Wireless Personal Communications

Joint Concurrent Routing and Multi-pointer Packet Scheduling in IEEE 802.16 Mesh Networks

--Manuscript Draft--

Manuscript Number:	WIRE-D-15-00387R1
Full Title:	Joint Concurrent Routing and Multi-pointer Packet Scheduling in IEEE 802.16 Mesh Networks
Article Type:	Manuscript
Keywords:	IEEE 802.16, WiMAX, packet scheduling, routing, mesh networks
Corresponding Author:	XUE JUN LI Auckland University of Technology Auckland, NEW ZEALAND
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Auckland University of Technology
Corresponding Author's Secondary Institution:	
First Author:	XUE JUN LI
First Author Secondary Information:	
Order of Authors:	XUE JUN LI
	Maode Ma, PhD
Order of Authors Secondary Information:	
Funding Information:	
Abstract:	IEEE 802.16, also known as Worldwide Interoperability for Microwave Access (WiMAX), is a standardization effort carried out by the IEEE to provide last-mile broadband access to end users. The IEEE 802.16 standard supports two medium access control (MAC) modes - a mandatory point to multipoint (PMP) mode and an optional mesh mode. In this paper, we propose an asymmetric interference aware routing algorithm and a new multipointer approach in implementing scheduling algorithms for IEEE 802.16 mesh networks. We modify three different centralized scheduling algorithms, fixed scheduling, ordered scheduling and per-slot scheduling using multipointer approach to allow for spatial reuse (SR) in IEEE 802.16 mesh networks. Simulation results reveal that fixed scheduling with SR provides the best performance.

Authors' reply to the reviewer's comments of the manuscript WIRE-D-15-00387 entitled "Joint Concurrent Routing and Multipointer Packet Scheduling in WiMAX Mesh Networks"

Xue Jun Li Maode Ma

October 27, 2015

We would like to thank the Editor to give us a chance to revise our paper submitted to Wireless Personal Communications. We appreciate the constructive comments from the reviewers, and hereby we provide the detailed response to each comment and indicate the changes have been made in our revised manuscript. **All changes are also highlighted with underline in our revised manuscript**. The major changes are summarized as follows:

1. We have rewritten the introduction part by adding 11 latest references, and cited them properly in the revised manuscript. Please refer to pp. 2-4.
2. We have highlighted our contribution in terms of three points. Please refer to pp. 3-4.
3. We have added the limitation of the proposed schemes in pg. 17.
4. We have changed the spelling of "Multipointer" to "Multi-pointer".
5. We have changed the word "WiMAX" in the title to "IEEE 802.16" to make it more accurate.

Reply to Reviewer

1. The authors propose an asymmetric interference aware routing algorithm and a new multi-pointer approach in implementing scheduling algorithms for WiMAX mesh networks.

The references in this paper are quite outdated. It is strongly suggested for the authors to add more updated references, to provide the existing examples in the academic and in industry, and to highlight the differences between the proposed system and the existing examples.

Authors' reply: We agree with the reviewer and have added 11 latest references in the revised manuscript. Please refer to [1], [6-14] and [18]. We have analyzed these existing works, and pointed out the main differences between our proposed schemes and the existing ones. Please refer to pp. 2-4.

2. The paper is well written and easy to follow. The innovation factor of the paper must be underlined by the author.

Authors' reply: We thank the reviewer for his/her positive comments on our work. The main contributions have been summarized into three items: We propose a new routing algorithm, asymmetric interference aware routing (AIAR), which is based on effective link rates and adopts the interference model to obtain optimum throughput while considering SR. (ii) We are particularly enlightened by the three suboptimal algorithms originally proposed in [19], namely fixed allocation scheme, ordering scheme and per slot maximum transmission scheme. Our proposed scheduling algorithms are based on similar philosophy, but they are essentially different from those three proposed in [19]. To emphasize the difference, we deliberately rename them as fixed scheduling (FS), ordered scheduling (OS) and per-slot scheduling (PSS). We further modify FS, OS and PSS to implement SR and obtain FS with SR (FS-SR), OS with SR (OS-SR) and PSS with SR (PSS-SR), respectively. (iii) By taking buffer size and hardware delay into account, we propose a new multipointer approach to implement scheduling algorithms so as to minimize scheduling time, to reduce end-to-end delay and to improve system throughput. Simulation results demonstrate the feasibility of these algorithms. Please refer to pp. 3-4.

3. What is the limitation of proposed schemes?

Authors' reply: We have added the limitation of proposed schemes. Briefly, in our proposed schemes, the number of pointers is M, which is equal to the scheduling validity. It will work well if the scheduling period is not excessively long. However, for large IEEE 802.16 mesh networks with fairly fixed network topology, the proposed multi-pointer approach might have a scalability issue. We also need test the proposed algorithms in a WiMAX testbed in order to investigate their real-time performance. Please refer to pg. 17.

Wireless Personal Communications manuscript No.
(will be inserted by the editor)

Joint Concurrent Routing and Multi-pointer Packet Scheduling in IEEE 802.16 Mesh Networks

Xue Jun Li · Maode Ma

Received: 26 February 2015 / Accepted: date

Abstract IEEE 802.16, also known as Worldwide Interoperability for Microwave Access (WiMAX), is a standardization effort carried out by the IEEE to provide last-mile broadband access to end users. The IEEE 802.16 standard supports two medium access control (MAC) modes - a mandatory point to multipoint (PMP) mode and an optional mesh mode. In this paper, we propose an asymmetric interference aware routing algorithm and a new multipointer approach in implementing scheduling algorithms for IEEE 802.16 mesh networks. We modify three different centralized scheduling algorithms, fixed scheduling, ordered scheduling and per-slot scheduling using multipointer approach to allow for spatial reuse (SR) in IEEE 802.16 mesh networks. Simulation results reveal that fixed scheduling with SR provides the best performance.

Keywords WiMAX, IEEE 802.16, packet scheduling, routing, mesh networks, wireless networking, throughput

1 Introduction

The IEEE 802.16 Standard [1], better known as Wireless Interoperability for Microwave Access (WiMAX) was designed to provide last-mile wireless broadband access to users in remote areas. Its performance is comparable to that of a cable network or digital subscriber line (DSL). A WiMAX network is cheap

---

Xue Jun Li (✉)
Private Bag 92006, Auckland 1142, New Zealand
Tel.: +64-9-921-9999
Fax: +64-9-921-9973
E-mail: xuejun.li@aut.ac.nz

Maode Ma
50 Nanyang Avenue, Singapore, 639798
Tel.: +65-67904385
Fax: +65-67920415
E-mail: emdma@ntu.edu.sg

to deploy and easy to maintain. With the advent of other wireless technologies, WiMAX is considered to be a promising and easy alternative to wirelines infrastructure [2].

A IEEE 802.16 network consists of a base station (BS) with backhaul access to the network and many substations (SSs) with advanced multiple-input multiple-output (MIMO) transceivers. The BS and SSs maintain connectivity with all nodes which are within transmission range. The IEEE 802.16 [3] network operates in two modes - the mandatory *point-to-multipoint (PMP) mode* and the optional *mesh mode*. In the *PMP mode*, SSs interact only with the BS, and all traffic exchange occurs only between a SS and the BS. In the *mesh mode*, traffic is allowed to travel via SSs to their final destination. Compared to the *PMP mode*, the *mesh mode* exhibits better scalability, enhanced coverage, higher throughput and stronger resilience to node failures.

Inherently, the IEEE 802.16 standard adopts time division multiple access (TDMA), under which each frame is divided into a number of transmission opportunities that are allocated to SSs using a scheduling algorithm. Scheduling can be centrally carried out at the BS or distributively completed by individual SS using the three-way handshake. Since the IEEE 802.16 standard does not specify any particular scheduling algorithm, scheduling in IEEE 802.16 mesh networks has attracted considerable amount of attention from both academia and industrial communities in recent years.

Liao et al. proposed a clique partitioning approach for centralized scheduling in IEEE 802.16 mesh networks, which was to optimize the schedule length, maximize number of concurrent transmissions and minimize the buffer size required at stations [4]. Huang et al. studied fair rate-balance in order to ensure link fairness and network stability in [5]. First-come-first-serve algorithm was proposed to be used in conjunction with priority algorithm [6]. Joint bandwidth allocation and packet scheduling was studied in order to improve throughput [7]. Mnif et al. evaluated the performance of various scheduling algorithms using OPNET in [8], and proposed enhanced adaptive proportional fairness as a new scheduling algorithm. Similarly, a comparative study was presented in [9]. Channel aware cross-layer scheduling for WiMAX in *PMP mode* was proposed in [10], and a similar concept for mobile WiMAX was presented in [11]. Zubairi et al. studied fair scheduling in WiMAX and Long Term Evolution (LTE) in [12]. Akashdeep et al. presented a survey on scheduling algorithms in IEEE 802.16 networks [13], and Yadav et al. presented their classifications in [14].

Besides scheduling, routing is another important issue in IEEE 802.16 mesh networks, which was missed out in the works mentioned above. Different routing algorithms were proposed for IEEE 802.16 mesh networks. Wei et al. proposed an interference-aware routing scheme and a centralized mesh scheduling scheme in [15]. They mentioned that interference-aware design resulted in better spatial reuse (SR). In [16], Tao et al. proposed to use the protocol interference model to enhance throughput with concurrent transmission. However, none of them considered the effect of scheduling algorithms. Xie et al. investigated video-on-demand streaming over WiMAX and proposed a multicast routing technique [17], where scheduling was achieved with admission control and SR was not included. Guo et al. studied interference-avoidance scheduling for two-tier cluster based routing tree [18], where the intra-cluster scheduling and extra-cluster scheduling was managed by the clusterhead and BS, respectively.

It becomes even more challenging to achieve joint optimization in routing and scheduling in IEEE 802.16 mesh networks. Shetiya and Sharma studied joint routing and centralized scheduling and proposed several simpler suboptimal scheduling algorithms [19]. Each algorithm had a specific node selection mechanism that was used in a vanilla BGreedy algorithm. However, they did not consider buffer constraints of IEEE 802.16 mesh networks. Due to certain hardware buffer queueing and processing, data received in a frame 'n' cannot be made available for transmission until the next frame '$n + 1$'. Furthermore, they did not study the performance of their proposed algorithms in terms of end-to-end delay. Jin et al. showed the NP-completeness of the problem of joint packet scheduling and routing in general topology, and proposed routing/scheduling algorithms for mesh networks based on their study of a linear chain network [20]. Later, Lo and Ou studied the application of BGreedy algorithm [20] in a tree-topology [21]. However, SR was not considered. Nahle and Malouch proposed a joint routing and scheduling algorithm to maximize network throughput [22]. However, they did not discuss the performance of their algorithm in terms of end-to-end delay. Study of schedule length was discussed in [23], where the authors proposed maximum spatial reuse (MSR) algorithm to maximize the SR by minimizing the schedule length and allowing as many concurrent transmission as possible.

We noticed the following three issues through careful study of existing works: (i) optimal scheduling schemes were dependent on the instantaneous queue length [20] [22] [23]. However, for practical implementations of IEEE 802.16 mesh networks, parameters like queue length and network load are not readily available at the BS. We should take this into account when designing scheduling algorithms. (ii) The optimal scheduling algorithms were based on the assumption that all links have equal data rates. However, effective link rates are asymmetric due to varying medium access control (MAC) layer encoding/decoding. (iii) Conventional packet scheduling is based one single pointer, which shifts from the first available slot towards the last one. This works well in IEEE 802.16 networks operating in PMP mode. However, for IEEE 802.16 networks operating in mesh mode, practical buffer constraints and multihop distances could render single-pointer based iteration useless. As a result, multi-pointer based implementation of scheduling algorithms should be adopted. Corresponding to these three issues, the main contributions of this paper are threefold: (i) We propose a new routing algorithm, asymmetric interference aware routing (AIAR), which is based on effective link rates and adopts the interference model to obtain optimum throughput while considering SR. (ii) We are particularly enlightened by the three suboptimal algorithms originally proposed in [19], namely fixed allocation scheme, ordering scheme and per slot maximum transmission scheme. Our proposed scheduling algorithms are based on similar philosophy, but they are essentially different from those three proposed in [19]. To emphasize the difference, we deliberately rename them as fixed scheduling (FS), ordered scheduling (OS) and per-slot scheduling (PSS). We further modify FS, OS and PSS to implement SR and obtain FS with SR (FS-SR), OS with SR (OS-SR) and PSS with SR (PSS-SR), respectively. (iii) By taking buffer size and hardware delay into account, we propose a new multipointer approach to implement scheduling algorithms so as to minimize scheduling time, to reduce end-to-end delay and to improve system throughput. Simulation results demonstrate the feasibility of these algorithms.

The rest of this paper is organized as follows. Section 2 presents the system model with underlying assumptions. In Section 3, we analyze the interference model and propose the AIAR scheme. Section 4 introduces multipointer approach and discusses our proposed FS, OS, PSS, FS-SR, OS-SR and PSS-SR schemes. Simulation design is presented in Section 5. In Section 6, we present the results from the OMNeT++ simulations and evaluate the performance of the multipointer scheduling algorithms. Finally, Section 7 concludes the paper.

2 System Model

In this paper, we consider a IEEE 802.16 mesh network with a BS and 10 SSs as shown in Fig. 1. The network is modeled as a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ is the set of SSs in the network and $\mathcal{E}$ is the set of links with $(i, j) \in \mathcal{E}$ iff SS_i and SS_j are within transmission range. All links in the network are bidirectional, thus, $(i, j) \in \mathcal{E} \Rightarrow (j, i) \in \mathcal{E}$.

In the *mesh mode*, traffic can flow among SSs without going through the BS. Two types of physical layers were defined in the standard, namely *WirelessMAN-OFDM* and *WirelessHUMAN*, operating in the licensed spectrum and unlicensed spectrum under 11GHz, respectively. They both use 256 point Fast Fourier Transform (FFT) in a TDMA/TDM structure for channel access. They support adaptive coding and modulation, and the link rates vary according to the channel conditions.

Both uplink and downlink scheduling can be achieved in a centralized fashion, while we focus on scheduling uplink packets. Scheduling of downlink packets can be done similarly using the same algorithm. We assume that each node is allowed to transmits at a maximum power. Uplink scheduling is carried out using mesh request (*MSH-REQ*) packets generated from each SS once every scheduling period (SP), which is defined as the period over which the generated scheduling map is deemed valid. Each scheduling is hence valid for a SP of K frames. Each frame consists of N time slots, among which *MSH-CTRL-LEN* time slots are reserved for centralized scheduling messages like *MSH-REQ*, *MSH-GRANT*, *MSH-NENT*, etc.

Each node on arrival interacts with the nodes within transmission range and chooses a sponsor node as the parent node, which helps register the new node at the BS. Once the node is registered, it can participate in scheduling by requesting bandwidth from the BS. This is done via the bandwidth

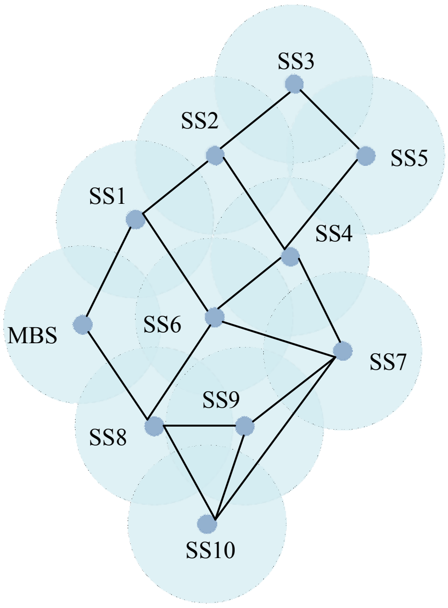


Fig. 1 Mesh network architecture with underlying graph structure

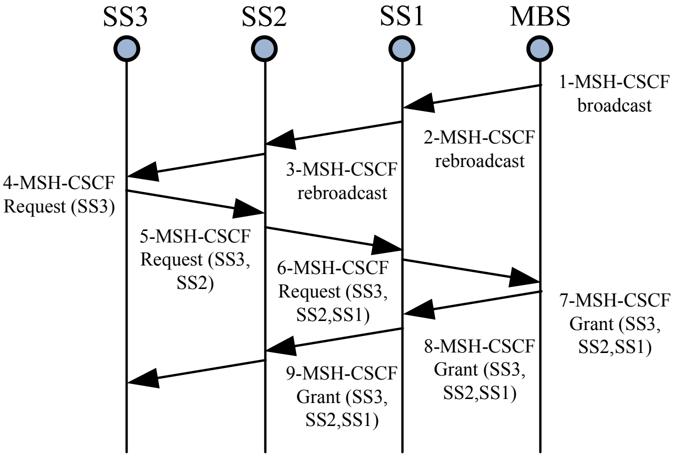


Fig. 2 Bandwidth Request/Grant Mechanism in Mesh Networks

request/grant mechanism. Each node forwards its bandwidth request to its parent node. A parent node assimilates the requests of its children nodes in its own request and sends it to its parent node. This repeats recursively until the BS receives the bandwidth request. The BS executes the scheduling algorithm and broadcasts bandwidth grants to all nodes. This grant is broadcasted throughout the network by intermediate nodes. As illustrated in Fig. 2, due to buffering limitations, the grant/request received from a node in the previous

Table 1 List of Notations

C	concurrent transmission set
H	connection matrix
k	frame number
λ_i	mean number of bits arriving at node i at the beginning of frame k
M	number of SSs in the mesh network
N	number of slots in a frame
n_i	number of slots assigned to node i
$Q_i(k)$	queue length at the beginning of frame k
$r_i\ (k)$	transmission rate of the medium during frame k
r_{ij}	the data rate of the link between SS_i and SS_j
$E[r_{ij}]$	the average data rate of the link (i, j)
σ_i	order of transmission of node i as given in the configuration file
SP	scheduling period
SQ	spillover queue
w	fixed scheduling weighting factor
$X_i(k)$	external arrival at node i during frame k
$Y_i(k)$	arrival from other nodes to node i during frame k

frame can be forwarded only in the next frame. Hence, an entire *MSH-REQ* and *MSH-GRANT* cycles takes two scheduling periods. Hence, packets originating in scheduling period k, will request for resources in scheduling period $k + 1$ and get granted resources in scheduling period $k + 2$. Hence, resources to these packets are actually assigned in scheduling period $k + 3$. This allows us to bound the delay to a maximum value if we provide an efficient routing and scheduling algorithm.

2.1 Spatial Reuse and Interference Model

The IEEE 802.16 standard allows us to implement SR. Two or more nodes can transmit concurrently if they do not interfere with one another. We follow the protocol model and define rules under which links are considered to be interfering. There are two types of interference that a link can experience: (i) primary interference and (ii) secondary interference. All the links in our network are bidirectional. A transmitting node cannot receive at the same time and vice-versa. This type of interference is called primary interference. Secondary interference occurs when an active link interferes with other links which do not share the same source/destination.

