On the Construction of a WiMAX Mesh Tree

Salim Nahle, Luigi Iannone, Benoit Donnet and Naceur Malouch

Abstract— The IEEE 802.16 protocol, also known as WiMAX, has been designed to support long-range communications with high bitrates, using two operation modes: Point-to-Multi-Point (PMP) and Mesh. In the mesh mode, Subscriber Stations (SSs) can directly communicate with each other, thus forming a tree, and can be used to forward other's data packets in a multi-hop fashion. On the contrary, in the PMP mode only one hop communication toward the Base Station (BS) is allowed. In this paper, we investigate the performance of the mesh mode by proposing an algorithm for constructing the WiMAX mesh tree. Our algorithm increases routes' effective throughput by splitting long links into multiple shorter ones. We show through simulations that this approach leads to improving the throughput capacity of WiMAX-based wireless mesh networks.

Index Terms—

I. INTRODUCTION

MESH networks show salient advantages such as ubiquitous coverage, rapid and cost-efficient deployment, and robustness ([1], [2]). Mesh networks are built using various technologies, however, the most commonly used are WiFi (based on the IEEE 802.11 [3] family) and WiMAX (based on IEEE 802.16 [4], [5]). Using IEEE 802.11 for the wireless backbone leads to dense and suboptimal deployments due to the short transmission range of the standard and, consequently, low aggregate throughput capacity can be obtained. WiMAX, on the other hand, promises to transcend this limitation by a transmission range of several kilometers with high data rates. This, in turn, enlarges the possible deployment scenarios for wireless mesh networks (e.g., distant rural-areas, large industrial sites, etc).

WiMAX supports two modes of communication: *Point-to-Multi-Point* (PMP) communication and *mesh mode*. The latter enables direct communications between *Subscriber Stations* (SSs), that can also be used as relays to forward other's data. Two kinds of scheduling are used to coordinate transmissions: *Centralized* and *Distributed*. The centralized approach organizes all the nodes of a WiMAX network in a tree structure rooted at a particular node, namely the *Base Station* (BS).

The way this tree is built and the choice of the links used have a deep impact on the capacity that a WiMAX backbone may offer. For instance, the length of the links (in meters) significantly affects the bitrate. Long links can only support

low bitrates, hence end-to-end throughput between source and destination may increase if relay SSs are used instead of direct communication. In a previous work, we showed that it is possible to have better throughput if more hops were used to reach the BS [6].

In this paper we propose an algorithm for constructing the mesh tree that maximizes the throughput capacity. It uses link transmission rates as an input metric, and chooses paths that have maximal end-to-end data rates. We show by simulation the efficiency of our algorithm. Deeper mesh trees (i.e., where more hops are allowed) can achieve better throughput than the case where all the stations connect to the BS with a single hop. In particular, we observe that trees allowing four hops perform better than the single-hop case.

The remainder of this paper is organized as follows: Sec. II presents our algorithm for constructing the mesh tree; Sec. III evaluates our algorithm through simulations; finally, Sec. IV concludes this paper by summarizing its main contributions and discussing directions for further work.

II. MESH TREE CONSTRUCTION ALGORITHM

A. Single Hop vs. Multi Hop

The IEEE 802.16 standard adapts different modulations and coding techniques on different transmissions, which implicitly means that different bitrates can be attained. In fact, as the physical distance increases, due to signal attenuation, more robust modulations are needed and hence lower data rates. Consequently, a connection between an SS and a BS may have better throughput if multiple hops are used rather than a single long hop. Assume for example a source and a destination that are 4 km apart, then QPSK 1/2 (*Quadrature Phase Shift Keying*) that offers about 2 Mbps is used [7]. However, if another SS, which is placed in between that is 2 km apart from each, is used, 16 QAM 3/4 (*Quadrature Amplitude Modulation*) can be used to provide about 7 Mbps on each link, leading to 3.5 Mbps effective end-to-end data rate.

On the other hand, and according to the standard [4], increasing the number of hops to the BS implicitly introduces more control and scheduling overhead. Furthermore, as we will show later, above a certain threshold, the overhead and the fragmented transmission lead to a drastic drop in performances.

In an earlier version of this work [6], we showed that deeper trees may result in better throughput for the intra mesh communications (i.e., communications between SSs). For the communications via the BS, which represents the major traffic in wireless mesh networks, the results show that improvements are obtained in some specific cases. We conjecture that the reason of not realizing the expected benefit of multi-hopping in

other cases is that we tried to construct a balanced or a quasi-balanced tree. In this paper we propose a clever algorithm to build the mesh tree, in order to maximize data rates on links between SSs and the BS.

B. Construction Algorithm

In this section, we propose an algorithm to construct the mesh tree. The algorithm, formally defined in Algorithm 1, aims at maximizing throughput capacity. The objective is to select links that have the highest data rates among the set of all possible paths between an SS and the BS.

