A Multimedia Traffic Scheduler for IEEE 802.16 Point-to-Multipoint Networks

Spyros Xergias+, Nikos Passas†, Anastasia Lygizou† and Apostolis K. Salkintzis*
† University of Athens, Greece       * Motorola

Abstract—In this paper, we describe and evaluate an advanced traffic scheduler for IEEE 802.16 Point-to-Multipoint networks, referred to as “Colored Frame Registry Tree Scheduler” (C-FRTS). The main characteristics of this scheduler are i) the use of a tree structure to prepare the creation of time frame maps and reduce processing requirements at the beginning of each frame, and ii) a colored discrimination of packets scheduled for transmission to guarantee the bandwidth reserved for different connections. Simulation results show that C-FRTS can guarantee major QoS parameters, such as the maximum latency and the minimum and maximum reserved bandwidth, as well as provide differentiated treatment to the various service types.

Index Terms—IEEE 802.16, traffic scheduling, tree structure, deadlines.

I. INTRODUCTION

IEEE 802.16 [1] is a standard that aims at filling the gap between local and wide area networks, by introducing an advanced system for metropolitan environments. In such a system, both point-to-multipoint (cellular) and mesh mode configurations can be supported, while node mobility is also covered by 802.16e [2]. One of the main advantages of the standard is the large degree of flexibility it provides by supporting a wide range of traffic classes with different quality of service (QoS) requirements. Being out of its scope, the standard does not describe a specific traffic scheduler to utilize these parameters.

In this paper, a traffic scheduling algorithm for IEEE 802.16 Point-to-Multipoint networks, referred to as the “Colored Frame Registry Tree Scheduler” (C-FRTS) is proposed and evaluated. The main characteristics of this scheduler are i) its ability to prepare in advance the structure of each time frame, and avoid complex processing before the frame header transmission, and ii) a packet discrimination scheme that allows fairer dropping of excess traffic during congestion. The scheduler is as an extension of the FRTS proposed in [10], that does not use the colored discrimination to differentiate packets.

The paper is organized as follows. Section II presents the main characteristics of IEEE 802.16. Section III describes the operation of the proposed scheduler, focusing on the actions required for frame preparation and creation. Section IV contains the description of the simulation model used to evaluate C-FRTS against FRTS and the obtained results. Section V contains the conclusions and plans for future work.

II. IEEE 802.16 BROADBAND ACCESS SYSTEMS

A. Basic Operation

The IEEE 802.16 standard covers the two lower layers of the protocol stack, i.e., the physical and the data link layer. The physical layer operates at 10–66 GHz (802.16) and 2–11 GHz (amendment 802.16e [2]) with data rates between 32 and 130 Mbps, depending on the channel bandwidth and modulation scheme. Three operation modes are defined: Point-to-Multipoint (PMP), Centralized Mesh and Distributed Mesh modes. This paper focuses on PMP mode where the system architecture consists of: i) base stations, that have direct connections to backhaul services and are responsible for a specific area cell, ii) stationary subscriber stations (SSs) and iii) mobile subscriber stations (MSSs).

In PMP mode, all communications between the BS and its (M)SSs are multiplexed either with Time Division Duplex (TDD) or Frequency Division Duplex (FDD). A TDD frame has a fixed duration, which may take several values (0.5, 1 or 2 msec for PMP mode). Various transmission parameters, including the modulation and coding schemes, may be adjusted individually for each SS on a frame-by-frame basis. Additionally, each frame is divided into a downlink (DL) subframe and an uplink (UL) subframe and consists of an integral number of Physical Slots (PSs) that represent the bandwidth allocation units.

The schedule of each frame is carried through the DL-MAP and UL-MAP fields that define the PSs allocated to each individual subscriber on the downlink and uplink respectively. Both DL-MAP and UL-MAP are transmitted during one or more downlink bursts, since each one of them can refer to different modulation or coding types. Each SS receives and decodes the DL-MAP looking for MAC headers that indicate data for that particular SS in the remainder of the downlink subframe. Similarly, in the UL-MAP each SS receives information on its transmission opportunities in the uplink subframe. Allocations in both directions should be compatible with the traffic characteristics and QoS requirements of each active connection and satisfy, where possible, time-varying transmission requests.

