
Contents lists available at ScienceDirect
Journal of Computer and System Sciences
www.elsevier.com/locate/jcss

The design and evaluation of fair scheduling in wireless mesh networks

Jason B. Ernst*, Mieso K. Denko

Department of Computing and Information Science, University of Guelph, Guelph, Ontario, N1G 2W1, Canada

A R T I C L E   I N F O

Article history:
Received 15 September 2009
Received in revised form 15 December 2009
Available online 25 February 2010

Keywords:
Fair scheduling
Gateways
Wireless mesh networks
Scheduling

A B S T R A C T

In this paper we address the problem of scheduling in wireless mesh networks. First, we provide a comparison of existing scheduling algorithms and classify them based on the degree of fairness, the scheduling techniques and their implementation frameworks. Then we propose a fair scheduling approach using multiple gateways. The proposed scheduling approach consists of four important steps, namely, requirement tables, requirement propagation, clique generation and schedule generation. Simulation experiments are conducted to compare the performance of fair scheduling with the method that does not use fair scheduling. The simulation results confirm that the proposed scheduling has better performance with respect to the metrics used for performance evaluation.

© 2010 Elsevier Inc. All rights reserved.

1. Introduction

Wireless mesh networks (WMN) are convenient and easy to setup and maintain. They are quickly replacing traditional wired networks for many forms of communication. For instance, cellular phone service is quickly becoming more popular than traditional land based telephone services. This is especially true in developing countries where infrastructure is non-existent and prohibitively expensive. Additionally, wireless local area networks (WLAN) are gaining popularity compared with older technologies such as Ethernet for data communications in both residential and businesses due to the decreased cost and ease of setup compared with laying wires. These same attractive features are also the reason why wireless technology is used in the military and in disaster situations. More recently, wireless mesh networks have become the focus of much research since they allow for increased coverage range while retaining the attractive features of low cost and easy deployment [9]. However, there are still many challenges left in order to achieve all of the applications that the technology is capable of. In particular this paper will focus on the challenge of scheduling in wireless mesh networks.

Scheduling is an important challenge to deal with, especially in commercial wireless mesh network applications. Many current deployments are optimized with respect to throughput, delay or some other feature that gives little regard to fairness. The focus of this paper is on fair scheduling techniques which use multiple gateways. Thus, the contributions of this paper are two-fold. First we give an in-depth comparison and analysis of existing techniques in the area. Second we provide our own fair scheduling algorithms for WMNs with multiple gateways. The implementation of fair scheduling for WMNs with multiple gateways is presented and evaluated along with experimental results.

The remainder of this paper is organized as follows: Section 2 gives background, related work and motivation for studying the problem. It also provides analysis of assumptions in existing solutions. Section 3 provides the detailed description of the proposed approach. Section 4 presents the performance evaluation of our proposed approach. Finally in Section 5, we give conclusions and discuss areas for future research.

* Corresponding author.
E-mail addresses: jernst@uoguelph.ca (J.B. Ernst), mdenko@uoguelph.ca (M.K. Denko).
2. Background and related work

In this section we first define fairness with respect to wireless mesh networks. We then give a brief introduction to fair scheduling techniques. This is followed by a classification, comparison, and analysis of current scheduling solutions. The literature review establishes where our proposals stand in comparison to the existing work. We identify areas where more research could be accomplished in the future. Moreover we identify work that is most similar to our own. Lastly, we describe how cross-layer design can be used to further improve scheduling in wireless mesh networks and why a mixed-biased cross-layer approach is a promising technique for cross-layer scheduling.

2.1. Fairness in wireless networks

A number of scheduling and resource allocation techniques have been proposed for WMN in literature [8,15,16,18,22–24]. The trend is a tradeoff between the throughput and fairness using a constant weighting system or a dynamic weighting system that changes the weights over time to achieve a long-term fairness. It is important to note that fairness could occur at different points in a wireless mesh network. Some researchers have proposed per-mesh-router fairness or per-link fairness [26]. There is also a notion of “uplink-downlink fairness” [19,20,28] because the mechanisms in some current solutions, such as IEEE distributed coordination function (DCF) [28] allow for inequality between the directions of flow in WMNs. In other words, an improvement in downlink throughput may severely affect performance of the uplink or vice-versa. However, more recently [26,31] have focused on per-client fairness. The motivation behind this is that in commercial applications each user is paying an equal amount of money for services from the network so each user should get equal Quality of Service (QoS). It is also important to consider which metrics fairness is being defined with respect to. For example, a scheduling algorithm could provide fairness in terms of the possible throughput available but the delay may not be equal. Certain nodes in the network may remain starved for traffic while other nodes are free to communicate for various reasons. It is also important to consider that fairness and scheduling is affected by intruders in the system. Many of the existing solutions for scheduling in WMNs rely on the assumption of co-operation between nodes, and this is not always the case in real world networks.

