur, which is referred to as the secondary queuing delay in this article. The secondary queuing delay occurs when multiple flows pass through an outbound link of a relay node. The secondary QP shown in Figure 1 represents the point where the secondary queuing delay occurs.

For example, in Figure 1, multiple flows pass through the Node R1-to-R2 link. These flows share a slot for transferring packets. Assume that the network employs the minimum length schedule. Then, Node R1 allocates only one slot for transferring packets to node R2. It is also assumed that Node S1 and Node S2 are supposed to send a packet to Node R1 in the 1st slot and the 2nd slot, respectively. Moreover, Node R1 is scheduled to send a packet to Node R2 in the 3rd slot. It is also assumed that, in each node, the arrival of a packet from application layer is concurrent with the start of a frame. In the 3rd slot in the frame, Node R1 has two packets to send: one from Node S1 and another from Node S2. However, Node R1 can transfer the packet received from node S1 in the current frame and can transfer the packet received from Node S2 in the

next frame because it can transfer only one packet per link in a frame as prescribed by the minimum length schedule. In conclusion, the minimum length schedule may work well in single-hop wireless networks with high throughput and short delay. However, in multihop wireless networks, it may cause secondary queuing delay because only one common slot is allocated for multiple flows.

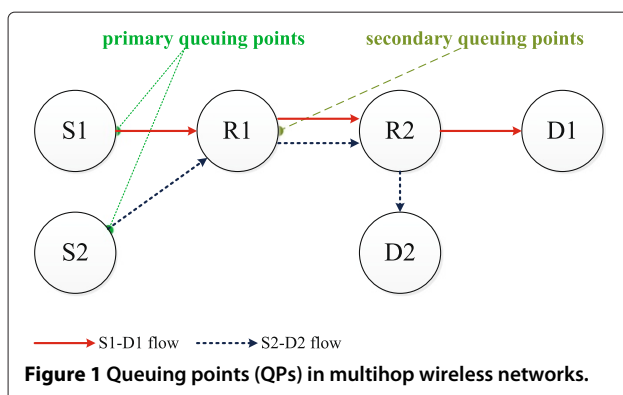
Therefore, this article proposes a new distributed scheduling scheme to eliminate the secondary queuing delay, thereby ultimately reducing the multihop packet transmission delay.

5 Proposed scheduling scheme

In this section, the operational procedures for the proposed scheduling scheme are described in detail. The operational procedures are classified into two phases: Phase I and Phase II. In Phase I, each node obtains paths using the ad-hoc on-demand distance vector (AODV) routing protocol [20] and gathers the information on its one-hop neighbors. These two tasks are also performed in the conventional schemes [9,21] prior to the TDMA scheduling, although both the routing algorithms and the approaches for obtaining the neighbor information are slightly different. Moreover, these tasks are generally excluded during the overhead analysis in the conventional tasks, as these tasks are considered as input parameters for scheduling. Therefore, we also exclude Phase I during the overhead analysis for performance comparison with the conventional schemes. However, we have performed these tasks during the simulation.

On the other hand, Phase II involves three steps for slot allocation: an initial frame length synchronization (IFLS) process, a multihop slot allocation (MSA) process, and a global frame length synchronization (GFLS). First, IFLS determines the initial frame length L_{init} , which can be used initially for slot allocation in the network. Second, the MSA process allocates a different slot to each flow in a link to eliminate the secondary queuing delay. For example, in Figure 1, two flows exist between Node R1 and Node R2. Therefore, the proposed scheme allocates two different slots for two flows in the Node R1-to-R2 link. Although such a slot allocation can eliminate the secondary queuing delay, it results in a large frame length. In such a case, it is important to consider the order of the slots allocated on a path. Therefore, we show the delay effect by an allocation order as well as demonstrate the manner in which slots should be allocated sequentially in a flow after the routing path is established.

Meanwhile, the multihop nature of WMNs allows spatial reuse of the TDMA slots. Different nodes can use the same time slot if they do not interfere with each other [22]. Finally, after the MSA process, we obtain the minimum



frame length, which can be used globally in the network. After all three processes are completed, all the nodes in the network transfer packets in their allocated slots. This article describes the details of Phase II from the following section onward. The description of two tasks performed in Phase I is not included, because it is beyond the scope of this article as mentioned before.

5.1 Initial frame length synchronization: IFLS

From this section onward, the three steps of Phase II are described in detail. First, for TDMA scheduling, it is necessary to determine the initial frame length, which can be used commonly throughout the TDMA network. In [21], the authors assume that the initial frame length is sufficiently large for the allocation of slots to all the nodes. However, all the nodes carrying out the proposed MSA process must exchange a frame map with each other, where the frame map represents the bit-wise expression of the slot allocation status. A detailed description is provided in the following section. Thus, our assuming too large frame length results in the large overhead in the proposed scheme. Therefore, in this study, each node obtains information on the initial frame length by broadcasting hop information. In order to obtain the initial frame length L_{init} , each node gathers the hop information by broadcasting $h(i)$. $h(i)$ indicates the hop information to the destination, which a source node i obtains after the completion of routing. When a node hears $h(i)$, if the received hop value is higher than the one it currently knows, it rebroadcasts the new value. After some predetermined dissemination time, all nodes obtain the highest hop value (h_{max}) from $h(i)$. In the proposed MSA process, each node allocates a different slot to each flow in a link. Therefore, under the condition that each node manages

only one flow, the upper bound of the initial frame length can be determined by

$$L_{init} = h_{max} \cdot N, \tag{1}$$

where N is the total number of nodes in the network. Once initial frame length synchronization is completed, then each node carries out the MSA process using the same initial frame length L_{init} .

5.2 Multihop slot allocation: MSA

In this section, we describe the proposed MSA process in detail. First, we assume that each node follows global slot synchronization. Thus, all nodes know the starting time of each frame. Before the description of the proposed MSA process, we first introduce the delay effect caused by an allocation sequence of TDMA slots and show that the allocation sequence in a flow is critical for reducing the multihop packet transmission delay in an environment where a node allocates a different slot to each flow in a link.

