Parag Mogre, Matthias Hollick, Ralf Steinmetz, Viraj Dadia, Somnath Sengupta:
Distributed Bandwidth Reservation Strategies to Support Efficient Bandwidth Utilization and QoS on a Per-Link Basis in IEEE 802.16 Mesh Networks. In: Proceedings of the 34th IEEE Conference on Local Computer Networks (IEEE LCN 2009), p. to appear, IEEE, October 2009.

Distributed Bandwidth Reservation Strategies to Support Efficient Bandwidth Utilization and QoS on a Per-Link Basis in IEEE 802.16 Mesh Networks

Parag S. Mogre, Matthias Hollick, and Ralf Steinmetz
Email: parag.mogre@kom.tu-darmstadt.de
Multimedia Communications Lab (KOM),
Technische Universität Darmstadt, Germany

Viraj Dadia and Somnath Sengupta
Electronics & Electrical Communication Engineering Dept.,
Kalpana Chaola Space Technology Cell,
Indian Institute of Technology, Kharagpur, India

***Abstract*—The IEEE 802.16 standard specifies a MeSH mode of operation which permits the setup of Wireless Mesh Networks (WMN) with per-link QoS support. The standard specifies both distributed as well as centralized reservation schemes. Distributed scheduling is highly flexible, and enables operation of the WMN even in the absence of a central controlling instance or base station. A systematic study of strategies for distributed scheduling in the IEEE 802.16 MeSH mode is, however, missing. In this paper we model the individual links in the 802.16 WMN and design and derive efficient strategies for distributed scheduling to reserve bandwidth required for transmission on the modelled link. Additionally, we evaluate our proposed reservation model using simulations, study the impact of key parameters and identify issues for further research in WiMAX based WMNs.**

I. Introduction and Motivation

State-of-the-art standards supporting mesh topologies for forming Wireless Mesh Networks (WMNs) have proposed Time Division Multiple Access/Time Division Duplex (TDMA/TDD) based reservation mechanisms to support QoS and next-generation multimedia applications. Some examples of such standards are the IEEE 802.16 standard's MeSH[1] mode of operation, the IEEE 802.11s MDA mode of operation, and the WirelessHART standard. In this paper we will focus on the IEEE 802.16 standard's MeSH mode, and its distributed bandwidth reservation schemes in particular.

Although the MeSH mode specifies the protocols and primitives for bandwidth reservation, the implementation details and the framework for bandwidth reservation in order to support QoS is left open to permit vendor optimization. To the best of our knowledge till date there is a lack of a systematic study of distributed bandwidth reservation strategies for the MeSH mode (see Sec. II for a brief overview of the MeSH mode, further details are presented in [2]). Especially, detailed and systematic study of distributed bandwidth reservation strategies and their implications for QoS support and bandwidth utilization for the MeSH mode are missing. In this paper we address the above gap. Our contributions are as follows:

- We present a system model (Sec. III) for the distributed per-link bandwidth reservation process.
- We design exemplary distributed bandwidth reservation strategies using the developed system model (Sec. IV).
- We present an optimized bandwidth reservation strategy which supports provision of class based QoS, similar to the 802.16 Point to Multipoint (PMP) mode (Sec. IV).
- We evaluate the designed bandwidth reservation framework and reservation strategies via a simulation study (Sec. V) providing useful insights for future work.

[1]In this paper we will use the notation MeSH to refer to the mesh mode of operation of the IEEE 802.16 [1] standard

II. Overview of Distributed Scheduling

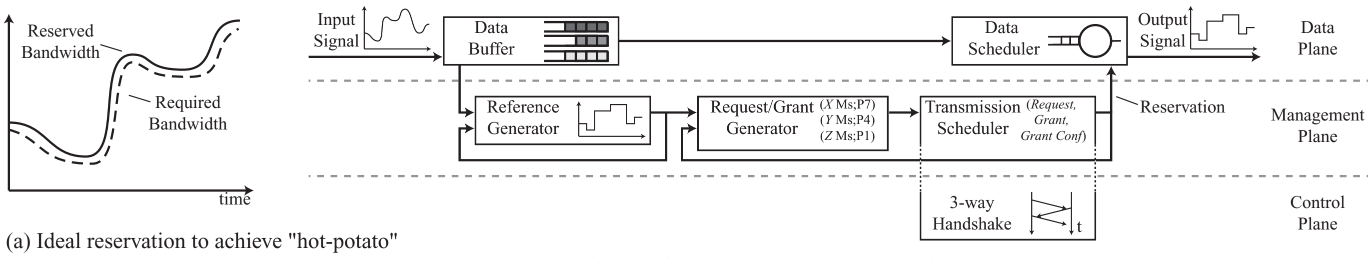
The MeSH mode uses TDMA/TDD to arbitrate medium access. Time is thereby divided into frames, where each frame is divided into a control subframe (for reservation control messages) and a data subframe. The IEEE 802.16 MeSH mode specifies both centralized (limited to a tree rooted at the mesh base station) as well as distributed means of bandwidth reservation or scheduling. We focus only on distributed scheduling which is of more relevance to wireless mesh networks as compared to the centralized scheduling scheme.