Scheduling algorithms usually give preference to flows which are least expensive by some criteria. These criteria may be distance from the gateway, delay, small flows and other similar metrics. However, this approach may allow for starvation or reduced QoS for flows which do not meet the criteria. Preference is may be given to greedy flows. On one extreme is absolute or hard-fairness. This side gives little priority to throughput and ensures that each client gets a fair share of the network resources. This may be achieved by using a time division mechanism or other similar approaches.

The problem with this approach is that not all flows require the same amount of resources at all times so the resources may remain unused at times resulting in poor throughput. One approach which aims for a balance between the competing goals of fairness and throughput, denoted as max–min fairness [35] works by maximizing the minimum data rates for each flow. It results in higher throughput than hard-fairness, however, the overall throughput is still much less than maximum throughput and leaves much to be desired. The most interesting definition of fairness then is a compromise between hard-fairness and maximum throughput. In [4,19,22,29,31,33] this approach has been denoted as proportional fairness. Proportional fairness assigns priority to certain flows based on criteria such as the number of hops or amount of resources requested. Similarly, the max–min approach has also been modified with a proportional factor as well yielding improved results. A new approach called mixed-bias is a hybrid approach which emphasizes throughput while still providing a basic level of fairness. In the scheme proposed in [31], a portion of the resources are assigned to a strong biasing against nodes which are far away from each other. In order to prevent starvation, however, another portion is assigned to a proportional or max–min scheme as well. This is one of the first approaches that can offer a minimum level of fairness while retaining throughput which is often even greater than of proportional fairness or max–min.

2.2. Motivation for fair scheduling in wireless mesh networks

The first motivation for studying fairness in WMNs is in networks where users are paying equal amounts of money for service and expect a similar quality of service (QoS). Often existing solutions focus on either the problem of throughput or the problem of fairness. It is often difficult to create a solution which addresses both of these problems since they are divergent goals. Recently, however, with works like those of [31], it is possible to have high throughput solutions that avoid node starvation. However in this paper we do not intend to do both. For details on joint optimization of fairness and throughput see [24].

Mesh Clients (MCs) (see Fig. 1), which are far away from the gateways (in terms of number of hops) often receive much lower QoS than those which are very close. This is because while the farther users’ packets are traversing all of the hops along the path, there is a transmission and queuing delay at each hop. The nodes which are close to the gateways do not experience this and can often transmit many packets while the farther nodes are still waiting for one packet to arrive. However, if we give each node enough time to transmit regardless of distance the throughput of the network decreases dramatically. This is because the delay increases greatly by giving each node enough time to transmit regardless of distance to the GW. Some nodes may end up waiting almost indefinitely while other nodes are transmitting. On the other hand, we want to avoid collisions and retransmissions. A reduction in retransmission will significantly increase performance...
since retransmission and the exponential back-off function can significantly increase the time it takes a packet to reach its destination [1].

2.3. Classification of fair scheduling by degree of fairness

Fair scheduling protocols for wireless mesh networks can be classified by fairness into five categories. These categories in order of fairness from the most fair to the least fair are: Hard-fairness [4,22,26,28,30], max–min [31,35], proportional fairness [4,19,22,29,31,33], mixed-bias [31] and maximum throughput [3].

2.3.1. Hard fairness

Hard fairness [4,22,26,28,30] is also known as round-robin scheduling. Homogeneity of resources is not a requirement in this type of scheduling in wireless mesh networks. In traditional distributed systems this may be true when providing CPU resource scheduling. Often the resource being scheduling in the wireless mesh network is time or frequency so this is not the case. It has been used in some of the earliest wireless networks and in simplistic network models since it is the least complex. It is the fairest scheme since each node is guaranteed exactly equal amount of time in order. In networks where the nodes only require a small proportion of resources hard fairness causes problems. Since each node is given time to transmit at regular intervals, if the node does not have any data to send, the time is wasted. This leads to very low overall throughput. At the same time, however, the problem of node starvation does not exist. Resources are assigned to each node inversely proportional to the number of flows through the node.

2.3.2. Max–min fairness

Max–min fairness [31,35] allocates resources in order of increasing demand. The minimum amount of resources assigned to each node is maximized. So if there are more than enough resources for each node, every node gets what it needs. If there is not, the resources are split evenly. This means that the nodes which require fewer resources get a higher proportion of their need satisfied. The nodes which require more resources end up dropping many packets and thus the network ends up with still quite low packet delivery ratio. This type of scheme works best in situations where there is not large differences in resources requested at each node. This can be a problem in a mesh network because intuitively, the nodes closer to the gateways will experience much higher traffic than those on the outside of the network, yet may end up dropping many of the packets anyway. This may be partially solved by increasing the resource capacity of nodes closest to the gateways.