3 System Stability and Proposed Routing Scheme

Let us define the following notations in Table 1. Then, the queue length at node i can be calculated as

$$Q_i\left(k+1\right) = \left\{Q_i\left(k\right) - n_i r_i\left(k\right)\right\}^+ + X_i\left(k\right) + Y_i\left(k\right) \tag{1}$$

For the sake of system stability, we need to ensure the capacity of the network must be greater than the demand of the nodes; otherwise, the queue length may continue to increase and the system might become unstable. Thus,

$$n_i E\left[r_i\right] > E\left[X_i + Y_i\right] = \lambda_i + E\left[Y_i\right] \tag{2}$$

In (2), $E[r_i]$ is the output data transmission rate of node i, and the total data originating at other nodes and passing through node i is given by $E\left[Y_i\right] = \sum_{j=1}^{m_i} \lambda_{a_{i,j}}$, where $\{a_{i,1}, a_{i,2}, a_{i,3} \ldots a_{i,m_i}\}$ are the nodes whose data passes through node i. Hence, we get the number of slots required by node i,

$$n_i > \frac{\lambda_i + \sum_{j=1}^{m_i} \lambda_{a_{i,j}}}{E\left[r_i\right]} \tag{3}$$

To be within the stability range, the total number of slots allocated throughout the network should not exceed 'N'- the total number of available slots, i.e., $\sum_{i=1}^{M} n_i \leq N$. By rearranging the terms in (3), we can obtain,

$$\sum_{i=1}^{M} \left(\lambda_i \sum_{j=1}^{h_i} \left(E\left[r_{p_{i,j}}\right] \right)^{-1} \right) < N \tag{4}$$

where $\{p_{i,1}, p_{i,2}, p_{i,3} \ldots p_{i,h_i}\}$ are the intermediate nodes for which data from node i is routed, including node i. From (4), we get the upper bound on the arrival rate λ_i of each node. This is the maximum load on the network. Hence, to maximize the stability region, the value of $\sum_{j=1}^{h_i} \left(E\left[r_{p_{i,j}}\right] \right)^{-1}$ should be minimized. This translates to shortest path routing, which essentially resembles a path that has the maximum rate of transfer to the MBS.

3.1 Asymmetric Interference Aware Routing

Routing in WiMAX can be either fixed or adaptive. However, using adaptive routing in such a dynamic environment like WiMAX will lead to continuous path changes [24]. These changes would affect data delivery. Packets traveling from one node to another could experience out of order delivery and high jitter. Moreover, most data transfer (video and voice, etc) require that packets arrive in the same order in which they have been transmitted. Therefore, fixed routing is adopted for most deployments. Assuming that data from a given node follows a single path, we can obtain an optimum tree routing structure. To enable SR and optimize throughput, we propose a routing algorithm, namely asymmetric interference aware routing (AIAR).

As shown in Algorithm 1, consider a graph $\mathcal{G}\left(\mathcal{V}, \mathcal{E}\right)$ where link $\mathcal{E}_i$ with source $s_{\mathcal{E}i}$ and destination $d_{\mathcal{E}i}$. A link $\mathcal{E}_1$ is said to be non-interfering with another link $\mathcal{E}_2$ if: (i) $d_{\mathcal{E}_2} \notin \mathit{neighbour}\left(s_{\mathcal{E}_1}\right)$, i.e., no receiver in the neighbourhood of the transmitter; (ii) $s_{\mathcal{E}_2} \notin \mathit{neighbour}\left(d_{\mathcal{E}_1}\right)$, i.e., no transmitter in the neighbourhood of the receiver; (iii) $d_{\mathcal{E}_2} \neq d_{\mathcal{E}_1}$, i.e., one node cannot receive from two transmitters.

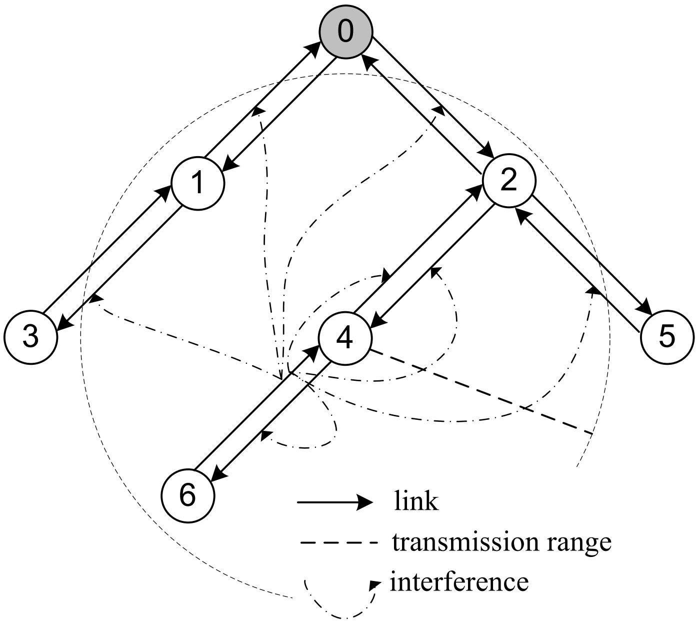


Fig. 3 Mesh Network Architecture with underlying Graph Structure

We use a parameter *link interference parameter*, $\mathcal{L}(x, y)$ to denote the sum of data rates of all links that it interferes with. A node on entry calculates the $\mathcal{L}(x, y)$ of every link it is involved in. It then selects a suitable node as its parent node using *node blocking parameter* $\mathcal{B}(n)$

$$B(n) = \arg \min_{\forall n_i \in \text{neighbours}(n)} \{B(n_i) + L(n, n_i)\} \tag{5}$$

The parent node is hence the node that offers minimal interference with other links. This algorithm is dependent on the order in which the nodes enter the network. Let H denote the hop count of the farthest node from BS, if we execute the algorithm H times, we can obtain an optimal routing tree. Our following discussion on scheduling algorithms is based on an assumption that the minimum interference routing tree is pre-determined.

4 Enhanced Scheduling using Multi-pointer Approach

In this section, the new multipointer scheduling algorithm is presented, followed by modifications to the three scheduling algorithms in order to take into account SR. Finally, a multipointer scheduling example is presented.

Algorithm 1 Asymmetric Interference Aware Routing

```
Input: Graph of the network including all edges and nodes - $\mathcal{G}(\mathcal{V},\mathcal{E})$
Output: Routing tree with parents of each node - $\mathcal{G}^{'}(\mathcal{V},\mathcal{E}^{'})$
Initialize: $\mathcal{V} \leftarrow \{1, 2, 3, ..., M\}$, $\mathcal{V}^{'} \leftarrow \{0\}$
For every edge in $\mathcal{E}$, calculate $\mathcal{L}(x, y)$ from the above given conditions
$\mathcal{B}(n)$=0 for $n$=0; otherwise infinity
while ($\mathcal{V} \neq \emptyset$) do
    Choose a node $i$ from $\mathcal{V}$ with least number of hops to the MBS
    for $j$ = each neighbour of $i$ in $\mathcal{V}$ do
        $\mathcal{B}(i)$ = minimum$\{\mathcal{B}(i), \mathcal{L}(i, j)+\mathcal{B}(j)\}$
        $parent(i) = j$, for which $\mathcal{B}(i)$ is minimum
    end for
    $\mathcal{V} \leftarrow \mathcal{V} - \{i\}$
    $\mathcal{V}^{'} \leftarrow \mathcal{V}^{'} + \{i\}$
    $\mathcal{E}^{'} \leftarrow \mathcal{E}^{'} + \{i, parent(i)\}$
end while
```

4.1 Multi-pointer Algorithm for Packet Scheduling

In most algorithms that we have seen, the scheduling of resources is done from the first available slot towards the last available slot. This is acceptable for the *PMP mode* [25]. However, for the *mesh mode*, buffering constraints and the multihop distance to the destination render the normal iteration useless. Under these circumstances, we need to develop a different approach. Scheduling validity is defined as the number of frames over which a particular schedule grant is valid. For WiMAX deployments with centralized scheduling, the scheduling validity is set to the hop count of the farthest node from the BS. As shown in Algorithm 2, the number of pointers used for scheduling is equal to the scheduling validity. Consequently, we require one pointer for each frame in the scheduling period. When all the slots in that particular frame have been allocated, the pointer is merged with the pointer in the next frame.

Algorithm 2 Multi-pointer Algorithm Structure

```
Initialize: multipointer $P_i \leftarrow$ the first data slot in $i^{th}$ frame, $n^i_j \leftarrow$ the number of slots
    required by the $j^{th}$ node in the $i^{th}$ frame, where $j \leq M, i \leq SP$
for $i = 1 : SP$ do
    while $n^i_j > 0$ do
        $P_i \leftarrow j$
        $n^i_j \leftarrow n^i_j - 1$
        $P_i \leftarrow P_i + 1$
        if $p_i$ = end of frame then
            $P_i \leftarrow P_{i+1}$
        end if
    end while
end for
```

Let us look at how a multipointer algorithm works. Consider a IEEE 802.16 network with four nodes as given in Fig. 4. Since the maximum hop count of

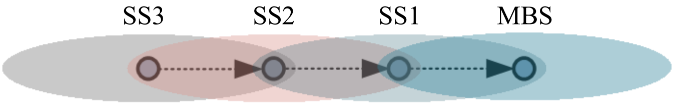


Fig. 4 IEEE 802.16 network with four nodes

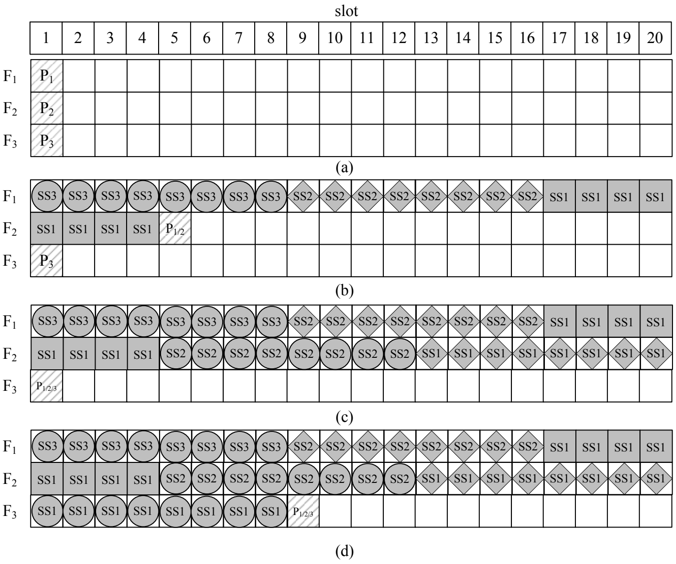


Fig. 5 Working of Multi-pointer Algorithm: (a) before scheduling (b) after first frame of scheduling (c) after second frame of scheduling (d) after third frame of scheduling

the farthest node to the MBS is 3, we use a scheduling period of 3 frames. As depicted in Fig. 5(a), for the sake of simplicity, let us assume that each frame has 20 time slots, where the pointers in each of these frames are denoted by P_1, P_2 and P_3, and initialized to the first slot in the 1^{st}, 2^{nd} and 3^{rd} frame, respectively. Assuming that each node has a request of eight slots, we can obtain the results are shown in Fig. 5 after running the multipointer algorithm. In Fig. 5, a slot with a circle mark inside indicates its allocation to a node; a slot with a diamond mark inside indicates during which data transfer is being done; a shaded slot without any mark indicates during which data is received for a node.

4.2 Fixed Scheduling with Spatial Reuse

FS follows the principle of allocating resources proportional to the arrival rate at a node. The arrival rate at the node comprises of external arrivals from other nodes and arrivals from users of the node itself. The number of slots required for node i without concurrency is given in (3). We calculate the number of slots for all nodes and then allocate slots to the nodes in order of their hop count from the BS, as specified by σ_i.

To incorporate SR in terms of concurrent transmissions, we define a child matrix $c_{i,j}$, to denote whether or not a node j is a child/sub-child of node i. This helps in calculating the total arrival rate at a given node. Then, we use $I_{i,j}$ to define the interference matrix, where if link i interferes with link j, then $I_{i,j}$ is equal to 1. We should note that the interference matrix is a symmetric matrix. Hence, $I_{i,j}$=$I_{j,i}$$\leq$ 1.

As we consider only uplink scheduling and a routing tree, each node has only one uplink, and this link from node i to its parent node can be denoted as $L_{i,parent(i)}$. Since no other link from i can serve as an uplink, we can simplify the notation as L_i. The order in which each node transmits is given in *MSH-CSCF*. This is denoted by σ_i for node i. The scheduling output gives a set of nodes that can transmit concurrently during a slot.

FS-SR follows a staggered approach. The arrival rate to each node is calculated as a sum of arrival rate to itself and its children nodes. The number of slots hence required in total at each node is the sum arrival rate divided by the rate of data transfer per slot. In particular, if the total number of slots cannot be serviced by the scheduling period, the number of slots granted to each node is multiplied by a scheduling weight, w, which is defined as the ratio of the number of slots in the scheduling period to the total number of slots required.

Algorithm 3 Fixed Scheduling with Spatial Reuse

Input: arrival rates, children matrix, order of each node **Output:** scheduling grant for all nodes

```
 1: calculate $n_i$ and $w$
 2: if $w < 1$ then
 3:     $n_i = n_i \times w, i = 1 : M$
 4: end if
 5: Initialize: multipointers for all frames $P_i, i = 1 : SP$
 6: set $Q_i = \lambda_i \times SP$
 7: for i=1:SP do
 8:     for j=1:N do
 9:         find node $n_s$=arg $\min_{i=1...M} \{\sigma_i\}$, $Q_{n_s} > 0$ and $I_{n_s n_t} = 0\ \forall\ n_t \in C$
10:         $C \leftarrow \{C, n_s\}$
11:         continue steps 8-10 until all nodes have been considered
12:         if $C = \emptyset$ then
13:             break
14:         end if
15:         allocate slot $P_i$ to C
16:         $P_i \leftarrow P_i + 1$
17:         $Q_k = Q_k - r_k$ and $SQ_{parent(k)} = SQ_{parent(k)} + r_k, \forall k \in C$
18:         if $P_i$=end of frame then
19:             $P_i = P_{i+1}$
20:         end if
21:     end for
22:     $Q_k = Q_k + SQ_k, k = 1 : M$
23: end for
```

4.3 Ordered Scheduling with Spatial Reuse

In OS, the scheduling weight for node is defined as

$$w_i = \sum\nolimits_{j=1}^{h_i} \left(E \left[r_{p_{i,j}} \right] \right)^{-1} \tag{6}$$

Given the parameter w_i, interference matrix I, and individual queue lengths Q_i, OS orders the nodes based on their w_i values and assigns each slot to the first node that can fully utilize that slot. Preference is given to a node with a lower w_i value. Furthermore, we propose OS-SR to implement SR (See Algorithm 4). The concurrent set 'C' allocated to each slot contains the set of all nodes that can transmit during that slot.

For each slot, OS-SR first selects the node that has the minimum w_i value and queue length longer enough to use the entire slot. Then, it selects other nodes that can fully utilize a slot and can transmit concurrently with the first selected node in order of their respective parameter values. The corresponding uplinks of these nodes are added into the concurrent set 'C'. This method continues until all nodes are scheduled.

Algorithm 4 Ordered Scheduling with Spatial Reuse

Input: arrival rates, children matrix, order of each node
Output: scheduling grant for all nodes

```
 1: calculate $w_i$
 2: **Initialize:** multipointers $P_i, i = 1 : SP$
 3: set queue length of all nodes to $Q_i$ from $MSH - REQ$ packet received
 4: **for** i=1:SP **do**
 5:     **for** j=1:N **do**
 6:         find node $n_s$=$\arg\min_{i=1 \ldots M} \{w_i\}$, $Q_{n_s} > r_{n_s}$ and $I_{n_s n_t} = 0 \; \forall \; n_t \in C$
 7:         $C \leftarrow \{C, n_s\}$
 8:         continue steps 6-8 until all nodes have been considered
 9:         **if** $C = \emptyset$ **then**
10:             find node $n_s$=$\arg\min_{i=1 \ldots M} \{w_i\}$, $Q_{n_s} > 0$ and $I_{n_s n_t} = 0 \; \forall \; n_t \in C$
11:             $C \leftarrow \{C, n_s\}$
12:             continue steps 9-11 until all nodes have been considered
13:         **end if**
14:         **if** $C = \emptyset$ **then**
15:             break
16:         **end if**
17:         allocate slot $P_i$ to C
18:         $P_i \leftarrow P_i + 1$
19:         $Q_k = Q_k - min\{Q_k, r_k\}$ and $SQ_{parent(k)} = SQ_{parent(k)} + min\{Q_k, r_k\}, \forall k \in C$
20:         **if** $P_i$=end of frame **then**
21:             $P_i = P_{i+1}$
22:         **end if**
23:     **end for**
24:     $Q_k = Q_k + SQ_k, k = 1 : M$
25: **end for**
```

4.4 Per Slot Scheduling with Spatial Reuse

PSS works on a dynamic programming principle (See Algorithm 5) to schedule one slot at a time to a node which can transmit the maximum number of bits in that given slot. With SR, a slot is scheduled to nodes in the concurrent set 'C', which can transmit simultaneously without experiencing severe interference.

One advantage of PSS is that the average resources required at nodes which have a higher egress rate is low. However, the corresponding disadvantage is that under high loads, the nodes with a lower transmission rate are allocated resources towards the end of the scheduling frame. Hence, these nodes have longer queue lengths. The data drop probability at these nodes is also high due to the prioritized scheduling of nodes with higher egress rates.