With distributed scheduling, to reserve bandwidth for a link, nodes use a three-way handshake. To compute conflict free schedules, nodes associate a slot status with each slot and use the control (handshake) messages overheard from their neighbours as well as generated by themselves to update the slot states to reflect the scheduled transmissions. Bandwidth reservations are for a range of slots for a range of frames, where the number of frames for a reservation is chosen from a range of permitted values. Access to the control subframe is contention free and is regulated by a distributed mesh election algorithm. For further details of the MeSH mode see [2], [1].

III. System Model for Per-Link Reservation

As discussed previously, the standard [1] provides the protocol messages required for minislot reservation, but leaves crucial issues unanswered like when, how and how much bandwidth to reserve on a link according to data arrivals. To tackle this, we present a single link model (Fig. 1) for designing bandwidth reservation strategies.

Ideally, to support real time flows, the bandwidth reservations in each frame should be equal or exceed the required demand (see Fig. 1 (a)). As shown in Fig. 1 (b), in the data



(a) Ideal reservation to achieve "hot-potato" forwarding for real-time traffic

(b) Model for distributed bandwidth reservation for a single link in the WMN

Fig. 1. Idealized bandwidth reservation scenario for supporting real-time services and a system model for distributed bandwidth reservation

plane, we model data-bytes arriving for transmission at the MAC layer on the modeled link as a discrete time series with frame-level granularity (*input signal*). This data is buffered in the *data buffer* before transmission by the *scheduler*. The *scheduler* transmits this data according to reservations done in every frame (giving the *output signal*).

In the control plane, the 3-way handshake takes place to reserve minislots. But decisions about how many minislots to reserve, and when, are taken in the management plane. To reduce transmission latencies for real-time data, we propose proactive requesting, i.e. reserving minislots before actual data arrivals, according to a reference signal. The *reference generator* provides this reference signal based on past arrivals and current buffer size information. In an ideal scenario this reference would perfectly represent the future arrivals and also never allow buffer-overflow (see Fig. 1 (a)). Hence, this block plays a vital role in measuring efficiency of the model in terms of bandwidth usage and delay (or data latency). Depending on this reference, the *request/grant generator* block takes decisions of how to generate requests in terms of persistence levels and minislots, and confirm grants. The *transmission scheduler* is responsible for transmitting the generated grant-confirms, grants and requests with the required availabilities. In a multi-link scenario, this block also decides the link for which the available transmission opportunity is used.

We propose an efficient bandwidth reservation policy for distributed scheduling in the MeSH mode using the model presented in the previous section. Due to space limitations, we provide only an overview of the functional blocks, details can be found in [3]. The functional blocks of the proposed *differential bandwidth requester* (DBREQ) (Fig. 2) provide functionalities corresponding to the *reference generator* and the *request/grant generator* blocks in Fig 1. DBREQ classifies traffic arriving at the MAC layer for transmission on a link into different service classes (QoS classes) similar to those in the PMP mode. This classification permits different scheduling schemes as well as different reference generators to be used in combination with different bandwidth reservation policies tailored to each traffic class.

We classify bandwidth requesting policies into *proactive* and *reactive* policies respectively. In proactive policies, the bandwidth requests are sent in advance in anticipation of future data arrivals so that the bandwidth is available when the data arrives for transmission (useful for real-time services). With reactive bandwidth request policies, bandwidth reservation requests are sent as a reaction to data arrival for transmissions involving additional control latency (sufficient for non-real time services). Considering the above design criteria, as shown in Fig. 2, different reference generators are used for the different service classes. These are briefly outlined next.