5.2.1 Delay effect produced by an allocation sequence

In this study, the MSA process allocates a different slot to each flow in a link in order to eliminate the secondary queuing delay. In such a slot allocation, we must consider the sequence of the slots allocated. Figure 2 shows the delay effect produced by the allocation sequence of the TDMA slots on a path. In Figure 2, the number of nodes, q , is 4 and the number of hops, h , is 3. h represents the distance between a source node and a destination node. It is assumed that the source node p_1 is supposed to transfer a packet z_1 from the i th frame. If the slot allocation is randomly carried out ($e_2e_1e_3$), the destination, p_4 , will

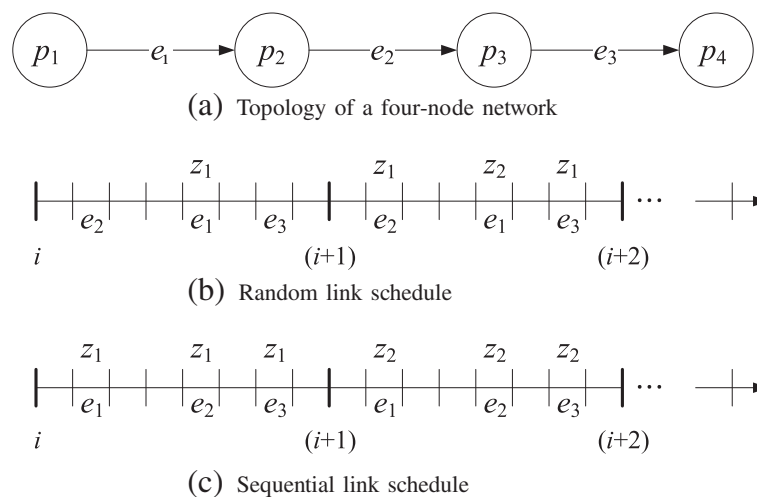


Figure 2 Example of random/sequential link schedules. (a) Topology of a four-node network. (b) Random link schedule. (c) Sequential link schedule.

receive z_1 in the $(i + 1)$ th frame (Figure 2b). In this article, such a link schedule is referred to as a random link schedule. However, as shown in Figure 2c, z_1 will be transferred to p_4 in the i th frame because the slots on the path are allocated $(e_1e_2e_3)$ sequentially. Such a link schedule is referred to as a sequential link schedule in this article. Therefore, if the sequential link schedule is not considered, the multihop packet transmission delay may be quite large as the number of hops increases.

Now, we derive the multihop packet transmission delay caused by a random link schedule. We assume that the traffic flow between all the node pairs in the network is uniform and that the processes of new packet arrival at the different nodes are independent. Therefore, we now concentrate on the characteristics of one node, and thus, it is assumed that the node transmits a packet in the first slot of each frame. Consider a typical packet generated by a node. Because we assume the stop-and-go queuing system [23], the total packet transmission delay suffered by a packet, D_r , can be obtained by using the following three components (Figure 3): (1) the time between its generation and the end of the current frame; (2) the average frame delay, which indicates the number of frames required to transfer a packet from the source to the destination; and (3) the time between the start of the last frame and its reception at the destination. Given that all the frames are of equal length, the average time between the packet generation time and the end of the current frame is $0.5T_M$, where T_M is the frame length. Next, assuming that the slot for the destination is randomly allocated in a frame, we observe that the time between the start of the last frame and packet reception at the destination is $0.5T_M + T$. Finally, the average frame delay $F_h (h \geq 2)$ can be calculated as

$$F_h = \left(\prod_{k=3}^{h+1} k \right) / h! = \left(\frac{h+1}{2} \right). \quad (2)$$

Proof. Now, we prove the above equation. It is assumed that a node is supposed to transfer z_1 from the i th frame in Figure 2. If the slot allocation is completed randomly and $h=3$, then cases exists as many as

3!. For each case, the frame delay is $T_M (e_1e_2e_3)$, $2T_M (e_1e_3e_2, e_2e_1e_3, e_2e_3e_1, e_3e_1e_2)$, and $3T_M (e_3e_2e_1)$ long. Thus, F_3 is $12T_M/3! = 2T_M$. If we consider each $h(\geq 2)$, we obtain

$$F_2 = 3T_M/2!, F_3 = 12T_M/3!, F_4 = 60T_M/4!, \dots$$

□

Accordingly, the total multihop packet transmission delay suffered by a packet, D_r , is given by

$$D_r = 0.5T_M + (F_h - 1)T_M + 0.5T_M + T \quad (3) \\ = F_hT_M + T.$$

Figure 4 shows the frame delay (number of frames) with an increase in the number of hops. In Figure 4, *Lower bound* and *Upper bound* indicate that the links are scheduled sequentially in the order of $e_1 \rightarrow e_2 \rightarrow \dots \rightarrow e_h$ and $e_h \rightarrow e_{h-1} \rightarrow \dots \rightarrow e_1$, respectively. From this result, it can be found that the allocation sequence of the TDMA slots on a path may be an important factor in multihop wireless networks if the delay bound must be considered. Therefore, the objective of this study is to sequentially allocate the slots, such that the allocated link sequence becomes $e_1 \rightarrow e_2 \rightarrow \dots \rightarrow e_h$ on a path within one frame, resulting in reducing the multihop packet transmission delay.

5.2.2 MSA process

To efficiently describe the proposed MSA process, we show two types of examples using an algorithm and a figure. To carry out the proposed MSA process, three packets are used: a map request (MAP_{REQ}) packet, a map response (MAP_{REP}) packet, and a slot allocation (SA) packet. The MAP_{REQ} and MAP_{REP} packets are used by a node to request the frame map of its neighbors and to respond to the request, respectively. The SA packet is employed for transferring the information on the allocated slot index to the next node. The followings are some terminologies used for describing the proposed MSA process.

- forward/reverse path: path to the destination/source node.

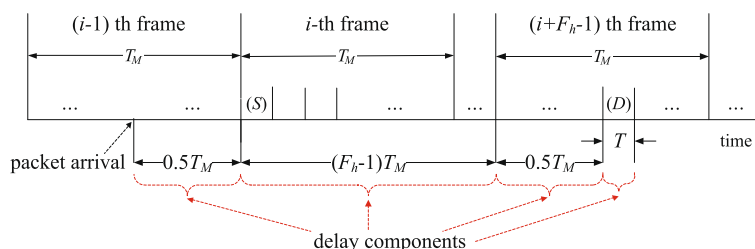
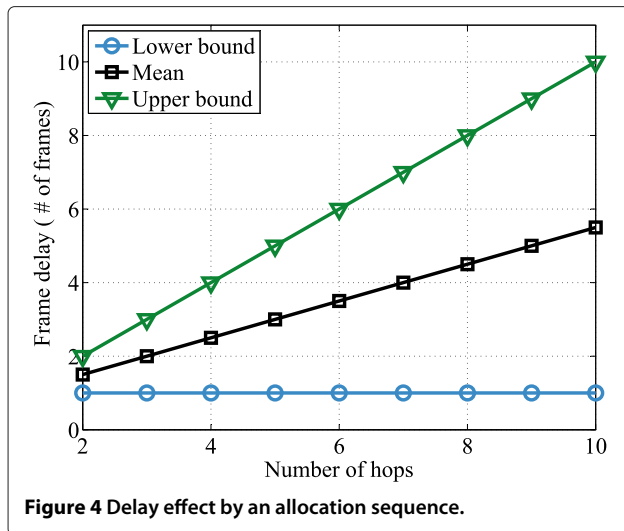


Figure 3 Delay components of a random link schedule.