A new node that enters the mesh network will first search for mesh network configuration MSH-NCFG messages. In this work we consider that all the SSs are in the range of the BS and we try to find if it is better to join the BS directly or to multi-hop through different SSs. The *metric* we aim at maximizing is the effective data rate on the path towards the BS. The effective data rate of a certain node refers to the rate that it can send to the BS. Due to multi-hopping, it is computed as the average data rate on the links on its path towards the BS, divided by the number of hops (links) on this path, which is normally carried in the MSH-NCFG message.

We suppose that each node knows the transmission rate of the links connecting it to the different candidate parent nodes, which in the WiMAX terminology are called candidate Sponsoring Nodes (SN). We also assume that the effective data rate of each SN is known for the new nodes.

Now assume that r_{ij} is the data rate on link (i,j) between SS_i and SS_j, r_i is the effective data rate of SS_i, P_i is the set of links on the path from SS_i towards the BS, and n_i is the number of links on this path. Thus, $r_i = \left(\sum_{(j,k)\in P_i} r_{jk} \right) / n_i^2$, where $\left(\sum_{(j,k)\in P_i} r_{jk} \right) / n_i$ is the average data rate on the path P_i. We denote by CN_w the set of candidate SN for a new joining node SS_w. Then node SS_w chooses, as a parent, the SN in CN_w that maximizes its effective data rate r_w as follows:

$$\max_{\forall i \in CN_w} \frac{r_{wi} + n_i^2 \times r_i}{(n_i + 1)^2} . \tag{1}$$

This means that each node chooses among the different candidate SSs the one that maximizes its effective data rate with the BS. The number of hops is squared in the numerator and the denominator, since we account for the average data rate along the path of an SS. Eqn. 1 naturally allows the distinction between two routes that have the same average data rate but different number of hops.

A new SS joining the network may alter the whole configuration since an existing SS that has already chosen a parent may notice that there exists a better path with higher throughput. Consequently, it changes its parent. The tree structure is broadcasted in the *Mesh Centralized Scheduling Configuration* MSH-CSCF. Based on the information carried in this message (namely the burst profile for each link which represents the modulation and coding schemes used), the achievable rates of the different links are known, then and as a consequence, an already existing SS may find a higher-throughput path to the BS. In this case only, it chooses to change its SN by sending new mesh network entry message (MSH-NENT) to the new candidate SN.

Algorithm 1 Mesh tree construction

Require: p, the current parent of the SS_w. $p = \emptyset$ if the node is joining the mesh. r, the current data rate. $r = 0$ if the node is joining the mesh.

1: **procedure** PARENTSELECTION(p, r)
2:     $CN \leftarrow$ MESH-NCFG
3:     **for all** $i \in CN$ **do**
4:         metric $\leftarrow \frac{r_{wi} + n_i^2 \times r_i}{(n_i+1)^2}$
5:         **if** metric $> r$ **then**
6:             $p \leftarrow i$
7:             $r \leftarrow$ metric
8:         **end if**
9:     **end for**
10:     send(p, MSH-NENT);
11: **end procedure**

III. EVALUATION

A. Methodology

The performance of any network is strongly dependent on the traffic pattern generated by its nodes. Throughout this study, we consider that all the stations generate the same amount of uplink traffic and hence request the same amount of resources. Similarly, all the nodes request the same amount of downlink traffic. This is a reasonable assumption considering all nodes having the same average behavior on the long term. It is worth to remark that the assumptions we made do not imply an equal uplink traffic volume and downlink traffic volume. We evaluate our algorithm by implementing our own simulator in Java. We implement the centralized scheduling algorithm of the mesh mode as described in the IEEE 802.16 standard which operates in *Time Division Duplex* (TDD) to access the channel. Based on this algorithm, the BS collects the requests of all SSs in the mesh tree and allocates grants by distributing the available minislot space accordingly. The BS adopts proportional fairness. That is, the allocations are done proportional to links' data rates. Two links that must support the same user's traffic but have different data rates are not allocated the same number of minislots. Links (consequently SSs) with lower rates are allocated more slots. Each MAC frame consists of 256 minislots. The scheduling control subframes are only used by centralized scheduling (since we do not account for distributed scheduling). For the physical layer, we use the bitrate distance function described by Betancur et al. [7]. In order to lower the complexity of the problem, we use only three modulation schemes, namely 64QAM 3/4, 16QAM 1/2 and QPSK 1/2. These schemes offer, respectively, 11, 4.8, 2.2 Mbps respectively.

The number of nodes n is kept constant during the simulations. It is set to 49 nodes distributed on a 7×7 rectangular grid. The BS is the upper-left-corner node on the square grid. The maximum distance is set to 5 km.