For predicting aggregate UGS arrivals we use a simple *proactive forecaster* which uses the average traffic arrival rate for the past few (m) frames as the future expected arrival rate for the next few (n) frames. For the presented simulations we chose m to be 25 to allow quick response and n to be 150 so that the value is greater than all the finite request persistences in the standard to allow ease of packing (explained later). For self-similar real-time traffic aggregates (e.g. rtPS traffic) we used a combination of Haar-Wavelet filter bank and a *linear autoregressive predictor* (see [4]) (of order p=100) with a inverse Haar-Wavelet (see [5]) filtering (see Fig. 3) to predict the traffic arrivals for the next few frames. This allows efficient use of a simple linear prediction scheme to predict arrivals over a number of future frames (256). For a detailed explanation of the reasons for the design as well as the finer details readers are referred to [3].

IV. Differential Bandwidth Request Framework

Both the above reference generators will produce a signal, which can be looked at as a curve specifying the amount of bytes of traffic arrival expected for the look-ahead frames in

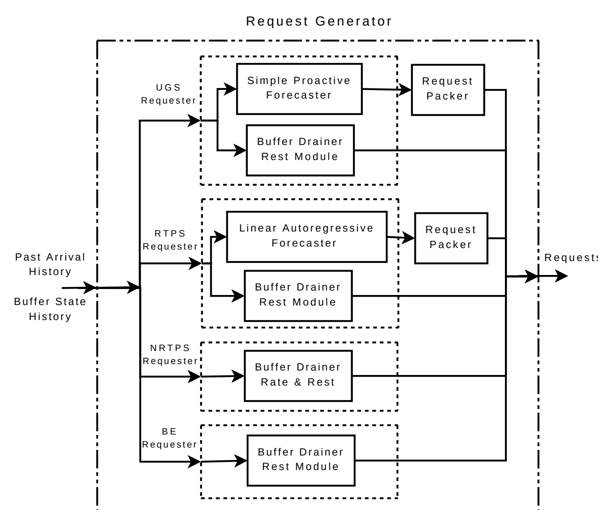


Fig. 2. Proposed Differential Requester

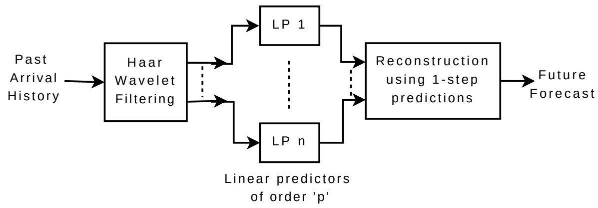


Fig. 3. Combined Haar Wavelet filtering with 1-step Linear Predictor

the future. This is used as an input by the *request packer* in Fig. 2 to produce a set of requests which will satisfy the estimated future demand. The request packer tries to tessellate the area under the predicted demand curve using rectangles of length (demand persistence, number of frames) and width (number of minislots) permissible within the scope of the MeSH mode. Details of the tessellation algorithm used by the request packer are out of scope of the current discussion and may be found in [3]. The aim of the packing algorithm is to cover the demand with as few rectangles as possible as each rectangle corresponds to a request which must be transmitted in the MSH-DSCH message.

For non-real-time traffic (BE/nrtPS) we use reactive requesting policies with the reference (required reservation) being generated by analysis of the data buffer. Here, based on the traffic class we use a suitable combination of polices which consider either just the bytes in the data queue, or also look additionally into the rate of change of the buffer occupancy over the past few frames when generating the requests. Both the above aim at draining (transmitting) the bytes which have already arrived and are termed *buffer drainer rest/rate* functional blocks respectively.

Finally, at the granter we aim to grant as much of the requested bandwidth as possible. Once the node has bandwidth reserved for transmissions on a link, this information is used by the data scheduler to actually transmit data on the link. We use a simple hierarchical data transmission scheduler, which gives equal priority to both the real-time service classes (UGS/rtPS). Bandwidth leftover from their reservations is then used in a strict priority order: UGS, rtPS, nrtPS, BE. Bandwidth reserved for a particular service class is used primarily for data belonging to that service class.

To enable benchmarking of the performance of the above, we use the following bandwidth request policies:

- *oracle*: It is an "ideal" *differential requester* where the reference generators give perfect forecasts.
- *simple proactive*: is similar to the simple proactive bandwidth requester discussed for the UGS class in the Differential requester, only here, the requester is responsible for the entire traffic (for all types of service classes) and does not differentiate the arriving traffic.
- *simple reactive*: is similar to the buffer drainer rest module discussed for the differential requester and is a prototype for a simple reactive bandwidth request strategy, it too does not differentiate traffic classes and uses a simple FIFO data scheduler.