In all transmissions, data bits are randomized, FEC encoded, and mapped to one of the mandatory (Spread BPSK, BPSK, QPSK, 16-QAM, 64-QAM) or optional (256-QAM) signal constellations.
B. Scheduling Services in IEEE 802.16

IEEE 802.16 can support multiple communication services (data, voice, video, etc.) with different QoS requirements organized into different connections. Each connection is associated with a single service flow and specifies a set of traffic and QoS parameters that quantify its traffic behavior and QoS expectations. This set includes the minimum reserved traffic rate (mrtr), the maximum sustained traffic rate (mstr), the maximum latency and jitter, the traffic priority, etc. Every connection should belong to one of the different services as defined by the standard: Unsolicited Grant Service (UGS), real-time Polling Service (rtPS), non-real-time Polling Service (nrtPS), and Best Effort service (BE). Compressed real-time video is usually characterized as rtPS, in contrast to constant-bit-rate voice, which is set to UGS and video-on-demand, which is set to nrtPS. In [2], a new service, referred to as enhanced rtPS (ertPS), was defined to better support real-time service flows that generate variable size data packets on a periodic basis, e.g., VoIP with silence suppression.

The traffic scheduler located at the BS decides on the allocation of the PSs in each time frame. Uplink scheduling is performed by the BS with the aim of providing each SS with enough bandwidth for uplink transmissions or opportunities for extra transmission requests. When a SS needs additional bandwidth, it utilizes its transmission opportunities during contention periods or when it is polled by the BS, depending on its agreed QoS characteristics, to pass its transmission requests. Downlink scheduling on the other hand, considers packets waiting for transmission at the BS as implicit requests for bandwidth allocation.

C. Scheduling Algorithms for IEEE 802.16

A specific scheduling algorithm is not described in the IEEE 802.16 standard, because it is not included among the mandatory modules required for the standardized system’s operation. On the other hand, the operation of the scheduler is important for the performance of the whole system, and this is why it attracts growing attention over the last couple of years. A number of articles can be found in the literature, proposing scheduling algorithms for 802.16. These proposals are based mostly on extensions and combinations of ideas already applied in systems prior to IEEE 802.16, such as the IEEE 802.11 WLAN.

A Call Admission Control (CAC) mechanism together with a scheduler for PMP mode are described in [3], where Earliest Deadline First (EDF) is used for rtPS connections and Weighted Fair Queuing (WFQ) is used for nrtPS connections. In [4], a scheduler based on the CAC mechanism of [3] is proposed, that uses token buckets to characterize the traffic flows. Based on a queuing analytical model, [5] proposes a queue-aware adaptive uplink bandwidth allocation and rate control mechanisms for polling service in IEEE 802.16. In [6], an adaptive bandwidth reservation method based on data mining techniques is described. Two-layer service flow management architecture is proposed in [7], which are based on Deficit Fair Priority Queue. [8] proposes an interference-aware framework for 802.16 Mesh mode, based on an interference-aware route construction algorithm. Finally, in [9] an analytical model for the distributed scheduling algorithm in the IEEE 802.16 Mesh mode is presented.

III. COLORED FRAME REGISTRY TREE SCHEDULER

Considering the complexity of the QoS provision mechanism in IEEE 802.16, efficient traffic scheduling implies complex procedures and calculations. This means significant computational requirements and expensive hardware. The packet scheduler described here, referred to as the “Colored Frame Registry Tree Scheduler” (C-FRTS), is an extension of the previously proposed Frame Registry Tree Scheduler (FRTS) [10] aiming to efficiently guarantee 802.16 QoS parameters and distribute over time the computational complexity required for the time frame preparation. Additionally, the scheduler can easily deal with possible modifications of one or more transmission characteristics of a connection, such as the modulation.

More specifically, the main objectives of C-FRTS are:

1. QoS agreements between BS and SSs should be guaranteed (mrtr, mstr, maximum latency, jitter, etc.).
2. Conforming traffic should be protected against excess traffic regardless of the service priority.
3. In every frame, transmissions should be organized in an increasing modulation order, from the lower (e.g., BPSK) to the higher (e.g., 256-QAM), in both downlink and uplink subframes (as indicated in the standard).
4. Transmissions should also be organized per SS and per connection.
5. A per QoS service treatment of the transmissions should be possible, based on a specific scheduling scheme (strict, weighted, portioned, etc.).
6. The required processing at the beginning of a time frame should be limited.

To accomplish the above objectives, C-FRTS uses the tree structure of Figure 1, referred to as the Frame Registry Tree. In its full version, the tree consists of six levels, although some of them can be omitted, depending on the specific network configuration:

1st Level: Represents the time frames immediately following the present one, in a sequential order (i.e., TFj is immediately after the present frame, TFj; the next one, etc.). The maximum number of time frames in the tree depends on the maximum latency of all active connections.

2nd Level: Represents the direction (uplink or downlink).

3rd Level: Corresponds to the available modulation types. This level can consist of a maximum of twelve nodes, each one representing one of the six possible modulation types both for uplink and downlink.

4th Level: In this level, connections are organized per SS. Every SS can have a maximum of two nodes at this level in every time frame subtree, one for the uplink and one for the downlink.
5th Level: Represents the different kinds of QoS services (UGS, ertPS, rtPS, nrtPS, BE), per SS.

