



Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

OLSR-aware channel access scheduling in wireless mesh networks^{☆,☆☆}

Miray Kas^a, Ibrahim Korpeoglu^{b,*}, Ezhan Karasan^c

^a Department of Electrical and Computer Engineering, Carnegie Mellon University, United States

^b Department of Computer Engineering, Bilkent University, Turkey

^c Department of Electrical and Electronics Engineering, Bilkent University, Turkey

ARTICLE INFO

Article history:

Available online 2 December 2010

Keywords:

Cross-layer design

Spatial TDMA

OLSR

MAC

Centralized channel access scheduling

Distributed channel access scheduling

ABSTRACT

Wireless mesh networks (WMNs) have emerged as a key technology having various advantages, especially in providing cost-effective coverage and connectivity solutions in both rural and urban areas. WMNs are typically deployed as backbone networks, usually employing spatial TDMA (STDMA)-based access schemes which are suitable for the high traffic demands of WMNs. This paper aims to achieve higher utilization of the network capacity and thereby aims to increase the application layer throughput of STDMA-based WMNs. The central idea is to use optimized link state routing (OLSR)-specific routing layer information in link layer channel access schedule formation. This paper proposes two STDMA-based channel access scheduling schemes (one distributed, one centralized) that exploit OLSR-specific information to improve the application layer throughput without introducing any additional messaging overhead. To justify the contribution of using OLSR-specific information to the throughput, the proposed schemes are compared against one another and against their non-OLSR-aware versions via extensive ns-2 simulations. Our simulation results verify that utilizing OLSR-specific information significantly improves the overall network performance both in distributed and in centralized schemes. The simulation results further show that OLSR-aware scheduling algorithms attain higher end-to-end throughput although their non-OLSR-aware counterparts achieve higher concurrency in slot allocations.

© 2010 Elsevier Inc. All rights reserved.

1. Introduction

A wireless mesh network (WMN) is a multi-hop communication network in which the nodes are organized to form a mesh topology, providing communication over multiple wireless links. In the last decade, wireless mesh networking has emerged as a rapidly deployable network infrastructure to provide better services in wireless networks, especially in military and urban scenarios [28].

In the direction in which the wireless networks evolve, cross-layer networking is another increasingly important paradigm, and it is currently one of the most active research areas in wireless networking [47,44,20]. In cross-layer networking, the strict layered network implementation is relaxed and the knowledge is shared among loosely coupled layers through stricter cooperation in order to provide efficient allocation of network resources.

Improving the performance of multi-hop wireless mesh networks, especially in terms of the overall application layer throughput, is a very active research area. Our goal in this paper is to improve the end-to-end performance of multi-hop WMNs by means of cross-layer interaction between the MAC layer and the routing layer. The key idea is to unite spatial TDMA (STDMA) and optimized link state routing (OLSR) protocol. More specifically, we propose utilizing the information collected by OLSR in providing better channel access schedules and medium access coordination. In our approach, in order to increase the scalability of the MAC layer, we target low-overhead scheduling schemes that would exploit the readily available information without introducing any communication overhead.

In WMNs, typically FDMA-, CDMA- or TDMA-based mechanisms are employed for multiple access coordination as CSMA-based schemes are known to result in inferior performance in multi-hop networks with high traffic demands [51]. Given that the forthcoming WMN standards such as WiMAX [18] and 802.11s [17] consider TDMA-based MAC mechanisms and WMNs operate in multi-hop environments, in this paper, we focus on spatial TDMA (STDMA)-based schemes at the MAC layer.

At the network layer, we consider the use of OLSR as the routing protocol as it is one of the most widely used proactive MANET routing protocols in wireless ad hoc and mesh networks [11,31,46]. It is an optimized link state routing protocol where not all nodes

[☆] A preliminary version of this paper was presented at the IEEE Wireless Communications and Networking Conference (WCNC), Budapest, Hungary, 5–8 April 2009.

^{☆☆} This work is supported in part by European Union FP7 Framework Program FIRESENSE Project 244088.

* Corresponding author.

E-mail addresses: mkas@ece.cmu.edu (M. Kas), korpe@cs.bilkent.edu.tr (I. Korpeoglu), ezhan@ee.bilkent.edu.tr (E. Karasan).

flood the network with link state updates but only selected special nodes, called multi-point relay (MPR) nodes, disseminate the link state information. OLSR is used as the reference protocol for DARPA's tactical networks [8] and as the network layer protocol in Freifunk, which is a popular OLSR-based urban mesh network solution with several hundreds of users volunteering to join in order to form community mesh networks in many cities all around the world [46]. Additionally, OLSR is acknowledged by IETF as a promising MANET routing protocol as its second version is currently under development to provide more flexible and efficient signaling framework [9].

To the best of our knowledge, there has been no research done on STDMA-based distributed or centralized channel access scheduling schemes which employ OLSR-specific MPR information in a cross-layer manner in node scheduling to improve the application layer throughput. In our approach, we focus on acknowledging the importance of forwarding nodes for the whole network operation. In OLSR-based systems, MPR nodes are the forwarding nodes. In the proposed access scheduling schemes, we prioritize MPR nodes using our proposed MPR-based weighting scheme and demonstrate how this affects the overall system throughput through our simulation results.

In this paper, we make the following contributions.

- We propose a cross-layer weighting scheme called the MPR-based weighting scheme that makes use of OLSR-specific routing layer information with the central idea of giving more weight to the forwarding nodes, which are definitively identifiable in OLSR-based systems.
- We propose one centralized algorithm and one fully distributed pseudo-random channel access scheduling algorithm that primarily aim to improve the overall application layer throughput by means of cross-layer interaction. The algorithms proposed in this paper use our MPR-based weighting scheme and distinguish themselves from other cross-layer scheduling algorithms by using simple network layer information (i.e. MPR information), which is disseminated at no additional cost.
- In the current literature, there are many studies which only focus on maximizing the number of concurrent slot allocations to be able to maximize the overall throughput [48,27,45]. Through our simulation results, we show that maximizing the number of concurrent slot allocations does not necessarily maximize the application layer throughput, and OLSR-aware scheduling schemes perform significantly better than their non-OLSR-aware counterparts, although non-OLSR-aware algorithms achieve higher transmission concurrency.

The rest of the paper is organized as follows. In Section 2, we provide a brief background on STDMA and OLSR and review related studies in the literature. In Section 3, details of the targeted network model and MPR-based weighting scheme are presented. Section 4 focuses on centralized algorithms and presents OLSR-aware centralized scheduling along with its non-OLSR-aware version. Similarly, Section 5 presents the OLSR-aware distributed scheduling algorithm and its non-OLSR-aware version. In Section 6, details of the simulation implementation are given, and Section 7 reports the simulation results along with an in-depth qualitative analysis. Section 8 concludes the paper emphasizing the key insights derived from the simulation results.

2. Background and related work

The first part of this section starts with a short introduction of the OLSR protocol, followed by a discussion about the state-of-the-art research on OLSR and other cross-layer studies. In the second part, background on STDMA scheduling is provided and STDMA-related research in the literature is briefly reviewed. The third part notes the standing of our work in the literature, discussing its main differences from previous work.

2.1. OLSR

OLSR is a routing protocol which is mainly aimed at mobile wireless networks. OLSRv1 was developed at INRIA and standardized by IETF in RFC 3626 in 2003 [10]. Inherited from open shortest path first (OSPF) [30], OLSR provides an optimization of the classical link state routing protocol. In traditional link state routing protocols, the overhead introduced by the transmission of broadcast packets to inform all nodes about link states is quite high, and OLSR significantly reduces the overall messaging overhead by using multi-point relay (MPR) nodes.

A node is called an MPR node if it is chosen by one or more of its 1-hop neighbors to forward their messages. Each node keeps information about the nodes it has selected as its MPRs and the nodes that have selected it as MPR in its MPR Set and MPR Selector Set, respectively. The collection of MPR nodes forms a connected backbone, and MPR nodes are the only nodes which have to relay/forward messages.

In OLSR, each node also detects its neighborhood and periodically broadcasts HELLO messages that contain the list of its 1-hop neighbors. Therefore, each node gets the opportunity to learn about its 2-hop neighborhood. Nodes keep the information about their 1-hop and 2-hop neighbors in their Neighbor Set and 2-hop Neighbor Set, respectively. In other words, OLSR handles detection of 1-hop and 2-hop neighbors for each node, and this information is readily available for the use of scheduling algorithms with no additional overhead.

