This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2009 proceedings.

OLSR-Aware Distributed Channel Access Scheduling for Wireless Mesh Networks

Miray Kas
Dept. of Computer Engineering
Bilkent University
Ankara, Turkey 06800
Email: miray@cs.bilkent.edu.tr

Ibrahim Korpeoglu
Dept. of Computer Engineering
Bilkent University
Ankara, Turkey 06800
Email: korpe@cs.bilkent.edu.tr

Ezhan Karasan
Dept. of Electrical & Electronics Engineering
Bilkent University
Ankara, Turkey 06800
Email: ezhan@ee.bilkent.edu.tr

***Abstract*—In this paper, we present OA-TDMA (OLSR-Aware TDMA), a TDMA based cross-layer channel access scheduling scheme which uses the information collected by the OLSR routing protocol. In OA-TDMA, each node makes decisions in a distributed manner with no central control, using the local information disseminated by the OLSR protocol. The distinctive feature of the OA-TDMA protocol lies in its weighting scheme where OA-TDMA approximates the traffic passing through each node by using the local topology information collected by OLSR. Our simulations on ns-2 confirm the significant performance improvement achieved by the combination of OLSR and OA-TDMA over other scheduling schemes considered in this paper.**

***Index Terms*—OLSR, Distributed Scheduling, TDMA, MAC, Cross-layer approach.**

I. Introduction

A very important element of a wireless mesh network (WMN) required to keep the network functional is the routing protocol. Since WMNs typically have static topology, a proactive routing protocol which collects various information that can also be used in a cross-layer manner might be selected. As OLSR is one such protocol which is among the most popular routing protocols in ad-hoc and mesh networks, in this study we investigate the channel access problem in WMNs with OLSR as the routing protocol [2].

In the current literature, most channel access protocols are contention-based (CSMA, ALOHA) or schedule based (TDMA, FDMA, CDMA) [7]. The contention based CSMA schemes are shown to provide lower performance and unfair channel access as the number of contending nodes increases.

TDMA based protocols have advantages of being able to provide collision free transmission, while achieving higher throughput and fairness. Due to such advantages, TDMA based scheduling has been studied extensively in the literature despite the difficulty of synchronization and efficient use of time slots. TDMA protocols arrange the transmission of the nodes in the network based on a static or dynamic schedule. This schedule can be formed in a topology dependent, topology transparent, randomized, or in some other prearranged manner. Finding an optimal conflict-free access time table (schedule) has shown to be NP-complete and the problem is usually reduced to node or edge coloring problem [7]. The proposed polynomial algorithms are known to achieve suboptimal results by means of randomized approaches or heuristics based on particular features of the graph [4].

The arrangement of nodes or links (in our case, nodes) for transmission may be obtained through the use of a centralized or distributed protocol. In most of the early studies in this area, centralized approach is adapted where a central site is required that computes the TDMA schedule for the whole network [18]. This situation also implies the requirement of a significant amount of message exchange between all the nodes in the network and the central site. Therefore, this approach has certain drawbacks such as introducing a significant amount of control overhead, possible requirement of a control channel and having a single point of failure.

On the other hand, there is no single point of failure in distributed systems. Moreover, distributed approaches provide more flexibility and scalability as the network size increases. However, most of the current distributed protocols also introduce quite high overhead for control message exchanges due to contention among neighboring nodes [18].

In this paper, we propose a distributed scheme for transmission scheduling of nodes in a wireless mesh network having the goal of increasing simultaneous transmissions (concurrency) so that a higher overall throughput in the network can be achieved. Our scheme does not require the exchange of MAC level control messages to learn about neighboring nodes and local topology. This is achieved by utilizing the topology information that is already collected by a link state routing protocol, i.e., OLSR in our particular study here. As simulation results show, such a cross-layer scheduling scheme that utilizes the information gathered by the routing protocol while assigning time slots to nodes can achieve a good performance in terms of throughput and delay.

The rest of the paper is structured as follows. In Section II, information regarding the related work in the literature is given. In Section III, OA-TDMA is presented along with the details about the network model, the weighting scheme, the node access scheduling algorithm and the implementation issues. Section IV presents the simulation results and finally Section V concludes the paper pointing out possible research directions.