6th level: Consists of one leaf for every active connection in every time frame subtree, holding the number of data packets scheduled for transmission in that time frame. Optionally, more information could be stored (e.g., the number of symbols required for the transmission of the packets) in order to save processing complexity at the beginning of a time frame.

The operation of the scheduler can be divided into two main procedures: packet/request arrival and frame map creation.

1) Packet/request arrival
Since the scheduler treats both new packets arriving for the downlink and new requests for packet transmissions from the uplink in the same way, only the downlink case is described here. Newly arriving packets are scheduled for transmission in either the last time frame before their deadline or the last time frame of the tree at the time of the packet arrival, depending on whether their maximum latency is given (e.g. for UGS, ertPS, rtPS) or not (e.g. for nrtPS and BE). Moreover, it is assumed that a traffic policing mechanism is not necessarily available at the BS (e.g., leaky bucket) to ensure that the incoming traffic is consistent to each connection’s declarations during setup. Consequently, packets arriving for transmission are colored by the scheduler according to the value of their mrtr and mstr QoS parameters to indicate conforming or non-conforming traffic.

More specifically, for every packet to be transmitted:

\( \text{Deadline}(P_{ij}) = \text{ArrivalTime}(P_{ij}) + \text{Latency}(C_{ij}) \)

where \( P_{ij} \) is the \( i \)-th packet of connection \( C_j \). In this case, the leaf of connection \( C_j \) in the last time frame before the packet’s deadline is updated (i.e., the number of packets scheduled for transmission in that leaf is incremented by one). If the corresponding leaf does not exist at that time (e.g., this is the first scheduled transmission of a time frame), the appropriate path is created.

For nrtPS and BE services, no specific deadline can be calculated. In this case, the leaf of \( C_j \) in the last existing time frame in the tree at the time of the packet arrival is updated.

As an example, let us assume the case of a downlink packet arrival at the BS for connection \( C_j \) that is of type UGS and should be sent to subscriber \( SS_6 \) using 64-QAM. If the last frame that this packet can be sent to \( SS_6 \) based on its deadline, is \( TF_5 \), the required path to be updated is \( TF_5 \rightarrow \text{DL} \rightarrow 64\)-QAM \( \rightarrow SS_6 \rightarrow \text{UGS} \rightarrow C_3 \). If the leaf of \( C_j \) exists in \( TF_5 \), its counter increases by one. In different case, the required path is created and the counter is set to one. The packet/request arrival procedure does not have to deal with the details of time frame sizes and structure. It simply places packet transmissions to the appropriate leaves.

Moreover, to differentiate conforming from non-conforming traffic, each packet placed in the tree is colored in accordance to the value of its connection’s throughput at the time of its arrival, compared to the values of mrtr and mstr for that connection. By using a colored discrimination of packets getting into the tree, the traffic is divided into three classes: i) red packets represent traffic higher than the maximum declared, ii) unpainted packets represent traffic between the minimum and maximum declared, and iii) green packets represent traffic lower than mean. Thus, for every arriving packet \( P_{ij} \) belonging to connection \( C_j \) we distinguish three cases:

i. If \( \text{Thr}(P_{ij}) < \text{mrtr}(C_j) \), the packet is colored red.
ii. If \( \text{Thr}(P_{ij}) > \text{mstr}(C_j) \), the packet is colored green.
iii. If \( \text{mrtr}(C_j) \leq \text{Thr}(P_{ij}) \leq \text{mstr}(C_j) \), the packet is uncolored.

2) Frame Map Creation
The frame map creation procedure is responsible to decide on the contents and structure of the next time frame, according to the information stored in the Frame Registry Tree. Let us assume that at the beginning of a time frame there are \( n \) time frame subtrees (\( TF_1, \ldots, TF_m \)). The next frame will be generated mainly from the first subtree (\( TF_1 \)), and in case of empty slots these can be filled with packets from the next subtrees (\( TF_2, TF_3, \ldots, TF_m \)). Three cases can be distinguished for \( TF_1 \):

i. The packets under subtree \( TF_1 \) exactly fit into the time frame. This is the best possible case, where the frame can be constructed without further processing, exactly as the leaves of \( TF_1 \) indicate.
The packets under subtree $TF_0$ are less than the capacity of the time frame. In this case, the algorithm can fill the empty space with packets from the next frames. It starts by making at most three passes in $TF_2$ moving from UGS to rtPS packets (from left to right in the subtree). First it chooses green packets, then unpainted ones and last it chooses red packets. If there are still empty slots in $TF_2$, the same procedure is repeated with the next subtrees (i.e. $TF_3$).