In the current literature, OLSR is examined in many studies, including several papers that compare different routing protocols' effects on the performance of wireless networks [7,4]. The results presented in these papers show that OLSR provides better performance in terms of data packet delivery ratio, throughput, packet latency and routing overhead when compared against other MANET routing protocols such as AODV [36], DSDV [37], and DSR [19], contributing to the popularity of OLSR.

Apart from these, there are also other studies in the literature that mainly focus on OLSR. [21] proposes a modified version of OLSR that uses the link cost values in the establishment of routes. Link cost value involves maximum signal strength (RSSI), link capacity and contention information. [1] and [33] discuss another modification to the OLSR protocol, aiming to introduce scalability into OLSR through the use of fish-eye routing techniques. [14] uses MPRs for estimating node positions in heterogeneous WMNs through anchoring and [39] proposes an adaptive multi-channel OLSR based on topology maintenance, while [34,35] investigate the effects of interference on OLSR.

Moreover, OLSR is also examined in different studies that address various aspects of cross-layering. For instance, in [15], the authors discuss different cross-layer methodologies aiming to improve the performance of OLSR using the information available at the MAC layer, whereas [28] aims to mitigate interference on OLSR STDMA-based networks.

2.2. STDMA scheduling

Allowing multiple nodes to transmit during the same time slot as long as they are on the non-conflicting parts of the network (i.e. sufficiently separated in space) is a key to increasing the performance of multi-hop WMNs. Spatial TDMA (STDMA) allows concurrent transmissions on the non-conflicting parts of the network during a given time slot so that the spatial reusability is exploited and the number of packets that can be delivered in a collision-free manner is maximized [32].

To enable collision-free concurrent transmissions in multi-hop WMNs consisting of nodes with single half-duplex radios, the scheduling algorithm should avoid two main types of conflict.

1. *Primary conflict* is observed if a node is scheduled to transmit and receive at the same time.
2. *Secondary conflict* is observed if a node is scheduled to receive from two different nodes simultaneously.

To ensure that both kinds of conflict are avoided, in STDMA-operated multi-hop networks, no two nodes within the same 2-hop neighborhood should be scheduled to transmit at the same time slot [24]. The scheduling algorithms designed for STDMA-operated multi-hop networks can be broadly classified as centralized and distributed algorithms.

Centralized scheduling algorithms commonly require global knowledge about the network topology, and they are run at a central site. In off-line centralized scheduling algorithms, the scheduling problem is solved once and for all, whereas in adaptive centralized scheduling algorithms the central scheduler site solves the scheduling problem dynamically to be able to adapt to the topology changes.

The proposed centralized algorithms usually construct a tree of nodes that has the central scheduler as its root or employ graph-theoretic solutions such as link/vertex coloring. The tree-based centralized scheduling algorithms might prefer inferring the routing tree from the MAC level control messages [6] or constructing it through interference and/or some other metric-based cost assignments [49] or using traditional minimum spanning tree algorithms [12] or line graph approaches [22]. Some of the most recent STDMA-based studies also consider interference mitigation [28,16] or quality of service (QoS) provisioning [23].

On the other hand, in order to decrease the vulnerability to topological changes and improve flexibility, distributed algorithms are considered essential. In fully distributed algorithms, each node can simultaneously run the algorithm, essentially resulting in parallel computation. In the literature, there are many proposed distributed channel access schemes which can be further classified as cluster-based, hybrid [40], randomized [3] or graph-theoretic [29] schemes.

2.3. Contributions to the literature

The idea we propose in this paper is different from those discussed in Section 2.1 [21,1,33,14,35,39,34] in the sense that we target the maximization of application layer throughput by exploiting the information readily available in OLSR, while [21,1,33,14,35,39,34] focus on OLSR and propose methods/extensions that primarily aim at improving the performance of OLSR in various contexts.

The second set of papers that our work is related to focus on scheduling at the MAC layer, each following a different topology. As cross-layer interaction to improve the end-to-end performance is a relatively new research area, papers such as [3,13,41] focus only on traditional algorithms, without looking into ways of improving performance through cross-layer interaction. On the other hand, more recent studies such as [6,22] focus only on a single layer of the network implementation, while we consider a holistic view of the system, including the interaction between two different layers and its high-level effects on the end-to-end performance.

In the current literature, there are also other papers that consider cross-layer interaction among different layers. Although [28,47,44,20] consider interaction among different layers to improve their performance metrics, they all entail complex calculations, while our proposed cross-layer scheduling approach makes use of simple calculations, and it has no extra overhead.

3. OLSR-aware channel access scheduling

In this section, we describe the network model the proposed algorithms are intended for, and discuss the cross-layer weighting scheme used in the proposed scheduling algorithms and the dissemination of the weight information.

3.1. Network model

In our study, we consider a multi-hop wireless mesh network which can be modeled as an undirected graph $G = (N, L)$, where N is the set of nodes and L is the set of undirected links connecting the nodes in N . Each node represents a wireless mesh node with a wireless communication range of R . Link $l(i, j)$ exists if and only if the distance between the nodes i and j is less than or equal to R , enabling communication from i to j and from j to i . Besides, there exists a set of flows (connections) F that are active in the network. Each flow is identified by a pair of source and destination nodes, and a route determined by OLSR. From this point onwards, the terms flow and connection are used interchangeably.

The targeted system operates in discrete (or slotted) time. In any time slot, nodes may attempt transmission. A packet transmission attempt is considered successful if the receiver of the transmission is not interfered with by a simultaneous transmission from another node within its interference range. The interference range of a node is usually much larger than its transmission range, and nodes more than two hops away may also be involved in its interference range [50]. However, we simplify the problem here by ignoring such cases and use a 2-hop interference model. The 2-hop interference model assumes that the interference between the nodes that are more than two hops away from one another in the physical topology is negligible [25].

The basic features of our network model can be listed as follows.

- Each node is uniquely identifiable.
- Node and time synchronization are available. However, the methods for achieving synchronization are out of the scope of this paper.
- A maximum-sized packet can fit into a time slot.
- The nodes are stationary, and no further maintenance is done after deployment.
- Communication is established via omni-directional antennas over a single physical radio channel.
- Each node in the network has a single half-duplex radio, and the nodes' radios are always on.
- Each node keeps a single packet queue, not differentiating the packets from different connections.
- Each node is eligible to generate traffic destined to any other node.
- The routing protocol is OLSR.

3.2. The MPR-based weighting scheme

The MPR-based weighting scheme is one of the most important factors that affect the performances of the proposed scheduling schemes. The MPR-based weighting scheme utilizes MPR information available at nodes due to use of the OLSR routing protocol. In [5], the authors show that, in most cases, 75% of all MPRs are elected in the first round. Since MPR selection is mandatory for route calculations, and the process converges quite quickly, we propose using MPR-related information in the MAC layer, within the slot allocation decision phase.

The key idea of this weighting scheme is to give more opportunities (time slots) to the nodes that are liable to carry the traffic generated by other nodes. In order to achieve this, we assign W_X , the weight of a node X , as in Eq. (1):

$$W_X = \text{Size}(\text{MPR Selector Set}(X)) + 1. \quad (1)$$

In a multi-hop mesh network, the nodes that forward data on behalf of other nodes carry more traffic than nodes which do not forward others' data, only dealing with their own traffic. The number of nodes in an MPR node's MPR Selector Set indicates the number of nodes which will possibly forward their incoming

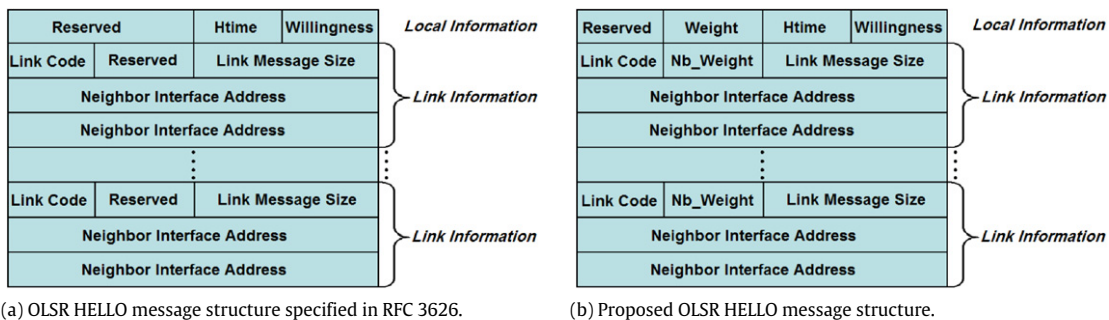


Fig. 1. OLSR HELLO message formats.

packets to that particular MPR. Without loss of generality, we assume that all nodes are eligible to generate traffic. For each node that has selected node X as MPR, 1 unit of weight is added to the weight of node X along with 1 unit of weight for node X itself. The number of slots assigned to node X , $Slots_X$, is proportional to its weight, W_X ; approximately proportional if a randomized algorithm is used; and exactly proportional if a deterministic algorithm is preferred. Eq. (2) shows the relationship between weights of nodes and their slot allocation.

$$\frac{Slots_X}{\sum_{k \in N} Slots_k} \propto \frac{W_X}{\sum_{k \in N} W_k}. \quad (2)$$

As MPR information is compulsory for route calculation, it is readily available and disseminated by the routing layer. This information can be collected by the MAC layer at *no* extra messaging cost. Apart from its having no extra overhead, the main reason for selecting such a weighting scheme is that it can be used to approximate the traffic passing through each node without traffic monitoring overhead, since all the nodes in the network are eligible to send packets to any other node in the network. Another reason for selecting the size of the MPR Selector Set as the weight indicator is that, if the network is not too mobile, once the network stabilizes, the MPR Selector Set will remain the same most of the time. Therefore, the weights calculated by (1) will mostly be stable.