978-1-4244-2948-6/09/$25.00 ©2009 IEEE

II. Background and Related Work

In this section, information about OLSR is given and some of the TDMA based MAC schemes are discussed briefly.

A. OLSR

OLSR (Optimized Link State Routing) is an optimization of the classical link state routing protocol, designed for mobile wireless networks [9] and standardized by IETF in RFC3626 [6]. Every node periodically broadcasts HELLO messages that contain its links and neighbor list. The distinguishing feature of the OLSR protocol is the use of multi-point relays (MPRs). An MPR is a node that is chosen by other nodes to forward their messages and MPRs periodically broadcast the list of nodes that select them as MPR. Via this technique, the message overhead is reduced in comparison to pure flooding mechanism.

The link state information is only generated by the MPRs and disseminated throughout the network by TC (Topology Control) messages. Each node has an overview of the network, and this overall network information is used by each node to calculate the shortest path (in terms of number of hops) to each known destination kept in the routing table. The protocol is particularly suitable for large and dense networks as more optimization is achieved through the technique of MPRs. This situation also causes the selection of MPRs to be the most critical part of the protocol. A node selects its MPRs so that there is no two-hop neighbor it cannot reach.

The nodes usually tend to select same nodes as their MPRs. This feature of the protocol enables us to develop a unique weighting scheme which is explained in detail in Section III-B. Note that the MPR set is not necessarily minimum, as finding the minimum coverage set is NP-complete. A heuristic for MPR selection given is in RFC3626 [6].

B. MAC schemes

In the literature, there is considerable amount of research on TDMA based access schemes. Some of these schemes are cluster based, some are hybrid like Z-MAC (ZEBRA-MAC [15]), some of them are randomized (DRAND [16], NAMA [3]) and there are some others as HBS, TRAMA that perform node access scheduling taking design specific metrics into account.

NCR (Neighbor-Aware Contention Resolution) is a MAC protocol proposed by Bao et al. [3]. In NCR, it is assumed that knowledge about two-hop neighborhood (that is, the contenders) is achieved by some means and the nodes have mutual knowledge. A seed value for random value generation is formed as the combination (node-id, contended-slot-number) for each of the 2-hop neighbors and a winner, the node that draws the highest random value for the contended slot, is elected as the winner of the slot. The priority value of each node is calculated by Eq (1) below where t is the contended-slot-number and k is the node-id:

$$p_k^t = Rand(k \oplus t) \oplus k \tag{1}$$

To give a couple of examples to NCR based protocols, NAMA [3] is one such protocol where every node runs NCR and no two nodes within the same 2-hop neighborhood transmit simultaneously.

HBS (History Based Scheduling) is another protocol using NCR algorithm, improving on NAMA [7] by using the ratio of the number of slots in a frame used by a node to transmit to the number of slots in a frame given to that node as a success indicator. If this ratio drops below some pre-set threshold, the weight of the node is decremented. Similarly, the weight of a node is incremented if the ratio is larger than another pre-set threshold.

TRAMA (Traffic Adaptive Medium Access) [13] is another TDMA protocol which assigns time slots to the nodes through the use of one-hop traffic information and two-hop neighborhood information. For each transmission time, each node selects one of the transmitting, receiving, stand-by nodes. The nodes with no data to send are not involved in the election.

III. Proposed Channel Scheduling Scheme

In this section, the details of the proposed distributed cross-layer scheduling scheme OA-TDMA are presented.

A. Network Model

The following assumptions hold for the network model we are working on:

- Every node has a unique id, and node synchronization is available.
- All the nodes have the same capabilities, there is no central base station.
- No further maintenance is done after deployment and the nodes are stationary.
- Communication is established via omnidirectional antennas over a single channel.
- Routing protocol is OLSR.

There are many studies that examine OLSR from different aspects, such as investigating the effects of interference on the OLSR protocol [12], making OLSR resource aware [10], designing multichannel OLSR [14] and many others. However, to the best of our knowledge, there is no distributed scheduling based TDMA MAC scheme that uses OLSR-specific information in the decision process in a cross-layer manner.