2.3.3. Proportional fairness

Proportional fairness [4,19,22,29,31,33] allocates resources proportional to some characteristic in the network. For example, one may choose to give priority to nodes which are close to the gateways in a wireless mesh network. The amount of resources allocated then would be proportional to how close the node is to the gateway. The strength of the proportionality can be controlled depending on the proportionality factor as can be seen in Eq. (1) [31].

\[
R = \frac{1}{c^\beta},
\]

where: \(R\) is the resources allocated to the node; \(c\) is the characteristic which priority is given to, \(c > 0\); \(\beta\) is the proportionality factor, \(\beta > 0\).
2.3.4. Mixed-bias scheduling

Mixed-bias [31] scheduling allows for different levels of control over resources. Rather than just allowing for one bias, this scheme mixes two different biasing levels together. A certain proportion of the resources are assigned to one factor and the rest to another factor as shown in Eq. (2). This allows the scheduling algorithm to provide two different biasing levels or “mixed-biasing” against a certain characteristic. Rather than just strongly biasing against that characteristic which may result in certain nodes to be starved, the mixed-biasing allows for a combination of weak and strong biasing meaning that a portion of the resources are reserved to provide a minimum service level, even for the nodes which are undesirable in terms of certain characteristics. This is shown mathematically in Eq. (2).

\[ R = \frac{\alpha}{c^{\beta_1}} + \frac{1 - \alpha}{c^{\beta_2}}, \]

where: \( R \) is the resources allocated to the node; \( c \) is the characteristic which priority is given to, \( c > 0 \); \( \beta_1, \beta_2 > 0 \); \( \alpha \) is the fraction of resources assigned to each bias \( \alpha \geq 0 \).

2.3.5. Maximum throughput

Maximum throughput [3] scheduling has only one goal. As the name suggests, this goal is to maximize throughput. This is the only concern of this particular type of scheduling. Whichever node requires the most resources, or can transmit the fastest or most data gets access to the resources first. This ensures a very high throughput, however, there is a limitation with this approach. Nodes which have less priority, such as those far away from gateways, those with fewer users, fewer flows, or less demanding traffic, are essentially ignored. If enough time passes, all of the packets waiting in the queues at these MRs are dropped causing some nodes to be starved for traffic. This causes performance problem and should be avoided.

2.4. Classification by scheduling control

A scheduling algorithm can be classified based on whether or not they are centralized and the metric or mechanisms they use in scheduling. In [13] there is a comparison between the key features of centralized and distributed approaches for scheduling. The distributed approach because of three observations: (i) To eliminate the coordinator from the network to reduce overhead, in case of nodes not being able to communicate with it and because of coordinator failure due to battery life failure. In WMNs the only observation that still holds is lowered overhead. (ii) The observation that some nodes will not be able to communicate with the coordinator node does not hold with WMN because of the multi-path nature of the network unless the MRs are allowed to have mobility. So for situations where the MRs are anticipated to be static, it may be easier and more beneficial to use centralized scheduling. In contrast, when the MRs are allowed mobility, it may be better to make use of a distributed approach in case the network becomes partitioned due to the movement of MRs.

2.5. Classification by metrics and mechanisms

We will identify some of the common metrics and mechanisms used for fair scheduling in WMNs. For fair scheduling in WMNs many of the algorithms use the concept of a bandwidth allocation vector [19] or similar approaches [31,35] to determine how much of the network resource is required from a flow, when to schedule the resources and how. Also fair scheduling algorithms which attempt to avoid collision altogether make use of a compatibility matrix to determine which nodes can communicate at the same time without collisions [4,13].