3.2.1. Dissemination of weight information

OLSR exchanges periodic HELLO messages and collects 2-hop neighborhood and MPR information to be able to construct the routes (see Section 2.1). This mechanism can be extended easily to carry the weight information.¹

Fig. 1(a) presents the HELLO message structure specified in RFC 3626. The ‘Htime’ field holds the HELLO emission interval (HELLO_INTERVAL), the time until the next HELLO message transmission, and the ‘Willingness’ field defines the willingness of a node to carry or forward traffic on behalf of other nodes. ‘Link Code’ specifies information about a particular link. It is formed as the combination of Neighbor Type and Link Type. ‘Link Message Size’ specifies the message length between two consecutive ‘Link Code’ fields. Finally, ‘Neighbor Interface Address’ specifies the address of the neighbor node’s associated interface.

In the HELLO message structure specified in RFC 3626, the ‘Reserved’ fields are unused, and are filled with zeros. The ‘Reserved’ field within the local information section is 2 bytes long while the ‘Reserved’ field in the link information section is 1 byte long.

We propose extending this message structure to include weight information for the originating node itself and its listed 1-hop

neighbors. The proposed modified message structure is shown in Fig. 1(b). In the new message structure, the second half of the ‘Reserved’ field within the local information section is replaced with the ‘Weight’ field and the ‘Reserved’ field in the link information section is substituted with the ‘Nb_Weight’ field. In a single HELLO message, there is only one ‘Weight’ field, but there might be multiple ‘Nb_Weight’ fields, depending on the number of links advertised. Both the ‘Weight’ field and the ‘Nb_Weight’ field are 1 byte long. The ‘Weight’ field holds the weight information of the originating node. The ‘Nb_Weight’ field holds the weight information for the neighbor node associated with the advertised link.

Using this new HELLO message structure, every node is able to collect the weight information of all the nodes in its 2-hop neighborhood via the routing layer control messages without requiring the MAC layer to exchange any further messages. There is *no* extra overhead introduced by our proposed OLSR-aware schemes as the unused parts of HELLO messages are utilized for the dissemination of the weight information. In this way, we also ensure that there are no compatibility issues with the OLSR protocol as we only modify the unused ‘Reserved’ fields to embed scheduling related information.

4. OLSR-aware centralized channel access scheduling

In devising solutions for channel access scheduling, different forms of graph-coloring algorithms are widely used. Given an undirected graph $G = (N, L)$, *vertex coloring* is the assignment $\alpha : N \rightarrow C$ of colors (C) to vertices (N) such that no two adjacent vertices get the same color and the number of colors used is minimized [29,26]. Finding the minimum number of colors in this assignment process is shown to be NP-hard [42].

Since the slot assignment problem is NP-hard, there are several heuristics proposed to provide an approximate solution. Among these heuristics, First-Fit and Degree-Based Ordering are among the best-known solutions [2]. Algorithm 1 presents the most general form of the First-Fit Vertex Coloring Algorithm.

In the First-Fit Vertex Coloring Algorithm, there is a list associated with each color, holding the nodes that are already assigned to that color. Whenever an unassigned node i is to be assigned a color, the First-Fit Vertex Coloring Algorithm starts by checking the already assigned nodes list associated with each color and assigns the first suitable color j . A color j is called suitable for node i if node i does not conflict with any of the nodes that are already assigned color j . A new color is assigned to node i if all colors used so far are unsuitable. In the First-Fit Vertex Coloring Algorithm, no particular strategy is applied for the selection order of the nodes to be colored. On the other hand, in the Maximum Degree First (MDF) Vertex Coloring Algorithm, the vertex with the highest number of neighbors is selected first, providing an intuitively better coloring than the First-Fit Vertex Coloring Algorithm [2].

¹ OLSRv2 directly inherits the MPR and message structure from OLSRv1. Therefore, our mechanism works for both versions of OLSR.

Algorithm 1: First-Fit Vertex Coloring Algorithm

Data: Undirected graph $G = (N, L)$ where N is the set of nodes and L is the set of links connecting the nodes in N .

Result: Nodes in N are assigned colors such that no two conflicting nodes in N are assigned the same color.

```

1 for  $i \leftarrow 1$  to  $|N|$  do
2   foreach Color  $j$  do
3     if  $IsNonConflicting(assignedLst(j), i)$  then
4        $i.Color \leftarrow j$ ;
5        $assignedLst(j).Insert(i)$ ;
6       break;
7   end
8 end
9 end

```

On the other hand, *distance- d coloring* is a special form of vertex coloring. In distance- d coloring, the colors are assigned such that no two vertices of distance d or less share the same color. TDMA channel access scheduling using the 2-hop interference model reduces to distance-2 coloring when the time slots are perceived as colors and both the primary and the secondary conflicts need to be avoided.

Our OLSR-aware centralized scheduling scheme (OA-C) uses a modification of the Distance-2 Maximum Degree First Vertex Coloring Algorithm for slot allocation. As explained in Section 3.2, we argue that the size of the MPR Selector Set is a good predictor for the amount of traffic that can pass through a node, assuming that all nodes are eligible to generate traffic destined to any other node in the network. Therefore, in our OA-C scheme, we modify the Distance-2 Maximum Degree First Vertex Coloring Algorithm to integrate the MPR-based weighting scheme (see Algorithm 2). As a result, our OA-C algorithm has two major differences from the traditional Distance-2 Maximum Degree First Vertex Coloring Algorithm.

1. Each node $i \in N$ is associated with an MPR-based weight W_i and is assigned W_i time slots in a single scheduling cycle.
2. Nodes in N are sorted in a non-increasing order with respect to their MPR-based weights. In this way, the nodes whose assignments resolve more conflicts (both primary and secondary) are assigned first and the nodes that are assigned later are less likely to require new slots, resulting in a smaller scheduling cycle length.

In Algorithm 2, ($cycle_count + 1$) is the number of different slots that are used to schedule all nodes in the network, in other words, the length of the scheduling cycle. The resulting frame size for the centralized scheduling scheme varies. Depending on the network conditions, OA-C can be configured to run at the end of every frame so that the scheduling mechanism responds to the topological changes in the network (e.g. a new node entering the network) in a timely manner.

In Algorithm 2, we make use of a subfunction called *IsFeasible*, which is presented in Algorithm 3. The *IsFeasible* function ensures that no other nodes within the 2-hop neighborhood of the given node n_id are scheduled to transmit at the given time slot $slot_number$. In the function, N_{1,n_id} and N_{2,n_id} represent the 1-hop and 2-hop neighbors of node n_id , respectively.

On the other hand, the non-OLSR-aware centralized scheduling algorithm (NOA-C), which is the non-OLSR-aware version of the OA-C algorithm, does not make use of the MPR-based weighting scheme. NOA-C makes use of the above-mentioned Distance-2 Maximum Degree First heuristic which is widely used/extended in many studies [2,38]. In Algorithm 4, we present a high-level description of how we implement this algorithm in our framework so that a fair comparison of the scheduling schemes discussed (OA-C and NOA-C) is possible.

Algorithm 2: OLSR-Aware Centralized Scheduling

Data: Undirected graph $G = (N, L)$ where N is the set of nodes and L is the set of links connecting the nodes in N .

Data: W : Weight vector.