Basic features of the proposed OA-TDMA protocol are as follows:

- The number of slots in a frame is fixed.
- The radios are always on.
- Each node decides the time slots it will use for transmission based on local cross-layer information.
- The nodes and time slots are assumed to be synchronized. Methods for achieving synchronization are out of the scope of this paper.
- A packet with maximum size can fit into a time slot.

Although we keep the radios always on in our current implementation, it is possible to introduce energy preservation into OA-TDMA. Since the nodes need to hear OLSR HELLO messages from all of their 1-hop neighbors, the OA-TDMA schedule can be classified as a broadcast schedule. In a broadcast schedule, it would be enough for a node to only keep its radio open when the winner is itself or one of its 1-hop neighbors. Besides, by the use of the algorithm given in Section III-C, each node knows the winner of a particular slot within its 2-hop neighborhood. Therefore, it is possible to exploit this broadcast schedule property as an energy-preserving method in OA-TDMA.

B. Weighting Scheme

In [5], the authors show that in most cases 75% of all MPRs are elected in the first round. As the MPR selection converges quite fast, this information might be used by MAC in slot allocation decision. In OA-TDMA, we allocate the time slots to nodes in a multi-hop network in a distributed manner through the use of a weighted scheme. The weighting scheme models the idea that the nodes that are liable to carry the traffic generated by other nodes should be more likely to get more opportunities (time slots). To achieve this, we assign the weight of a node X as:

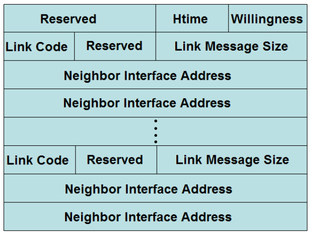
$$Weight(X) = Num\_of\_MPR\_Selectors(X) + 1 \tag{2}$$

Assuming that all nodes are equally likely 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 and 1 unit of weight for node X itself is added as well. Hence, the number of slots node X obtains, Slots(X), during the entire simulation will be approximately proportional to its weight, Weight(X).

$$Slots(X)/\sum_k Slots(k) \;\; \alpha \;\; Weight(X)/\sum_k Weight(k) \tag{3}$$

The main reason for selecting such a weighting scheme is that it can be used to approximate the traffic passing through each node under the scenarios where all the nodes in the network have packets to send to the others. Another reason for selecting size of Mpr_Selector_Set as the weight indicator is that, if the network is not too mobile, once the network stabilizes, the weight calculated in Eq (2) will be much less frequently changing hence consistent than the weight calculated through queue length or the amount of traffic generated. This unique weighting scheme is one of the most important parts of the OA-TDMA access scheduling scheme affecting the overall performance. OLSR already exchanges periodic HELLO messages and collects 2-hop neighborhood information. Hence, this mechanism can be easily extended to carry the weight information as well.

In RFC3626, the structure of an HELLO message is given as in Figure 1.

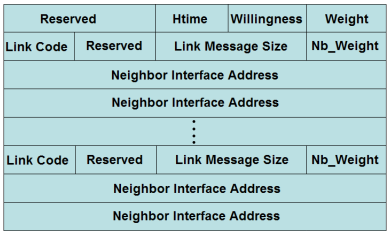


Reserved		Htime	Willingness
Link Code	Reserved	Link Message Size	
Neighbor Interface Address			
Neighbor Interface Address			
⋮			
Link Code	Reserved	Link Message Size	
Neighbor Interface Address			
Neighbor Interface Address			

Figure 1. RFC3626 OLSR HELLO Message Structure

Htime field holds the HELLO emission interval, the time until the next HELLO message transmission and 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 the Neighbor Type and the 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.

The message structure in Figure 1 is extended to include weight information for the originating node itself and its listed 1-hop neighbors. The extended message structure is shown in Figure 2.