Table 1 summarizes the three classifications discussed to allow quick comparison of recently proposed scheduling algorithms for WMNs. In order to keep the table to a manageable size the following abbreviations were used: RR: Round Robin, PF: Proportional Fairness, M–M: Max–Min, M-B: Mixed-Bias, D: Distributed, C: Centralized.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Type of fairness</th>
<th>Metric/mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>D. Koutsonikolas [4]</td>
<td>RR</td>
<td>interference threshold, SINR, centralized</td>
</tr>
<tr>
<td>L. Erwu [19]</td>
<td>PF</td>
<td>bandwidth requirement weight-factor, distributed</td>
</tr>
<tr>
<td>M. Cao [22]</td>
<td>PF when necessary</td>
<td>bandwidth-allocation vector, centralized/distributed</td>
</tr>
<tr>
<td>N.B. Salem [26]</td>
<td>RR with spatial re-use</td>
<td>compatibility matrix for collision avoidance, distributed</td>
</tr>
<tr>
<td>N.H. Viadya [27]</td>
<td>PF</td>
<td>back-off interval, distributed</td>
</tr>
<tr>
<td>P. Gupta [29]</td>
<td>PF with service levels</td>
<td>access threshold, distributed</td>
</tr>
<tr>
<td>S. Singh [31]</td>
<td>M-B</td>
<td>bias weight function, centralized</td>
</tr>
<tr>
<td>T.B. Sorensen [32]</td>
<td>PF, RR</td>
<td>TDMA with and without weighting, distributed</td>
</tr>
</tbody>
</table>
2.6. Analysis of assumptions and areas for further research

The following sections will provide an in-depth analysis of some of the assumptions and limitations of the current approaches for fair scheduling in WMNs. As mentioned previously, some of the assumptions this paper will focus on are: limited or no mobility for the MRs, fixed topology of MRs and gateways, the assumption of a single gateway and downlink and uplink equivalence. For more detail on which techniques make which assumptions see summary of Table 2 in Section 2.7.4.

2.6.1. The mobility of mesh routers

The assumption of limited or no mobility for mesh routers in a WMN is made in almost every piece of literature reviewed for this paper. This assumption is important in order to reduce complexities when developing the initial algorithms however in many cases, if this assumption is relaxed it allows for a more general solution which is more flexible and useful. Consider for example a WMN where the MRs are mounted on cars, trains and buses as part of a transit system. The MRs could provide Internet access to passengers on the transit system, allow for wireless surveillance systems on the vehicles to keep passengers safe, or to collect information on the locations of the vehicles to provide estimates on arrival times. In this system we could still make the assumption that the MRs have more resources (electricity, processing and memory) compared with the MCs. So this network would be something between a WMN and an ad-hoc network and could make use of techniques used in either of these as well as those from WLANs for scheduling and load balancing.

2.6.2. Network topology for mesh routers and gateways

Similarly to the assumption mentioned above, many papers assume that the topology of the MRs and gateways is either fixed or rarely changes and as such can be manually configured. This is contrary to one of the most important benefits of WMN. A WMN is supposed to be self-configuring, self-healing and flexible so MRs and gateways should be able to be added/removed and as mentioned above, mobile.

2.6.3. Number of gateways

Again, to keep the scheduling and load balancing algorithm simple, many publications chose to assume that there was only one gateway [26–29,31,35] in the WMN or even assumed that there was not gateway at all (the traffic was limited to local network traffic only) [4,21,22,31]. However, it has been pointed out that one of the main uses for WMN is to provide Internet access with expanded service areas from traditional WLANs so that means the majority of the traffic flow is between the gateways and the MCs [35]. Having only one gateway in this scenario is a major bottleneck so the existing solutions should be extended to be able to support any number of gateways to make a truly scalable WMN.

2.6.4. Downlink and uplink equivalence

Some recent papers have begun to explore whether uplink and downlink scheduling can be treated equally when it comes to scheduling and load balancing. The reasoning behind this is that there could be a flow which makes use of uplink traffic to large extent while hardly requiring any downlink traffic, so if there are different schedules for both uplink and downlink, perhaps a higher throughput and greater fairness could be achieved [28]. In earlier papers, and even many recent papers, it is just assumed that downlink and uplink are equivalent. In some papers only one is dealt with at once (for example just uplink scheduling) leaving the reader with the assumption that the opposite (for example downlink) may be dealt with in the exact same manner.

From the discussions in this section, we conclude that while certain characteristics and assumptions of WLANs and MANETs are very different, it may be possible to make changes that allow for previous solutions to be adapted for WMNs. For example, one promising technique that could be transferred from single hop WLAN to WMNs is proposed [28] where uplink and downlink scheduling are dealt with separately. Many of the limitations of cross-layering techniques are similar to those of normal scheduling techniques in wireless mesh networks. The solutions that currently exist make many assumptions including: single gateways (or no gateways) [2,12,34], limited or no mobility of mesh routers [11,12,23,25], non-overlapping cellular coverage areas [23], static topologies [11,23] and uplink–downlink equivalence. These assumptions leave much future work to be done in the area. If these assumptions are relaxed more general and flexible solutions could be designed. On the other hand as noted in [6,33], the complexity of cross-layered design may be the reason why so few solutions have been extended with cross-layering. This limitation can however be solved if the optimality requirement of the scheduling is relaxed. When this is the case, whole class of relatively simple and efficient scheduling can be implemented in a distributed fashion [33].