Result: Each node in i in N is assigned W_i many slots such that no two nodes within the same 2-hop neighborhood are assigned the same slots.

```

1  $N \leftarrow Sort(N, W, Nonincreasing)$ ;
2  $cycle\_count \leftarrow 0$ ;
3 for  $i \leftarrow 1$  to  $|N|$  do
4    $j \leftarrow 1$ ;
5   while  $j < W_i$  do
6      $count \leftarrow 0$ ;
7     while  $IsFeasible(i, count) = FALSE$  do
8        $count ++$ ;
9     end
10     $slots[count].Add(i)$ ;
11    if  $count > cycle\_count$  then
12       $cycle\_count \leftarrow count$ ;
13    end
14     $j ++$ ;
15  end
16 end

```

Algorithm 3: IsFeasible Function

Data: n_id : Node Identifier.

Data: $slot_count$: Slot Number.

Result: Returns a boolean value indicating whether the slot $slot_count$ is feasible for the node n_id .

```

1  $nbr\_index \leftarrow -1$ ;
2 if  $slots[slot\_count].Empty()$  then
3   return TRUE;
4 end
5  $nbrLst \leftarrow N_{1,n\_id} \cup N_{2,n\_id} \cup n\_id$ ;
6 foreach  $nbr \in nbrLst$  do
7   if  $slots[slot\_count].Contains(nbr)$  then
8     return FALSE;
9   end
10 end
11 return TRUE;

```

Algorithm 4: Non-OLSR-Aware Centralized Scheduling

Data: Undirected graph $G = (N, L)$ where N is the set of nodes and L is the set of links connecting the nodes in N .

Data: S : Neighborhood size vector.

Result: Nodes in N are assigned time slots such that no two nodes within the same 2-hop neighborhood are assigned the same slot.

```

1  $N \leftarrow Sort(N, S, Nonincreasing)$ ;
2  $cycle\_count \leftarrow 0$ ;
3 for  $i \leftarrow 1$  to  $|N|$  do
4    $count \leftarrow 0$ ;
5   while  $IsFeasible(i, count) = FALSE$  do
6      $count ++$ ;
7   end
8    $slots[count].Add(i)$ ;
9   if  $count > cycle\_count$  then
10     $cycle\_count \leftarrow count$ ;
11  end
12 end

```

5. OLSR-aware distributed channel access scheduling

In this section, we present our OLSR-aware distributed scheduling algorithm (OA-D). In OA-D, each node determines the time slots it will use for transmission based on the information about its

1-hop and 2-hop neighbors and their respective weights collected by OLSR. It is a pseudo-random weighted channel access scheme which requires no schedule negotiation messages and no negotiation delay. Since all the nodes have consistent data about their 2-hop neighborhood and their respective weights, nodes can run their algorithms without having to wait for their neighbors' approval signals.

For this access scheme, the number of slots in a frame is fixed ($FRAME_SIZE$), and OA-D is independently run by each node i at the end of every frame in order to select the slots it is eligible to transmit during the next frame. OA-D is presented as Algorithm 5.

Algorithm 5: OLSR-Aware Distributed Scheduling

Data: Topology and weight information for 2-hop neighborhood of node i .
Result: The set of time slots node i is eligible to transmit during the next frame.

```

1 localLst ← FormLocalAgents(i);
2 nbrLst ← FormNeighborAgents(i);
3 contenders ← nbrLst ∪ localLst;
4 for j ← 1 to FRAME_SIZE do
5   slotID ← FormSlotID(FrameCount, j);
6   res_set ← MeshElection(slotID, contenders);
7   winner ← FindMax(res_set);
8   if localLst.Contains(winner) then
9     slots[j].status ← WON;
10 end
11 end
```

In OA-D, schedule formation is a pseudo-randomized process in which each node i generates W_i agents. All agents of node i compete for winning time slots on its behalf, and a similar case holds for all agents. Therefore, nodes with a higher number of agents (e.g. a crowded agent group) have higher chances of winning a certain slot. Each agent of node i is assigned an *agentID*. The *AgentID* is formed as the concatenation of the node identifier and a number from 0 to $W_i - 1$. In the first two steps of the algorithm, *agentIDs* for the hosting node's agents and the *agentIDs* of its 1-hop and 2-hop neighbors' agents are generated and put into *localLst* and *nbrLst*, respectively. All the agents generated in these two steps are involved in all contentions held throughout the frame.

In the for loop, a separate contention is held for each time slot in a frame. The *MeshElection* function returns a set of pairs, where each pair involves the *agentID* and its corresponding *SmearValue*, which is described below. The agent with the largest *SmearValue* is the winner of the contended time slot. If the winner agent's *agentID* belongs to *localLst*, then the node marks the slot as one of the slots it is eligible to transmit (i.e. sets the slot's status to 'WON').

The *MeshElection* function in OA-D is adapted from the *MeshElection* algorithm specified in the 802.16-2004 standard [18] as a part of the distributed EBTT mechanism which is responsible for the allocation of control slots such that the control messages are transmitted in a collision-free manner in a 2-hop neighborhood without requiring explicit schedule negotiation. *MeshElection* function's first parameter, *slotID*, is formed by the *FormSlotID* function as the concatenation of the contended frame count, *FrameCount*, and the contended slot number, j . The *SmearValue* is obtained as

$$SmearValue = smear(agentID \wedge slotID), \quad (3)$$

where the *smear* function is the hashing function given in the 802.16-2004 standard [18] which converts a uniform value to an uncorrelated uniform hash value, through the use of mixing. The *smear* function uses only simple arithmetic operations. We preferred using the *smear* function over a random number generator as it can be computed very quickly in practice.

Recall that the weight of each node is calculated via Eq. (1). Each node might have at least 0 and at most $N - 1$ nodes in its MPR Selector Set, where $|N|$ is the number of nodes in the network. This also implies that the weight of any node remains within the range $[1, N]$. Therefore, the worst-case complexity of the *MeshElection* function is $O(N^2)$ and the worst-case complexity of OA-D is $O(FRAME_SIZE * N^2)$, where $FRAME_SIZE$ is the number of slots in a frame.

For comparison purposes, we also present the non-OLSR-aware distributed scheduling algorithm (NOA-D) as Algorithm 6, which does not use the MPR-based weighting scheme. In NOA-D, all node weights are equal to 1.

Algorithm 6: Non-OLSR-Aware Distributed Scheduling

Data: Topology information for 2-hop neighborhood of node i .
Result: The set of time slots node i is eligible to transmit during the next frame.

```

1 nbrLst ← FormNeighborAgents(i);
2 contenders ← nbrLst ∪ agentID;
3 for j ← 1 to FRAME_SIZE do
4   slotID ← FormSlotID(FrameCount, j);
5   res_set ← MeshElection(slotID, contenders);
6   winner ← FindMax(res_set);
7   if agentID = winner then
8     slots[j].status ← WON;
9   end
10 end
```

6. Simulation implementation

In this section, detailed information about the simulation implementations of the schemes discussed is given. All schemes discussed in this paper are implemented in the ns-2.31 environment as MAC classes. The implementation of each scheme is composed of two parts: (1) the implementation of the required changes in the OLSR module, (2) the implementation of the proposed MAC scheme.

For OLSR implementation, we use UM-OLSR-0.8.8 for ns-2.31, as it is compliant with RFC 3626 and provides MAC layer feedback support, which is useful in detecting lost links [43]. We replace the RFC 3626 specified HELLO message structure (Fig. 1(a)) with our proposed HELLO message structure (Fig. 1(b)) in order to disseminate the weight information of the originating node as well as its known neighbors'. In addition, we extend the OLSR table structures to hold the weight information.

At the MAC layer, we took a basic non-concurrent TDMA-based MAC protocol as the starting point, which comes with ns-2 implementations from ns-2.23 onwards. This protocol does not support concurrent transmissions; hence it does not exploit the spatial reusability available in multi-hop environments. In the implementation of this protocol, each TDMA frame contains data transmission slots, where the number of data transmission slots in a frame is equal to the number of nodes in the network. During each frame, each node takes its turn once even if it has no data to send.

However, this implementation has obvious drawbacks as it produces very low throughput, since it does not take the slot reusability and the traffic into account. It also does not model centralized scheduling accurately, as there is no central controlling node which dictates the schedules of the remaining nodes.

Building on this non-concurrent TDMA MAC implementation in ns-2, we implemented our distributed and centralized channel access schemes as MAC protocols, allowing multiple nodes to transmit concurrently.