Reserved		Htime	Willingness	Weight
Link Code	Reserved	Link Message Size		Nb_Weight
Neighbor Interface Address				
Neighbor Interface Address				
⋮				
Link Code	Reserved	Link Message Size		Nb_Weight
Neighbor Interface Address				
Neighbor Interface Address				

Figure 2. Extended OLSR HELLO Message Structure

Both Weight and Nb_Weight fields are of 8 bits length. Weight field holds the weight information of the originating node. The Nb_Weight field holds the advertised link's associated neighbor node's weight information. In a single HELLO message, there is only one Weight field, but there might be multiple Nb_Weight fields depending on the number of the links advertised. Using this extended HELLO message structure, every node will be able to collect the weight information of all the nodes in its 2-hop neighborhood via the routing layer without requiring the MAC layer to exchange any further control messages.

C. Node Access Scheduling

There are two main types of conflicts that should be avoided in order to achieve a collision free schedule:

1-) Primary Conflict: Observed if a node is scheduled to transmit and receive at the same time.

2-) Secondary Conflict: Observed if a node is scheduled to receive from two different nodes simultaneously.

In order to ensure that both kinds of conflicts are avoided, no two nodes within the same 2-hop neighborhood should be scheduled to transmit at the same time slot [18], [11].

In OA-TDMA, each node determines the time slots it will use to transmit based on the information about its 1-hop and

2-hop neighbors collected by OLSR. We propose a pseudo-random weighted 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 neighbors' respective weights, nodes can run their algorithms without having to wait for their neighbors' approval signals.

The OA-TDMA-scheduler algorithm is independently run by every node at the end of every frame in order to select the slots it is eligible to transmit during the next frame. The pseudocode of OA-TDMA-Scheduler is given in Algorithm 1.

Algorithm 1 OA-TDMA-Scheduler

```
$localAgentsLst \leftarrow FormLocalNodeAgents()$
$nbrAgentsLst \leftarrow FormNeighborNodesAgents()$
$nbrAgentsLst \leftarrow nbrAgentsList \bigcup localAgentsLst$
for $i = 1$ to $FRAME\_SIZE$ do
    $res\_set \leftarrow MeshElection(contestTime, nbrAgentsLst)$
    $winner \leftarrow find\_max(res\_set)$
    if $localAgentsLst.contains(winner)$ then
        $slots[i] \leftarrow WON$
    end if
end for
```

Each node has as many agents as its weight where all of its agents compete to win slots on its behalf. In the first two steps of OA-TDMA-Scheduler algorithm, the set of AgentIDs of the owner node's agents (localAgentsLst) and the set of AgentIDs of its 2-hop neighbors' agents (nbrAgentsLst) are generated. 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. MeshElection method returns a set of pairs where each pair involves the AgentID and its corresponding Smear_value. The agent with the largest Smear_value is then elected as the winner of the contended time slot. If the winner AgentID belongs to localAgentsLst, then the node marks the slot as one of the slots it is eligible to transmit.

The MeshElection function in OA-TDMA-Scheduler is adapted from the MeshElection algorithm given in 802.16-2004 standard [1] as a part of the distributed EBTT mechanism which is responsible for the allocation of control slots such that the control messages are transmitted collision-free manner in two-hop neighborhood without requiring explicit schedule negotiation. ContestTime, MeshElection function's first parameter, is formed as the concatenation of the contended frame_count with the contended slot_id. The Smear_value is obtained as

$$Smear\_value = inline\_smear(AgentID \wedge ContestTime) \tag{4}$$

where inline_smear function is the hashing function given 802.16-2004 standard [1] which converts a uniform value to an uncorrelated uniform hash value, through the use of mixing.

D. Implementation

We implemented and verified our proposed OA-TDMA scheme in ns-2.31 environment. The implementation is composed of two parts which are the implementation of the required changes in the OLSR module and the implementation of the proposed MAC scheme.

For the OLSR implementation, we chose UM-OLSR-0.8.8 for ns-2.31 as it is compliant with RFC3626 and provides MAC layer feedback support which is useful in detecting lost links [17]. We have extended the OLSR HELLO messages to disseminate weight information of the originating node as well as its known neighbors as mentioned in Section III-B.