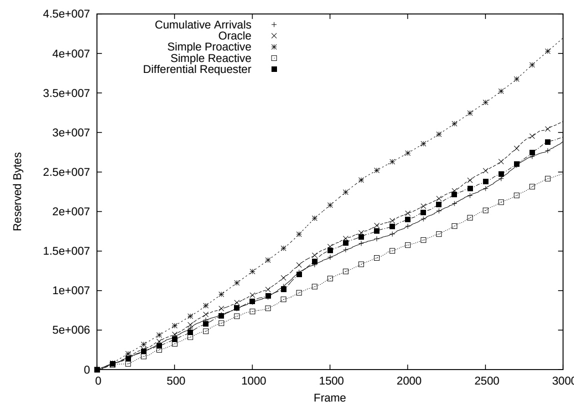


Fig. 4. Cumulative bandwidth reservations and cumulative aggregate traffic arrivals for a sample run for the High-High scenario

V. Evaluation

The two main criteria we use to evaluate the proposed bandwidth reservation policy are utilization (bandwidth used/bandwidth reserved) and delay (delay per hop at the MAC layer).

To evaluate our framework, we developed a custom, standard conform simulator for the IEEE 802.16 MeSH mode. All simulations are run for a total of 3000 frames, with each simulation setup being run 50 times with different random seeds to obtain statistically valid results. Due to space limitations we will present here only a selected set of results. Additional results for different setups and different network configuration parameters can be found in [3]. For presented simulation scenarios with the assumed configuration parameters (for details see [3]) we get a maximal data rate of approximatively 4.95 MB/s. The duration of a frame (control + data) is 10 ms with the data subframe having a length of approximately 8.81 ms. The data subframe has 172 minislots where 288 bytes can be transmitted per minislot, the control subframe has 10 transmission opportunities with the default modulation. We assume a buffer capacity of 1 MB for enqueueing all the packets arriving at a link for transmission before they can be transmitted. The focus of the evaluation will be at the level of a single link in the WMN, this link is termed as the primary link or link of interest. For all the simulations presented here, we chose a controlled representative (realistic) mix of traffic on the primary link. The aggregate traffic is composed of traffic which can be classified as belonging to each of the scheduling services UGS, rtPS, nrtPS and BE (see Sec. VI and [1] for details about the scheduling services). For simulations we used the following primary traffic mix (giving approximately on an average total primary traffic arrival rate of 7.6925 Mbps):

- UGS traffic: an aggregate of between 1-4 T1/E1 CBR flows showing on-off behaviour, with the on/off times for the flows randomly selected (average rate approx. 1.903 Mbps).
- rtPS traffic: the trace data for the MPEG stream news_.IPB (around 2.688 Mbps) (source see [6]).
- nrtPS traffic: the network traffic trace file BC-pAug89.TL (around 1.4016 Mbps) (source see [7]).

TABLE I
HIGH-HIGH SCENARIO: UTILIZATION AND AGGREGATE TRAFFIC DELAY.

Request Generator	Utilization (%)		Overall Delay (ms)	
	Mean	95% Confidence	Mean	95% Confidence
Differential Requester	95.15	0.61	349.87	10.59
Oracle	88.14	0.74	117.13	6.77
Simple Proactive	70.49	0.40	50.50	6.97
Simple Reactive	100.00	0.0	938.29	12.47

TABLE II
HIGH-HIGH SCENARIO: PER-CLASS DELAY (IN MS).

Traffic class	Differential Requester		Oracle	
	Mean	95% Confidence	Mean	95% Confidence
UGS	61.89	10.79	7.10	1.24
RTPS	45.24	1.91	18.97	0.66
NRTPS	189.75	9.90	52.10	4.36
BE	931.44	32.38	335.05	20.29

- BE traffic: the network traffic trace file BC-pOct89.TL (around 1.699 Mbps) (source see [7]).