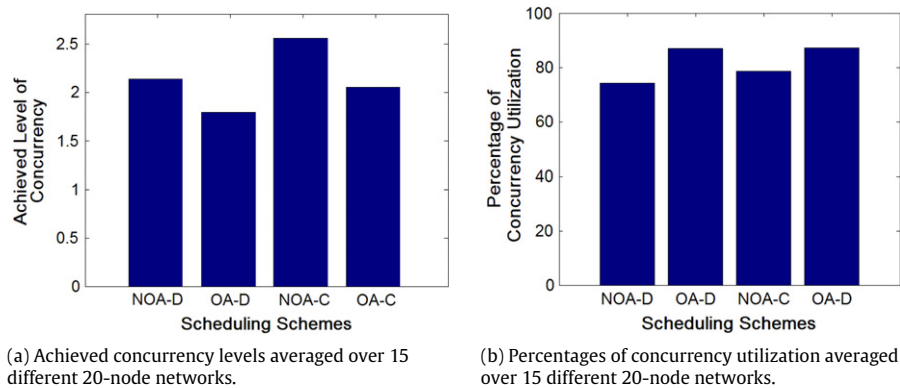


Fig. 2. Results on achieved/utilized concurrency levels.

Table 1

Ns-2 simulation parameters.

Parameter	Value
OLSR parameters (RFC)	
Hello interval	2 s
TC interval	5 s
MAC parameters	
Bandwidth	3 Mbps
Max packet length	1500 bytes
Frame size	50 slots
Traffic parameters	
Packet size	200 bytes
Traffic generation rate	50–700 bps

7. Simulation results and analysis

In this section, we report our ns-2-based simulation results and provide comparisons for the following scheduling schemes.

1. OLSR-Aware Centralized Scheduling (OA-C).
2. Non-OLSR-Aware Centralized Scheduling (NOA-C).
3. OLSR-Aware Distributed Scheduling (OA-D).
4. Non-OLSR-Aware Distributed Scheduling (NOA-D).

We define the following performance metrics and present simulation results that illustrate how these metrics change under both uniform and nonuniform traffic patterns while the packet generation rate, network size and the number of active flows in the network change.

1. *Packet delivery ratio*: The packet delivery ratio is calculated as the ratio of the number of packets delivered at the application layer to the number of packets generated at the application layer for the whole network, which is given by Eq. (4).

$$\text{Packet Delivery Ratio} = \frac{\text{Packets Delivered}}{\text{Packets Generated}} \quad (4)$$

This metric can be considered as the *normalized throughput* (normalized to the generated traffic). A network that can provide higher capacity will carry more traffic without loss or the same amount of traffic with a smaller loss rate compared to a network that provides lower capacity. Therefore, if a scheme performs well in terms of this metric, it means that the scheme provides good throughput.

2. *Average end-to-end delay*: End-to-end delay is the time taken for a packet to be transmitted across a network from source to destination. The average end-to-end delay is calculated for all packets that are successfully received at the application layer by the destination nodes.

Table 1 lists several parameters and their values used in our ns-2 simulations.

7.1. Concurrency levels achieved

Before presenting our simulation results with uniform/nonuniform traffic distributions, we discuss the average concurrency levels achieved by different scheduling schemes, which are presented in Fig. 2(a). Concurrency is defined as the average number of nodes that are able to transmit concurrently without conflicting, and it is calculated using Eq. (5). In Eq. (5), S denotes the set of time slots for the whole simulation time, while $|S|$ represents the size of S (i.e. the total number of time slots). $S[k].size()$ is the number of nodes that are *allowed* to transmit during the time slot k . However, allowing more nodes to transmit concurrently during the same slot does not necessarily increase the end-to-end throughput, since all the nodes allowed to transmit concurrently during a particular time slot do not necessarily have packets to transmit during that slot.

$$\text{Concurrency} = \frac{\sum_{k=1}^{|S|} S[k].size()}{|S|} \quad (5)$$

There are two main factors that affect the concurrency levels achieved by the scheduling schemes discussed.

1. Computation method (distributed/centralized).
2. Weighting scheme (MPR-based/uniform with no weights).

The computation method has a significant impact on the level of concurrency achieved by the scheduling schemes discussed, mostly because they use different levels of information. In centralized schemes, the central scheduler exploits global information, while in fully distributed schemes, the nodes make decisions using only the local information available.

In both of the distributed schemes (OA-D and NOA-D), each node locally runs an election (e.g. *MeshElection*) independent from the other nodes. Taking the topology in Fig. 3(a) as a reference, consider the scenario in which *node 9* loses to *node 7* because *node 7* has a larger *SmearValue* for the time slot T_s . In this situation, *node 9* refrains from transmission during T_s . For the same time slot, *node 7* also runs an independent local election, and assume that *node 12* has a larger *SmearValue* than that of *node 7*. In this case, *node 7* also refrains from transmission during T_s . Indeed, *node 9* and *node 12* could have transmitted concurrently as the distance between these two nodes is more than two hops. However, since they are more than two hops away from each other, none of them can predict what the other node's *SmearValue* will be. In contrast, centralized scheduling schemes are able to resolve such problems and prevent unnecessary refrainment. Hence, centralized scheduling schemes provide higher levels of concurrency than their distributed counterparts, which can be observed in Fig. 2(a).

Secondly, the selected weighting scheme (MPR based or uniform with no weights) also has a significant impact on the achieved

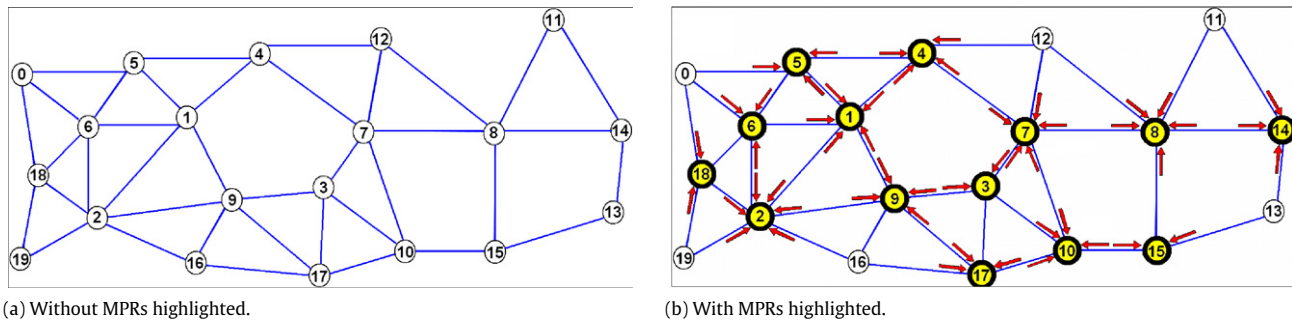


Fig. 3. Sample topology for a 20-node network.

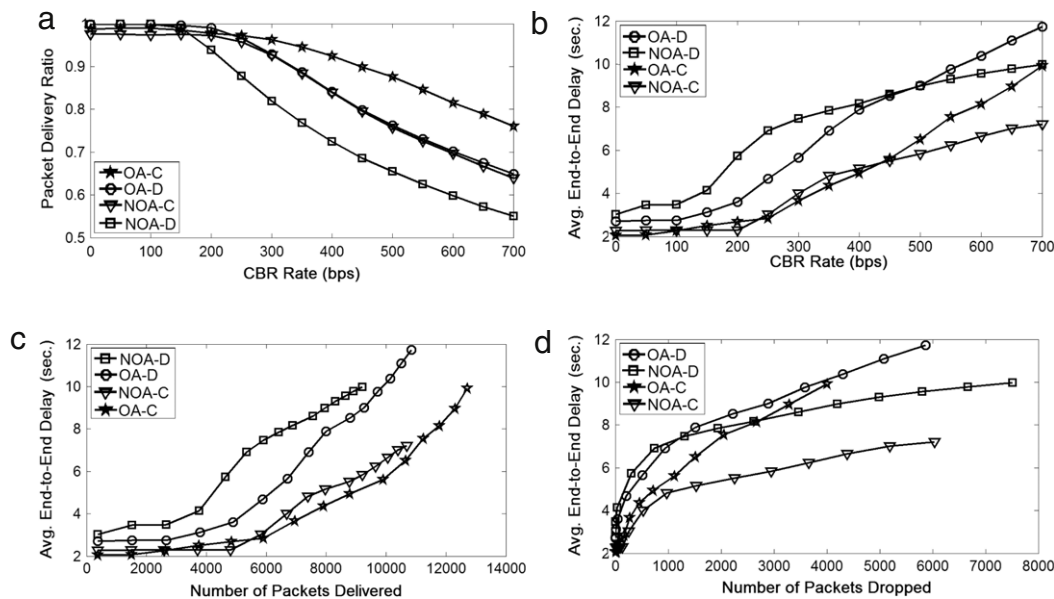


Fig. 4. Averaged uniform traffic simulation results on 20-node networks.

concurrency levels, as presented in Fig. 2(a). The nodes with large 2-hop neighborhoods can suppress the transmissions from many other nodes. Such nodes are usually selected as MPR nodes in OLSR-enabled networks as they are eligible to directly reach more nodes. In scheduling schemes using MPR-based weighting, these nodes have more weight and thus win more slots, leading to a decrease in the average number of nodes that can transmit concurrently during a time slot when compared to scheduling schemes not using any weighting scheme.