From ns-2.23 onwards, ns-2 implementation comes with a basic non-concurrent TDMA based MAC protocol [8]. This TDMA MAC protocol is a preamble based one-hop TDMA scheme which does not exploit the slot reusability in multi-hop environments. In this implementation, each TDMA frame contains a preamble and the data transmission slots. The number of data transmission slots in a frame is equal to the number of nodes in the network. During each frame, every node takes turn once even if it has no data to send. The preamble is used by each node to broadcast the destination id of the packet it will transmit next so that the nodes go to sleep during the slots they don't send or receive anything in order to reduce the power consumption.

However, this implementation has obvious drawbacks as it produces very low throughput, not taking the slot reusability and traffic into account and not modeling centralized scheduling to the full extent as there is no central controlling node that dictates the schedules of the remaining nodes.

Taking the non-concurrent TDMA MAC implementation in ns-2 as a starting point, we have implemented OA-TDMA as a MAC protocol which is multihop, allowing multiple nodes to transmit concurrently. At the end of each frame, every node independently runs OA-TDMA-Scheduler algorithm to decide the slots it will use to transmit during the next frame.

The only exception to the use of this algorithm is seen in the very first frame of the simulation. Since the nodes start with empty neighbor lists, each node assumes that it is the only node in the contention context and tends to select every slot of the first frame as eligible to transmit. In order to avoid collisions, only in the first frame, nodes act overly precautious and every node selects its slots for transmission as if the network is a one-hop network. In a scenario, where there are 20 nodes in the network and the FRAME_SIZE is set to 100, then a node, say *node-7* selects 7, 27, 47, 67, 87 for transmission. After the first frame, OA-TDMA-Scheduler algorithm is used by all the nodes during the whole simulation.

For comparison purposes, we have also implemented non-weighted OA-TDMA protocol. Both the time slot & frame structure and the election mechanism are kept the same as in OA-TDMA. The only difference between non-weighted OA-TDMA and OA-TDMA protocols lies in the election mechanism as the non-weighted OA-TDMA protocol does not use the weight information, assuming all nodes' weights are equal to 1.

IV. Simulation Results and Analysis

In this section we report our ns-2 simulation results to compare the proposed OA-TDMA scheme with two other schemes, non-weighted OA-TDMA and non-concurrent TDMA. The three protocols are simulated and compared under the same uniform traffic scenarios where every node generates a CBR connection to every other node in the network. In each of the simulation scenarios, there are $O(n^2)$ CBR connections, each of which start and end at the same time. The CBR rate (in bps for each CBR connection) and the packet size is kept the same (200 bytes) for all the CBR connections in a single scenario and a range of different CBR rates are applied in different simulations. In the first part of the simulation results (Figures 3-7), we use a 20-node network. We then extend these results in order to study the effects of changing network size.

Figure 3 illustrates the relationship between the offered traffic (total number of packets sent by the agent layer) and the overall throughput (delivered traffic; total number of packets received at agent layer). The results indicate that concurrent protocols perform much better than the non-concurrent TDMA protocol and using weight information in transmission scheduling provides additional gains.

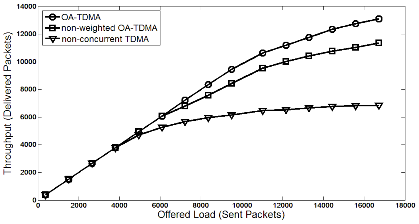


Figure 3. Delivered Traffic versus Offered Traffic

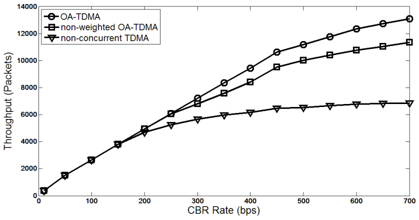


Figure 4. Throughput versus CBR Rate

Figure 4 demonstrates the relationship between the throughput and the CBR rate. This relation exhibits a very similar behavior to the relationship between the throughput and offered traffic.