Table 2 summarizes the previous categories to allow for easy comparison and identification of areas of future work for scheduling in WMNs. Assumption A is static mobility. Assumption B is static topology. Assumption C is the single gateway assumption. Lastly, assumption D is downlink–uplink equivalence.

3. The proposed approach

This section provides the proposed approach for our fair scheduling algorithm simulation. The performance evaluation was conducted using simulation. The fair scheduling algorithm used in the simulation is based on the algorithm provided in [26], extended to support multiple gateways.
Table 2
Summary of assumptions and previous research in fair scheduling in WMN.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Assumptions</th>
<th>Multi-hop</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>D. Koutsonikolas [4]</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>D. Nandiraju [5]</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>K.N. Ramachandran [17]</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>L. Erwu [18]</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>L. Popa [21]</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>M. Cao [22]</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>N.B. Salem [26]</td>
<td>some</td>
<td>some</td>
</tr>
<tr>
<td>N.H. Viadya [27]</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>N.S.P. Nandiraju [28]</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>P. Gupta [29]</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>S. Singh [31]</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>T.B. Sorensen [32]</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Y. Bejerano [35]</td>
<td>some</td>
<td>some</td>
</tr>
<tr>
<td>Y. Bejerano [36]</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

3.1. Assumptions and overview of the proposed approach

Like most existing research in the area, we have made some assumptions. We assumed that MRs and GWs are not mobile. Their positions are fixed throughout the simulation. This assumption is quite common in many of the existing solutions (see Table 2). There are many benefits and applications of this type of network. It could be used in transit systems, military applications or disaster relief. Rather than having to deal with multiple handoffs of many moving clients, the moving clients could associate with a moving MR. This would allow the network to focus on dealing with only one handoff while all of the MCs associated with the MR retain their attachment to the network. The topology of the network for this simulation remains fixed throughout the entire simulation. In contrast to existing solutions however, we assume that the network may contain multiple gateways. This is an important assumption because limiting the network to one gateway causes an extreme bottleneck at this gateway. Even if the traffic within the network is balanced and fair, having only one gateway can decrease the performance of the network. There are two solutions to this problem. One is to assume that the gateway always has enough capacity to serve the needs of the network, regardless of its size. The other option, which we have chosen in this experiment, is to allow multiple gateways so that the load of the traffic is spread around more evenly. For the initial results presented in this paper, we assume there is no load balancing mechanism within the gateways.

Lastly, the assumption of downlink and uplink equivalence is another common assumption with existing solutions. Several existing proposals only simulate one type of traffic and assume that the same approach could be taken with the other style. For example a proposal may simulate uplink scheduling and assume downlink will work similarly. In our approach, both uplink and downlink traffic are simulated.

3.2. Detailed description of the proposed approach

This section will provide a discussion of the fair scheduling approach with multiple gateways, highlighting the main contributions we have made to this approach.

We have proposed an enhancement of the original fair scheduling approach proposed by [26] which we call the distributed requirement table. The original work proposed only a scheduling, however, does not provide a mechanism for maintaining and collecting requirements. The requirements are required for generating the scheduling since this information tells how busy each link is. Thus we propose a distributed manner of accomplishing this. Each mesh router keeps track of a local requirement table. In this requirement table, the demand on each link between the router and a neighbour is kept. When a new schedule is requested, each gateway asks for the partial requirement tables from each mesh router associated with it.

The gateway then combines these tables to form one complete requirement table which it uses to generate cliques and eventually the scheduling. One main difference from [26] approach is that we assume multiple gateways. This means that each gateway in the network is responsible for scheduling all of the links which will forward packets towards it. The single gateway assumption is a significant one for two reasons: (i) The single gateway causes an extreme bottleneck in the network. All traffic which flows in and out of the network must use this node and so any scheduling work done in the network is limited by the single gateway. (ii) Similarly, the single gateway node causes a single point of failure in the network. If the gateway node is to go down in this scheme, there is no recovery. When multiple gateways are assumed, the bottleneck is eliminated. Not all of the traffic is destined to the same node in the network and is spread more evenly, especially with strategic gateway placement. With a more complex scheme than we proposed, one could further take advantage of the multiple gateways and perform load balancing on the multiple gateways so that under-utilized gateways could be taken advantage of for further performance improvements. Lastly, the single point of failure is eliminated as well. If one gateway experiences an outage, the network has the ability to reconfigure itself to forward packets and perform scheduling from
another gateway. Once the requirement table is formed, the gateway uses this information along with the clique information to form a scheduling plan. The clique information is all of the sets of links which may transmit at the same time without interfering with one another. The clique information is generated once before any transmissions occur in the network in a manner similar to the way neighbours are discovered in [14]. In our system model we assume static nodes and topology, so no nodes are added or removed and there is no mobility. Thus we do not need to generate this information more than once in the life of the simulation. This is important because this operation is very expensive computationally, because clique enumeration is known to be a difficult problem to compute. If we were to assume non-static topology, we may have to make an assumption of a certain network size based on the computational resources of the gateway nodes in the network. Using both the clique information and the requirement information, we can then determine which links should be activated together and for how long. A further modification of this scheme would be to use different characteristics other than demand on a link, for example the quality of the link and the distance from the gateway could also be taken into account using a biasing scheme as we have proposed.