- TN/RN: a transmitting node and a corresponding receiving node in an allocated slot.
- *Allocated Slot Index*: the slot index allocated.
- next node: a neighbor peer node to the destination node in a forward path.
- MAC_{NEXT} : medium access control (MAC) address of a next node.
- MAC_{SOURCE} : MAC address of a source node.
- $MAC_{DESTINATION}$: MAC address of a destination node.

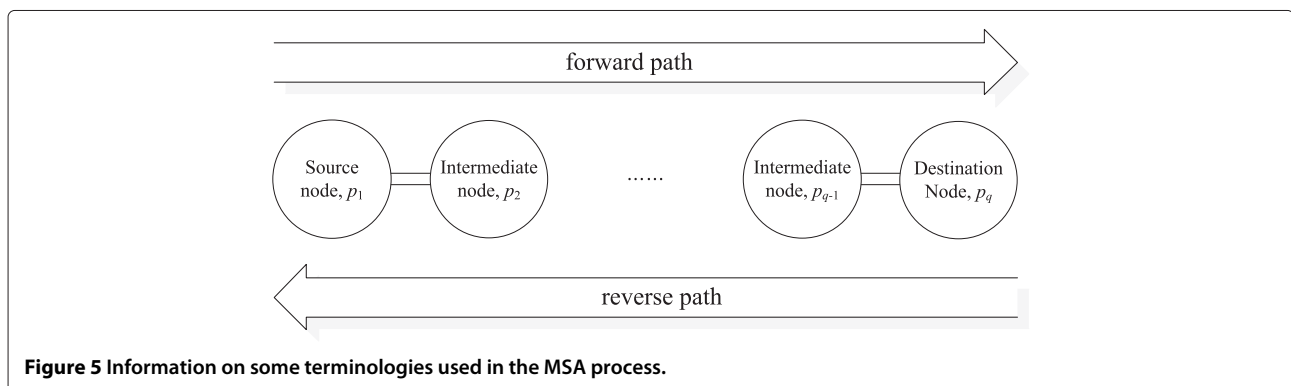
Some of the above terms are shown with regard to forward and reverse paths in Figure 5. During the proposed MSA process, all the nodes related to a flow (except the destination node) always have two tasks to perform. One is to obtain frame maps from all their one-hop neighbors, and the other is to transfer an SA packet to the next node on the path after allocating the slot(s) on the basis of the obtained frame maps. Each node knows which node it transfers the SA packet because the routing process has already been completed in Phase I. A MAP_{REQ}

Table 1 Field size of packets used for the proposed MSA process

Fields	Size (bits)
MAP_{REQ} ID	8
MAP_{REP} ID	8
SA ID	8
MAC_{NEXT}	48
<i>Allocated Slot Index</i>	16

packet includes the forwarding information on an identification (ID) field, a MAP_{REP} packet on an ID field and a frame map field, and an SA packet on the MAC_{NEXT} field and the *Allocated Slot Index* field. Table 1 lists the size of each field. The field size for an ID and an address is based on the values in [20]. As explained before, a frame map represents the bitwise expression of the slot allocation status in a node. The length of the frame map is equal to the frame length L_{init} . Initially, the frame maps of all the nodes are set as zeros. During the proposed MSA process, a node sets the part allocated as TN or RN in its frame map to '1'. A node's frame map is updated each time a node receives an SA packet. If a source node is ready for the slot allocation in a flow, then it unicasts the SA packet after allocating the slot(s) on the basis of the obtained frame maps.

Algorithm 1 shows the proposed MSA process initiated by a source node. Further, Algorithm 2 shows the proposed MSA process when either an intermediate node or a destination node receives an SA packet from node p_k ($1 \leq k < q, q \geq 2$) in a forward path. An intermediate node or a destination node invokes Algorithm 2 whenever it receives an SA packet wherein MAC_{NEXT} is equal to its MAC address. In Algorithm 2, when an intermediate node allocates the slot(s) as a TN, it is very important for the intermediate node to reserve the right-hand side slot for comparison with the *Allocated Slot Index* within the SA packet received, such that multihop links can be scheduled sequentially on a path.



Algorithm 1. MSA in a source node

- 1: **if** $k = 1$ **then**
- 2: p_1 first obtains frame maps of its one-hop neighbors by exchanging MAP_{REQ} and MAP_{REP} packets.
- 3: p_1 allocates the commonly available slot(s) for both p_1 and p_2 as a TN.
- 4: p_1 transfers the SA packet with *Allocated Slot Index* to p_2 .
- 5: **end if**

Algorithm 2. MSA in an intermediate/a destination node

- 1: **if** $1 < k < q$ **then**
- 2: Based on the received *Allocated Slot Index*, p_k allocates the slot(s) for both p_{k-1} and p_k as an RN.
- 3: Then, p_k obtains the frame map of its one-hop neighbors by exchanging MAP_{REQ} and MAP_{REP} packets.
- 4: p_k allocates the commonly available slot(s) for both p_k and p_{k+1} as a TN.
- 5: p_k transfers the SA packet with *Allocated Slot Index* to p_{k+1} .
- 6: **else if** $k = q$ **then**
- 7: Based on the received *Allocated Slot Index*, p_q allocates the slot(s) for both p_{q-1} and p_q as an RN.
- 8: **end if**

This article presents another example of the proposed MSA process for eliminating any ambiguity regarding the process. Figure 6 shows an example of the proposed MSA process. In Figure 6, p_1 and p_4 represent the source node and destination node in a flow, respectively, and p_2 and p_3 represent the relay nodes. In this example, it is assumed that each node has five (1 to 5) slots.

In each frame in Figure 6, the gray slots are those that have already been allocated by other flows. First, p_1 obtains the map information of its neighbors by exchanging MAP_{REQ} and MAP_{REP}. Then, p_1 assigns the 2nd slot as a TN and then sends SA {MAC_{NEXT}, *Allocated Slot Index*} to p_2 . When p_2 receives SA { $p_2, 2$ } from p_1 , it assigns the same 2nd slot as an RN. Second, for communication between p_2 and p_3 , p_2 assigns the 3rd slot as a TN after it obtains the map information from its one-hop neighbors. Then, p_2 sends SA { $p_3, 3$ } to p_3 . Next, for communication between p_3 and p_4 , p_3 first assigns the 5th slot as a TN after it obtains the map information from its one-hop neighbors. After finishing the slot allocation, p_3 sends SA { $p_4, 5$ } to p_4 . Finally, the destination node p_4 assigns the 5th slot as an RN. After the proposed MSA process is completed, the map status of p_1, p_2, p_3 , and p_4 becomes '11000', '01100', '00111', and '00001', respectively.