Algorithm 5 Per Slot Scheduling (Concurrency) with Spillover

Input: arrival rates, children matrix, order of each node
Output: scheduling scheme for all nodes

```
 1: Initialize: multipointers $P_i, i = 1 : SP$
 2: set queue length of all nodes to $Q_i$ from $MSH - REQ$ packet received
 3: for i=1:SP do
 4:     for j=1:N do
 5:         find node $n_s$=arg max$_{\forall i \notin C}$ $\{\min(Q_i, r_i)\}$ and $I_{n_s n_t} = 0 \ \forall \ n_t \in C$
 6:         $C \leftarrow \{C, n_s\}$
 7:         continue steps 6-8 until all nodes have been considered
 8:         if $C = \emptyset$ then
 9:             break
10:         end if
11:         allocate slot $P_i$ to C
12:         $P_i \leftarrow P_i + 1$
13:         $Q_k = Q_k - min\{Q_k, r_k\}$ and $SQ_{parent(k)} = SQ_{parent(k)} + min\{Q_k, r_k\}, \forall k \in C$
14:         if $P_i$=end of frame then
15:             $P_i = P_{i+1}$
16:         end if
17:     end for
18:     $Q_k = Q_k + SQ_k, k = 1 : M$
19: end for
```

5 Simulation Design

Our simulation in OMNeT++ used the IEEE 802.16 mesh network as shown in Fig. 1, designed using a Network Descriptor (NED). If a node was in transmission range of another node, then it was connected to that node via a link. All links were bidirectional and the link parameters were governed by their burst profiles as shown in Table 2. The burst profile governed the modulation, coding rate of that link. Hence, a different burst profile translated to a different data rate. The link rates (in Mbps) between any two nodes were predetermined, as specified in a NED, and they were aggregated in a connection matrix, H with element (i, j) corresponding to burst profile of link between node i and node j. $H_{11\times 11} =$

$$\begin{pmatrix}
0&6&0&0&0&0&0&0&4&0&0\\
6&0&3&0&0&0&2&0&0&0&0\\
0&3&0&6&4&0&0&0&0&0&0\\
0&0&6&0&0&3&0&0&0&0&0\\
0&0&4&0&0&4&0&6&0&0&0\\
0&0&0&3&4&0&0&0&0&0&0\\
0&2&0&0&0&0&0&3&2&0&0\\
0&0&0&0&6&0&3&0&0&2&2\\
4&0&0&0&0&0&2&0&0&3&3\\
0&0&0&0&0&0&0&2&3&0&0\\
0&0&0&0&0&0&0&2&3&0&0
\end{pmatrix}$$

Table 2 Physical Parameters of Burst Profile

Burst Profile No.	Modulation	Coding Rate	Uncoded bytes per OFDM Symbol	Uncoded bytes per slot
1	QPSK	1/2	24	96
2	QPSK	3/4	36	144
3	16QAM	1/2	48	192
4	16QAM	3/4	72	288
5	64QAM	2/3	96	384
6	64QAM	3/4	108	432

The maximum load on the network with respect to arrival rate at each SS was calculated using the method in Section 3 and the load is 1.74Mbps, which was corresponding to a normalized load of 1.0. The normalized load on the network was varied from 0 to 1.0 with a step size of 0.1. Poisson packet arrival was considered and the packet size was varied exponentially with respect to the mean. Traffic generation occurred in the *trafgen* submodule of the substation. The mean packet size was fixed at 768 bits. If all the nodes have the same average arrival rate λ_i, the average size of the packet is fixed at L and the packet inter-arrival time is calculated as

$$E[\tau] = \frac{L}{\lambda} \tag{7}$$

All traffic is directed towards the BS. The traffic generator can also be adapted to include priority, tardiness and drop probability of the packets, which has been left out for our future study. Upon receiving data packets, either self-generated or from other nodes, SSs put them into a reserved buffer, and then moved them into the transmitter buffer at the beginning of each frame. This perfectly simulated the buffer constraints of the IEEE 802.16 network. The queuing rule was *first-in first-out (FIFO)*. Other queuing methods like *earliest deadline first* and *fair scheduling* could also be employed.

Each packet was time-stamped on of creation at the traffic-generator, and given a time-to-live (TTL) value that was also varied exponentially with respect to the mean. We tested our algorithms for four mean TTL values: 60ms, 120ms, 240ms and 480ms. Packets which experienced a delay longer than their TTL values were dropped by intermediate nodes or the BS.

The simulation was run for $5000s$ or 5×10^6 frames. We can safely assume that simulations could accurately record the real-time performance of the IEEE 802.16 mesh network. Each frame was assigned a duration of $10ms$ and the scheduling period was set to 4 frames, as the maximum hop-count amongst all leaf nodes is 4 in Fig. 1.

During initialization, we first execute AIAR to obtain the optimal routing tree and then applied a scheduling algorithm. We simulated the request and grant process and data transmission with buffer constraints in the IEEE 802.16 mesh network, and analyzed the performance of each of the six scheduling algorithms discussed above, namely, FS, OS, PSS, FS-SR, OS-SR and PSS-SR.

6 Results Analysis

To analyze the performance of a IEEE 802.16 network, we mainly look into two main aspects: (i) *mean delay* – end-to-end time taken by a packet to reach its destination; and (ii) *throughput* – the number of packets which have been successfully delivered.

6.1 Mean Delay

The delay of a packet is defined as the time it takes from the time when it is created by the traffic generator to the time when it reaches its destination.

As shown in Fig. 6, at low traffic load with low TTL values (60ms∼120ms), FS outperformed other algorithms, which makes FS a suitable scheduling algorrithm for low-data-rate and time-critical applications like VoIP and Telnet. Next, scheduling priority under OS was to the node with the fastest path towards the BS; while under PSS was to the fastest link. Due to this *selective* nature of these algorithms, nodes with slower links/paths were allocated resource at the end of a scheduling period, leading to a higher mean delay. In addition, every scheme outperformed its corresponding counterpart when implemented with SR. This reduction in mean delay could be attributed to less mean medium access time due to the concurrent transmissions.

At low traffic load with high TTL value (240ms∼480ms), there was no apparent difference in the performance of the six scheduling schemes. Scheduling algorithms employing SR performed just marginally better because they increased the effective capacity of the network, thereby enabling more packets to reach the destination faster. Furthermore, we find that *proactive* scheduling algorithms like FS significantly outperformed the *reactive* scheduling algorithms like OS and PSS. If we define ‘reaction time’ as the number of scheduling periods taken by the mesh network to respond to a given mesh-request. As shown in Fig. 2, for PSS and OS that are based on the queue length of nodes, the reaction time is 2 scheduling periods. However, for FS, this is reduced to zero, as FS is based on arrival rate at each node. Therefore, FS offered the smallest delay amongst the six.

At high traffic load with low TTL values (60ms∼120ms), OS and PSS outperformed FS. This is because the fact that FS would allocate more resources to weaker links, giving rise to a longer waiting time for data with stronger links and an overall increase in average delay. In particular, the '*selective*' nature of OS and PSS did not favor scheduling to moderate and weak paths/links. Some packets were lost due to excessive delay and many packets were delayed as the algorithm allocated resources first to stronger links and paths and then to weaker links. The overall delay hence increases at high traffic load for these algorithms. Packets with slower links (See Fig. 6) were dropped and hence the average delay of packet delivery was comparable between OS and PSS. Next, with SR, OS-SR and PSS-SR outperformed OS and PSS, respectively. Furthermore, FS-SR provided the best performance. This was because of simultaneous allocations to strong and weak links at the same time. Since FS was not selective in nature, data were allocated slots uniformly and hence we had a lower average delay.

At high traffic load with high TTL value (240ms∼480ms), FS-SR offered the best average delay under all loads and all TTL conditions, and it could be used for almost all applications. Furthermore, PSS outperformed OS, while OS-SR and PSS-SR performed similarly.

6.2 Throughput

Throughput of a system is a measure of the data generated by the node compared to the amount of data received by the MBS. In our case, data which were not delivered with a TTL were dropped and hence an unfair scheduling algorithm may lead to a low throughput.

As shown in Fig. 7, at low traffic load, FS outperformed FS-SR, implying that SR does not improve throughput. Furthermore, the performance of other four schemes was about the same. Interestingly, the performance of six schemes were converging when the TTL values were increased at low traffic load.

At high traffic load, without SR, PSS provided the highest throughput as it strived to transmit the maximum number of bits in every slot. The path-selective OS provided a lower throughput than PSS; FS resulted in the lowest throughput. With SR, FS-SR offered the maximum throughput. The proactive scheduling saved '*reaction time*' and hence allowed faster delivery of nodes. Furthermore, OS-SR and PSS-SR offered comparable throughput. When the TTL values were increased, the performance of FS-SR, OS-SR and PSS-SR converged.

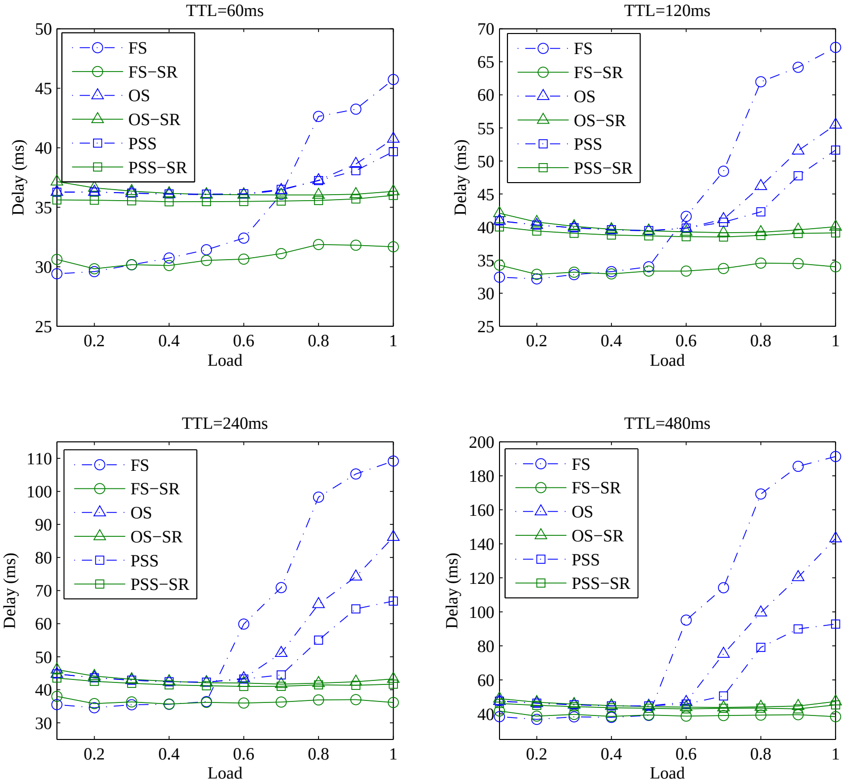


Fig. 6 Mean Delay Analysis

7 Conclusion

A concurrent routing scheme with spatial reuse (SR), asymmetric interference aware routing (AIAR) was proposed. Then, a multi-pointer approach of implementing scheduling algorithms for IEEE 802.16 mesh networks was proposed. Six scheduling algorithms, *fixed scheduling (FS)*, *ordered scheduling (OS)* and *per-slot scheduling (PSS)*, *FS-SR*, *OS-SR* and *PSS-SR* were proposed. Through computer simulations, we found that without SR, *PSS* provides highest throughput while *FS* gives lowest packet delay. While incorporating SR, the *FS-SR* scheme offers the best overall performance.

In our proposed multi-pointer approach of implementing scheduling algorithm, the number of pointers used for scheduling is equal to the scheduling validity. As such, it will work well if the scheduling period is not excessively long. However, for large IEEE 802.16 mesh networks with fairly fixed network topology, the proposed multi-pointer approach might have a scalability issue. This has been left for our future work in this topic. Furthermore, we also plan to test the proposed algorithms in a WiMAX testbed in order to investigate their real-time performance.

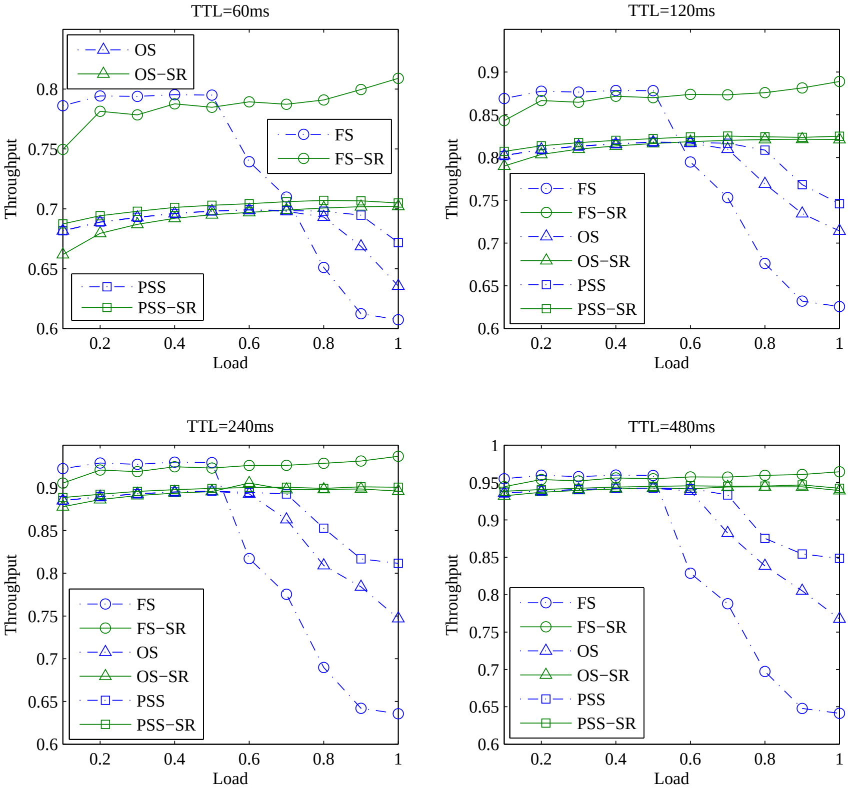


Fig. 7 Throughput Analysis

References

1. Ieee standard for local and metropolitan area networks part 16: Air interface for broadband wireless access systems amendment 3: Advanced air interface. *IEEE Std 802.16m-2011(Amendment to IEEE Std 802.16-2009)*, pages 1–1112, 2011.
2. Gorbenko, N. and Jean-Pierre, E. and Almuhtadi, W. and Srinivasan, A. Comparison of simulated and real network traffic results for multimedia streaming over WiMAX networks with QoS scheduling. In *Proc. of IEEE CCECE'15*, pages 1324–1328.
3. Ieee standard for local and metropolitan area networks - part 16: Air interface for fixed and mobile broadband wireless access systems - amendment 3: Management plane procedures and services (amendment to ieee 802.16-2004). *IEEE Approved Draft Std P802.16g/D9, Apr 07*, 2007.
4. Wen-Hwa Liao, S. P. Kedia, and A. K. Dubey. A centralized scheduling algorithm for wimax mesh network. In *Proc. of IEEE NOMS'10*, pages 858–861.
5. Yanbin Huang, Shoubao Yang, Da Teng, Weiqing He, and Lihua Xie. A fair rate-balance link scheduling in wimax mesh network. In *Proc. of IEEE WCNIS'10*, pages 355–359.
6. Kapila, Deepak and Gill, S. S. A hybrid approach for scheduling in WIMAX for reduction of process waiting time. *International Journal of Innovation and Applied Studies*, 9(4):1620–1625, 2014.
7. Kumar, Niharika and Algur, Siddu. Novel bandwidth allocation and packet scheduling strategy for WiMAX. *International Journal of Wireless and Mobile Computing*, 7(6):517–527, 2014.

8. Mnif, K. and Khdhir, R. and Kamoun, L. Evaluation and comparaison of scheduling algorithms in Wimax networks. In *Proc. of ICMCS'14*, pages 884–889.
9. Rehman, A. and Shakir, Mustafa. Comparative Analysis of Scheduling Algorithms in IEEE 802.16 WiMAX Radio. *International Journal of Computing and Digital Systems*, 3(2):161–172, 2014.
10. Randhini, R. and Devarajan, N. Comparison for WiMAX scheduling algorthims and proposal quality of service improvement in WiMAX networks. *American Journal of Applied Science*, 11(1):8–16, 2014.
11. Yang, Feng-Ming and Chen, Wei-Mei and Wu, Jean-Lien C. A dynamic strategy for packet scheduling and bandwidth allocation based on channel quality in IEEE 802.16e OFDMA system. *Journal of Network and Computer Applications*, 39:52–60, 2014.
12. Zubairi, Junaid Ahmed and Erdogan, Erdem and Reich, Shaun. Experiments in fair scheduling in 4G WiMAX and LTE. In *Proc. of HPCS'15*, pages 277–282.
13. Akashdeep and Kahlon, Karanjeet S. and Kumar, Harish. Survey of scheduling algorithms in IEEE 802.16 PMP networks. *Egyptian Informatics Journal*, 15(1):25–36, 2014.
14. Yadav, A. L. and Vyavahare, P. D. and Bansod, P. P. Review of WiMAX Scheduling Algorithms and Their Classification. *Journal of The Institution of Engineers (India): Series B*, 96(2):197–208, 2015.
15. Hung-Yu Wei, S. Ganguly, R. Izmailov, and Z. J. Haas. Interference-aware ieee 802.16 wimax mesh networks. In *Proc. of IEEE VTC'05-Spring*, volume 5, pages 3102–3106 Vol. 5.
16. Jian Tao, Fuqiang Liu, Zhihui Zeng, and Zhangxi Lin. Throughput enhancement in wimax mesh networks using concurrent transmission. In *Proc. of WCNM'05*, volume 2, pages 871–874.
17. Fei Xie, K. A. Hua, and Ning Jiang. Achieving true video-on-demand service in multi-hop wimax mesh networks. In *Proc. of IEEE LCN'07*, pages 287–294.
18. Guo, Ming-Huang and Deng, Der-Jiunn and Liaw, Horng-Twu and Park, Jong Hyuk. An efficient route scheduling mechanism for WiMAX network. *The Knowledge Engineering Review*, 29(04):452–462, 2014.
19. H. Shetiya and V. Sharma. Algorithms for routing and centralized scheduling in ieee 802.16 mesh networks. In *Proc. of IEEE WCNC'06*, volume 1, pages 147–152.
20. Fanchun Jin, A. Arora, Hwang Jinho, and H. A. Choi. Routing and packet scheduling in wimax mesh networks. In *Proc. of BROADNETS'07*, pages 574–582.
21. Shou-Chih Lo and Lyu-Chen Ou. Efficient algorithms for routing and centralized scheduling for ieee 802.16 mesh networks. In *Proc. of SCALCOM-EMBEDDEDCOM'09*, pages 212–217.
22. S. Nahle and N. Malouch. Joint routing and scheduling for maximizing fair throughput in wimax mesh network. In *Proc. of IEEE PIMRC'08*, pages 1–5.
23. J. El-Najjar, C. Assi, and B. Jaumard. Joint routing and scheduling in wimax-based mesh networks: A column generation approach. In *Proc. of IEEE WoWMoM'09*, pages 1–10.
24. So-In Chakchai, R. Jain, and A. K. Tamimi. Scheduling in ieee 802.16e mobile wimax networks: Key issues and a survey. *IEEE Journal on Selected Areas in Communications*, 27(2):156–171, 2009.
25. Najah Abu Ali, Pratik Dhrona, and Hossam Hassanein. A performance study of uplink scheduling algorithms in point-to-multipoint wimax networks. *Computer Communications*, 32(3):511–521, 2009.

Wireless Personal Communications manuscript No.
(will be inserted by the editor)

Joint Concurrent Routing and Multi-pointer Packet Scheduling in IEEE 802.16 Mesh Networks

Xue Jun Li · Maode Ma

Received: 26 February 2015 / Accepted: date

Abstract IEEE 802.16, also known as Worldwide Interoperability for Microwave Access (WiMAX), is a standardization effort carried out by the IEEE to provide last-mile broadband access to end users. The IEEE 802.16 standard supports two medium access control (MAC) modes - a mandatory point to multipoint (PMP) mode and an optional mesh mode. In this paper, we propose an asymmetric interference aware routing algorithm and a new multipointer approach in implementing scheduling algorithms for IEEE 802.16 mesh networks. We modify three different centralized scheduling algorithms, fixed scheduling, ordered scheduling and per-slot scheduling using multipointer approach to allow for spatial reuse (SR) in IEEE 802.16 mesh networks. Simulation results reveal that fixed scheduling with SR provides the best performance.