3.2.1. The requirement tables

The type of fairness used in this solution is round-robin style with spatial re-use. We use centralized schedule generation at the gateways which makes use of distributed routing tables located at the mesh routers. We propose the requirement propagation algorithm which allows each gateway to distribute the requirements and routing table for the scheduling into the network. At each mesh router, the path to the gateway is maintained. In this table, requirements for the links on this path are also maintained. For each client requesting to use this mesh router, each link along the way to the gateway in the local table is given a requirement. When the gateway signals the start time for new schedule generation, it requests the local requirement information from all of the mesh routers which are currently using it as their primary gateway. It then combines the requirements to help determine the scheduling as shown in Fig. 2. In Fig. 2, each mesh router has a local requirement table. This requirement table keeps track of the requirement for itself and for all the nodes on the path towards the gateway. A requirement is added when an MC sends data to an MR. At that particular MR, the requirement is incremented for itself and for all hops to the gateway in its local table since all of these nodes will have to relay the packet. A single gateway is responsible for generating the scheduling for all of the nodes which route through it. Then when a new scheduling must be generated, the gateways request the requirement from each table. Each gateway then combines the requirement information from each mesh router with the compatibility matrix. The compatibility matrix represents the links which may transmit simultaneously without interference and is computed or setup manually once when the network is setup. The gateway then computes the scheduling. After the scheduling is computed, START packets are sent to the MRs when they are free to transmit and END packets are sent to the same MRs when their transmission period has ended. This continues until the end of the current schedule and the process repeats.

In our solution, each gateway is responsible for generating a scheduling for the mesh routers making use of it to relay packets to the Internet. The schedule generation algorithm from [3,26] requires that a compatibility matrix be generated for the network before the algorithm operates. The compatibility matrix is a way of representing which links may be activated simultaneously without interference or collisions at MRs. One main difference with our approach is the use of distributed requirement tables located at each of the MRs.

The size of the requirement table at a particular MR is dependent on the number of clients which are associated with it. If there are many clients, the table could become quite large, however this could be solved by placing another MR at the location to alleviate some of the load placed on this particular MR. Similarly, the GW which is responsible for collecting the distributed tables, could end up with a large table as well. Again, this could be solved by introducing another GW to the network which would effectively cut the size of the table in half, if the location of the GW was positioned appropriately.
The consistency of the tables is not particularly related to the size, however an increased table size would likely affect the delay in the network since more information must be transmitted in order to generate a new scheduling.

3.2.2. Requirement propagation

The requirement propagation algorithm given in Algorithm 1, allows the gateway to keep track of the requirements across all of the links. At the MR, a table containing a partial representation of the network is kept for all of the MRs on the way to the gateway. When an MC associates with a given MR, the requirement is incremented for all the MRs along the way to the gateway in the local table. When a new schedule generation is to be completed, the GW requests for the requirements from all of the MRs and combines the results from the partial tables to determine which links must be activated and for how long.

Algorithm 1: Requirement propagation.

\begin{algorithm}
\begin{algorithmic}
\State 1: Associate MC with MR // issued when the MC connects to the network
\State 2: Generate a Client Requirement at MR for the MC // notify MC that MR requests resources
\For {each link between MR and GW}
\State 4: - Requirement(current-link) ++ // increment the requirement in the local table
\EndFor
\For {each Hop}
\State 7: - Requirement(current-link) -- // decrement the requirement in the local table
\EndFor
\State 9: On Drop: For each link between MR and GW
\State 10: - Requirement(current-link) -- // decrement the requirement in the local table
\end{algorithmic}
\end{algorithm}

Algorithm 1. Requirement propagation [10].

In this scheme, each gateway is responsible for generating the centralized scheduling for all of the links routing to it. The distribution and coordination of the scheduling is done through the use of START and END packets. The gateway sends a START packet to the MR when it has scheduled time to send and an END packet when it no longer has permission. It is assumed that these control packets are sent on a different channel from the data and thus do not interfere with data traffic. At the end of one cycle of scheduling, the process is repeated with a new scheduling plan being computed and distributed throughout the network.