5.3 Global frame length synchronization: GFLS

In this study, all the nodes undergoing the proposed MSA process employ the initial frame length L_{init} . L_{init} is calculated as the upper bound of the affordable frame length. After the proposed MSA process is completed, the frame length of all the nodes may be less than L_{init} because of slot reuse in the TDMA system [22]. Therefore, L_{init} needs to be reduced for efficiency. In the proposed MSA process, the slots are always allocated from each source-node side in a flow. Therefore, the slot index allocated by the destination node is the highest one in the flow. After the proposed MSA process is completed, all the destination nodes broadcast that slot index. If a destination node is related to multiple flows, the highest slot index among the slot indices allocated by the destination is broadcast. If a node hears a higher slot index than the one it currently knows, it rebroadcasts the new index. After some predetermined dissemination time, all the nodes calculate the

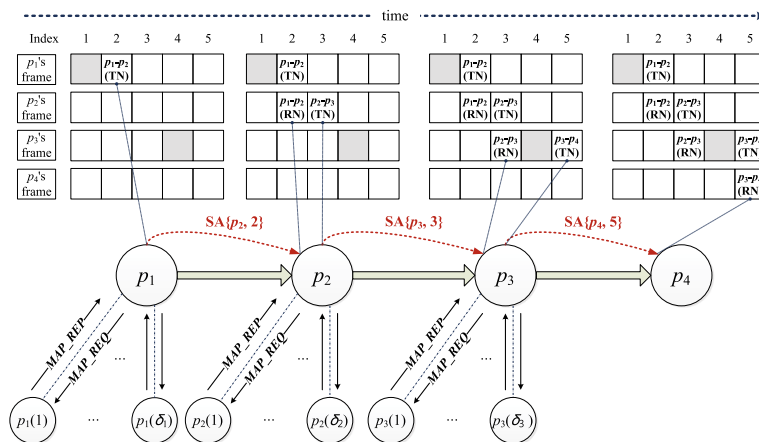


Figure 6 Example of the proposed MSA process.

global frame length on the basis of the highest slot index they learns, i.e., the global frame length $L_{\text{proposed}} =$ the highest slot index. Subsequently, all the nodes adjust their frame length to L_{proposed} .

6 Performance analysis

6.1 Overhead analysis

In this section, we evaluate the overhead of the proposed MSA process in terms of the power consumed for the scheduling by all the nodes in the network. The proposed scheme gets the map information of its one-hop neighbors in each hop on the path before transferring the SA packet to the next node. Considering all the possible combinations, we find that the total power consumed for scheduling, P_{total} , consists of the following two components.

- P_{MAP} : power consumed for exchanging map information by all the nodes.
- P_{SA} : power consumed for transferring SA packets by all the nodes.

First, P_{MAP} can be represented as

$$P_{\text{MAP}} = P_{\text{MAP-REQ}} + P_{\text{MAP-REP}}, \quad (4)$$

where $P_{\text{MAP-REQ}}$ and $P_{\text{MAP-REP}}$ denote the sum of both the transmitting power in transmitting nodes and the receiving power in receiving nodes when transferring the MAP_{REQ} and MAP_{REP} packets, respectively. In Figure 6, assuming that all the nodes in the network are evenly distributed, we find that $\delta = \delta(p_1) = \delta(p_2) = \delta(p_3) = \delta(p_4)$. Therefore, $P_{\text{MAP-REQ}}$ can be calculated as

$$P_{\text{MAP-REQ}} = \sum_{i=1}^N \sum_{j=1}^{h(i)} \left[\begin{array}{l} \sum_{k=1}^{\delta} \left(l_{\text{total-REQ}} \cdot p_{\text{tx}} \right. \\ \left. + l_{\text{total-REQ}} \cdot p_{\text{rx}} \cdot \delta \right) \\ \left. + \sum_{k=1}^{\delta} \left(l_{\text{ACK}} \cdot p_{\text{tx}} + l_{\text{ACK}} \cdot p_{\text{rx}} \cdot \delta \right) \right]. \quad (5)$$

In Equation (5), N denotes the total number of nodes; $h(i)$, the total number of hops obtained by the i th source

node to transfer its packet to its destination node; δ , the one-hop degree in a node which means the number of one-hop neighbors; l_{REQ} , the length of the MAP_{REQ} packet; $l_{\text{MAC+PHY}}$, the sum of the overhead in MAC and physical (PHY) layers; $l_{\text{total-REQ}}$, the sum of l_{REQ} and $l_{\text{MAC+PHY}}$; and l_{ACK} , the length of the acknowledgement (ACK) packet. Further, p_{tx} denotes the energy spent in transmitting a bit over a distance of 1-meter, and p_{rx} denotes the energy spent in receiving a bit.

We can also obtain the value of $P_{\text{MAP-REP}}$ by simply replacing $l_{\text{total-REQ}}$ with $l_{\text{total-REP}}$ as follows.

$$P_{\text{MAP-REP}} = \sum_{i=1}^N \sum_{j=1}^{h(i)} \left[\begin{array}{l} \sum_{k=1}^{\delta} \left(l_{\text{total-REP}} \cdot p_{\text{tx}} \right. \\ \left. + l_{\text{total-REP}} \cdot p_{\text{rx}} \cdot \delta \right) \\ \left. + \sum_{k=1}^{\delta} \left(l_{\text{ACK}} \cdot p_{\text{tx}} + l_{\text{ACK}} \cdot p_{\text{rx}} \cdot \delta \right) \right], \quad (6)$$

where l_{REP} denotes the length of the MAP_{REP} packet and $l_{\text{total-REP}}$ is the sum of l_{REP} and $l_{\text{MAC+PHY}}$.

Similar to the calculation of $P_{\text{MAP-REQ}}$ and $P_{\text{MAP-REP}}$, P_{SA} can be represented as

$$P_{\text{SA}} = \sum_{i=1}^N \sum_{j=1}^{h(i)} \left[\begin{array}{l} l_{\text{total-SA}} \cdot p_{\text{tx}} + l_{\text{total-SA}} \cdot p_{\text{rx}} \cdot \delta \\ \left. + l_{\text{ACK}} \cdot p_{\text{tx}} + l_{\text{ACK}} \cdot p_{\text{rx}} \cdot \delta \right], \quad (7)$$

where l_{SA} is the length of the SA packet and $l_{\text{total-SA}}$ is the sum of l_{SA} and $l_{\text{MAC+PHY}}$. Accordingly, the total power consumed for scheduling by all the nodes is given by

$$P_{\text{total}} = P_{\text{MAP}} + P_{\text{SA}}. \quad (8)$$

6.2 Analysis of multihop packet transmission delay

In this section, we derive the average multihop packet transmission delay of the proposed scheduling scheme.