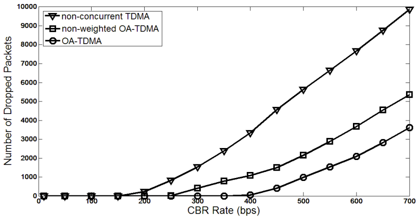


Figure 5. Number of Dropped Packets versus CBR Rate

All the packets that are not received within the simulation duration are considered as dropped. The number of dropped packets as a function of the CBR rate is given in Figure 5. We observe that OA-TDMA significantly reduces the number of dropped packets compared to the other scheduling schemes considered.

Figure 6 and 7 show how the average end-to-end delay changes as the CBR Rate and throughput changes respectively. The difference between the delay values achieved by the scheduling algorithms become more significant when we compare the resulting delays in order to obtain the same throughput using different algorithms, as depicted in Figure 7.

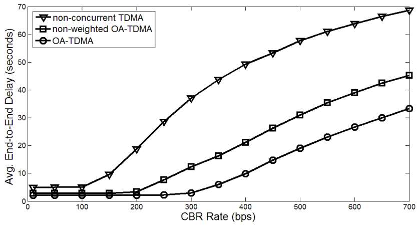


Figure 6. Average End-to-End Delay versus CBR Rate

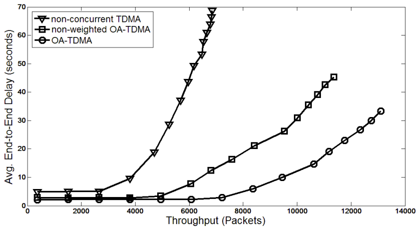


Figure 7. Average End-to-End Delay versus Throughput

For the next set of experiments, the CBR rate is fixed (set to 400 bps) and the number of nodes in the network is changed. Figures 8 - 10 present the results of these simulation experiments.

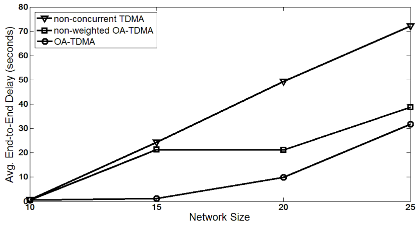


Figure 8. Average End-to-End Delay versus Network Size

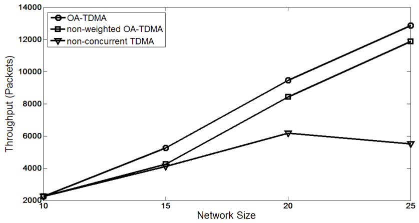


Figure 9. Throughput versus Network Size

Figure 8 illustrates how the average end-to-end delay changes when the CBR rate is fixed and the network size is increased. The gap between the concurrent protocols and the non-concurrent TDMA gets wider as the network size increases. Since each node generates *O(n)* traffic and has a large amount of data to send, concurrency gains more importance.

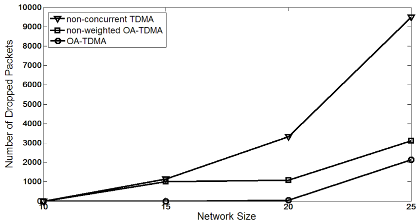


Figure 10. Number of Packets Dropped versus Network Size

In Figures 9 and 10, it is seen that weighting helps in reducing the number of packets dropped and increasing the throughput. As the number of nodes increases, the relay traffic becomes more dominant since paths typically use more hops. In OA-TDMA, since the nodes that relay more packets obtain more slots, the queues of these more heavily loaded nodes are consumed more quickly, and this is reflected in both throughput and dropped packets results.

The overhead OA-TDMA introduces into the OLSR control messages has also been investigated. The increase in the total size of sent OLSR messages has been checked to see how much extra overhead is introduced due to the addition of weight information in OLSR HELLO messages. The average overhead in a set of simulations using different sized networks (10 nodes to 25 nodes) turned out to be around 0.4% - 1.3% and the related results are presented in Figure 11.

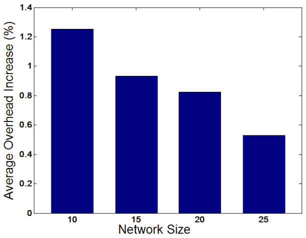


Figure 11. Average Percentages of Extra Overhead Introduced in Different Simulations

The percentages of overhead increases are quite stable as the network size remains the same; however they tend to decline as the network size increases. This is due to the fact in a small network, there is less information to be disseminated by the control messages. In larger networks, the overhead introduced by the addition of weight information is relatively small compared with the overall OLSR control traffic.