However, as discussed in the following subsections, the level of concurrency is not the only metric which determines the amount of traffic that can be delivered. In other words, there might be nodes provided with more concurrent transmission opportunities than they actually need. This situation causes the achieved level of concurrency to appear higher, although the expected performance gain in terms of the number of packets delivered successfully is not obtained because of not utilizing the allocated slots.

In Fig. 2(b), the percentages of slot utilization are presented. The presented values are calculated as the average of slot utilization under the same uniform scenario over 15 different 20-node networks. In the simulated scenarios, we used high traffic rates in order to ensure that most of the nodes will have packets in their queues most of the time, so that the impact of the scheduling scheme will become more accurately visible. Fig. 2(b) shows that OLSR-aware schemes improve the utilization of the allocated time slots over their non-OLSR-aware versions by around 8%–13%.

7.2. Simulation results with uniform traffic scenarios

The four scheduling schemes (OA-C, NOA-C, OA-D and NOA-D) are simulated and compared under the same uniform traffic scenarios in which every node generates a connection to every other node in the network using CBR traffic, resulting in $O(n^2)$ connections. The packet generation rates (in bps) and the packet sizes (in bytes) are kept the same for all connections in a single scenario, and different packet generation rates are applied over different simulation scenarios.

In Fig. 4, we report our uniform traffic simulation results. We present the average performance metrics achieved by the scheduling schemes, where the average is taken over 15 randomly generated connected topologies each with 20 nodes. Fig. 3(a) depicts a sample topology. In Fig. 3(b), the MPR nodes are highlighted, and arrows directed towards MPR nodes from each of their selectors are included.

Fig. 4(a) presents the packet delivery ratios (normalized throughput values) of the scheduling schemes discussed. In Fig. 4(a), it is observed that OLSR-aware schemes can deliver more packets (i.e. have less loss) than their non-OLSR-aware counterparts at the cost of higher delay, especially when the network load increases. The higher delay is, however, observed due to the fact that OLSR-aware scheduling schemes are able to deliver more packets even if they are delayed while non-OLSR-aware scheduling schemes have to drop them. On the other hand, Fig. 4(b)–(d) illustrate the delay behaviors of the scheduling schemes discussed. In Fig. 4(b), the average end-to-end delay turns out to be less in the

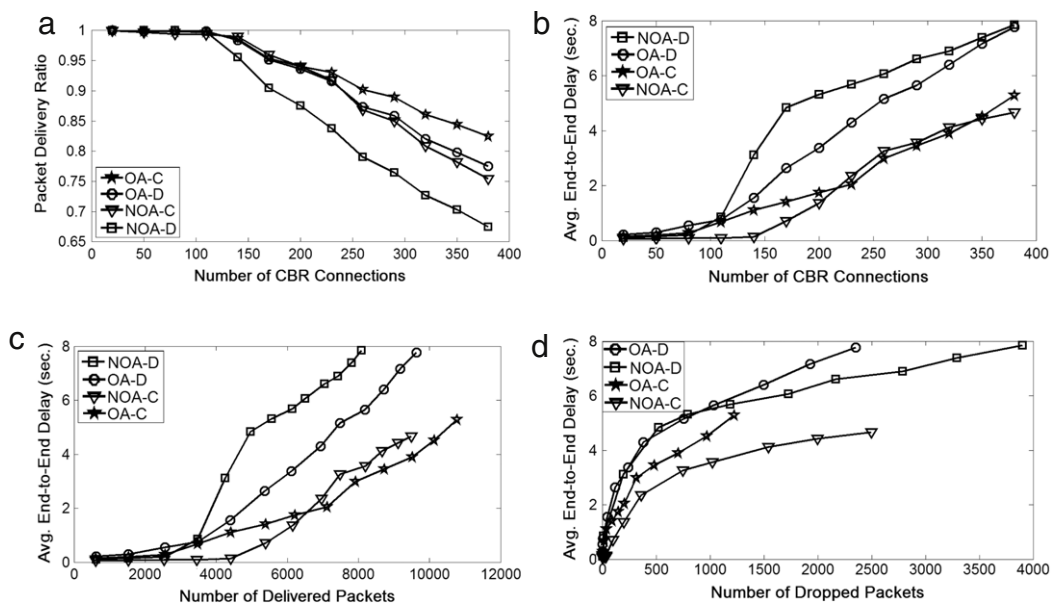


Fig. 5. Averaged nonuniform traffic simulation results on 20-node networks.

centralized schemes than it is in the distributed schemes. In addition, when the distributed and centralized cases are examined separately, in both cases we see that non-OLSR-aware algorithms exhibit a slower increase in the average end-to-end delay than OLSR-aware algorithms. This is due to the fact that the average end-to-end delay is calculated only for the packets that could be delivered successfully.

The results presented in Fig. 4(c) and (d) demonstrate the trade-off between the number of packets dropped/delivered and the average end-to-end packet delay. The data presented in Fig. 4(c) and (d) show how much each scheduling scheme can achieve in terms of absolute number of delivered and dropped packets within the given simulation duration. In Fig. 4(c), NOA-D has no data point after the 8000–10 000 range on the x-axis because it is unable to deliver that many packets, while the number of packets that OA-C can deliver goes up to the 12 000–14 000 range. When the end-to-end delays achieved by the distributed scheduling algorithms NOA-C and OA-C shown in Fig. 4(c) are considered, we observe that OA-C attains lower end-to-end delay compared to NOA-C for a given number of delivered packets. On the other hand, it can be observed from Fig. 4(d) that for a given end-to-end delay OA-C drops fewer packets compared to NOA-C. Similar arguments apply for the comparison of NOA-D and OA-D.

There are two main conclusions that can be drawn from the simulation results presented in Fig. 4. First, the centralized schemes outperform their distributed counterparts. Second, the OLSR-aware algorithms that use the MPR-based weighting scheme perform better than their non-OLSR-aware versions in terms of the packet delivery ratio in both the distributed and the centralized cases. The trends in packet delivery ratios are also observable in the absolute amount of traffic delivered by each scheduler algorithm.

In this respect, the results comply both with our expectations regarding the benefits of the proposed MPR-based weighting scheme and the simulation results on the respective concurrency levels achieved by different scheduling schemes.

The results presented in Fig. 4 also justify our argument that the level of concurrency is not the only metric affecting the end-to-end throughput and delay since OLSR-aware algorithms can deliver more packets despite the fact that non-OLSR-aware algorithms provide more concurrency. The level of concurrency achieved loses its importance when the transmission opportunities are not provided to the nodes that can utilize them effectively, as they are wasted.

7.3. Simulation results with nonuniform traffic scenarios

In this section, we present simulation results under a number of different nonuniform traffic scenarios in which some nodes either create or receive more traffic than the other nodes. In this set of simulations, CBR traffic is used, and CBR rate is fixed at 500 bps. Each simulation lasts for 200 s. All the connections start at sometime between 25th and 50th second and end at some time between the 125th and 150th second. The connection pairs are chosen randomly in each single simulation, and the number of connections is changed with a step size of 30.

The related simulation results (plotted in Fig. 5) represent the average of simulation results where the average is obtained using 15 different 20-node networks. Fig. 5(a) shows the packet delivery ratio while Fig. 5(b)–(d) depict the delay behaviors of the four scheduling schemes discussed.

All the results presented in Fig. 5 were obtained while the number of active connections in a network is changed. As the number of active connections in a network is increased, since the CBR rate is fixed, the load of the network increases. When the number of connections is 380 (19 × 20) for a 20-node network, the simulation scenario is very close to the uniform scenario, except for the start and end times of the connections.

The nonuniform traffic scenarios' results are in line with the results obtained under uniform traffic scenarios, and can be interpreted with similar reasoning. However, nonuniform traffic simulation results are more important than uniform traffic results for two reasons: (1) the tested traffic pattern is more realistic than the uniform traffic patterns, as the source and destination pairs are picked randomly and the load of the network is increased dynamically over time; (2) the performance improvement obtained by employing the MPR-based weighting scheme within the scheduling procedure shows that using the MPR-based weighting scheme and thereby prioritizing forwarding nodes in the slot allocations is useful not only in uniform traffic patterns, but also in general traffic patterns.