Similar to the analysis in the random link schedule, the delay suffered by a packet, D_s , can be obtained using the following four components (refer to Figure 7): (1) the time between its generation and the end of the current frame, (2) the primary queuing delay to allow all the packets

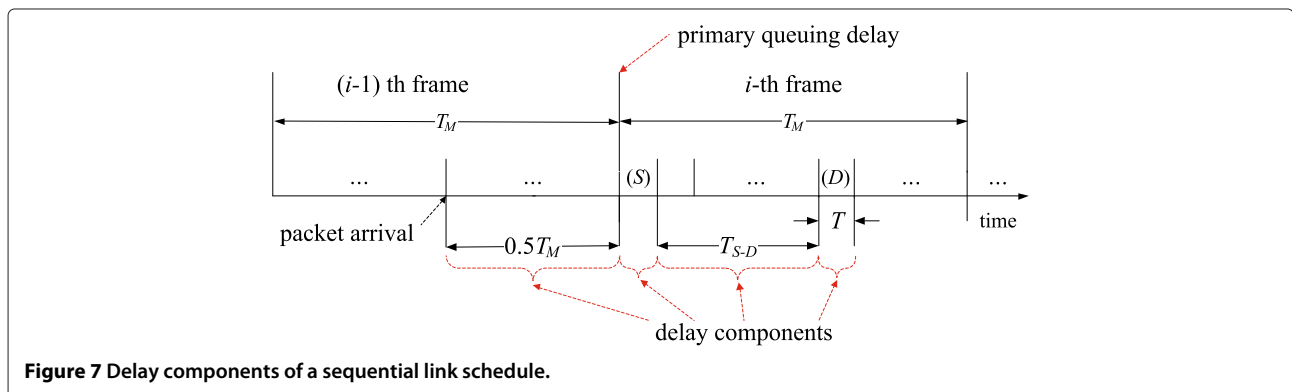


Figure 7 Delay components of a sequential link schedule.

already queued to be transmitted in a source node, (3) the distance between the first slot and the slot allocated for the destination node, and (4) packet transmission time in both the source and destination nodes. The first component is identical to that in case of the random link schedule, whose value is $0.5T_M$, and the fourth component, which is the packet transmission time in both the source and destination nodes, is $2T$.

To compute the second component, i.e., primary queuing delay, (once the end of the current frame is reached), we observe that the queue behaves exactly like the one with a deterministic service time of T_M . If we assume a Poisson arrival process of λ packets/s for a user and that the number of packets that can be stored in a queue is not bounded, then the primary queuing delay is identical to the queuing time in an M/D/1 queuing system in which the deterministic service time is T_M . Thus, the expected primary queuing time of a packet, W_q , is given by [24,25]

$$W_q = \frac{\rho}{2(1-\rho)} \cdot T_M = \frac{\rho}{2(1-\rho)} \cdot L_{\text{proposed}} \cdot T, \quad (9)$$

where $\rho = \lambda \cdot T_M$. If we consider a deterministic packet arrival and a deterministic service time, then W_q is equal to zero [15,23].

On the other hand, the third component, which is the distance between the first slot and the slot allocated for the destination node, T_{S-D} , can be calculated by averaging all the possible combinations. Figure 8 shows an example of the calculation of T_{S-D} when $h = 3$. First, if there is no vacant slot (A_0) among all the allocated slots on the path, then T_{S-D} is T and one case (${}_{3-2}C_{3-2}$) exists, where ${}_{\alpha}C_{\beta}$ denotes the combination of α things taken β at a time.

Second, when there is one vacant slot (A_1), T_{S-D} is $2T$ and two cases (${}_{4-2}C_{3-2}$) exist. Next, when there are two vacant slots (A_2), T_{S-D} is $3T$ and three cases (${}_{5-2}C_{3-2}$) exist. Therefore, for all N , T_{S-D} can be given by

$$T_{S-D} = \left[\left(\sum_{j=h}^N \binom{j-2}{h-2} \cdot (j-2) \right) / \left(\sum_{j=h}^N \binom{j-2}{h-2} \right) \right] \cdot T, \quad (10)$$

where $N = T_M/T$. Accordingly, the total multihop packet transmission delay suffered by a packet, D_s , is given by

$$D_s = 0.5T_M + W_q + T_{S-D} + 2T. \quad (11)$$

7 Performance evaluation

In this study, we compare the delay performance of the proposed scheme with that of the conventional scheme that uses distance-2 graph coloring. Distance-1 graph coloring causes the well-known hidden node problem [9,21]. Therefore, it is excluded in this article.

7.1 Simulation scenarios

For the performance evaluation, two scenarios are considered. In *Scenario #1*, we simulate five grid networks to evaluate the delay effect produced by the secondary queuing delay. In *Scenario #2*, a grid network and a random network are simulated for the performance comparison of both the proposed scheduling scheme and the conventional graph coloring. In *Scenario #2*, as a consideration of primary queuing delay, we also consider both a deterministic packet arrival (DPA) and a non-deterministic packet arrival (NON-DPA) having exponential distribution. These two cases are considered for observing the

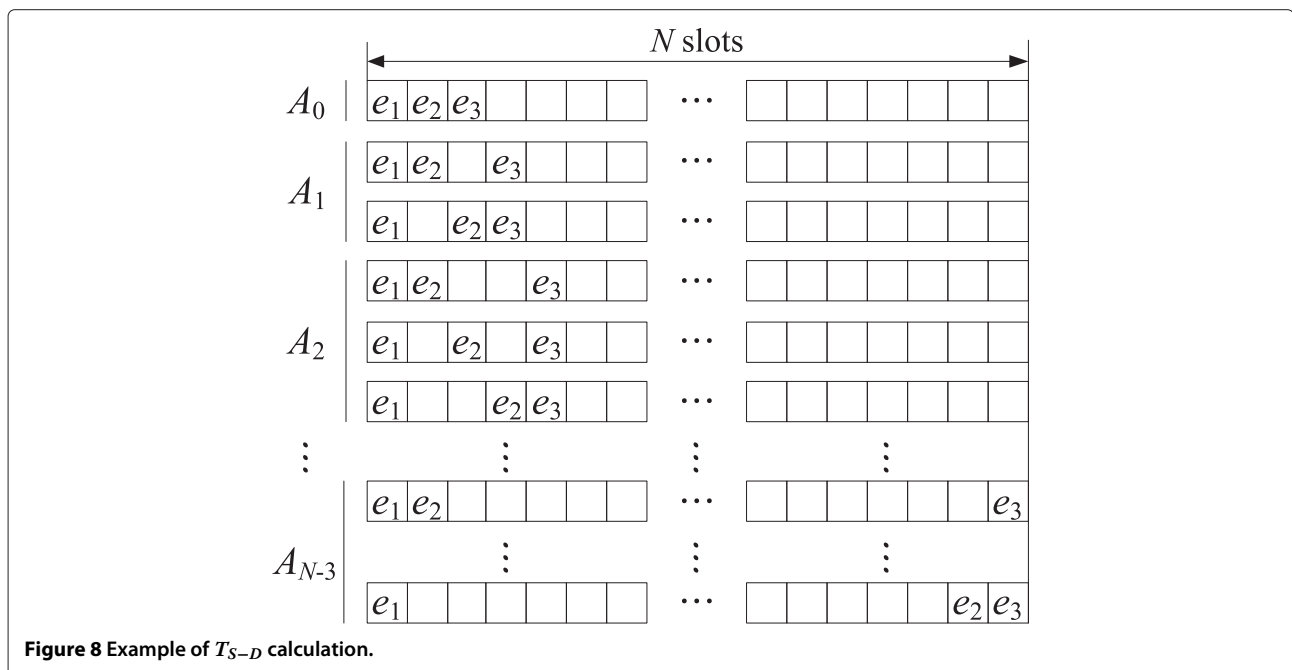


Figure 8 Example of T_{S-D} calculation.

behavior for both non-constant and constant packet inter-arrival characteristics.