Keywords WiMAX, IEEE 802.16, packet scheduling, routing, mesh networks, wireless networking, throughput

1 Introduction

The IEEE 802.16 Standard [1], better known as Wireless Interoperability for Microwave Access (WiMAX) was designed to provide last-mile wireless broadband access to users in remote areas. Its performance is comparable to that of a cable network or digital subscriber line (DSL). A WiMAX network is cheap

---

Xue Jun Li (✉)
Private Bag 92006, Auckland 1142, New Zealand
Tel.: +64-9-921-9999
Fax: +64-9-921-9973
E-mail: xuejun.li@aut.ac.nz

Maode Ma
50 Nanyang Avenue, Singapore, 639798
Tel.: +65-67904385
Fax: +65-67920415
E-mail: emdma@ntu.edu.sg

to deploy and easy to maintain. With the advent of other wireless technologies, WiMAX is considered to be a promising and easy alternative to wirelines infrastructure [2].

A IEEE 802.16 network consists of a base station (BS) with backhaul access to the network and many substations (SSs) with advanced multiple-input multiple-output (MIMO) transceivers. The BS and SSs maintain connectivity with all nodes which are within transmission range. The IEEE 802.16 [3] network operates in two modes - the mandatory *point-to-multipoint (PMP) mode* and the optional *mesh mode*. In the *PMP mode*, SSs interact only with the BS, and all traffic exchange occurs only between a SS and the BS. In the *mesh mode*, traffic is allowed to travel via SSs to their final destination. Compared to the *PMP mode*, the *mesh mode* exhibits better scalability, enhanced coverage, higher throughput and stronger resilience to node failures.

Inherently, the IEEE 802.16 standard adopts time division multiple access (TDMA), under which each frame is divided into a number of transmission opportunities that are allocated to SSs using a scheduling algorithm. Scheduling can be centrally carried out at the BS or distributively completed by individual SS using the three-way handshake. Since the IEEE 802.16 standard does not specify any particular scheduling algorithm, scheduling in IEEE 802.16 mesh networks has attracted considerable amount of attention from both academia and industrial communities in recent years.

Liao et al. proposed a clique partitioning approach for centralized scheduling in IEEE 802.16 mesh networks, which was to optimize the schedule length, maximize number of concurrent transmissions and minimize the buffer size required at stations [4]. Huang et al. studied fair rate-balance in order to ensure link fairness and network stability in [5]. First-come-first-serve algorithm was proposed to be used in conjunction with priority algorithm [6]. Joint bandwidth allocation and packet scheduling was studied in order to improve throughput [7]. Mnif et al. evaluated the performance of various scheduling algorithms using OPNET in [8], and proposed enhanced adaptive proportional fairness as a new scheduling algorithm. Similarly, a comparative study was presented in [9]. Channel aware cross-layer scheduling for WiMAX in *PMP mode* was proposed in [10], and a similar concept for mobile WiMAX was presented in [11]. Zubairi et al. studied fair scheduling in WiMAX and Long Term Evolution (LTE) in [12]. Akashdeep et al. presented a survey on scheduling algorithms in IEEE 802.16 networks [13], and Yadav et al. presented their classifications in [14].

Besides scheduling, routing is another important issue in IEEE 802.16 mesh networks, which was missed out in the works mentioned above. Different routing algorithms were proposed for IEEE 802.16 mesh networks. Wei et al. proposed an interference-aware routing scheme and a centralized mesh scheduling scheme in [15]. They mentioned that interference-aware design resulted in better spatial reuse (SR). In [16], Tao et al. proposed to use the protocol interference model to enhance throughput with concurrent transmission. However, none of them considered the effect of scheduling algorithms. Xie et al. investigated video-on-demand streaming over WiMAX and proposed a multicast routing technique [17], where scheduling was achieved with admission control and SR was not included. Guo et al. studied interference-avoidance scheduling for two-tier cluster based routing tree [18], where the intra-cluster scheduling and extra-cluster scheduling was managed by the clusterhead and BS, respectively.

It becomes even more challenging to achieve joint optimization in routing and scheduling in IEEE 802.16 mesh networks. Shetiya and Sharma studied joint routing and centralized scheduling and proposed several simpler suboptimal scheduling algorithms [19]. Each algorithm had a specific node selection mechanism that was used in a vanilla BGreedy algorithm. However, they did not consider buffer constraints of IEEE 802.16 mesh networks. Due to certain hardware buffer queueing and processing, data received in a frame 'n' cannot be made available for transmission until the next frame '$n+1$'. Furthermore, they did not study the performance of their proposed algorithms in terms of end-to-end delay. Jin et al. showed the NP-completeness of the problem of joint packet scheduling and routing in general topology, and proposed routing/scheduling algorithms for mesh networks based on their study of a linear chain network [20]. Later, Lo and Ou studied the application of BGreedy algorithm [20] in a tree-topology [21]. However, SR was not considered. Nahle and Malouch proposed a joint routing and scheduling algorithm to maximize network throughput [22]. However, they did not discuss the performance of their algorithm in terms of end-to-end delay. Study of schedule length was discussed in [23], where the authors proposed maximum spatial reuse (MSR) algorithm to maximize the SR by minimizing the schedule length and allowing as many concurrent transmission as possible.

We noticed the following three issues through careful study of existing works: (i) optimal scheduling schemes were dependent on the instantaneous queue length [20] [22] [23]. However, for practical implementations of IEEE 802.16 mesh networks, parameters like queue length and network load are not readily available at the BS. We should take this into account when designing scheduling algorithms. (ii) The optimal scheduling algorithms were based on the assumption that all links have equal data rates. However, effective link rates are asymmetric due to varying medium access control (MAC) layer encoding/decoding. (iii) Conventional packet scheduling is based one single pointer, which shifts from the first available slot towards the last one. This works well in IEEE 802.16 networks operating in PMP mode. However, for IEEE 802.16 networks operating in mesh mode, practical buffer constraints and multihop distances could render single-pointer based iteration useless. As a result, multi-pointer based implementation of scheduling algorithms should be adopted. Corresponding to these three issues, the main contributions of this paper are threefold: (i) We propose a new routing algorithm, asymmetric interference aware routing (AIAR), which is based on effective link rates and adopts the interference model to obtain optimum throughput while considering SR. (ii) We are particularly enlightened by the three suboptimal algorithms originally proposed in [19], namely fixed allocation scheme, ordering scheme and per slot maximum transmission scheme. Our proposed scheduling algorithms are based on similar philosophy, but they are essentially different from those three proposed in [19]. To emphasize the difference, we deliberately rename them as fixed scheduling (FS), ordered scheduling (OS) and per-slot scheduling (PSS). We further modify FS, OS and PSS to implement SR and obtain FS with SR (FS-SR), OS with SR (OS-SR) and PSS with SR (PSS-SR), respectively. (iii) By taking buffer size and hardware delay into account, we propose a new multipointer approach to implement scheduling algorithms so as to minimize scheduling time, to reduce end-to-end delay and to improve system throughput. Simulation results demonstrate the feasibility of these algorithms.

The rest of this paper is organized as follows. Section 2 presents the system model with underlying assumptions. In Section 3, we analyze the interference model and propose the AIAR scheme. Section 4 introduces multipointer approach and discusses our proposed FS, OS, PSS, FS-SR, OS-SR and PSS-SR schemes. Simulation design is presented in Section 5. In Section 6, we present the results from the OMNeT++ simulations and evaluate the performance of the multipointer scheduling algorithms. Finally, Section 7 concludes the paper.

2 System Model

In this paper, we consider a IEEE 802.16 mesh network with a BS and 10 SSs as shown in Fig. 1. The network is modeled as a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$. $\mathcal{V}$ is the set of SSs in the network and $\mathcal{E}$ is the set of links with $(i,j)\in\mathcal{E}$ iff SS_i and SS_j are within transmission range. All links in the network are bidirectional, thus, $(i,j)\in\mathcal{E}\Rightarrow(j,i)\in\mathcal{E}$.

In the *mesh mode*, traffic can flow among SSs without going through the BS. Two types of physical layers were defined in the standard, namely *WirelessMAN-OFDM* and *WirelessHUMAN*, operating in the licensed spectrum and unlicensed spectrum under 11GHz, respectively. They both use 256 point Fast Fourier Transform (FFT) in a TDMA/TDM structure for channel access. They support adaptive coding and modulation, and the link rates vary according to the channel conditions.

Both uplink and downlink scheduling can be achieved in a centralized fashion, while we focus on scheduling uplink packets. Scheduling of downlink packets can be done similarly using the same algorithm. We assume that each node is allowed to transmits at a maximum power. Uplink scheduling is carried out using mesh request (*MSH-REQ*) packets generated from each SS once every scheduling period (SP), which is defined as the period over which the generated scheduling map is deemed valid. Each scheduling is hence valid for a SP of K frames. Each frame consists of N time slots, among which *MSH-CTRL-LEN* time slots are reserved for centralized scheduling messages like *MSH-REQ*, *MSH-GRANT*, *MSH-NENT*, etc.

Each node on arrival interacts with the nodes within transmission range and chooses a sponsor node as the parent node, which helps register the new node at the BS. Once the node is registered, it can participate in scheduling by requesting bandwidth from the BS. This is done via the bandwidth

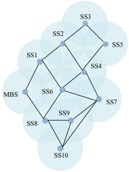


Fig. 1 Mesh network architecture with underlying graph structure

request/grant mechanism. Each node forwards its bandwidth request to its parent node. A parent node assimilates the requests of its children nodes in its own request and sends it to its parent node. This repeats recursively until the BS receives the bandwidth request. The BS executes the scheduling algorithm and broadcasts bandwidth grants to all nodes. This grant is broadcasted throughout the network by intermediate nodes. As illustrated in Fig. 2, due to buffering limitations, the grant/request received from a node in the previous frame can be forwarded only in the next frame. Hence, an entire *MSH-REQ* and *MSH-GRANT* cycles takes two scheduling periods. Hence, packets originating in scheduling period k, will request for resources in scheduling period $k+1$ and get granted resources in scheduling period $k+2$. Hence, resources to these packets are actually assigned in scheduling period $k+3$. This allows us to bound the delay to a maximum value if we provide an efficient routing and scheduling algorithm.

2.1 Spatial Reuse and Interference Model

The IEEE 802.16 standard allows us to implement SR. Two or more nodes can transmit concurrently if they do not interfere with one another. We follow the protocol model and define rules under which links are considered to be interfering. There are two types of interference that a link can experience: (i) primary interference and (ii) secondary interference. All the links in our

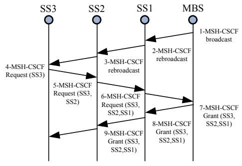


Fig. 2 Bandwidth Request/Grant Mechanism in Mesh Networks

network are bidirectional. A transmitting node cannot receive at the same time and vice-versa. This type of interference is called primary interference. Secondary interference occurs when an active link interferes with other links which do not share the same source/destination.

3 System Stability and Proposed Routing Scheme

Let us define the following notations in Table 1. Then, the queue length at node i can be calculated as

$$Q_i\left(k+1\right) = \left\{Q_i\left(k\right) - n_i r_i\left(k\right)\right\}^+ + X_i\left(k\right) + Y_i\left(k\right) \tag{1}$$

For the sake of system stability, we need to ensure the capacity of the network must be greater than the demand of the nodes; otherwise, the queue

Table 1 List of Notations

C	concurrent transmission set
H	connection matrix
k	frame number
λ_i	mean number of bits arriving at node i at the beginning of frame k
M	number of SSs in the mesh network
N	number of slots in a frame
n_i	number of slots assigned to node i
$Q_i(k)$	queue length at the beginning of frame k
$r_i\ (k)$	transmission rate of the medium during frame k
r_{ij}	the data rate of the link between SS_i and SS_j
$E[r_{ij}]$	the average data rate of the link (i, j)
σ_i	order of transmission of node i as given in the configuration file
SP	scheduling period
SQ	spillover queue
w	fixed scheduling weighting factor
$X_i(k)$	external arrival at node i during frame k
$Y_i(k)$	arrival from other nodes to node i during frame k

length may continue to increase and the system might become unstable. Thus,

$$n_i E\left[r_i\right] > E\left[X_i + Y_i\right] = \lambda_i + E\left[Y_i\right] \tag{2}$$

In (2), $E[r_i]$ is the output data transmission rate of node i, and the total data originating at other nodes and passing through node i is given by $E\left[Y_i\right] = \sum_{j=1}^{m_i} \lambda_{a_{i,j}}$, where $\{a_{i,1}, a_{i,2}, a_{i,3} \ldots a_{i,m_i}\}$ are the nodes whose data passes through node i. Hence, we get the number of slots required by node i,

$$n_i > \frac{\lambda_i + \sum_{j=1}^{m_i} \lambda_{a_{i,j}}}{E\left[r_i\right]} \tag{3}$$

To be within the stability range, the total number of slots allocated throughout the network should not exceed 'N'- the total number of available slots, i.e., $\sum_{i=1}^{M} n_i \leq N$. By rearranging the terms in (3), we can obtain,

$$\sum_{i=1}^{M} \left(\lambda_i \sum_{j=1}^{h_i} \left(E\left[r_{p_{i,j}}\right]\right)^{-1} \right) < N \tag{4}$$

where $\{p_{i,1}, p_{i,2}, p_{i,3} \ldots p_{i,h_i}\}$ are the intermediate nodes for which data from node i is routed, including node i. From (4), we get the upper bound on the arrival rate λ_i of each node. This is the maximum load on the network. Hence, to maximize the stability region, the value of $\sum_{j=1}^{h_i} \left(E\left[r_{p_{i,j}}\right]\right)^{-1}$ should be minimized. This translates to shortest path routing, which essentially resembles a path that has the maximum rate of transfer to the MBS.

3.1 Asymmetric Interference Aware Routing

Routing in WiMAX can be either fixed or adaptive. However, using adaptive routing in such a dynamic environment like WiMAX will lead to continuous path changes [24]. These changes would affect data delivery. Packets traveling from one node to another could experience out of order delivery and high jitter. Moreover, most data transfer (video and voice, etc) require that packets arrive in the same order in which they have been transmitted. Therefore, fixed routing is adopted for most deployments. Assuming that data from a given node follows a single path, we can obtain an optimum tree routing structure. To enable SR and optimize throughput, we propose a routing algorithm, namely asymmetric interference aware routing (AIAR).

As shown in Algorithm 1, consider a graph $\mathcal{G}\left(\mathcal{V},\mathcal{E}\right)$ where link $\mathcal{E}_i$ with source $s_{\mathcal{E}i}$ and destination $d_{\mathcal{E}i}$. A link $\mathcal{E}_1$ is said to be non-interfering with another link $\mathcal{E}_2$ if: (i) $d_{\mathcal{E}_2} \notin \mathit{neighbour}\left(s_{\mathcal{E}_1}\right)$, i.e., no receiver in the neighbourhood of the transmitter; (ii) $s_{\mathcal{E}_2} \notin \mathit{neighbour}\left(d_{\mathcal{E}_1}\right)$, i.e., no transmitter in the neighbourhood of the receiver; (iii) $d_{\mathcal{E}_2} \neq d_{\mathcal{E}_1}$, i.e., one node cannot receive from two transmitters.

We use a parameter *link interference parameter*, $\mathcal{L}(x,y)$ to denote the sum of data rates of all links that it interferes with. A node on entry calculates the $\mathcal{L}(x,y)$ of every link it is involved in. It then selects a suitable node as its parent node using *node blocking parameter* $\mathcal{B}(n)$

$$B\left(n\right) = \arg \min_{\forall n_i \in \text{neighbours}(n)} \left\{B\left(n_i\right) + L\left(n,n_i\right)\right\} \tag{5}$$

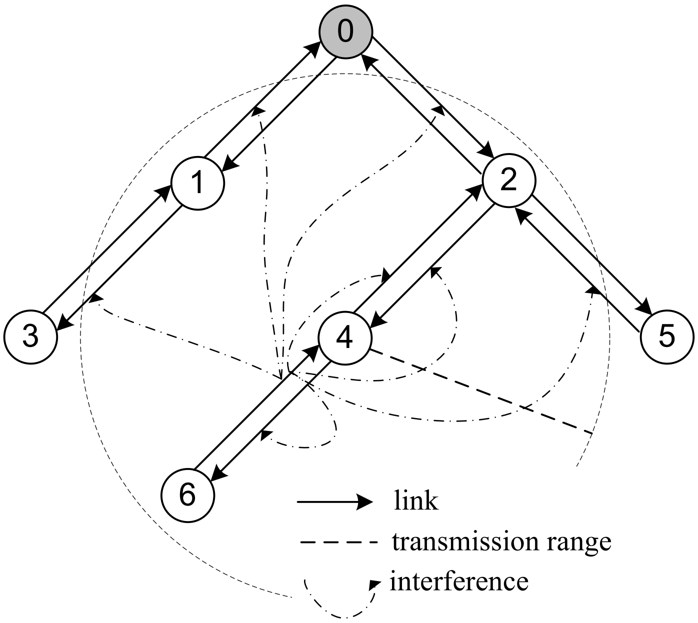
The parent node is hence the node that offers minimal interference with other links. This algorithm is dependent on the order in which the nodes enter the network. Let H denote the hop count of the farthest node from BS, if we execute the algorithm H times, we can obtain an optimal routing tree. Our following discussion on scheduling algorithms is based on an assumption that the minimum interference routing tree is pre-determined.

Algorithm 1 Asymmetric Interference Aware Routing

Input: Graph of the network including all edges and nodes - $\mathcal{G}(\mathcal{V},\mathcal{E})$

Output: Routing tree with parents of each node - $\mathcal{G}'(\mathcal{V},\mathcal{E}')$

```
Initialize: $\mathcal{V} \leftarrow \{1,2,3,...,M\}$, $\mathcal{V}' \leftarrow \{0\}$
For every edge in $\mathcal{E}$, calculate $\mathcal{L}(x,y)$ from the above given conditions
$\mathcal{B}(n)$=0 for $n$=0; otherwise infinity
while ($\mathcal{V} \neq \emptyset$) do
    Choose a node $i$ from $\mathcal{V}$ with least number of hops to the MBS
    for $j$ = each neighbour of $i$ in $\mathcal{V}$ do
        $\mathcal{B}(i)$ = minimum$\{\mathcal{B}(i), \mathcal{L}(i,j)+\mathcal{B}(j)\}$
        $parent(i) = j$, for which $\mathcal{B}(i)$ is minimum
    end for
    $\mathcal{V} \leftarrow \mathcal{V} - \{i\}$
    $\mathcal{V}' \leftarrow \mathcal{V}' + \{i\}$
    $\mathcal{E}' \leftarrow \mathcal{E}' + \{i, parent(i)\}$
end while
```


0
1
2
3
4
5
6
link
transmission range
interference

4 Enhanced Scheduling using Multi-pointer Approach

In this section, the new multipointer scheduling algorithm is presented, followed by modifications to the three scheduling algorithms in order to take into account SR. Finally, a multipointer scheduling example is presented.

4.1 Multi-pointer Algorithm for Packet Scheduling

In most algorithms that we have seen, the scheduling of resources is done from the first available slot towards the last available slot. This is acceptable for the *PMP mode* [25]. However, for the *mesh mode*, buffering constraints and the multihop distance to the destination render the normal iteration useless. Under these circumstances, we need to develop a different approach. Scheduling validity is defined as the number of frames over which a particular schedule grant is valid. For WiMAX deployments with centralized scheduling, the scheduling validity is set to the hop count of the farthest node from the BS. As shown in Algorithm 2, the number of pointers used for scheduling is equal to the scheduling validity. Consequently, we require one pointer for each frame in the scheduling period. When all the slots in that particular frame have been allocated, the pointer is merged with the pointer in the next frame.