The round-robin nature of the scheduling allows the solution to be simple compared to techniques that include weighting functions. At the same time, when compared with a naive round-robin technique, less time is wasted waiting for links which have no traffic to send since time is only allocated to links with requirements. Since we are concerned with fairness among clients who are paying similarly for equal service, this solution works well. Many existing solutions make use of similar round-robin style techniques [4,26,28,33] but none of them use of multiple gateways. Using a single GW to serve a large mesh network is impractical, however, since it becomes a bottleneck quickly as the network size grows. In [26] the solution was distributed in the sense that the scheduling had to be spread around the network to all the MRs from the centralized GW, however, the algorithm presented provided no means for the distribution to be accomplished. We provide a method for this in our solution.

3.2.3. Clique generation

In order to determine which groups of links should be scheduled together, a concept of gain which was introduced in [26] is used to select groups of links which have the greatest load. Gain is defined as the sum of the requirements of all the links minus the greatest requirement. The scheduling algorithm uses the path and requirements information to give permission to certain MRs to transfer at the required timeslots. When the fair scheduling algorithm is enabled, an MR may only send packets when it has permission to do so. If it does not have permission, it retries until a waiting threshold has been crossed at which point the packet is dropped. When collision occurs because a buffer is full the packet is dropped. The performance of the network could be improved further if a retry or backup mechanism was implemented or if load balancing was applied at the GWs.

3.2.4. Schedule generation

Scheduling is generated for all of the mesh routers in the network using the concept of a compatibility matrix similar to that used in [26,30]. The compatibility matrix is then used to determine which links can be enabled at the same time without causing interference. In our network model, this means that the two MRs do not have a common neighbor and are not neighbors with each other. Due to the positioning of the MRs and the communication ranges, if two MRs are not neighbors and do not share a common neighbor, they are not close enough to cause interference with each other and they do not compete for the resources of a common neighbor. This way both may communicate at the same time. The spatial TDMA scheduling allows multiple links to be activated at the same time when they do not interfere. So the network can be used far more efficiently than it could if only one link in the entire network were active [7]. Furthermore, since the algorithm uses the concept of compatibility, no two links are active that compete for resources so collisions are avoided. The solution presented here is different from many other TDMA solutions because it only allocates time for links which actually have requirements associated with them.
4. Performance evaluation of the proposed approach

In this section we will describe in detail the simulation environment, performance metrics and simulation parameters. This will be followed by a discussion of the results of the experiments.

4.1. Simulation environment

The performance evaluation was carried using simulation experiments. The simulation focuses on packet transmission from MRs to GWs. MCs are generated (using a uniform random distribution) at the start of the simulation and are randomly distributed within the simulation environment. Each MC is associated with the closest MR and each MR routes its packets to the closest GW. This means that any packets that experience a collision at the association stage are not counted in the reported results. We consider this problem separate from the one we are trying to address in this paper. In this paper we are concerned with fair scheduling among the MRs. The control packets for distributing the scheduling are assumed to be sent on another channel and thus do not impact the performance of the network. Additionally, the simulation environment acts as an omniscient observer in that it performs the scheduling and distributes in to the gateways. In a real-world implementation this would need to either be performed through a centralized GW or via some kind of distributed GW solution. The interference model assumes that two nodes interfere if they are within range and transmitting at the same time or if there is a buffer collision. When interference occurs, retransmission is allowed until a threshold timeout is reached.

4.2. Performance metrics and simulation parameters

This simulation study uses two performance metrics. The first metric is average packet delivery ratio. It is computed as the ratio of the total number of packets delivered to the total number of packets sent. The second metric used in the simulation is the average delay. It measures the time taken by a packet to reach its destination. These metrics can help to gauge the performance of the protocol effectively.

In order to keep the scheduling algorithm simple, many authors assume one or no gateways [26–28,30,31] in the WMN. However, one of the main uses for WMN is to provide Internet access with expanded service areas from traditional WLANs and hence the majority of the traffic flow is between the gateways and the MCs [35] via MRs. Having only one gateway in this scenario is a major bottleneck so the existing solutions should be extended to be able to support any number of gateways to make a truly scalable WMN.

There are several parameters used in this simulation. The main parameter settings are summarized in Table 3. The two main parameters varied during the simulation were the number of mesh routers and the number of gateways. The retry threshold is used when a collision occurs either from interference or buffer overflow. The packet is allowed to be retransmitted unless the retry threshold has expired. The retry threshold can be adjusted depending on the network conditions.