7.1.1 Scenario #1

This article simulates X by X grid networks, where X is set to 3, 4, 5, 6, and 7 to observe the queuing behavior of distance-2 graph coloring when only one packet is transferred from each source node. In this network, the vertical and horizontal distances between two adjacent nodes are 30 m, and the communication range of each node is 30 m. After each node allocates slots by distance-2 graph coloring, each source node transfers one packet in its allocated slot. If any intermediate nodes receive packets from the previous node on the path, they transmit the received packet in the allocated slot. This study considers ten different seeds for this scenario, and their simulation results are averaged.

7.1.2 Scenario #2

To compare the performance of the proposed scheduling scheme with that of the conventional scheme using distance-2 graph coloring as carried out in [15], this study simulates two TDMA networks with different topologies. One is the X by X grid network, where X is set as 7. In the grid network, the vertical and horizontal distances between two adjacent nodes are 30 m. The other is the network with 100 nodes randomly distributed in a square area of 200×200 m. In both networks, the communication range of each node is 30 m. As soon as all the source nodes complete the proposed MSA process successfully, they generate packets before transmitting them in the allocated slot. If intermediate nodes receive packets from the previous node on the path, they transmit the received packets in the allocated slot for each flow. For the grid network, this study considers five different seeds. In case of random topologies, this study considers ten different random topologies and their simulation results are averaged.

Table 2 summarizes some preliminary results obtained from the simulation. In Table 2, L_{coloring} denotes the

Table 2 Some preliminary results

unit slot time	Grid network ($N = 49$) 0.001 s	Random network ($N = 100$) 0.001 s
L_{coloring}	21	88
L_{init}	196	440
L_{proposed}	75	200
one-hop degree	3.35	7.5
two-hop degree	5.67	11.9
h	4	4.4

Table 3 Parameter for overhead calculation

Parameters	Value (bits)
$I_{\text{MAC}} + \text{PHY}$	496
I_{ACK}	496
I_{REQ}	8
I_{REP}	$8 + L_{\text{proposed}}$
I_{SA}	$8 + 48 + 8$
$I_{\text{total}} - \text{REQ}$	$I_{\text{MAC}} + \text{PHY} + I_{\text{REQ}}$
$I_{\text{total}} - \text{REP}$	$I_{\text{MAC}} + \text{PHY} + I_{\text{REP}}$
$I_{\text{total}} - \text{SA}$	$I_{\text{MAC}} + \text{PHY} + I_{\text{SA}}$

frame length of the conventional schedule using distance-2 graph coloring and L_{proposed} denotes the frame length of the proposed scheduling scheme.

7.2 Numerical and simulation results and discussions

In this section, we first discuss the results of overhead analysis. Next, the simulation results from *Scenario #1* are discussed. Finally, the simulation results from *Scenario #2* and the related analysis results are discussed.

7.2.1 Results from the overhead evaluation

Table 3 lists the parameters used for calculating the MSA overhead. As same in [9,26,27], p_{tx} and p_{rx} are set as 0.1 nJ/bit- m^2 and 50 nJ/bit, respectively. Table 4 shows the total energy spent by all the nodes for the proposed MSA process. Typically, a distributed network having 100 nodes needs a couple of joules of energy for scheduling [26,27]. In the proposed MSA process, all nodes exchange their table maps for each hop in a flow. Therefore, the energy consumed is slightly more than that consumed in the conventional schemes. In particular, if the overhead for the frame length synchronization in Phase II is included, the energy consumed may be slightly greater than that in the results shown in Table 4. However, it is believed that these results are within the acceptable range.

7.2.2 Results from scenario #1

Table 5 summarizes some results from the simulation of *Scenario #1*, where each source node transfers only one

Table 4 MSA overhead: power consumption

	49 nodes (Joules)	100 nodes (Joules)
$P_{\text{MAP}} - \text{REQ}$	0.1120	1.2474
$P_{\text{MAP}} - \text{REP}$	0.1203	1.4967
P_{SA}	0.0356	0.1769
P_{total}	0.2679	2.9210

Table 5 Some preliminary results in graph coloring

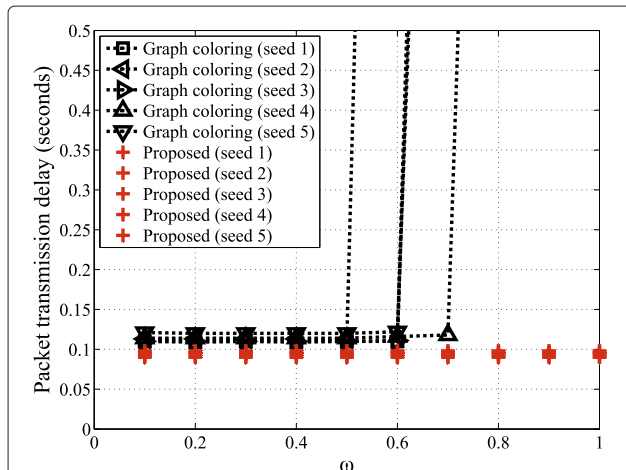
# of nodes	9	16	25	36	49
L_{coloring}	6	7	18	21	22
h	1.44	2.13	2.72	3.22	4.10
frame delay	2.2	3.5	5.7	7.5	9.4

packet in the first frame. In Table 5, L_{coloring} denotes the number of slots allocated by distance-2 graph coloring, h denotes the average number of hops, and frame delay denotes the number of frames elapsed for all the destination nodes to receive one packet from each source node. In conventional minimum length schedules where one common slot is allocated to multiple flows in a link, the factors that may influence frame delay are primary and secondary queuing delays. In this scenario, no primary queuing delay