Algorithm 2 Multi-pointer Algorithm Structure

```
Initialize: multipointer $P_i \leftarrow$ the first data slot in $i^{th}$ frame, $n_j^i \leftarrow$ the number of slots
    required by the $j^{th}$ node in the $i^{th}$ frame, where $j \leq M, i \leq SP$
for $i = 1 : SP$ do
    while $n_j^i > 0$ do
        $P_i \leftarrow j$
        $n_j^i \leftarrow n_j^i - 1$
        $P_i \leftarrow P_i + 1$
        if $p_i =$ end of frame then
            $P_i \leftarrow P_{i+1}$
        end if
    end while
end for
```

Let us look at how a multipointer algorithm works. Consider a IEEE 802.16 network with four nodes as given in Fig. 4. Since the maximum hop count of the farthest node to the MBS is 3, we use a scheduling period of 3 frames. As depicted in Fig. 5(a), for the sake of simplicity, let us assume that each frame has 20 time slots, where the pointers in each of these frames are denoted by P_1, P_2 and P_3, and initialized to the first slot in the 1^{st}, 2^{nd} and 3^{rd} frame, respectively. Assuming that each node has a request of eight slots, we can obtain the results are shown in Fig. 5 after running the multipointer algorithm. In Fig. 5, a slot with a circle mark inside indicates its allocation to a node; a slot with a diamond mark inside indicates during which data transfer is being

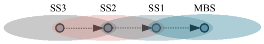


Fig. 4 IEEE 802.16 network with four nodes

done; a shaded slot without any mark indicates during which data is received for a node.

4.2 Fixed Scheduling with Spatial Reuse

FS follows the principle of allocating resources proportional to the arrival rate at a node. The arrival rate at the node comprises of external arrivals from other nodes and arrivals from users of the node itself. The number of slots required for node i without concurrency is given in (3). We calculate the number of slots for all nodes and then allocate slots to the nodes in order of their hop count from the BS, as specified by σ_i.

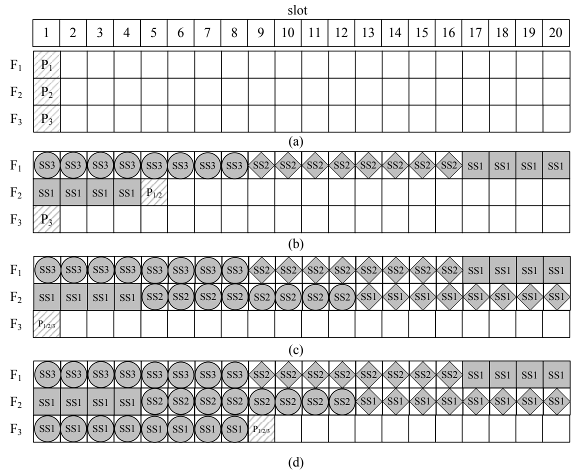


Fig. 5 Working of Multi-pointer Algorithm: (a) before scheduling (b) after first frame of scheduling (c) after second frame of scheduling (d) after third frame of scheduling

To incorporate SR in terms of concurrent transmissions, we define a child matrix $c_{i,j}$, to denote whether or not a node j is a child/sub-child of node i. This helps in calculating the total arrival rate at a given node. Then, we use $I_{i,j}$ to define the interference matrix, where if link i interferes with link j, then $I_{i,j}$ is equal to 1. We should note that the interference matrix is a symmetric matrix. Hence, $I_{i,j}$=$I_{j,i}$$\leq 1$.

As we consider only uplink scheduling and a routing tree, each node has only one uplink, and this link from node i to its parent node can be denoted as $L_{i,parent(i)}$. Since no other link from i can serve as an uplink, we can simplify the notation as L_i. The order in which each node transmits is given in *MSH-CSCF*. This is denoted by σ_i for node i. The scheduling output gives a set of nodes that can transmit concurrently during a slot.

FS-SR follows a staggered approach. The arrival rate to each node is calculated as a sum of arrival rate to itself and its children nodes. The number of slots hence required in total at each node is the sum arrival rate divided by the rate of data transfer per slot. In particular, if the total number of slots cannot be serviced by the scheduling period, the number of slots granted to each node is multiplied by a scheduling weight, w, which is defined as the ratio of the number of slots in the scheduling period to the total number of slots required.

Algorithm 3 Fixed Scheduling with Spatial Reuse

Input: arrival rates, children matrix, order of each node **Output:** scheduling grant for all nodes

```
 1: calculate $n_i$ and $w$
 2: if $w < 1$ then
 3:     $n_i = n_i \times w, i = 1 : M$
 4: end if
 5: Initialize: multipointers for all frames $P_i, i = 1 : SP$
 6: set $Q_i = \lambda_i \times SP$
 7: for i=1:SP do
 8:     for j=1:N do
 9:         find node $n_s$=arg $\min_{i=1...M} \{\sigma_i\}$, $Q_{n_s} > 0$ and $I_{n_s n_t} = 0 \; \forall \; n_t \in C$
10:         $C \leftarrow \{C, n_s\}$
11:         continue steps 8-10 until all nodes have been considered
12:         if $C = \emptyset$ then
13:             break
14:         end if
15:         allocate slot $P_i$ to C
16:         $P_i \leftarrow P_i + 1$
17:         $Q_k = Q_k - r_k$ and $SQ_{parent(k)} = SQ_{parent(k)} + r_k, \forall k \in C$
18:         if $P_i$=end of frame then
19:             $P_i = P_{i+1}$
20:         end if
21:     end for
22:     $Q_k = Q_k + SQ_k, k = 1 : M$
23: end for
```

4.3 Ordered Scheduling with Spatial Reuse

In OS, the scheduling weight for node is defined as

$$w_i = \sum\nolimits_{j=1}^{h_i} \left(E \left[r_{p_{i,j}} \right] \right)^{-1} \tag{6}$$

Given the parameter w_i, interference matrix I, and individual queue lengths Q_i, OS orders the nodes based on their w_i values and assigns each slot to the first node that can fully utilize that slot. Preference is given to a node with a lower w_i value. Furthermore, we propose OS-SR to implement SR (See Algorithm 4). The concurrent set 'C' allocated to each slot contains the set of all nodes that can transmit during that slot.

For each slot, OS-SR first selects the node that has the minimum w_i value and queue length longer enough to use the entire slot. Then, it selects other nodes that can fully utilize a slot and can transmit concurrently with the first selected node in order of their respective parameter values. The corresponding uplinks of these nodes are added into the concurrent set 'C'. This method continues until all nodes are scheduled.

Algorithm 4 Ordered Scheduling with Spatial Reuse

```
Input: arrival rates, children matrix, order of each node
Output: scheduling grant for all nodes
 1: calculate $w_i$
 2: Initialize: multipointers $P_i, i = 1 : SP$
 3: set queue length of all nodes to $Q_i$ from $MSH - REQ$ packet received
 4: for i=1:SP do
 5:     for j=1:N do
 6:         find node $n_s$=$\arg\min_{i=1...M}\{w_i\}$, $Q_{n_s} > r_{n_s}$ and $I_{n_s n_t} = 0\ \forall\ n_t \in C$
 7:         $C \leftarrow \{C, n_s\}$
 8:         continue steps 6-8 until all nodes have been considered
 9:         if $C = \emptyset$ then
10:             find node $n_s$=$\arg\min_{i=1...M}\{w_i\}$, $Q_{n_s} > 0$ and $I_{n_s n_t} = 0\ \forall\ n_t \in C$
11:             $C \leftarrow \{C, n_s\}$
12:             continue steps 9-11 until all nodes have been considered
13:         end if
14:         if $C = \emptyset$ then
15:             break
16:         end if
17:         allocate slot $P_i$ to C
18:         $P_i \leftarrow P_i + 1$
19:         $Q_k = Q_k - min\{Q_k, r_k\}$ and $SQ_{parent(k)} = SQ_{parent(k)} + min\{Q_k, r_k\}, \forall k \in C$
20:         if $P_i$=end of frame then
21:             $P_i = P_{i+1}$
22:         end if
23:     end for
24:     $Q_k = Q_k + SQ_k, k = 1 : M$
25: end for
```

4.4 Per Slot Scheduling with Spatial Reuse

PSS works on a dynamic programming principle (See Algorithm 5) to schedule one slot at a time to a node which can transmit the maximum number of bits in that given slot. With SR, a slot is scheduled to nodes in the concurrent set 'C', which can transmit simultaneously without experiencing severe interference.

One advantage of PSS is that the average resources required at nodes which have a higher egress rate is low. However, the corresponding disadvantage is that under high loads, the nodes with a lower transmission rate are allocated resources towards the end of the scheduling frame. Hence, these nodes have longer queue lengths. The data drop probability at these nodes is also high due to the prioritized scheduling of nodes with higher egress rates.

Algorithm 5 Per Slot Scheduling (Concurrency) with Spillover

```
Input: arrival rates, children matrix, order of each node
Output: scheduling scheme for all nodes
 1: Initialize: multipointers $P_i, i = 1 : SP$
 2: set queue length of all nodes to $Q_i$ from $MSH - REQ$ packet received
 3: for i=1:SP do
 4:     for j=1:N do
 5:         find node $n_s$=arg $\max_{\forall i \notin C} \{\min(Q_i, r_i)\}$ and $I_{n_s n_t} = 0 \; \forall \; n_t \in C$
 6:         $C \leftarrow \{C, n_s\}$
 7:         continue steps 6-8 until all nodes have been considered
 8:         if $C = \emptyset$ then
 9:             break
10:         end if
11:         allocate slot $P_i$ to C
12:         $P_i \leftarrow P_i + 1$
13:         $Q_k = Q_k - min\{Q_k, r_k\}$ and $SQ_{parent(k)} = SQ_{parent(k)} + min\{Q_k, r_k\}, \forall k \in C$
14:         if $P_i$=end of frame then
15:             $P_i = P_{i+1}$
16:         end if
17:     end for
18:     $Q_k = Q_k + SQ_k, k = 1 : M$
19: end for
```

5 Simulation Design

Our simulation in OMNeT++ used the IEEE 802.16 mesh network as shown in Fig. 1, designed using a Network Descriptor (NED). If a node was in transmission range of another node, then it was connected to that node via a link. All links were bidirectional and the link parameters were governed by their burst profiles as shown in Table 2. The burst profile governed the modulation, coding rate of that link. Hence, a different burst profile translated to a different data rate. The link rates (in Mbps) between any two nodes were predetermined, as specified in a NED, and they were aggregated in a connection matrix, H with element (i, j) corresponding to burst profile of link between node i and node j. $H_{11\times11} = \begin{pmatrix} 0&6&0&0&0&0&0&0&4&0&0 \\ 6&0&3&0&0&0&2&0&0&0&0 \\ 0&3&0&6&4&0&0&0&0&0&0 \\ 0&0&6&0&0&3&0&0&0&0&0 \\ 0&0&4&0&0&4&0&6&0&0&0 \\ 0&0&0&3&4&0&0&0&0&0&0 \\ 0&2&0&0&0&0&0&3&2&0&0 \\ 0&0&0&0&6&0&3&0&0&2&2 \\ 4&0&0&0&0&0&2&0&0&3&3 \\ 0&0&0&0&0&0&0&2&3&0&0 \\ 0&0&0&0&0&0&0&2&3&0&0 \end{pmatrix}$

The maximum load on the network with respect to arrival rate at each SS was calculated using the method in Section 3 and the load is 1.74Mbps, which was corresponding to a normalized load of 1.0. The normalized load on the network was varied from 0 to 1.0 with a step size of 0.1. Poisson packet arrival

Table 2 Physical Parameters of Burst Profile

Burst Profile No.	Modulation	Coding Rate	Uncoded bytes per OFDM Symbol	Uncoded bytes per slot
1	QPSK	1/2	24	96
2	QPSK	3/4	36	144
3	16QAM	1/2	48	192
4	16QAM	3/4	72	288
5	64QAM	2/3	96	384
6	64QAM	3/4	108	432

was considered and the packet size was varied exponentially with respect to the mean. Traffic generation occurred in the *trafgen* submodule of the substation. The mean packet size was fixed at 768 bits. If all the nodes have the same average arrival rate λ_i, the average size of the packet is fixed at L and the packet inter-arrival time is calculated as

$$E\left[\tau\right] = \frac{L}{\lambda} \tag{7}$$

All traffic is directed towards the BS. The traffic generator can also be adapted to include priority, tardiness and drop probability of the packets, which has been left out for our future study. Upon receiving data packets, either self-generated or from other nodes, SSs put them into a reserved buffer, and then moved them into the transmitter buffer at the beginning of each frame. This perfectly simulated the buffer constraints of the IEEE 802.16 network. The queuing rule was *first-in first-out (FIFO)*. Other queuing methods like *earliest deadline first* and *fair scheduling* could also be employed.

Each packet was time-stamped on of creation at the traffic-generator, and given a time-to-live (TTL) value that was also varied exponentially with respect to the mean. We tested our algorithms for four mean TTL values: 60ms, 120ms, 240ms and 480ms. Packets which experienced a delay longer than their TTL values were dropped by intermediate nodes or the BS.

The simulation was run for $5000s$ or 5×10^6 frames. We can safely assume that simulations could accurately record the real-time performance of the IEEE 802.16 mesh network. Each frame was assigned a duration of $10ms$ and the scheduling period was set to 4 frames, as the maximum hop-count amongst all leaf nodes is 4 in Fig. 1.

During initialization, we first execute AIAR to obtain the optimal routing tree and then applied a scheduling algorithm. We simulated the request and grant process and data transmission with buffer constraints in the IEEE 802.16 mesh network, and analyzed the performance of each of the six scheduling algorithms discussed above, namely, FS, OS, PSS, FS-SR, OS-SR and PSS-SR.

6 Results Analysis

To analyze the performance of a IEEE 802.16 network, we mainly look into two main aspects: (i) *mean delay* – end-to-end time taken by a packet to reach its destination; and (ii) *throughput* – the number of packets which have been successfully delivered.

6.1 Mean Delay

The delay of a packet is defined as the time it takes from the time when it is created by the traffic generator to the time when it reaches its destination.

As shown in Fig. 6, at low traffic load with low TTL values (60ms∼120ms), FS outperformed other algorithms, which makes FS a suitable scheduling algorrithm for low-data-rate and time-critical applications like VoIP and Telnet. Next, scheduling priority under OS was to the node with the fastest path towards the BS; while under PSS was to the fastest link. Due to this *selective* nature of these algorithms, nodes with slower links/paths were allocated resource at the end of a scheduling period, leading to a higher mean delay. In addition, every scheme outperformed its corresponding counterpart when implemented with SR. This reduction in mean delay could be attributed to less mean medium access time due to the concurrent transmissions.

At low traffic load with high TTL value (240ms∼480ms), there was no apparent difference in the performance of the six scheduling schemes. Scheduling algorithms employing SR performed just marginally better because they increased the effective capacity of the network, thereby enabling more packets to reach the destination faster. Furthermore, we find that *proactive* scheduling algorithms like FS significantly outperformed the *reactive* scheduling algorithms like OS and PSS. If we define ‘reaction time’ as the number of scheduling periods taken by the mesh network to respond to a given mesh-request. As shown in Fig. 2, for PSS and OS that are based on the queue length of nodes, the reaction time is 2 scheduling periods. However, for FS, this is reduced to zero, as FS is based on arrival rate at each node. Therefore, FS offered the smallest delay amongst the six.

At high traffic load with low TTL values (60ms∼120ms), OS and PSS outperformed FS. This is because the fact that FS would allocate more resources to weaker links, giving rise to a longer waiting time for data with stronger links and an overall increase in average delay. In particular, the ‘*selective*’ nature of OS and PSS did not favor scheduling to moderate and weak paths/links. Some packets were lost due to excessive delay and many packets were delayed as the algorithm allocated resources first to stronger links and paths and then to weaker links. The overall delay hence increases at high traffic load for these algorithms. Packets with slower links (See Fig. 6) were dropped and hence the average delay of packet delivery was comparable between OS and PSS. Next, with SR, OS-SR and PSS-SR outperformed OS and PSS, respectively. Furthermore, FS-SR provided the best performance. This was because of si-

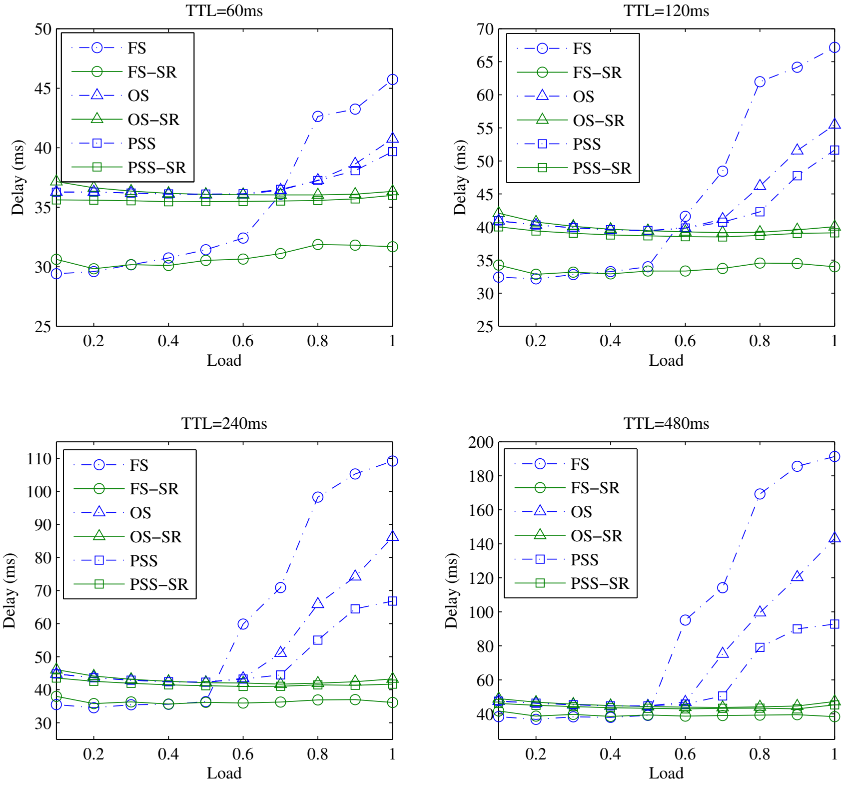


Fig. 6 Mean Delay Analysis

multaneous allocations to strong and weak links at the same time. Since FS was not selective in nature, data were allocated slots uniformly and hence we had a lower average delay.

At high traffic load with high TTL value (240ms∼480ms), FS-SR offered the best average delay under all loads and all TTL conditions, and it could be used for almost all applications. Furthermore, PSS outperformed OS, while OS-SR and PSS-SR performed similarly.

6.2 Throughput

Throughput of a system is a measure of the data generated by the node compared to the amount of data received by the MBS. In our case, data which were not delivered with a TTL were dropped and hence an unfair scheduling algorithm may lead to a low throughput.

As shown in Fig. 7, at low traffic load, FS outperformed FS-SR, implying that SR does not improve throughput. Furthermore, the performance of other four schemes was about the same. Interestingly, the performance of six schemes were converging when the TTL values were increased at low traffic load.