7.4. The effects of network size

In order to study the effects of changing the network size, we extended our simulation results with general traffic patterns

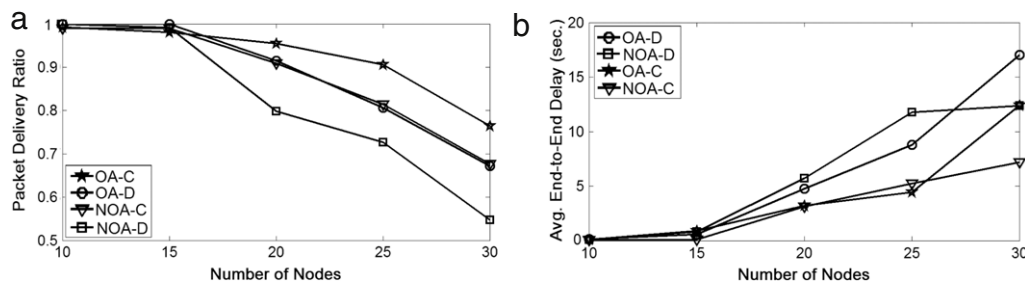


Fig. 6. The effects of network size.

(i.e. nonuniform traffic scenarios). In this set of experiments, the CBR rate is fixed (set to 500 bps) while the number of nodes is changed between 10 and 30 with a step size of 5. For each network size, we created 10 different random topologies and averaged their results. Fig. 6(a) and (b) illustrate how the packet delivery ratio and the average end-to-end delay, respectively, change while the network size changes.

In the results presented, OA-C again performs better than the other schemes in terms of the number of delivered/dropped packets and the packet delivery ratio, while NOA-D is the scheme with the lowest performance. In addition, the delay results are also consistent with the uniform/nonuniform traffic simulation results reported in earlier sections. The distributed schemes incur more delay than the centralized schemes, and the delays for the OLSR-aware algorithms tend to increase more steeply when compared against their non-OLSR-aware versions, since they can deliver more packets within the given simulation time.

8. Conclusions and key insights

This paper proposes an MPR-based cross-layer weighting scheme, two routing layer (OLSR)-aware STDMA-based channel access schemes (one distributed, one centralized), and provides detailed performance comparisons for the scheduling schemes discussed throughout the paper. Considering the results presented throughout the paper, several comments are in order.

- Considering our simulation results, using the MPR-based weighting scheme in the slot allocation procedure improves the total number of packets that can be delivered successfully at the application layer, that is, the total application layer throughput. In a multi-hop wireless mesh network, to be able to improve the overall application layer throughput, the nodes that forward more data should be given more transmission opportunities. In OLSR, this can be done by giving more weight (priority) to MPR nodes. In other words, MPR information provides a good approximation for the expected traffic, and granting more transmission opportunities to MPR nodes improves the overall performance of the network.
- Many studies in the literature focus on maximizing the number of concurrent allocations at the link layer [48,27,45]. On the other hand, maximizing the application layer throughput is more difficult, usually requiring dynamic calculations. Therefore, simpler methods such as handling the problem of maximizing the application layer throughput via the maximization of concurrent allocations at the link layer are very commonly preferred. However, the achieved level of concurrency is not the only metric that affects the overall application layer throughput. In our simulation results, we show that maximizing the number of concurrent slot allocations at the link layer does not necessarily increase the application layer throughput.

- Our simulation results also confirm that it is possible to achieve higher concurrency with a deterministic centralized algorithm compared with a pseudo-random distributed algorithm. In our simulations, the average concurrency levels achieved by the centralized schemes improve over the distributed schemes' concurrency levels by approximately 15% in OLSR-aware schemes and 20% in non-OLSR-aware schemes.
- The idea proposed in this paper explores centralized/distributed ways of adapting cross-layer information to improve the application layer throughput by only using a simple network layer parameter (i.e. MPR information in OLSR) in a very simple way. MPR information remains static as long as the network topology is static. However, through the use of periodic HELLO messages, MPR information is seamlessly adjusted to handle mobility. Since we perform cross-layer scheduling only by making use of a single parameter which is disseminated at *no extra overhead*, in terms of the implementation complexity, our work becomes advantageous over other cross-layer studies in the literature that use complex dynamic calculations.

References

- [1] C. Adjih, E. Baccelli, T. Clausen, P. Jacquet, G. Rodolakis, Fish eye OLSR scaling properties, in: *Mobile Ad Hoc Wireless Networks*, IEEE Journal of Communications and Networks (JCN) 6 (4) (2004) 343–351 (special issue).
- [2] H. Al-Omari, K.E. Sabri, New graph coloring algorithms, *American Journal of Mathematics and Statistics* 2 (4) (2006) 739–741.
- [3] L. Bao, J. Garcia-Luna-Aceves, A new approach to channel access scheduling for ad hoc networks, in: *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, 2001, pp. 210–221.
- [4] B.R. Arun Kumar, L. Reddy, P. Hiremath, Performance comparison of wireless mobile ad-hoc network routing protocols, *International Journal of Computer Science and Network Security* 8 (6) (2008) 337.
- [5] A. Bussan, N. Mitton, E. Fleury, An analysis of the multi-point relays selection in OLSR (RR-5468), Tech. Rep., Institut National De Recherche En Informatique Et En Automatique, 2005.
- [6] S. Cheng, P. Lin, D. Huang, S. Yang, A study on distributed/centralized scheduling for wireless mesh network, in: *International Conference on Communications and Mobile Computing*, 2006, pp. 599–604.
- [7] J. Chen, Y. Lee, D. Maniezzo, M. Gerla, Performance comparison of AODV and OLSR in wireless mesh networks, in: *Proceedings of the IFIP Med-Hoc-Net Conference*, 2006, pp. 14–17.
- [8] T. Clausen, et al. High performance communications, online. Accessed from: <http://hipercom.thomasclausen.net>.
- [9] T. Clausen, C. Dearlove, P. Jacquet, The optimized link state routing protocol version 2, in: *IETF Internet Draft*. <http://ietfreport.isoc.org/idref/draft-ietf-manet-olsrv2/>.
- [10] T. Clausen, P. Jacquet, Optimized link state routing protocol (OLSR): RFC 3626, in: *IETF Internet Draft*. <http://www.ietf.org/rfc/rfc3626.txt>.
- [11] S.M. Das, H. Pucha, K. Papagiannaki, Y.C. Hu, Studying wireless routing link metric dynamics, in: *IMC'07: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, 2007, pp. 327–332.
- [12] P. Djukic, S. Valaee, Centralized scheduling algorithms for 802.16 mesh networks, in: *WiMax/MobileFi: Advanced Research and Technology*.
- [13] Y. Du, Y. Bao, J.J. Garcia-Luna-Aceves, A history-based scheduling protocol for ad-hoc networks, in: *Proceedings of the 12th International Conference on Computer Communications and Networks*, 2003, pp. 223–228.
- [14] E. Ermel, P. Muhlethaler, Using OLSR multipoint relays (MPRs) to estimate node positions in a wireless mesh network, Tech. Rep., Institut National De Recherche En Informatique Et En Automatique, 2006.
- [15] Y. Gan, S. Masson, G. Guibe, B. Marin, C. Le Martret, Cross-layer optimization of OLSR with a clustered MAC, in: *Military Communications Conference, MILCOM*, 2006, pp. 1–7.
- [16] A. Gore, S. Jagabathula, A. Karandikar, C. Tanguy, On high spatial reuse link scheduling in STDMA wireless ad-hoc networks, in: *GLOBECOM*, 2007, pp. 736–741.