V. Conclusions and Future Work

In this paper, we presented OA-TDMA, a new scheduling-based channel access protocol with a topology-dependent weighting scheme utilizing the topology information gathered by OLSR routing protocol. Our simulations on ns-2 show that OA-TDMA performs well in uniform traffic scenarios where the weight of a node can be a good indicator of its traffic load. In order to enhance the performance obtained in other traffic scenarios where dynamic traffic patterns and mobility are involved it is possible to improve the proposed weighting scheme to provide more adaptability to traffic at the cost of increasing complexity. In this respect, the effects of metrics such as queue size, SINR and packet delivery ratio might be investigated.

VI. Acknowledgements

This work is partially supported by The Scientific and Technological Research Council of Turkey (TUBITAK) with grant number EEEAG-104E028.

References

[1] "IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems," *IEEE Std 802.16-2004*, pp. 1–895, 2004.

[2] T. Aure and F. Y. Li, "Load-balancing in Multi-homed OLSR Networks using Airtime Metric and Radio-aware Path Selection: Implementation and Testing," Tech. Rep., February 2008.

[3] L. Bao and 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*. ACM Press New York, NY, USA, 2001, pp. 210–221.

[4] ——, "Distributed Channel Access Scheduling for Ad Hoc Networks," *Book Chapter in 'Algorithms and Protocols for Wireless and Mobile Networks'*, 2004.

[5] A. Busson, N. Mitton, and E. Fleury, "An Analysis of the Multi-Point Relays Selection in OLSR," *INRIA Research report*, 2005.

[6] T. Clausen and P. Jacquet, "RFC3626: Optimized Link State Routing Protocol (OLSR)," 2003.

[7] Y. Du, Y. Bao, and 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.

[8] K. Fall, K. Varadhan, *et al.*, "The ns-2 manual," *the VINT Project, http://www.isi.edu/nsnam/ns/doc/ June 2008*.

[9] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized Link State Routing Protocol for Ad Hoc Networks," in *Proceedings of IEEE International Multi Topic Conference*, 2001, pp. 62–68.

[10] K. Kowalik, B. Keegan, and M. Davis, "Making OLSR Aware of Resources," in *Proceedings of Wireless Communications, Networking and Mobile Computing*, 2007, pp. 1488–1493.

[11] E. Lloyd, "Broadcast Scheduling for TDMA in Wireless Multihop Networks," *Handbook of Wireless Networks and Mobile Computing*, pp. 347–370, 2002.

[12] D. Nguyen and P. Minet, "Interference Effects on the OLSR Protocol: ns-2 Simulation Results," in *Mediterranean Ad Hoc Networking Workshop*, 2004.

[13] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves, "Energy-efficient, Collision-free Medium Access Control for Wireless Sensor Networks," *ACM SenSys*, pp. 181–192, 2003.

[14] Z. Ren, Y. Zhou, and W. Guo, "An Adaptive Multichannel OLSR Routing Protocol Based on Topology Maintenance," in *Proceedings of IEEE International Mechatronics and Automation Conference*, 2005.

[15] I. Rhee, A. Warrier, M. Aia, and J. Min, "Z-MAC: a Hybrid MAC for Wireless Sensor Networks," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*. ACM Press New York, NY, USA, 2005, pp. 90–101.

[16] I. Rhee, A. Warrier, J. Min, and L. Xu, "DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad-hoc Networks," in *Proceedings of the Seventh ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM Press New York, NY, USA, 2006, pp. 190–201.

[17] F. Ros, "UM-OLSR implementation (version 0.8.8) for ns-2."

[18] Y. Wang and I. Henning, "A Deterministic Distributed TDMA Scheduling Algorithm for Wireless Sensor Networks," in *Proceedings of Wireless Communications, Networking and Mobile Computing Conference*, 2007, pp. 2759–2762.