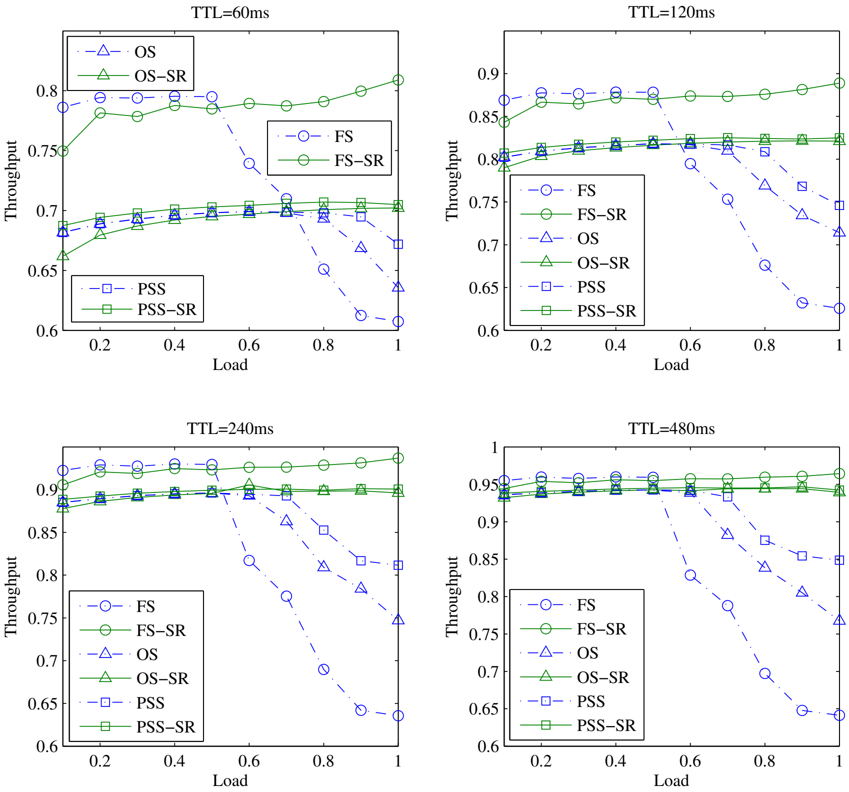


Fig. 7 Throughput Analysis

At high traffic load, without SR, PSS provided the highest throughput as it strived to transmit the maximum number of bits in every slot. The path-selective OS provided a lower throughput than PSS; FS resulted in the lowest throughput. With SR, FS-SR offered the maximum throughput. The proactive scheduling saved '*reaction time*' and hence allowed faster delivery of nodes. Furthermore, OS-SR and PSS-SR offered comparable throughput. When the TTL values were increased, the performance of FS-SR, OS-SR and PSS-SR converged.

7 Conclusion

A concurrent routing scheme with spatial reuse (SR), asymmetric interference aware routing (AIAR) was proposed. Then, a multi-pointer approach of implementing scheduling algorithms for IEEE 802.16 mesh networks was proposed. Six scheduling algorithms, *fixed scheduling (FS)*, *ordered scheduling (OS)* and *per-slot scheduling (PSS)*, *FS-SR*, *OS-SR* and *PSS-SR* were proposed. Through computer simulations, we found that without SR, *PSS* provides highest throughput while *FS* gives lowest packet delay. While incorporating SR, the *FS-SR* scheme offers the best overall performance.

In our proposed multi-pointer approach of implementing scheduling algorithm, the number of pointers used for scheduling is equal to the scheduling validity. As such, it will work well if the scheduling period is not excessively long. However, for large IEEE 802.16 mesh networks with fairly fixed network topology, the proposed multi-pointer approach might have a scalability issue. This has been left for our future work in this topic. Furthermore, we also plan to test the proposed algorithms in a WiMAX testbed in order to investigate their real-time performance.

References

1. Ieee standard for local and metropolitan area networks part 16: Air interface for broadband wireless access systems amendment 3: Advanced air interface. *IEEE Std 802.16m-2011(Amendment to IEEE Std 802.16-2009)*, pages 1–1112, 2011.
2. Gorbenko, N. and Jean-Pierre, E. and Almuhtadi, W. and Srinivasan, A. Comparison of simulated and real network traffic results for multimedia streaming over WiMAX networks with QoS scheduling. In *Proc. of IEEE CCECE'15*, pages 1324–1328.
3. Ieee standard for local and metropolitan area networks - part 16: Air interface for fixed and mobile broadband wireless access systems - amendment 3: Management plane procedures and services (amendment to ieee 802.16-2004). *IEEE Approved Draft Std P802.16g/D9, Apr 07*, 2007.
4. Wen-Hwa Liao, S. P. Kedia, and A. K. Dubey. A centralized scheduling algorithm for wimax mesh network. In *Proc. of IEEE NOMS'10*, pages 858–861.
5. Yanbin Huang, Shoubao Yang, Da Teng, Weiqing He, and Lihua Xie. A fair rate-balance link scheduling in wimax mesh network. In *Proc. of IEEE WCNIS'10*, pages 355–359.
6. Kapila, Deepak and Gill, S. S. A hybrid approach for scheduling in WIMAX for reduction of process waiting time. *International Journal of Innovation and Applied Studies*, 9(4):1620–1625, 2014.
7. Kumar, Niharika and Algur, Siddu. Novel bandwidth allocation and packet scheduling strategy for WiMAX. *International Journal of Wireless and Mobile Computing*, 7(6):517–527, 2014.
8. Mnif, K. and Khdhir, R. and Kamoun, L. Evaluation and comparaison of scheduling algorithms in Wimax networks. In *Proc. of ICMCS'14*, pages 884–889.
9. Rehman, A. and Shakir, Mustafa. Comparative Analysis of Scheduling Algorithms in IEEE 802.16 WiMAX Radio. *International Journal of Computing and Digital Systems*, 3(2):161–172, 2014.
10. Randhini, R. and Devarajan, N. Comparison for WiMAX scheduling algorthims and proposal quality of service improvement in WiMAX networks. *American Journal of Applied Science*, 11(1):8–16, 2014.
11. Yang, Feng-Ming and Chen, Wei-Mei and Wu, Jean-Lien C. A dynamic strategy for packet scheduling and bandwidth allocation based on channel quality in IEEE 802.16e OFDMA system. *Journal of Network and Computer Applications*, 39:52–60, 2014.
12. Zubairi, Junaid Ahmed and Erdogan, Erdem and Reich, Shaun. Experiments in fair scheduling in 4G WiMAX and LTE. In *Proc. of HPCS'15*, pages 277–282.
13. Akashdeep and Kahlon, Karanjeet S. and Kumar, Harish. Survey of scheduling algorithms in IEEE 802.16 PMP networks. *Egyptian Informatics Journal*, 15(1):25–36, 2014.
14. Yadav, A. L. and Vyavahare, P. D. and Bansod, P. P. Review of WiMAX Scheduling Algorithms and Their Classification. *Journal of The Institution of Engineers (India): Series B*, 96(2):197–208, 2015.
15. Hung-Yu Wei, S. Ganguly, R. Izmailov, and Z. J. Haas. Interference-aware ieee 802.16 wimax mesh networks. In *Proc. of IEEE VTC'05-Spring*, volume 5, pages 3102–3106 Vol. 5.

16. Jian Tao, Fuqiang Liu, Zhihui Zeng, and Zhangxi Lin. Throughput enhancement in wimax mesh networks using concurrent transmission. In *Proc. of WCNM'05*, volume 2, pages 871–874.
17. Fei Xie, K. A. Hua, and Ning Jiang. Achieving true video-on-demand service in multihop wimax mesh networks. In *Proc. of IEEE LCN'07*, pages 287–294.
18. Guo, Ming-Huang and Deng, Der-Jiunn and Liaw, Horng-Twu and Park, Jong Hyuk. An efficient route scheduling mechanism for WiMAX network. *The Knowledge Engineering Review*, 29(04):452–462, 2014.
19. H. Shetiya and V. Sharma. Algorithms for routing and centralized scheduling in ieee 802.16 mesh networks. In *Proc. of IEEE WCNC'06*, volume 1, pages 147–152.
20. Fanchun Jin, A. Arora, Hwang Jinho, and H. A. Choi. Routing and packet scheduling in wimax mesh networks. In *Proc. of BROADNETS'07*, pages 574–582.
21. Shou-Chih Lo and Lyu-Chen Ou. Efficient algorithms for routing and centralized scheduling for ieee 802.16 mesh networks. In *Proc. of SCALCOM-EMBEDDEDCOM'09*, pages 212–217.
22. S. Nahle and N. Malouch. Joint routing and scheduling for maximizing fair throughput in wimax mesh network. In *Proc. of IEEE PIMRC'08*, pages 1–5.
23. J. El-Najjar, C. Assi, and B. Jaumard. Joint routing and scheduling in wimax-based mesh networks: A column generation approach. In *Proc. of IEEE WoWMoM'09*, pages 1–10.
24. So-In Chakchai, R. Jain, and A. K. Tamimi. Scheduling in ieee 802.16e mobile wimax networks: Key issues and a survey. *IEEE Journal on Selected Areas in Communications*, 27(2):156–171, 2009.
25. Najah Abu Ali, Pratik Dhrona, and Hossam Hassanein. A performance study of uplink scheduling algorithms in point-to-multipoint wimax networks. *Computer Communications*, 32(3):511–521, 2009.

Wireless Personal Communications manuscript No.
(will be inserted by the editor)

Joint Concurrent Routing and Multipointer Packet Scheduling in WiMAX Mesh Networks

Xue Jun Li · Maode Ma

Received: 26 February 2015 / Accepted: date

Abstract Worldwide Interoperability for Microwave Access (WiMAX) is a standardization effort carried out by the IEEE to provide last-mile broadband access to end users. The WiMAX standard supports two medium access control (MAC) modes - a mandatory point to multipoint (PMP) mode and an optional mesh mode. In this paper, we propose an asymmetric interference aware routing algorithm and a new multipointer approach in implementing scheduling algorithms for WiMAX mesh networks. We modify three different centralized scheduling algorithms, fixed scheduling, ordered scheduling and per-slot scheduling using multipointer approach to allow for spatial reuse (SR) in WiMAX mesh networks. Simulation results reveal that fixed scheduling with SR provides the best performance.

Keywords WiMAX, IEEE 802.16, packet scheduling, routing, mesh networks, wireless networking, throughput

1 Introduction

The IEEE 802.16 Standard [1], better known as Wireless Interoperability for Microwave Access (WiMAX) was designed to provide last-mile wireless broadband access to users in remote areas. Its performance is comparable to that of a cable network or digital subscriber line (DSL). A WiMAX network is cheap

---

Xue Jun Li (✉)
Private Bag 92006, Auckland 1142, New Zealand
Tel.: +64-9-921-9999
Fax: +64-9-921-9973
E-mail: xuejun.li@aut.ac.nz

Maode Ma
50 Nanyang Avenue, Singapore, 639798
Tel.: +65-67904385
Fax: +65-67920415
E-mail: emdma@ntu.edu.sg

to deploy and easy to maintain. With the advent of other wireless technologies, WiMAX is considered to be a promising and easy alternative to wirelines infrastructure.

A WiMAX network consists of a base station (BS) with backhaul access to the network and many substations (SSs) with advanced multiple-input multiple-output (MIMO) transceivers. The BS and SSs maintain connectivity with all nodes which are within transmission range. The IEEE 802.16 [9] network operates in two modes - the mandatory *point-to-multipoint (PMP) mode* and the optional *mesh mode*. In the *PMP mode*, SSs interact only with the BS, and all traffic exchange occurs only between a SS and the BS. In the *mesh mode*, traffic is allowed to travel via SSs to their final destination. Compared to the *PMP mode*, the *mesh mode* exhibits better scalability, enhanced coverage, higher throughput and stronger resilience to node failures.

Inherently, the IEEE 802.16 Standard adopts time division multiple access (TDMA), under which each frame is divided into a number of transmission opportunities that are allocated to SSs using a scheduling algorithm. Scheduling can be centrally carried out at the BS or distributively completed by individual SS using the three-way handshake. Since the IEEE 802.16 does not specify any particular scheduling algorithm, scheduling in WiMAX mesh networks has attracted considerable amount of attention from both academia and industrial communities in recent years. Besides scheduling, routing is another important issue in WiMAX mesh networks.

Various literatures proposed different routing algorithms for WiMAX mesh networks. Wei et al. proposed an interference-aware routing scheme and a centralized mesh scheduling scheme in [13]. They mentioned that interference-aware design resulted in better spatial reuse (SR). In [14], Tao et al. proposed to use the protocol interference model to enhance throughput with concurrent transmission. However, none of them explored the actual scheduling algorithms. Xie et al. investigated video-on-demand streaming over WiMAX and proposed a multicast routing technique [11], where scheduling was achieved with admission control and SR was not included.

It becomes even challenging when one considers joint routing and scheduling in WiMAX mesh networks. Shetiya and Sharma studied joint routing and centralized scheduling and proposed several simpler suboptimal scheduling algorithms [12]. Each algorithm had a specific node selection mechanism that was used in a vanilla BGreedy algorithm. They did not consider buffer constraints of WiMAX mesh networks. Due to certain hardware buffer queueing and processing, data received in a frame ‘n’ cannot be made available for transmission until the next frame ‘$n + 1$’. Furthermore, they did not study the performance of their proposed algorithms in terms of end-to-end delay. Jin et al. showed the NP-completeness of the problem of joint packet scheduling and routing in general topology, and proposed routing/scheduling algorithms for mesh networks based on their study of a linear chain network [10]. Later, Lo and Ou studied the application of BGreedy algorithm [10] in a tree-topology [4]. However, SR was not considered. Liao et al. proposed a clique partitioning approach for centralized scheduling in WiMAX mesh networks, which was to optimize the schedule length, maximize number of concurrent transmissions and minimize the buffer size required at stations [2]. However, they did not study routing schemes. Huang et al. studied fair rate-balance in order to ensure link fairness and network stability in [3]. However, they did not consider SR and routing. Nahle and Malouch proposed a joint routing and scheduling algorithm to maximize network throughput [8]. However, they did not discuss the performance of their algorithm in terms of end-to-end delay. Study of schedule length was discussed in [5], where the authors proposed maximum spatial reuse (MSR) algorithm to maximize the SR by minimizing the schedule length and allowing as many concurrent transmission as possible.

From aforementioned works, we noticed that optimal scheduling schemes were dependent on the instantaneous queue length [10] [8] [5]. However, for practical implementations of WiMAX mesh networks, parameters like queue length and network load are not readily available at the BS, which motivates us to take this into account when designing scheduling algorithms. Moreover, the optimal scheduling algorithms were based on the assumption that all links have equal data rates. However, effective link rates are asymmetric due to varying medium access control (MAC) layer encoding/decoding.

The main contributions of this paper are threefold: (i) We propose a new routing algorithm, asymmetric interference aware routing (AIAR), which is based on effective link rates and use the interference model to obtain optimum throughput while considering SR. (ii) We are particularly enlightened by the three suboptimal algorithms originally proposed in [12], namely fixed allocation scheme, ordering scheme and per slot maximum transmission scheme. Our proposed scheduling algorithms are based on similar philosophy, but they are essentially different from those three proposed in [12]. To emphasize the difference, we deliberately rename them as fixed scheduling (FS), ordered scheduling (OS) and per-slot scheduling (PSS). We further modify FS, OS and PSS to implement SR and obtain FS with SR (FS-SR), OS with SR (OS-SR) and PSS with SR (PSS-SR), respectively. (iii) By taking buffer size and hardware delay into account, we propose a new multipointer approach to implement scheduling algorithms so as to minimize scheduling time, to reduce end-to-end delay and to improve system throughput. Simulation results demonstrate the feasibility of these algorithms.

The rest of this paper is organized as follows. Section 2 presents the system model with underlying assumptions. In Section 3, we analyze the interference model and propose AIAR scheme. Section 4 introduces multipointer approach and discusses our proposed FS, OS, PSS, FS-SR, OS-SR and PSS-SR schemes. Simulation design is presented in Section 5. In Section 6, we present the results from the OMNeT++ simulations and evaluate the performance of the multipointer scheduling algorithms. Finally, Section 7 concludes the paper.

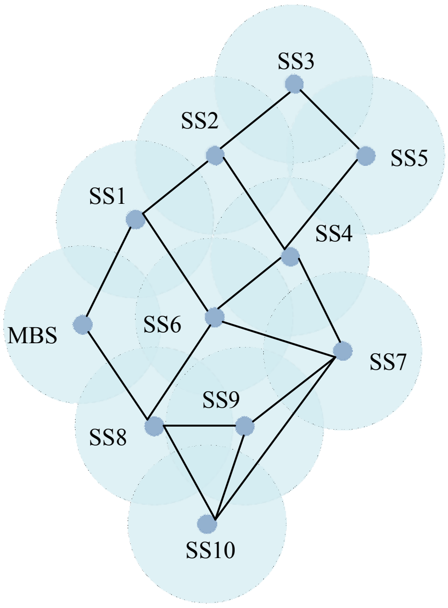


Fig. 1 Mesh network architecture with underlying graph structure

2 System Model

In this paper, we consider a WiMAX mesh network with a BS and 10 SSs as shown in Fig. 1. The network is modeled as a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$. $\mathcal{V}$ is the set of SSs in the network and $\mathcal{E}$ is the set of links with $(i,j)\in\mathcal{E}$ iff SS$_i$ and SS$_j$ are within transmission range. All links in the network are bidirectional, thus, $(i,j)\in\mathcal{E}\Rightarrow(j,i)\in\mathcal{E}$.

In the *mesh mode*, traffic can flow among SSs without going through the BS. Two types of physical layers were defined in the standard, namely *WirelessMAN-OFDM* and *WirelessHUMAN*, operating in the licensed spectrum and unlicensed spectrum under 11GHz, respectively. They both use 256 point Fast Fourier Transform (FFT) in a TDMA/TDM structure for channel access. They support adaptive coding and modulation, and the link rates vary according to the channel conditions.

Both uplink and downlink scheduling can be achieved in a centralized fashion, while we focus on scheduling uplink packets. Scheduling of downlink packets can be done similarly using the same algorithm. We assume that each node is allowed to transmits at a maximum power. Uplink scheduling is carried out using mesh request (*MSH-REQ*) packets generated from each SS once every scheduling period (SP), which is defined as the period over which the generated scheduling map is deemed valid. Each scheduling is hence valid for a SP of K frames. Each frame consists of N time slots, among which *MSH-CTRL-LEN* time slots are reserved for centralized scheduling messages like *MSH-REQ*, *MSH-GRANT*, *MSH-NENT*, etc.