- [17] IEEE 802.11s Task Group, Draft Amendment to Standard for Information Technology—Telecommunications and Information Exchange Between Systems—LAN/MAN Specific Requirements—Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment: ESS Mesh Networking, IEEE P802.11s/D1.0, IEEE Std 802.11s-2006.
- [18] IEEE standard for local and metropolitan area networks part 16: air interface for fixed broadband wireless access systems, IEEE Std 802.16-2004, 2004, pp. 1–895.
- [19] D. Johnson, D. Maltz, J. Broch, et al., DSR: the dynamic source routing protocol for multi-hop wireless ad hoc networks, *Ad Hoc Networking* 5 (2001) 139–172.
- [20] A. Khreishah, C. Wang, N. Shroff, Cross-layer optimization for wireless multihop networks with pairwise inter-session network coding, *IEEE Journal on Selected Areas in Communications* 27 (5) (2009) 606–621.
- [21] K. Kowalik, B. Keegan, M. Davis, Making OLSR aware of resources, in: *Wireless Communications, Networking and Mobile Computing Conference, WiCom, 2007*, pp. 1488–1493.
- [22] N. Kumar, A. Gore, A. Karandikar, Link scheduling in STDMA wireless networks: a line graph approach, in: *Proc. National Conference on Communications, 2008*, pp. 108–111.
- [23] M. Leoncini, P. Santi, P. Valente, An STDMA-based framework for qos provisioning in wireless mesh networks, in: *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS, 2008*, pp. 223–232.
- [24] E. Lloyd, Broadcast scheduling for TDMA in wireless multihop networks, in: *Handbook of Wireless Networks and Mobile Computing, 2002*, pp. 347–370.
- [25] M. Macedo, A. Grilo, M. Nunes, Distributed latency-energy minimization and interference avoidance in TDMA wireless sensor networks, *Computer Networks* 53 (5) (2009) 569–582.
- [26] E. Malaguti, The vertex coloring problem and its generalizations, *4OR: A Quarterly Journal of Operations Research* (2007) 1–4.
- [27] R. Mangharam, R. Rajkumar, Max: a maximal transmission concurrency MAC for wireless networks with regular structure, in: *3rd International Conference on Broadband Communications, Networks and Systems, BROADNETS, 2006*, pp. 1–10.
- [28] R. McTasney, D. Grunwald, D. Sicker, Interference mitigation in wireless mesh networks through STDMA wormhole switching, in: *3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 2008*, pp. 1–6.
- [29] S. Mecke, MAC Layer and Coloring, in: *Lecture Notes in Computer Science, vol. 4621, 2007*, p. 63.
- [30] J. Moy, OSPF version 2 (RFC 2328), in: *IETF Internet Draft*. <http://www.ietf.org/rfc/rfc2328.txt>.
- [31] S. Nanda, D. Kotz, Mesh-Mon: a multi-radio mesh monitoring and management system, *Computer Communications* 31 (8) (2008) 1588–1601.
- [32] R. Nelson, L. Kleinrock, Spatial TDMA—a collision-free multihop channel access protocol, *IEEE Transactions on Communications* 33 (9) (1985) 934–944.
- [33] D. Nguyen, P. Minet, Scalability of OLSR protocol with the fish eye extension, in: *Sixth International Conference on Networking, ICN, 2007*, p. 88.
- [34] D. Nguyen, P. Minet, Interference effects on the OLSR protocol: ns-2 simulation results, in: *Proceedings of the IFIP Med-Hoc-Net Conference, 2004*, pp. 428–435.
- [35] D. Nguyen, P. Minet, Interference-aware QoS OLSR for mobile ad-hoc network routing, in: *Proceedings of SNPD/SAWN, vol. 5, 2005*, pp. 428–435.
- [36] C. Perkins, E. Belding-Royer, S. Das, Ad hoc on-demand distance vector (AODV) routing: RFC 3561, in: *IETF Internet Draft*. <http://www.ietf.org/rfc/rfc3561.txt>.
- [37] C. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, *ACM SIGCOMM—Computer Communication Review* 24 (4) (1994) 234–244.
- [38] S. Ramanathan, E.L. Lloyd, Scheduling algorithms for multihop radio networks, *IEEE/ACM Transactions on Networking* 1 (2) (1993) 166–177.
- [39] Z. Ren, Y. Zhou, W. Guo, An adaptive multichannel OLSR routing protocol based on topology maintenance, in: *IEEE International Conference on Mechatronics and Automation, vol. 4, 2005*, pp. 2222–2227.
- [40] I. Rhee, A. Warrier, M. Aia, J. Min, Z-MAC: a hybrid MAC for wireless sensor networks, in: *Proceedings of the 3rd International Conference on Embedded Networks Sensor Systems, 2005*, pp. 90–101.
- [41] I. Rhee, A. Warrier, J. Min, L. Xu, DRAND: distributed randomized TDMA scheduling for wireless ad-hoc networks, in: *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2006*, pp. 190–201.
- [42] I. Rhee, A. Warrier, L. Xu, Randomized dining philosophers to TDMA scheduling in wireless sensor networks, *Tech. Rep.*, Computer Science Department, North Carolina State University, Raleigh, NC, 2004.
- [43] F.J. Ros, UM-OLSR implementation (version 0.8.8) for ns-2, UM-OLSR Project. <http://masimum.inf.um.es/?Software:UM-OLSR>.
- [44] H. Su, X. Zhang, Cross-layer based opportunistic MAC protocols for QoS provisionings over cognitive radio wireless networks, *IEEE Journal on Selected Areas in Communications* 26 (1) (2008) 118–129.
- [45] J. Tao, F. Liu, Z. Zeng, Z. Lin, Throughput enhancement in WiMAX mesh networks using concurrent transmission, vol. 2, 2005, pp. 871–874.
- [46] S.-O. Tucke, et al. Freifunk firmware for mesh routers, online. Accessed from: <http://www.freifunk.net>.
- [47] M. Vutukuru, H. Balakrishnan, K. Jamieson, Cross-layer wireless bit rate adaptation, *ACM SIGCOMM—Computer Communication Review* 39 (4) (2009) 3–14.
- [48] K. Wang, M. Peng, W. Wang, Distributed scheduling based on polling policy with maximal spatial reuse in multi-hop WMNs, *The Journal of China Universities of Posts and Telecommunications* 14 (3) (2007) 22–27.
- [49] H. Wei, S. Ganguly, R. Izmailov, Z. Haas, Interference-aware IEEE 802.16 WiMAX mesh networks, in: *IEEE Vehicular Technology Conference, VTC-Spring, 2005*, pp. 3102–3106.
- [50] K. Xu, L. Qi, Y. Shu, Enhancing TCP fairness in ad-hoc wireless networks using neighborhood RED, in: *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, 2003*, pp. 16–28.
- [51] S. Xu, T. Saadawi, Does the IEEE 802.11 MAC protocol work well in multihop wireless ad-hoc networks? *IEEE Communications Magazine* 39 (6) (2001) 130–137.



Miray Kas received her B.S. and M.S. degrees from the Department of Computer Engineering, Bilkent University, Ankara, Turkey, in 2007 and 2009, respectively. She is currently a Ph.D. student in the Electrical and Computer Engineering Department, Carnegie Mellon University. Her current research interests include wireless ad hoc/mesh networks and on-chip networks.



Ibrahim Korpeoglu received his Ph.D. and M.S. degrees from the University of Maryland at College Park, both in Computer Science. He is currently an assistant professor in the Computer Engineering Department of Bilkent University, Ankara, Turkey. Prior to joining Bilkent University, he worked in Ericsson, IBM T.J. Watson Research Center, Bell Labs, and Telcordia Technologies, in the USA. He has served on the program committees of several conferences and published numerous papers in the area of computer networking. He received a TUBITAK (The Scientific and Technological Research Council of Turkey) Young Scientist Career Award in 2004, Bilkent University Distinguished Teaching Award in 2006, and IBM Faculty Award in 2009. His research interests include computer networks, wireless ad hoc and sensor networks, wireless mesh networks, distributed systems, and P2P networks.



Ezhan Karasan received his B.S. degree from Middle East Technical University, Ankara, Turkey, his M.S. degree from Bilkent University, Ankara, Turkey, and his Ph.D. degree from Rutgers University, Piscataway, New Jersey, USA, all in Electrical Engineering, in 1987, 1990, and 1995, respectively. During 1995–1996, he was a post-doctorate researcher at Bell Labs, Holmdel, New Jersey, USA. From 1996 to 1998, he was a Senior Technical Staff Member in the Lightwave Networks Research Department at AT&T Labs-Research, Red Bank, New Jersey, USA. He has been with the Department of Electrical and Electronics Engineering at Bilkent University since 1998, where he is currently an associate professor. Dr. Karasan is a member of the Editorial Board of the journal *Optical Switching and Networking*. He is the recipient of the 2004 Young Scientist Award from Turkish Scientific and Technical Research Council (TUBITAK), the 2005 Young Scientist Award from Mustafa Parlar Foundation and a Career Grant from TUBITAK in 2004. Dr. Karasan received a fellowship from the NATO Science Scholarship Program for overseas studies in 1991–1994. His current research interests are in the application of optimization and performance analysis tools for the design engineering and analysis of optical networks and wireless ad hoc/mesh/sensor networks.