The tree is constructed based on the algorithm we discussed in Sec. II-B. However, in order to learn the full potential of multi-hopping, we added the constraint of fixing the maximum allowed hop number m which is the tree depth. We start by 1 which represents the single hop case and increase this value until 7, since the *number of hops* is a 3-bits field sent in the
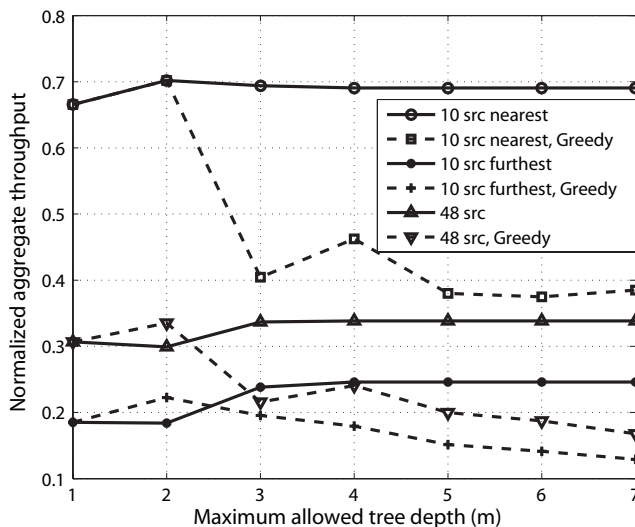
Fig. 1. Normalized aggregate throughput for ten and 48 sources.

`MSH-NCFG` message. For instance, $m = 2$ means that we set, while constructing the mesh tree, the maximum value to two hops. An SS that wants to know the best path to the BS will just compare the available bitrate upon connecting directly to the BS, and the effective data rate upon connecting to an SS that is only one hop away from the BS.

B. Results

Fig. 1 gives the *normalized aggregate throughput*. It is the aggregate throughput divided by the maximal possible throughput which is the maximal data rate namely 11 Mbps in our case. Fig. 1 plots six curves, three curves in bold solid lines showing the throughput obtained based on the previously presented algorithm, and three others in dashed lines presenting the throughput obtained based on another algorithm that we discuss later.

The curves labeled *10 src* represent the case where ten SSs are sending traffic to the BS while the curves labeled *48 src* represent the case when all the SSs in the system send traffic to the BS. The two curves labeled *10 src-nearest* depict the throughput when the SSs are close to the BS. The two curves labeled *10 src-furthest* correspond to simulations with the furthest SSs from the BS.

Looking first at the three curves based on our algorithm, we notice that the throughput obtained thanks to multiple hops outperforms the throughput of a single hop for a tree depth between two and four. This is specially true when the traffic sources are far from the BS. In this last case, the throughput improvement with four hops being allowed is about 30% compared to the single-hop case. The mesh tree obtains the highest throughput when four hops are allowed except for the closest ten stations case, where limiting the number of hops at two, yields more throughput than three and four hops. This

is reasonable since these nodes are only one hop or two hops from the BS according to our algorithm. A small decrease occurs with three and four hops, since the control information increases without gaining from increased data rate as SSs with higher number of hops are not transmitting.

When the tree depth is above four ($m > 4$), we notice that there is no change in the obtained throughput. Our algorithm will never allow an SS to choose an SN which is more than four hops away from the BS since no more data rate gain is obtained. In order to evaluate the throughput capacity when more than four hops are used, we use another algorithm that forces each node to choose the SN that minimizes the average hop distance (in meters), i.e., all SSs connect to their nearest neighbor in a way to exploit the use of m hops. We call this algorithm *Greedy algorithm*. The dashed curves show the throughput obtained with this algorithm. The best performance is obtained for $m = 2$. Nevertheless, the highest throughput never exceeds the one obtained by our algorithm. Moreover, when the tree depth is above four ($m > 4$), the performance of multi-hopping strongly decreases.

IV. CONCLUSION

WiMAX mesh mode may play an important role in providing access in wide areas. The way mesh trees are built affects their performances. In particular, the presence of long links reduces the aggregate throughput capacity due to the reduced data rates. We proposed in this paper an algorithm that constructs mesh trees by maximizing effective data rates on different links. In particular, long links are split into multiple shorter links that support higher data rates. We showed by simulations that multi-hopping based on our algorithm provides better throughput than single hop communications even when all the stations are within the BS range. Our current efforts consist in studying our algorithm on random topologies, exploring the impact of the placement of nodes on the throughput capacity and extending it to account for multiple-channels use and frequency spatial reuse.

REFERENCES

[1] I. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networs Elsevier Science*, vol. 47, no. 4, Jan. 2005.
[2] D. Beyer, "Fundamental characteristics and benefits of wireless routing (Mesh) networks," in *Proc. 8th Wireless Communications Association (WCA) International Technical Symposium*, Jan. 2002.
[3] IEEE 802 Standard Working Group, "Wireless lan medium access control (MAC) and physical layer (PHY) specifications," IEEE, Standard 802.11-1999, June 1999.
[4] , "IEEE standard for local and metropolitan area networkspart 16: air interface for fixed broadband wireless access systems," IEEE, Standard 802.16d-2004, Oct. 2004.
[5] WiMAX forum, available online: http://www.wimaxforum.org/home.
[6] S. Nahle, L. Iannone, B. Donnet, and T. Friedman, "Investigating depthfanout trade-off in WiMAX mesh networks," in *Proc. 1st WEIRD Workshop*, May 2007.
[7] L. Betancur, R. Hincapie, and R. Bustamante, "WiMAX channel: PHY model in network simulator 2," in *Proc. Workshop on ns-2: The IP Network Simulator*, Oct. 2006.