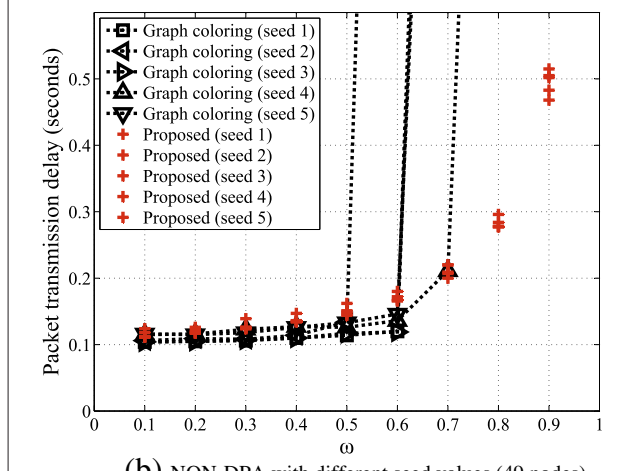
occurs because each node generates only one packet. Intuitively, it is said that if the number of hops from a source node to a destination is h , the frame delay is h . However, in Table 5, each network (when X is 3, 4, 5, 6, and 7) needs 2.2, 3.5, 5.7, 7.5, and 9.4 frames on an average when h is 1.44, 2.13, 2.72, 3.22, and 4.10, respectively. Accordingly, the increase of the frame delay in Scenario #1 is all caused by the secondary queuing delay.

7.2.3 Results from scenario #2

Figure 9 shows the multihop packet transmission delay with different seed values with an increase of the value of ω . In Figure 9, the symbols indicate the simulation results. The packet interarrival time is set as $(\text{unit slot time} \times L_{\text{proposed}})/\omega$, where ω increases from 0.1 to 1 in steps of 0.1. $(\tau \times L_{\text{proposed}})$ means a frame length in terms of time and is the fixed value. If the value of ω is 0.5, then each

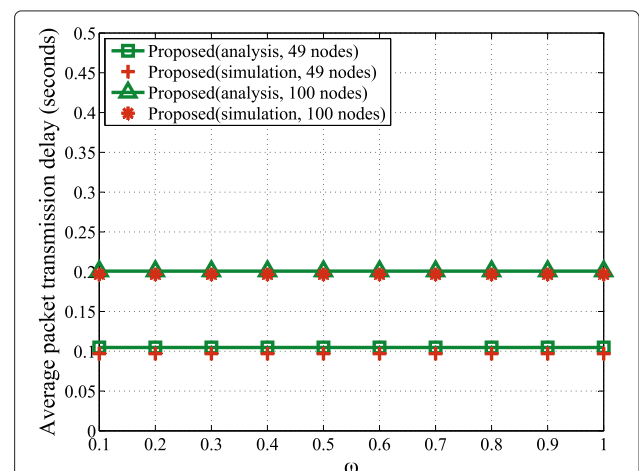


(a) DPA with different seed values (49 nodes)

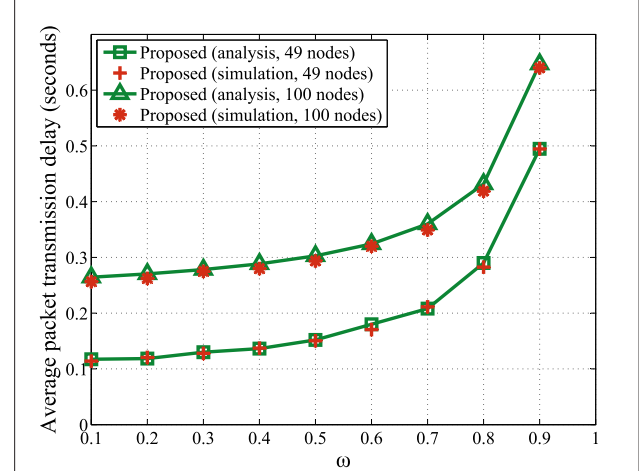


(b) NON-DPA with different seed values (49 nodes)

Figure 9 Packet transmission delay (seconds). (a) DPA with different seed values (49 nodes). (b) NON-DPA with different seed values (49 nodes).



(a) DPA



(b) NON-DPA

Figure 10 Average packet transmission delay (seconds). (a) DPA. (b) NON-DPA.

node generates one packet per two-frame. On the other hand, if the value of ω is 1, then each node generates one packet per one frame. Therefore, increasing the value of ω is the same as the increase of packet interarrival rate. For both DPA and NON-DPA cases, graph coloring shows intolerable delay for some values of ω at different seed values. For example, for seed 1, the multihop packet transmission delay is stable until ω is 0.6. In case of seed 4, the delay is stable until ω is 0.7. This is because the network using graph coloring is overloaded in the intermediate nodes. Therefore, the result graph has been drawn with different seed values.

In case of DPA, both scheduling schemes show a stable performance except in the intolerable cases. However, the proposed scheduling scheme shows a shorter multihop packet transmission delay, even though it starts the packet transmission with slightly greater frame length than that in case of graph coloring. As mentioned before, graph coloring shares slots (resources) for multiple flows in a link [9]. Therefore, an increase in the frame delay caused by the secondary queuing delay causes an increase in the multihop packet transmission delay. For the NON-DPA case, the proposed scheme shows slightly longer multihop packet transmission delay. The proposed scheduling scheme is more efficient when there is only one packet waiting for the packet transmission at the beginning of each frame. If there are more than two packets at the beginning of each frame, then all packets but one packet to be transferred in the current frame experience a long delay because of long frame length. When considering the NON-DPA case in the source node, each source node has the chance to see more than two packets; that is, the primary queuing delay may occur. Therefore, the proposed scheduling scheme shows a slightly lower delay performance for the NON-DPA case. However, the proposed scheme is more tolerable for a high packet interarrival rate than graph coloring.

Figure 10 shows the average multihop packet transmission delay of the proposed scheme with an increase of the packet interarrival rate. In Figure 10, the lines indicate the numerical results and the symbols indicate the simulation results. In this case, we average the results from ten different seeds. It can be observed that the analytic results are similar to the simulation ones for both cases.

8 Conclusions

This article proposed a new delay-efficient TDMA-based distributed scheduling scheme to eliminate the secondary queuing delay, which may occur in the conventional minimum length scheduling schemes. We derived the multihop packet transmission delay of the proposed scheduling scheme and validated it through a simulation. Finally,

we compared the performance of the proposed scheme with that of the conventional minimum length scheduling scheme that employs distance-2 graph coloring. The important contributions of this study are as follows:

- An intuitive method for eliminating the secondary queuing delay.
- Analysis of the delay effect caused by an allocation sequence of the TDMA slots.
- A distributed method to sequentially allocate slots on a path.
- Analysis of the proposed scheme and its simulation.

According to the simulation and analysis results, for the DPA case, the proposed scheme works well irrespective of the packet interarrival rate and outperforms the conventional graph coloring. However, in case of NON-DPA, the multihop packet transmission delay of the proposed scheme is slightly longer than that of the conventional graph coloring because the probability that each node has more than two packets increases at the beginning of the frame. However, the proposed scheduling scheme is more tolerant for a high packet interarrival rate.