4.3. Analysis of the experimental results

The performance of the fair scheduling was studied using the two simulation parameters described in the preceding section. The result presented compare both fair scheduling against no scheduling and fair scheduling with multiple gateways against fair scheduling with a single gateway. Uplink traffic only is considered for these results since we consider downlink scheduling a separate problem which can take advantage of caching and multicast to yield further improvements.

Fig. 3 shows the average packet delivery ratio as a function of the number of mesh routers in the network. Results are plotted for the case with a single gateway and five gateways for both fair scheduling and no scheduling. As the network size increases, the difference between the techniques becomes more pronounced. The cases with multiple gateways have the greatest packet delivery ratio. The results show a single gateway with no scheduling performs very poorly delivering only 30% of the packets successfully to the Internet. It is interesting to note that using multiple gateways without fair scheduling can actually perform better than fair scheduling with a single gateway as can be seen in Fig. 4. This is because despite the use of fair scheduling, the single gateway remains the major bottleneck in the network. This demonstrates how important it is to consider the case of multiple gateways.

| Table 3 |
| Simulation parameters. |
| Parameter | Value |
| Environment dimensions | 1000 m × 1000 m |
| Node range | 250 m |
| Number of mesh routers | 10 to 55 |
| Number of mesh clients | 250 |
| Number of gateways | 1 to 6 |
| Mean packet arrival | 0.01 s |
| Mean hop delay | 0.01 s |
| Retry threshold | 0.01 s |
In Fig. 4, average delay as a function of the number of mesh routers is displayed. The case of a single gateway without fair scheduling is again the worst case. The results are similar to Fig. 3 in that multiple gateways have the greatest performance in terms of average delay as well. With small network sizes (under about 30 nodes) all four techniques perform similarly, however once the network becomes larger fair scheduling and multiple gateways maintain the greatest performance in terms of delay. As the number of MRs grows past 35, the Single GW, FS case experiences a drastic decrease in performance. This is likely due to the GW becoming a bottleneck which would not occur until there were many more MRs in the network with multiple GWs. If we extrapolate the graph further, we might expect to see similar performance in the 5 GW, FS case when the number of MRs approaches 150 MRs since this is roughly 30–35 MRs per GW.

Fig. 5 shows the average packet delivery ratio as a function of the number of gateways in the network. These results were compiled with 55 mesh routers because larger network sizes are affected by a lack of gateways the most. This is reinforced by the results in Fig. 6 which show a large difference between the performance with 1 and 6 gateways with both fair scheduling and no scheduling. Additional this figure shows that fair scheduling gives greater performance than no scheduling.

Similarly, Fig. 6 shows the average delay as a function of the number of gateways with a network size of 55. In this case, the results of fair scheduling can best be seen with a single gateway. This is likely because as each additional gateway
is added, both the fair scheduling and the no scheduling cases benefit significantly by easing congestion on the single bottleneck gateway.

Fig. 7 shows a comparison of packet delivery ratio as a function of the number of mesh routers for a single gateway and multiple gateways. The purpose of this result is to highlight the importance of multiple gateways and to compare our approach to that of [26]. As expected, multiple gateways yield higher delivery ratios for all network sizes from 10 to 55. This is likely because on average there are fewer hops between any given MR and its GW. This is important because each hop increases the likelihood of encountering an MR that is busy which could result in packet loss at the worst or delay at best.

Fig. 8 shows the average delay as function of the number of mesh routers and once again compares the case with a single gateway to that with multiple gateways. Again the case with multiple gateways performs better in terms of delay. As in the previous figure, this is likely due to the lower average hops any given node must take to get a gateway. The delay is not only accumulated because of a greater amount of hops in the single GW case but also because of the time spent waiting for a free buffer.
5. Conclusions and future work

We have presented the state of the art in fair scheduling techniques in WMNs. It was noted that it is important to achieve per-client fairness and that fairness should be a balance between hard-fairness and maximum throughput so that end users perceive fair service while the network resources are used efficiently. The techniques investigated in this study were classified according to the type of scheduling and load balancing, metrics or mechanism used, and the management approaches (centralized or distributed). We have proposed and evaluated a fair scheduling technique using multiple GWs. The experimental results have shown that the performance of the network is much better with the fair scheduling enabled than without in terms of packet delivery ratio.

In the future, we plan to experiment with the proposed approach in a test-bed environment. It is often difficult to predict how a protocol or algorithm performs with real hardware. Another goal is that eventually the assumption of static nodes could be relaxed resulting in a mobile mesh network where the mesh clients and mesh routers are not fixed and the topology of the network is extremely dynamic. To further enhance the proposed approaches, load balancing and cross-layer design approaches will be used to reduce the gateway load and also enable intelligent scheduling by exchanging network and link layer status information.