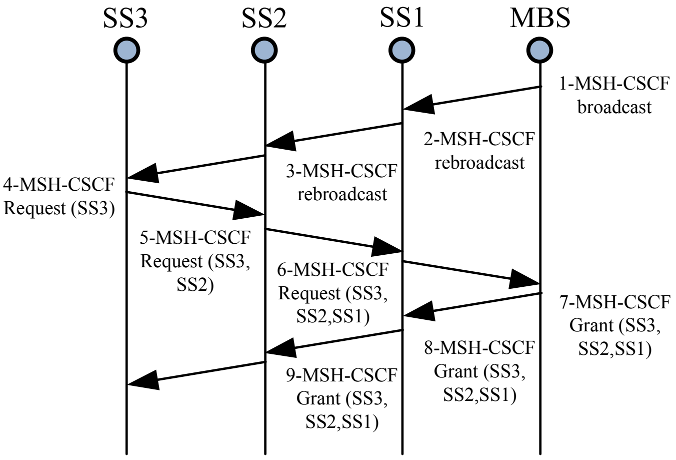


Fig. 2 Bandwidth Request/Grant Mechanism in Mesh Networks

Each node on arrival interacts with the nodes within transmission range and chooses a sponsor node as the parent node, which helps register the new node at the BS. Once the node is registered, it can participate in scheduling by requesting bandwidth from the BS. This is done via the bandwidth request/grant mechanism. Each node forwards its bandwidth request to its parent node. A parent node assimilates the requests of its children nodes in its own request and sends it to its parent node. This repeats recursively until the BS receives the bandwidth request. The BS executes the scheduling algorithm and broadcasts bandwidth grants to all nodes. This grant is broadcasted throughout the network by intermediate nodes. As illustrated in Fig. 2, due to buffering limitations, the grant/request received from a node in the previous frame can be forwarded only in the next frame. Hence, an entire *MSH-REQ* and *MSH-GRANT* cycles takes two scheduling periods. Hence, packets originating in scheduling period k, will request for resources in scheduling period $k + 1$ and get granted resources in scheduling period $k + 2$. Hence, resources to these packets are actually assigned in scheduling period $k + 3$. This allows us to bound the delay to a maximum value if we provide an efficient routing and scheduling algorithm.

2.1 Spatial Reuse and Interference Model

The WiMAX standard allows us to implement SR. Two or more nodes can transmit concurrently if they do not interfere with one another. We follow the protocol model and define rules under which links are considered to be interfering. There are two types of interference that a link can experience: (i) primary interference and (ii) secondary interference. All the links in our network are bidirectional. A transmitting node cannot receive at the same time and vice-versa. This type of interference is called primary interference.

Table 1 List of Notations

C	concurrent transmission set
H	connection matrix
k	frame number
λ_i	mean number of bits arriving at node i at the beginning of frame k
M	number of SSs in the mesh network
N	number of slots in a frame
n_i	number of slots assigned to node i
$Q_i(k)$	queue length at the beginning of frame k
$r_i\ (k)$	transmission rate of the medium during frame k
r_{ij}	the data rate of the link between SS_i and SS_j
$E[r_{ij}]$	the average data rate of the link (i, j)
σ_i	order of transmission of node i as given in the configuration file
SP	scheduling period
SQ	spillover queue
w	fixed scheduling weighting factor
$X_i(k)$	external arrival at node i during frame k
$Y_i(k)$	arrival from other nodes to node i during frame k

Secondary interference occurs when an active link interferes with other links which do not share the same source/destination.

3 System Stability and Proposed Routing Scheme

Let us define the following notations in Table 1. Then, the queue length at node i can be calculated as

$$Q_i\left(k+1\right) = \left\{Q_i\left(k\right) - n_i r_i\left(k\right)\right\}^+ + X_i\left(k\right) + Y_i\left(k\right) \tag{1}$$

For the sake of system stability, we need to ensure the capacity of the network must be greater than the demand of the nodes; otherwise, the queue length may continue to increase and the system might become unstable. Thus,

$$n_i E\left[r_i\right] > E\left[X_i + Y_i\right] = \lambda_i + E\left[Y_i\right] \tag{2}$$

In (2), $E[r_i]$ is the output data transmission rate of node i, and the total data originating at other nodes and passing through node i is given by $E\left[Y_i\right] = \sum_{j=1}^{m_i} \lambda_{a_{i,j}}$, where $\{a_{i,1}, a_{i,2}, a_{i,3} \ldots a_{i,m_i}\}$ are the nodes whose data passes through node i. Hence, we get the number of slots required by node i,

$$n_i > \frac{\lambda_i + \sum_{j=1}^{m_i} \lambda_{a_{i,j}}}{E\left[r_i\right]} \tag{3}$$

To be within the stability range, the total number of slots allocated throughout the network should not exceed 'N'- the total number of available slots, i.e., $\sum_{i=1}^{M} n_i \leq N$. By rearranging the terms in (3), we can obtain,

$$\sum_{i=1}^{M} \left(\lambda_i \sum_{j=1}^{h_i} \left(E\left[r_{p_{i,j}}\right]\right)^{-1} \right) < N \tag{4}$$

where $\{p_{i,1}, p_{i,2}, p_{i,3} \ldots p_{i,h_i}\}$ are the intermediate nodes for which data from node i is routed, including node i. From (4), we get the upper bound on the arrival rate λ_i of each node. This is the maximum load on the network. Hence, to maximize the stability region, the value of $\sum_{j=1}^{h_i} \left(E\left[r_{p_{i,j}}\right]\right)^{-1}$ should be minimized. This translates to shortest path routing, which essentially resembles a path that has the maximum rate of transfer to the MBS.

3.1 Asymmetric Interference Aware Routing

Routing in WiMAX can be either fixed or adaptive. However, using adaptive routing in such a dynamic environment like WiMAX will lead to continuous path changes [6]. These changes would affect data delivery. Packets traveling from one node to another could experience out of order delivery and high jitter. Moreover, most data transfer (video and voice, etc) require that packets arrive in the same order in which they have been transmitted. Therefore, fixed routing is adopted for most deployments. Assuming that data from a given node follows a single path, we can obtain an optimum tree routing structure. To enable SR and optimize throughput, we propose a routing algorithm, namely asymmetric interference aware routing (AIAR).

As shown in Algorithm 1, consider a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where link $\mathcal{E}_i$ with source $s_{\mathcal{E}i}$ and destination $d_{\mathcal{E}i}$. A link $\mathcal{E}_1$ is said to be non-interfering with another link $\mathcal{E}_2$ if: (i) $d_{\mathcal{E}_2} \notin \mathit{neighbour}\,(s_{\mathcal{E}_1})$, i.e., no receiver in the neighbourhood of the transmitter; (ii) $s_{\mathcal{E}_2} \notin \mathit{neighbour}\,(d_{\mathcal{E}_1})$, i.e., no transmitter in the neighbourhood of the receiver; (iii) $d_{\mathcal{E}_2} \neq d_{\mathcal{E}_1}$, i.e., one node cannot receive from two transmitters.

We use a parameter *link interference parameter*, $\mathcal{L}(x, y)$ to denote the sum of data rates of all links that it interferes with. A node on entry calculates the $\mathcal{L}(x, y)$ of every link it is involved in. It then selects a suitable node as its parent node using *node blocking parameter* $\mathcal{B}(n)$

$$B\,(n) = \arg \min_{\forall n_i \in \text{neighbours}(n)} \{B\,(n_i) + L\,(n, n_i)\} \tag{5}$$

The parent node is hence the node that offers minimal interference with other links. This algorithm is dependent on the order in which the nodes enter the network. Let H denote the hop count of the farthest node from BS, if we execute the algorithm H times, we can obtain an optimal routing tree. Our following discussion on scheduling algorithms is based on an assumption that the minimum interference routing tree is pre-determined.

4 Enhanced Scheduling using Multipointer Approach

In this section, the new multipointer scheduling algorithm is presented, followed by modifications to the three scheduling algorithms in order to take into account SR. Finally, a multipointer scheduling example is presented.

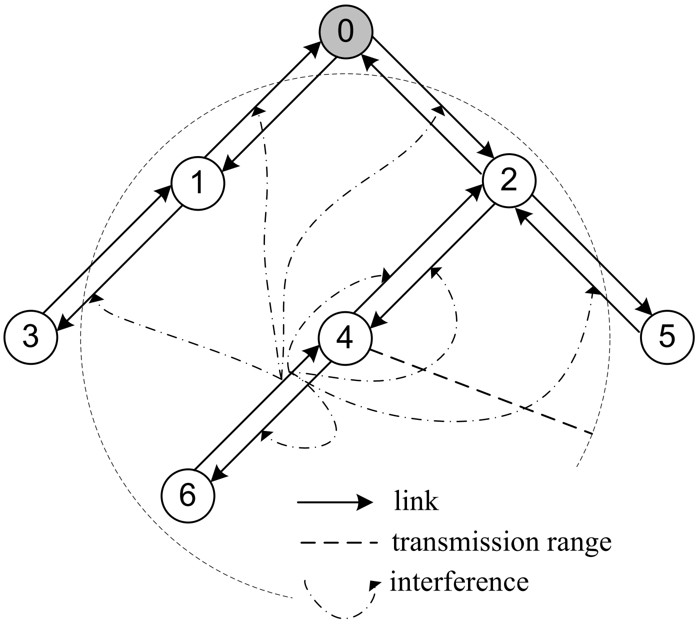


Fig. 3 Mesh Network Architecture with underlying Graph Structure

Algorithm 1 Asymmetric Interference Aware Routing

```
Input: Graph of the network including all edges and nodes - $\mathcal{G}(\mathcal{V},\mathcal{E})$
Output: Routing tree with parents of each node - $\mathcal{G}'(\mathcal{V},\mathcal{E}')$
Initialize: $\mathcal{V} \leftarrow \{1,2,3,...,M\}$, $\mathcal{V}' \leftarrow \{0\}$
For every edge in $\mathcal{E}$, calculate $\mathcal{L}(x,y)$ from the above given conditions
$\mathcal{B}(n)$=0 for $n$=0; otherwise infinity
while ($\mathcal{V} \neq \emptyset$) do
    Choose a node $i$ from $\mathcal{V}$ with least number of hops to the MBS
    for $j$ = each neighbour of $i$ in $\mathcal{V}$ do
        $\mathcal{B}(i)$ = minimum{$\mathcal{B}(i)$, $\mathcal{L}(i,j)$+$\mathcal{B}(j)$}
        $parent(i) = j$, for which $\mathcal{B}(i)$ is minimum
    end for
    $\mathcal{V} \leftarrow \mathcal{V} - \{i\}$
    $\mathcal{V}' \leftarrow \mathcal{V}' + \{i\}$
    $\mathcal{E}' \leftarrow \mathcal{E}' + \{i, parent(i)\}$
end while
```

4.1 Multipointer Algorithm for Packet Scheduling

In most algorithms that we have seen, the scheduling of resources is done from the first available slot towards the last available slot. This is acceptable for the *PMP mode* [7]. However, for the *mesh mode*, buffering constraints and the multihop distance to the destination render the normal iteration useless. Under these circumstances, we need to develop a different approach. Scheduling validity is defined as the number of frames over which a particular sched-

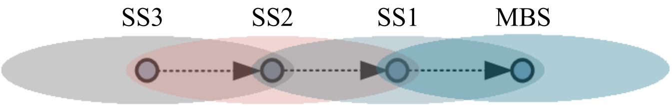


Fig. 4 WiMAX network with four nodes

ule grant is valid. For WiMAX deployments with centralized scheduling, the scheduling validity is set to the hop count of the farthest node from the BS. As shown in Algorithm 2, the number of pointers used for scheduling is equal to the scheduling validity. Consequently, we require one pointer for each frame in the scheduling period. When all the slots in that particular frame have been allocated, the pointer is merged with the pointer in the next frame.

Algorithm 2 Multipointer Algorithm Structure

```
Initialize: multipointer $P_i \leftarrow$ the first data slot in $i^{th}$ frame, $n_j^i \leftarrow$ the number of slots
    required by the $j^{th}$ node in the $i^{th}$ frame, where $j \leq M, i \leq SP$
for $i = 1 : SP$ do
    while $n_j^i > 0$ do
        $P_i \leftarrow j$
        $n_j^i \leftarrow n_j^i - 1$
        $P_i \leftarrow P_i + 1$
        if $p_i =$ end of frame then
            $P_i \leftarrow P_{i+1}$
        end if
    end while
end for
```

Let us look at how a multipointer algorithm works. Consider a WiMAX network with four nodes as given in Fig. 4. Since the maximum hop count of the farthest node to the MBS is 3, we use a scheduling period of 3 frames. As depicted in Fig. 5(a), for the sake of simplicity, let us assume that each frame has 20 time slots, where the pointers in each of these frames are denoted by P_1, P_2 and P_3, and initialized to the first slot in the 1^{st}, 2^{nd} and 3^{rd} frame, respectively. Assuming that each node has a request of eight slots, we can obtain the results are shown in Fig. 5 after running the multipointer algorithm. In Fig. 5, a slot with a circle mark inside indicates its allocation to a node; a slot with a diamond mark inside indicates during which data transfer is being done; a shaded slot without any mark indicates during which data is received for a node.

4.2 Fixed Scheduling with Spatial Reuse

FS follows the principle of allocating resources proportional to the arrival rate at a node. The arrival rate at the node comprises of external arrivals from other nodes and arrivals from users of the node itself. The number of slots required for node i without concurrency is given in (3). We calculate the number of

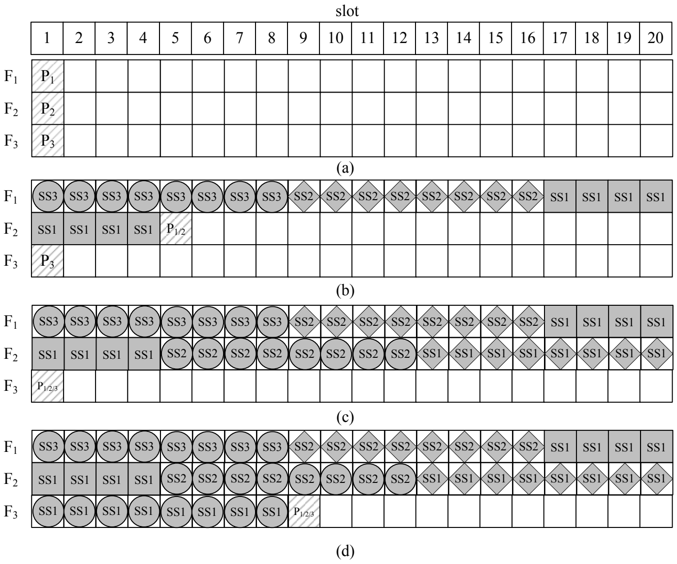


Fig. 5 Working of Multipointer Algorithm: (a) before scheduling (b) after first frame of scheduling (c) after second frame of scheduling (d) after third frame of scheduling

slots for all nodes and then allocate slots to the nodes in order of their hop count from the BS, as specified by σ_i.

To incorporate SR in terms of concurrent transmissions, we define a child matrix $c_{i,j}$, to denote whether or not a node j is a child/sub-child of node i. This helps in calculating the total arrival rate at a given node. Then, we use $I_{i,j}$ to define the interference matrix, where if link i interferes with link j, then $I_{i,j}$ is equal to 1. We should note that the interference matrix is a symmetric matrix. Hence, $I_{i,j}$=$I_{j,i}\leq$ 1.

As we consider only uplink scheduling and a routing tree, each node has only one uplink, and this link from node i to its parent node can be denoted as $L_{i,parent(i)}$. Since no other link from i can serve as an uplink, we can simplify the notation as L_i. The order in which each node transmits is given in *MSH-CSCF*. This is denoted by σ_i for node i. The scheduling output gives a set of nodes that can transmit concurrently during a slot.

FS-SR follows a staggered approach. The arrival rate to each node is calculated as a sum of arrival rate to itself and its children nodes. The number of slots hence required in total at each node is the sum arrival rate divided by the rate of data transfer per slot. In particular, if the total number of slots cannot be serviced by the scheduling period, the number of slots granted to each node is multiplied by a scheduling weight, w, which is defined as the ratio of the number of slots in the scheduling period to the total number of slots required.

Algorithm 3 Fixed Scheduling with Spatial Reuse

Input: arrival rates, children matrix, order of each node **Output:** scheduling grant for all nodes

```
 1: calculate $n_i$ and $w$
 2: if $w < 1$ then
 3:     $n_i = n_i \times w, i = 1 : M$
 4: end if
 5: Initialize: multipointers for all frames $P_i, i = 1 : SP$
 6: set $Q_i = \lambda_i \times SP$
 7: for i=1:SP do
 8:     for j=1:N do
 9:         find node $n_s$=$\arg\min_{i=1...M} \{\sigma_i\}$, $Q_{n_s} > 0$ and $I_{n_s n_t} = 0 \; \forall \; n_t \in C$
10:         $C \leftarrow \{C, n_s\}$
11:         continue steps 8-10 until all nodes have been considered
12:         if $C = \emptyset$ then
13:             break
14:         end if
15:         allocate slot $P_i$ to C
16:         $P_i \leftarrow P_i + 1$
17:         $Q_k = Q_k - r_k$ and $SQ_{parent(k)} = SQ_{parent(k)} + r_k, \forall k \in C$
18:         if $P_i$=end of frame then
19:             $P_i = P_{i+1}$
20:         end if
21:     end for
22:     $Q_k = Q_k + SQ_k, k = 1 : M$
23: end for
```

4.3 Ordered Scheduling with Spatial Reuse

In OS, the scheduling weight for node is defined as

$$w_i = \sum\nolimits_{j=1}^{h_i} \left(E\left[r_{p_{i,j}} \right] \right)^{-1} \tag{6}$$

Given the parameter w_i, interference matrix I, and individual queue lengths Q_i, OS orders the nodes based on their w_i values and assigns each slot to the first node that can fully utilize that slot. Preference is given to a node with a lower w_i value. Furthermore, we propose OS-SR to implement SR (See Algorithm 4). The concurrent set 'C' allocated to each slot contains the set of all nodes that can transmit during that slot.

For each slot, OS-SR first selects the node that has the minimum w_i value and queue length longer enough to use the entire slot. Then, it selects other nodes that can fully utilize a slot and can transmit concurrently with the first selected node in order of their respective parameter values. The corresponding uplinks of these nodes are added into the concurrent set 'C'. This method continues until all nodes are scheduled.

4.4 Per Slot Scheduling with Spatial Reuse

PSS works on a dynamic programming principle (See Algorithm 5) to schedule one slot at a time to a node which can transmit the maximum number of bits in

Algorithm 4 Ordered Scheduling with Spatial Reuse

Input: arrival rates, children matrix, order of each node
Output: scheduling grant for all nodes

```
 1: calculate $w_i$
 2: Initialize: multipointers $P_i, i = 1 : SP$
 3: set queue length of all nodes to $Q_i$ from $MSH - REQ$ packet received
 4: for i=1:SP do
 5:     for j=1:N do
 6:         find node $n_s$=$\arg\min_{i=1...M} \{w_i\}$, $Q_{n_s} > r_{n_s}$ and $I_{n_s n_t} = 0 \; \forall \; n_t \in C$
 7:         $C \leftarrow \{C, n_s\}$
 8:         continue steps 6-8 until all nodes have been considered
 9:         if $C = \emptyset$ then
10:             find node $n_s$=$\arg\min_{i=1...M} \{w_i\}$, $Q_{n_s} > 0$ and $I_{n_s n_t} = 0 \; \forall \; n_t \in C$
11:             $C \leftarrow \{C, n_s\}$
12:             continue steps 9-11 until all nodes have been considered
13:         end if
14:         if $C = \emptyset$ then
15:             break
16:         end if
17:         allocate slot $P_i$ to C
18:         $P_i \leftarrow P_i + 1$
19:         $Q_k = Q_k - min\{Q_k, r_k\}$ and $SQ_{parent(k)} = SQ_{parent(k)} + min\{Q_k, r_k\}, \forall k \in C$
20:         if $P_i$=end of frame then
21:             $P_i = P_{i+1}$
22:         end if
23:     end for
24:     $Q_k = Q_k + SQ_k, k = 1 : M$
25: end for
```

that given slot. With SR, a slot is scheduled to nodes in the concurrent set 'C', which can transmit simultaneously without experiencing severe interference.