9 Future study

In the future studies, first, we intent to extend the proposed scheduling scheme to an autonomous environment where either new nodes can efficiently assign time slots or existing nodes can release their slots on a path in a distributed manner. Second, the proposed scheme has the characteristics that each node allocates a different slot to each flow in a link. Therefore, it needs lots of slots; however, the distance-2 coloring leads to slightly smaller frame length because each node allocates one common slot for multiple flows in a link. This study have considered a situation wherein each node has only one flow. However, it is somewhat unrealistic. If two flows are considered per node, the frame length of the proposed scheme is up to two times greater than that when considering one flow per node. Accordingly, we are also interested in reducing the frame size and in concurrently reducing the network load by using conventional network coding (NC) schemes.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

"This research was supported by the Ministry of Knowledge Economy (MKE), Korea, under the Information Technology Research Center (ITRC) support program supervised by the National IT Industry Promotion Agency (NIPA)" (NIPA-2012-(C1090-1221-0011)), "This study had been supported by National GNSS Research Center program of Defense Acquisition Program Administration and Agency for Defense Development".

Received: 4 June 2012 Accepted: 5 November 2012
Published: 26 December 2012

References

1. V Gabale, B Raman, P Dutta, S Kalyanraman, A classification framework for scheduling algorithms in wireless mesh networks. *IEEE Commun. Surv. Tutor.* **PP**(99), 1–24 (2012)
2. K Jain, J Padhye, VN Padmanabhan, L Qiu, Impact of interference on multi-hop wireless network performance. *Wirel. Netw.* **11**, 4471–4487 (2005)
3. S Ramanathan, A unified framework and algorithm for channel assignment in wireless networks. *Wirel. Netw.* **5**(2), 81–94 (1999)
4. S Ramanathan, EL Lloyd, Scheduling algorithms for multihop radio networks. *IEEE/ACM Trans. Netw.* **1**(2), 166–177 (1993)
5. B Hajek, G Sasaki, Link scheduling in polynomial time. *IEEE Trans. Inf. Theory.* **34**(5), 910–917 (1988)
6. T Salonidis, L Tassiulas, in *Proc. ACM MobiHoc*. Distributed dynamic scheduling for end-to-end rate guarantees in wireless ad hoc networks (IL, USA, 2005), pp. 145–156
7. M Kodialam, T Nandagopal, Characterizing achievable rates in multi-hop wireless mesh networks with orthogonal channels. *IEEE/ACM Trans. Netw.* **13**(4), 868–880 (2005)
8. G Sharma, R Mazumdar, N Shroff, in *Proc. ACM Mobicom*. On the complexity of scheduling in wireless networks (CA, USA, 2006), pp. 227–238
9. S Gandham, M Dawande, R Prakash, in *Proc. IEEE INFOCOM*, vol. 4. Link scheduling in sensor networks: Distributed edge coloring revisited (TX, USA, 2005), pp. 2492–2501
10. S Sarkar, K Kar, in *Proc. 44th Annu. Allerton Conf. Communication, Control and Computing*. Achieving 2/3 throughput approximation with sequential maximal scheduling under primary interference constraints (IL, USA, 2006), pp. 5342–5347
11. M Sanchez, J Zander, T Giles, in *Proc. WPMC*, vol. 2. Combined routing and scheduling for spatialTDMA in multihop ad hoc networks (Honolulu, Hawaii, 2003), pp. 781–785
12. H-Y Wei, S Ganguly, R Izmailov, Z Haas, in *Proc. IEEE VTC Spring*, vol. 5. Interference-aware IEEE 802.16 WiMax mesh networks (Stockholm, Sweden, 2005), pp. 3102–3106
13. B Han, W Jia, L Lin, Performance evaluation of scheduling in IEEE 802.16 based wireless mesh networks. *Comput. Commun.* **30**, 782–792 (2007)
14. D Ghosh, A Gupta, P Mohapatra, in *Proc. Mobile Adhoc and Sensor Systems*. Admission control and interference-aware scheduling in multi-hop wimax networks (Pisa, Italy, 2007), pp. 1–9
15. P Djukic, S Valaee, Delay aware link scheduling for multi-hop tdma wireless networks. *IEEE/ACM Trans. Netw.* **17**(3), 870–883 (2009)
16. J Zou, D Zhao, Real-time cbr traffic scheduling in ieee 802.16-based wireless mesh networks. *Wirel. Netw.* **15**(1), 65–72 (2009)
17. G Narlikar, G Wilfong, L Zhang, in *Proc. INFOCOM*. Designing multihop wireless backhaul networks with delay guarantees (Catalunya, SPAIN, 2006), pp. 1–12
18. A Sahoo, P Goyal, in *Proc. COMSNETS*. A scheduling and call admission control algorithm for wimax mesh network with strict qos guarantee (Bangalore, India, 2010), pp. 20–29
19. V Gabale, A Chiplunkar, Raman Bhaskaran, Dutta Partha, in *Proc. COMSNETS*. Delaycheck: Supporting voice over multi-hop multichannel wireless mesh (Bangalore, India, 2011), pp. 1–10
20. IEEE 802.11s, *Draft Amendment: ESS Mesh Networking*, (2006)
21. I Rhee, A Warrier, J Min, L Xu, DRAND: distributed randomized TDMA scheduling for wireless ad hoc networks. *IEEE Trans. Mob. Comput.* **8**, 1384–1396 (2009)
22. C Zhu, MS Corson, in *Proc. IEEE INFOCOM*. A five-phase reservation protocol (FPRP) for mobile ad hoc networks (CA, USA, 1998), pp. 322–331
23. SJ Golestani, A framing strategy for congestion management. *IEEE J. Sel. Areas Commun.* **9**(7), 1064–1077 (1991)
24. R Rom, M Sidi, *Multiple Access Protocols: Performance and Analysis*. (Springer Verlag, New York, 1990)
25. S Dastangoo, TG Macdonald, D Reinharth, C Burns, in *Proc. MILCOM*. Performance analysis of distributed time division multiple access protocols in mobile ad hoc environments, (USA, 2009), pp. 1–7
26. S Gandham, M Dawande, R Prakash, S Venkatesan, in *Proc. IEEE Globecom*. Energy efficient schemes for wireless sensor networks with multiple mobile stations (San Francisco, USA, 2003), pp. 1108–1120
27. WR Heinzelman, A Chandrakasan, H Balakrishnan, in *Proc. HICSS*. Energy-efficient communication protocol for wireless microsensor networks (Hawaii, USA, 2000), pp. 3005–3014

doi:10.1186/1687-1499-2012-369

Cite this article as: Kim et al.: New delay-efficient TDMA-based distributed schedule in wireless mesh networks. *EURASIP Journal on Wireless Communications and Networking* 2012 **2012**:369.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com