The packets under subtree $TF_1$ are more than the capacity of the time frame. In this case, the algorithm has to remove some packets from $TF_1$. It starts by making at most three passes in $TF_1$ moving from BE to UGS packets (from right to left in the subtree). First it removes red packets, then unpainted ones and last it chooses green packets. Every packet that is removed can be either dropped or moved in the next time frames, depending on its service type. UGS, ertPS and rtPS packets cannot be moved in the next subtree because they will expire, so they are dropped. Using a tree structure like the Frame Registry Tree, “moving” means a simple change of pointers and update of leaves. On the contrary, nrtPS or BE packets can be placed in $TF_2$.

Other procedures required in the overall scheduler operation, such as a modulation change, can be easily performed through simple subtree re-positioning.

IV. SIMULATIONS

In order to measure the performance of the proposed scheduler, a simulation program was constructed in C++. The program simulates the full operation of IEEE 802.16 PMP mode, as well as the C-FRTS and FRTS schedulers.

We considered a simulation scenario that is based on WirelessMAN-SC MAC system profile with multiple types of traffic per SS, including video, compressed and uncompressed voice, ftp and http, to evaluate different features of the standard and the scheduler. Although a specific bandwidth was agreed for each kind of service (mrtr and mstr), all connections were configured to create more that the admitted bandwidth except those of compressed voice. Table 1 shows the traffic QoS characteristics of each traffic type.

All connections of a certain type were considered independent and statistically identical. The time frame length was set to 1 msec, the packet size to 54 bytes and the modulation to 64-QAM for all SSs, leading to a transmission speed of 120 Mbps (as indicated in the standard). To limit the simulation into reasonable numbers, we assumed that only 50% of the bandwidth was available for the above traffic. Our intention was to show that the per-service type differentiated treatment provided by C-FRTS, together with its deadline-based scheduling and colored discrimination, can attain better performance by guaranteeing more QoS parameters.

In Figure 2, the throughput of the five different service types per SS for C-FRTS is depicted. For small numbers of subscribers, even exceeding traffic is served without any differentiation among service types. From 140 to 160 SSs, C-FRTS drops red packets starting from BE. Near to 160 subscribers, all service types enjoy throughput close to their agreed mstr. After 160 SSs, C-FRTS starts dropping unpainted packets up to mtrr starting from BE and moving towards UGS. For more than 260 SSs, the scheduler is forced to drop green packets, starting from nrtPS. For example, let us focus on nrtPS. Table 1 shows that the mean input traffic per connection for nrtPS equals to 120Kbps, whereas mrtr and mstr equal to 60 and 110Kbps respectively. Up to 140 SSs, all created traffic is served. From 140 to 190 SSs only traffic equal to mstr is served. From 190 to 220 SSs the throughput drops linearly to mtrr and remains stable until all higher service types reach their mtrr value as well. Above 260 SSs, the traffic is so high that the scheduler is forced to service less than mtrr for nrtPS.

Figure 3 shows the throughput per SS for FRTS in the same scenario. Both algorithms have almost the same performance in total throughput. What differentiates FRTS from C-FRTS in relation to throughput is that it gives most of the bandwidth to higher priority service types, without being able to control excess traffic, leading lower priority ones to starvation very quickly. This is mainly due to the strict priority policy it follows and the lack of a mechanism to support the agreed QoS parameters. Thus, for more than 170 SSs, very few BE packets are served, and the same happens to nrtPS after 240 SSs. On the contrary, C-FRTS performs according to bandwidth agreements for all service types, respecting all their QoS characteristics. Thus, even for large numbers of subscribers, the lower priority service types enjoy a reasonable service.

Finally, Figures 4 and 5 show the mean and maximum delays of all five service types using C-FRTS and FRTS respectively. As both algorithms use the same delay-based logic for classes with a given maximum latency (UGS, ertPS, rtPS), their performance is similar for these classes. On the other hand, for classes with no given latency (nrtPS and BE), the performance of the two algorithms is differentiated, as a result of their operation. Figure 4 shows that all service types enjoy reasonable mean and maximum delays with the use of C-FRTS for larger number of SSs, in contrast to Figure 5 that shows a rapid increase of nrtPS and BE delays even for small number of SSs when FRTS is used.
V. CONCLUSIONS

The scheduler proposed in this paper, referred to as C-FRTS (Colored Frame Registry Tree Scheduler), aims at guaranteeing the maximum latency required by delay sensitive service types, by transmitting packets in the last time frame before their deadline. At the same time, lower priority conforming traffic is protected against higher priority excess traffic through a coloring discrimination. Additionally, the scheduler distributes the required processing over time, without significantly increasing memory requirements. A notable advantage of the tree structure used by the scheduler is its scalability to support future changes of the standard. For example, it can easily support more QoS services, or more modulation types that may be added in the future.

REFERENCES