One advantage of PSS is that the average resources required at nodes which have a higher egress rate is low. However, the corresponding disadvantage is that under high loads, the nodes with a lower transmission rate are allocated resources towards the end of the scheduling frame. Hence, these nodes have longer queue lengths. The data drop probability at these nodes is also high due to the prioritized scheduling of nodes with higher egress rates.

5 Simulation Design

Our simulation in OMNeT++ used the WiMAX mesh network as shown in Fig. 1, designed using a Network Descriptor (NED). If a node was in transmission range of another node, then it was connected to that node via a link. All links were bidirectional and the link parameters were governed by their burst profiles as shown in Table 2. The burst profile governed the modulation, coding rate of that link. Hence, a different burst profile translated to a different data rate. The link rates (in Mbps) between any two nodes were predetermined, as specified in a NED, and they were aggregated in a connection matrix, H with

Algorithm 5 Per Slot Scheduling (Concurrency) with Spillover

```
Input: arrival rates, children matrix, order of each node
Output: scheduling scheme for all nodes
 1: Initialize: multipointers $P_i, i = 1 : SP$
 2: set queue length of all nodes to $Q_i$ from $MSH-REQ$ packet received
 3: for i=1:SP do
 4:     for j=1:N do
 5:         find node $n_s$=arg $\max_{\forall i \notin C} \{\min(Q_i, r_i)\}$ and $I_{n_s n_t} = 0 \ \forall \ n_t \in C$
 6:         $C \leftarrow \{C, n_s\}$
 7:         continue steps 6-8 until all nodes have been considered
 8:         if $C = \emptyset$ then
 9:             break
10:         end if
11:         allocate slot $P_i$ to C
12:         $P_i \leftarrow P_i + 1$
13:         $Q_k = Q_k - min\{Q_k, r_k\}$ and $SQ_{parent(k)} = SQ_{parent(k)} + min\{Q_k, r_k\}, \forall k \in C$
14:         if $P_i$=end of frame then
15:             $P_i = P_{i+1}$
16:         end if
17:     end for
18:     $Q_k = Q_k + SQ_k, k = 1 : M$
19: end for
```

element (i, j) corresponding to burst profile of link between node i and node j.

$$H_{11\times11} = \begin{pmatrix} 0&6&0&0&0&0&0&0&4&0&0 \\ 6&0&3&0&0&0&2&0&0&0&0 \\ 0&3&0&6&4&0&0&0&0&0&0 \\ 0&0&6&0&0&3&0&0&0&0&0 \\ 0&0&4&0&0&4&0&6&0&0&0 \\ 0&0&0&3&4&0&0&0&0&0&0 \\ 0&2&0&0&0&0&0&3&2&0&0 \\ 0&0&0&0&6&0&3&0&0&2&2 \\ 4&0&0&0&0&0&2&0&0&3&3 \\ 0&0&0&0&0&0&0&2&3&0&0 \\ 0&0&0&0&0&0&0&2&3&0&0 \end{pmatrix}$$

Table 2 Physical Parameters of Burst Profile

Burst Profile No.	Modulation	Coding Rate	Uncoded bytes per OFDM Symbol	Uncoded bytes per slot
1	QPSK	1/2	24	96
2	QPSK	3/4	36	144
3	16QAM	1/2	48	192
4	16QAM	3/4	72	288
5	64QAM	2/3	96	384
6	64QAM	3/4	108	432

The maximum load on the network with respect to arrival rate at each SS was calculated using the method in Section 3 and the load is 1.74Mbps, which was corresponding to a normalized load of 1.0. The normalized load on the network was varied from 0 to 1.0 with a step size of 0.1. Poisson packet arrival was considered and the packet size was varied exponentially with respect to the mean. Traffic generation occurred in the *trafgen* submodule of the substation. The mean packet size was fixed at 768 bits. If all the nodes have the same average arrival rate λ_i, the average size of the packet is fixed at L and the packet inter-arrival time is calculated as

$$E[\tau] = \frac{L}{\lambda} \tag{7}$$

All traffic is directed towards the BS. The traffic generator can also be adapted to include priority, tardiness and drop probability of the packets, which has been left out for our future study. Upon receiving data packets, either self-generated or from other nodes, SSs put them into a reserved buffer, and then moved them into the transmitter buffer at the beginning of each frame. This perfectly simulated the buffer constraints of the WiMAX network. The queuing rule was *first-in first-out (FIFO)*. Other queuing methods like *earliest deadline first* and *fair scheduling* could also be employed.

Each packet was time-stamped on of creation at the traffic-generator, and given a time-to-live (TTL) value that was also varied exponentially with respect to the mean. We tested our algorithms for four mean TTL values: 60ms, 120ms, 240ms and 480ms. Packets which experienced a delay longer than their TTL values were dropped by intermediate nodes or the BS.

The simulation was run for $5000s$ or 5×10^6 frames. We can safely assume that simulations could accurately record the real-time performance of the WiMAX mesh network. Each frame was assigned a duration of $10ms$ and the scheduling period was set to 4 frames, as the maximum hop-count amongst all leaf nodes is 4 in Fig. 1.

During initialization, we first execute AIAR to obtain the optimal routing tree and then applied a scheduling algorithm. We simulated the request and grant process and data transmission with buffer constraints in the WiMAX mesh network, and analyzed the performance of each of the six scheduling algorithms discussed above, namely, FS, OS, PSS, FS-SR, OS-SR and PSS-SR.

6 Results Analysis

To analyze the performance of a WiMAX network, we mainly look into two main aspects: (i) *mean delay* – end-to-end time taken by a packet to reach its destination; and (ii) *throughput* – the number of packets which have been successfully delivered.

6.1 Mean Delay

The delay of a packet is defined as the time it takes from the time when it is created by the traffic generator to the time when it reaches its destination.

As shown in Fig. 6, at low traffic load with low TTL values (60ms$\sim$120ms), FS outperformed other algorithms, which makes FS a suitable scheduling algorrithm for low-data-rate and time-critical applications like VoIP and Telnet. Next, scheduling priority under OS was to the node with the fastest path towards the BS; while under PSS was to the fastest link. Due to this *selective* nature of these algorithms, nodes with slower links/paths were allocated resource at the end of a scheduling period, leading to a higher mean delay. In addition, every scheme outperformed its corresponding counterpart when implemented with SR. This reduction in mean delay could be attributed to less mean medium access time due to the concurrent transmissions.

At low traffic load with high TTL value (240ms$\sim$480ms), there was no apparent difference in the performance of the six scheduling schemes. Scheduling algorithms employing SR performed just marginally better because they increased the effective capacity of the network, thereby enabling more packets to reach the destination faster. Furthermore, we find that *proactive* scheduling algorithms like FS significantly outperformed the *reactive* scheduling algorithms like OS and PSS. If we define 'reaction time' as the number of scheduling periods taken by the mesh network to respond to a given mesh-request. As shown in Fig. 2, for PSS and OS that are based on the queue length of nodes, the reaction time is 2 scheduling periods. However, for FS, this is reduced to zero, as FS is based on arrival rate at each node. Therefore, FS offered the smallest delay amongst the six.

At high traffic load with low TTL values (60ms$\sim$120ms), OS and PSS outperformed FS. This is because the fact that FS would allocate more resources to weaker links, giving rise to a longer waiting time for data with stronger links and an overall increase in average delay. In particular, the '*selective*' nature of OS and PSS did not favor scheduling to moderate and weak paths/links. Some packets were lost due to excessive delay and many packets were delayed as the algorithm allocated resources first to stronger links and paths and then to weaker links. The overall delay hence increases at high traffic load for these algorithms. Packets with slower links (See Fig. 6) were dropped and hence the average delay of packet delivery was comparable between OS and PSS. Next, with SR, OS-SR and PSS-SR outperformed OS and PSS, respectively. Furthermore, FS-SR provided the best performance. This was because of simultaneous allocations to strong and weak links at the same time. Since FS was not selective in nature, data were allocated slots uniformly and hence we had a lower average delay.

At high traffic load with high TTL value (240ms$\sim$480ms), FS-SR offered the best average delay under all loads and all TTL conditions, and it could be used for almost all applications. Furthermore, PSS outperformed OS, while OS-SR and PSS-SR performed similarly.

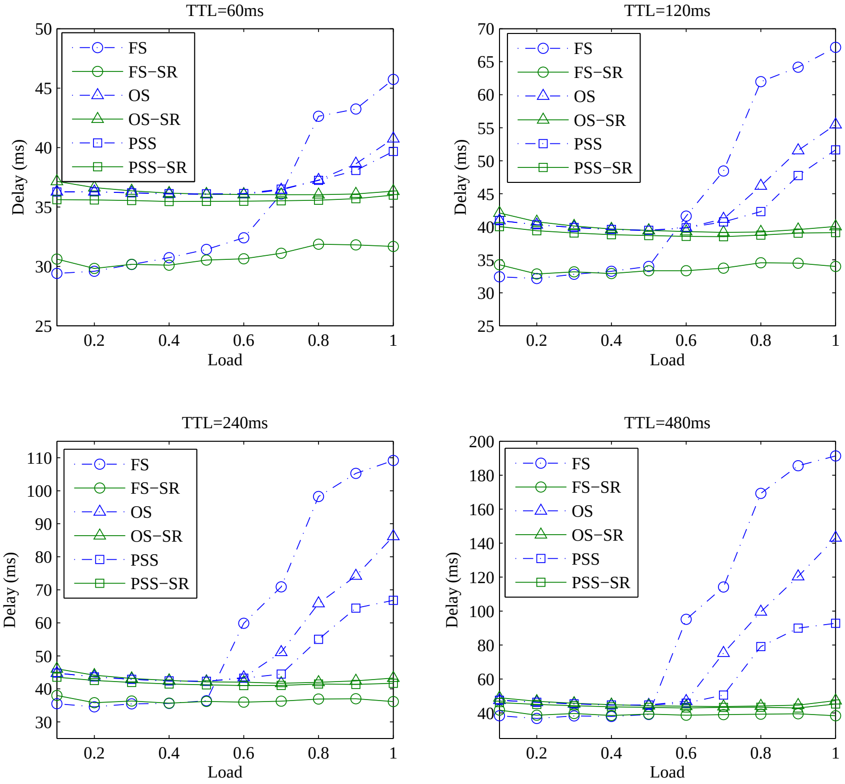


Fig. 6 Mean Delay Analysis

6.2 Throughput

Throughput of a system is a measure of the data generated by the node compared to the amount of data received by the MBS. In our case, data which were not delivered with a TTL were dropped and hence an unfair scheduling algorithm may lead to a low throughput.

As shown in Fig. 7, at low traffic load, FS outperformed FS-SR, implying that SR does not improve throughput. Furthermore, the performance of other four schemes was about the same. Interestingly, the performance of six schemes were converging when the TTL values were increased at low traffic load.

At high traffic load, without SR, PSS provided the highest throughput as it strived to transmit the maximum number of bits in every slot. The path-selective OS provided a lower throughput than PSS; FS resulted in the lowest throughput. With SR, FS-SR offered the maximum throughput. The proactive scheduling saved '*reaction time*' and hence allowed faster delivery of nodes. Furthermore, OS-SR and PSS-SR offered comparable throughput. When the TTL values were increased, the performance of FS-SR, OS-SR and PSS-SR converged.

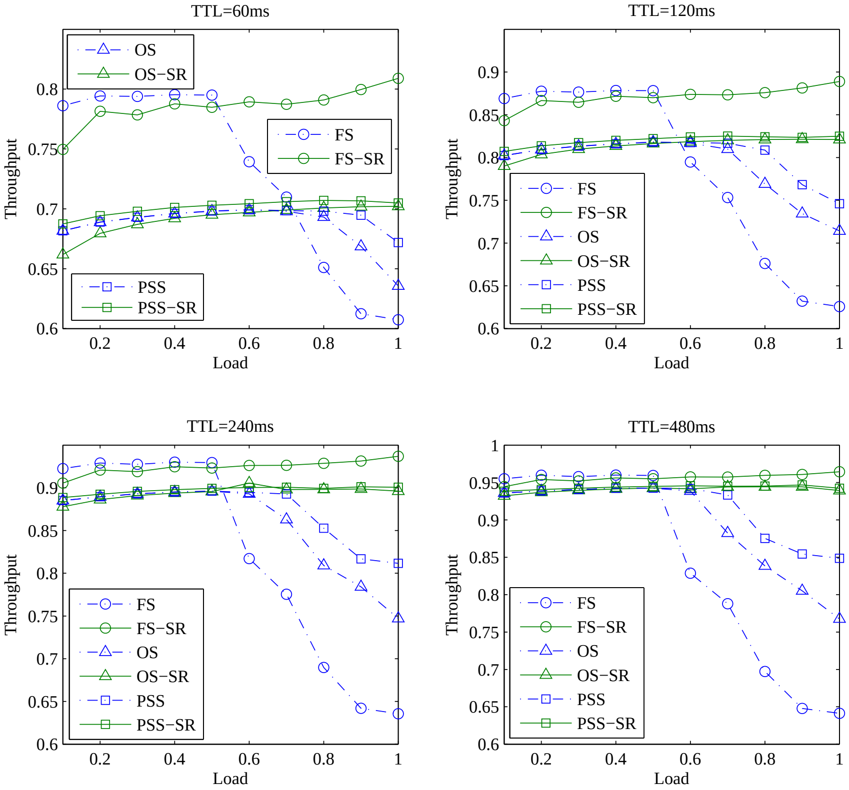


Fig. 7 Throughput Analysis

7 Conclusion

A concurrent routing scheme with spatial reuse (SR), asymmetric interference aware routing (AIAR) was proposed. Then, a multipointer approach of implementing scheduling algorithms for WiMAX mesh networks operating in the *mesh mode* was proposed. Six scheduling algorithms, *fixed scheduling (FS)*, *ordered scheduling (OS)* and *per-slot scheduling (PSS)*, *FS-SR*, *OS-SR* and *PSS-SR* were proposed. Through computer simulations, we found that without SR, *PSS* provides highest throughput while *FS* gives lowest packet delay. While incorporating SR, the *FS-SR* scheme offers the best overall performance.

References

1. IEEE Standard (2011). IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Broadband Wireless Access Systems Amendment 3: Advanced Air Interface, IEEE Std 802.16m-2011(Amendment to IEEE Std 802.16-2009), 1-1112.
2. W.-H. Liao, S. P. Kedia and A. K. Dubey. (2010) A centralized scheduling algorithm for WiMAX mesh network. In Proc. of IEEE NOMS'10 (pp. 858-861). Osaka, Japan, 19-23 April.

3. Y. Huang, S. Yang, D. Teng, W. He and L. Xie. (2010) A fair rate-balance link scheduling in WiMAX mesh network. In Proc. of IEEE WCNIS'10 (pp. 355-359). Beijing, China, 25-27 June.
4. S.-C. Lo and L.-C. Ou. (2009) Efficient Algorithms for Routing and Centralized Scheduling for IEEE 802.16 Mesh Networks. In Proc. of SCALCOM-EMBEDDEDCOM'09 (pp. 212-217). Dalian, China, 25-27 September.
5. J. El-Najjar, C. Assi and B. Jaumard. (2009) Joint routing and scheduling in WiMAX-based mesh networks: A column generation approach. In Proc. of IEEE WoWMoM'09 (pp. 1-10). Kos, Greece, 15-19 June.
6. S.-I. Chakchai, R. Jain and A. K. Tamimi. (2009). Scheduling in IEEE 802.16e mobile WiMAX networks: key issues and a survey, IEEE Journal on Selected Areas in Communications, 27, 156-171.
7. N. A. Ali, P. Dhrona and H. Hassanein. (2009). A performance study of uplink scheduling algorithms in point-to-multipoint WiMAX networks, Computer Communications, 32, 511-521.
8. Joint routing and scheduling for maximizing fair throughput in WiMAX mesh network. In Proc. of IEEE PIMRC'08 (pp. 1-5). Cannes, France, 15-18 September.
9. IEEE Standard (2007). Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems - Amendment 3: Management Plane Procedures and Services (Amendment to IEEE 802.16-2004), IEEE Approved Draft Std P802.16g/D9, Apr 07.
10. F. Jin, A. Arora, H. Jinho and H. A. Choi. (2007) Routing and packet scheduling in WiMAX mesh networks. In Proc. of BROADNETS'07 (pp. 574-582). Raleigh, NC, 10-14 September 2007.
11. F. Xie, K. A. Hua and N. Jiang. (2007) Achieving true video-on-demand service in multi-hop WiMax mesh networks. In Proc. of IEEE LCN'07 (pp. 287-294). Dublin, Ireland, 15-18 October.
12. H. Shetiya and V. Sharma. (2006) Algorithms for routing and centralized scheduling in IEEE 802.16 mesh networks. In Proc. of IEEE WCNC'06 (pp. 147-152). Las Vegas, NV, 3-6 April.
13. H.-Y. Wei, S. Ganguly, R. Izmailov and Z. J. Haas. (2005) Interference-aware IEEE 802.16 WiMax mesh networks. In Proc. of IEEE VTC'05-Spring (pp. 3102-3106 Vol. 5). Stockholm, Sweden, 30 May-1 June 2005.
14. J. Tao, F. Liu, Z. Zeng and Z. Lin. (2005) Throughput enhancement in WiMax mesh networks using concurrent transmission. In Proc. of WCNM'05 (pp. 871-874). Wuhan, China, 23-26 Sept. 2005.

Xue Jun Li received his B.Eng (First Class Honors) and Ph.D in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore, in 2004 and 2008, respectively. Between November 2007 and July 2008, he worked as Research Engineer and then as Research Fellow in Network Technology Research Centre, NTU. Between August 2008 and September 2008, he worked as Research Scientist in Temasek Laboratories @ NTU. From September 2008 to May 2011, he was with School of Electrical and Electronic Engineering, NTU as a faculty member. Between June 2011 and January 2013, he worked as Research Scientist in Institute for Infocomm Research (I^2R), Agency for Science, Technology and Research (A*STAR), Singapore. Since January 2013, he has been with School of Engineering, Auckland University of Technology (AUT), where he is a Senior Lecturer. His research interests include design/analysis of wireless networking protocols, modelling/design of radio frequency integrate circuits, and system optimizations.

Maode Ma received the B.E. degree in computer engineering from Tsinghua University, Beijing, China, in 1982, the M.E. degree in computer engineering from Tianjin University, Tianjin, China, in 1991, and the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, Hong Kong, in 1999. He is currently a Tenured Associate Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include wireless networking and network security.