Additionally, we consider also background traffic in the neighbourhood of the primary link. Due to space limitations we only present results for the High-High scenario (i.e. high number of neighbouring nodes for the primary nodes + high traffic on the neighbouring links, for additional results see [3]). Consider the primary link to be link (N_1,N_2) between nodes N_1 and N_2. In the latter scenario both these nodes have 4 common neighbour nodes and 5 exclusive neighbour nodes. Data connections between the background nodes are chosen at random based on the random seed, and the background traffic is randomly generated such that for this scenario between 10–15 Mbps of background traffic exists during the period of the simulation. From Table I we see that all proactive requesting policies perform better than the reactive bandwidth request policy as far as the delay is concerned. This is due to high control latency for the reactive schemes. In fact as seen in Fig. 4 the cumulative bandwidth reservations for the *simple reactive* bandwidth requesting policy is never able to catch up with the cumulative primary traffic arrivals. The figure also shows that the proposed *differential request* framework's bandwidth reservations are able to closely follow the cumulative data arrivals, thus supporting the forecast (reference generation) and bandwidth reservation architecture. One interesting aspect to be noted from Table I is that the utilization of the *oracle requester* is less than that of the *differential requester*. The cause for this is exact requesting by the oracle, which, if some requests are not granted leads to traffic backlog, which leads to excessive reservations by the buffer drainer leading to wasted bandwidth.

Table II presents the delays in ms for each traffic class for the *oracle* and the *differential requesters*. We do not present per class results for the *simple proactive*, and the *simple reactive* bandwidth requesters as these do not classify and differentiate traffic into different service classes. One can see that the *oracle* due to its perfect forecast has a better delay performance for each service class, but pays for the low delay with lower bandwidth utilization.

VI. RELATED WORK

IEEE 802.16 based WMNs are expected to support a mix multimedia traffic requiring QoS. To maintain QoS class compatibility to the PMP mode of operation, we classified the traffic arriving at the MAC convergence sublayer (see [1], [2] for details) into the same service classes as the PMP mode (e.g. using the IP TOS field). Traffic seen in WMNs on a link is essentially an aggregation of multiple traffic sources as a result of the multihop forwarding of traffic, and is bursty in nature due to possible route changes, flows entering and leaving the network, etc. Traffic in general in such networks is known to have self-similarity properties [8]. This environment is very challenging for any bandwidth reservation framework. Bandwidth reservation policies should strive to maintain a balance between high network utilization and low latency [9]. The authors in [9] conclude that dynamic algorithms allowing periodic adjustment of bandwidth allocation (which is very much supported by our proposal) have a clear advantage over static algorithms. To support real-time services in WMNs, control latency needs to be overcome; this can be done via traffic modelling. However, most of these models are not suitable for the MeSH mode due to their high complexity, the differing time granularities or the peculiarities of the MeSH mode. There is a lack of detailed study of reservation policies for distributed scheduling in the MeSH mode in the literature with most of the work focusing on the control subframe. This paper address the above gap, and has presented a model for designing flexible bandwidth reservation policies able to support carrier-grade QoS in the MeSH mode.

VII. SUMMARY AND CONCLUSION

The proposed *differential bandwidth requester* shows excellent bandwidth utilization and at the same time is able to provide low delays for real-time traffic. It is only slightly outperformed by the ideal *oracle bandwidth requester* which is not possible to realize in practice. In future we will investigate aspects such as the fairness and also look at the end-to-end delays and bandwidth utilization over multiple hops.

REFERENCES

[1] IEEE, "802.16 Standard," IEEE Std. 802.16-2004, October 2004.

[2] P. S. Mogre et al., "The IEEE 802.16-2004 MeSH Mode Explained, KOM-TR-2006-08," KOM, TU Darmstadt, Germany, ftp.kom.tu-darmstadt.de/pub/TR/KOM-TR-2006-08.pdf, Tech. Rep., 2006.

[3] V. M. Dadia, "Bandwidth Reservation Strategies for Distributed Scheduling in the IEEE 802.16 Mesh Mode," Master's thesis, Indian Institute of Technology Kharagpur, 2008.

[4] J. Makhoul, "Linear Prediction: A Tutorial Review," *Proceedings of the IEEE*, vol. Volume: 63, pp. 561– 580, 1975.

[5] I. Daubechies, *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992.

[6] O. Rose, "Statistical Properties of MPEG Video Traffic and Their Impact on Traffic Modelling in ATM Systems," University of Wuerzburg, Tech. Rep., 1995.

[7] H. J. Fowler and W. E. Leland, "Local Area Network Traffic Characteristics with Implications for Broadband Network Congestion Management," *IEEE JSAC*, vol. 9, pp. 1139–1149, 1991.

[8] M. S. Taqqu et al., "Estimators for Long Range Dependence: An Empirical Study," *Fractals*, vol. 3, pp. 785–788, 1995.

[9] Y. Afek et al., "Dynamic Bandwidth Allocation Policies," in *IEEE Infocom*, 1996.