A PRAGMATIC APPROACH FOR QOS IN WIRELESS MESH NETWORKS

Andre Herms and Georg Lukas
Department of Distributed Systems
University of Magdeburg, Germany

Keywords: Wireless Mesh Network, IEEE-802.11, QoS, clustering.

Abstract: In this paper we present a QoS routing protocol for IEEE-802.11 based mesh networks. The main challenge for providing QoS in terms of bandwidth and latency is that the medium is shared between all nodes in close range, which complicates reservation of medium time. Furthermore, the use of standard compliant hardware components requires an integration of the existing medium access mechanism, which is designed for best-effort communication only. A cluster-based structure is used for representing the local domains of the shared medium and allowing reservation of medium time. On top of this reservation an optimistic reactive algorithm is used for discovery and reservation of routes that fulfil the application specified QoS requirements. Simulation results are presented that prove the correctness of this approach.

1 INTRODUCTION

Wireless LANs based on the IEEE 802.11 standard (IEEE-802.11, 1999) are widely used for wireless communication. Based on this, mature multi-hop wireless routing protocols like AODV or OLSR (Liu and Kaiser, 2003) exist and the upcoming IEEE 802.11s standard promises wide availability of multi-hop technology. These multi-hop networks allow efficient deployment and good usability due to their self-organization abilities. Besides of typical applications like web browsing, services like video streaming or VoIP are commonly used today.

For use in automation the Wireless LAN Standard seems to be a promising choice. It allows wireless connectivity in a free frequency band, has a sufficiently large communication range, and allows relatively high data rates. Furthermore, a huge number of products exist that can be easily integrated into existing systems. The self-x properties of mesh routings – self-organizing, self-configuring, self-healing – would allow further flexibility and better deployment of such systems. However, applications in automation typically require QoS, in terms of bandwidth guarantees and timely delivery. This does not necessarily mean a maximal average throughput of all connection. Instead, the packet loss and latency of individual packets should be limited by a known enforcible bound.

Many approaches exist that aim to provide QoS in wireless mesh networks (see section 7). However, these are either based on different wireless technology or require extensions and modifications that are incompatible with the Wireless LAN standard. Thus, they cannot be used with hardware off the shelf. In this paper we present a pragmatic approach for providing QoS in wireless mesh networks, which allows the use of standard hardware. It requires only a corresponding routing software. The protocol has to deal with the behavior of standard compliant hardware that was not developed to provide QoS at all. Of course, this requires some compromises and workarounds so that it will not outperform other, specialized solutions. The main contributions of this paper is that it shows the possibility of using standard hardware for QoS routing in MANETS. Furthermore, various aspects of the medium access are discussed that are crucial for the provision of QoS.

The rest of the paper is organized as follows: Section 2 describes the principles of medium access in wireless networks and their consequences for the spatial separation of the shared medium. The principle of operation is explained in section 3. In section 4 our clustering protocol is presented. This is followed by a

Herms A. and Lukas G. (2007).
A PRAGMATIC APPROACH FOR QOS IN WIRELESS MESH NETWORKS.
In *Proceedings of the Second International Conference on Wireless Information Networks and Systems*, pages 57-64
DOI: 10.5220/0002147000570064

description of our routing protocol in section 5 and its evaluation in section 6. The paper ends with a presentation of related work in section 7 and a conclusion in section 8.

2 MEDIUM ACCESS IN MANETs

In opposite to modern wired networks, which use point-to-point duplex links, the wireless medium is shared between the participating stations. The MAC layer is responsible for managing access to the medium and the IEEE 802.11 standard uses a CSMA/CA mechanism, which is based on carrier sensing:

Before a station starts sending, it checks if there is an ongoing transmission on the medium and if necessary defers the packet start. Two different carrier sense strategies are combined for this – a physical carrier sense and a logical carrier sense. The logical carrier uses the transmission duration entry of packets it receives. These are stored in a register and compared to an internal timer. Combined with a four-way handshaking (RTS/CTS/DATA/ACK) it can even be used to prevent the hidden station problem (Tanenbaum, 2002).

The physical carrier is sensed using the receiver unit for detecting the carrier signal and setting an internal busy state. This mechanisms even detects an ongoing transmission when the packet cannot be correctly received due to bit errors. In experiments with standard hardware Mahrenholz was able to show that the range where the carrier can be detected is about twice as large as the communication range(Mahrenholz, 2006). In order to prevent that multiple stations start a transmission immediately after the medium becomes available, a random back off is used. Similar to the Ethernet standard, the time interval it is selected from is exponentially increased, when packet loss happens. This is detected by the missing ACK of unicast data packets.

As our protocol uses standard hardware, the medium is still controlled by this CSMA/CA mechanisms. We do not define a new MAC layer, but try to cope with the existing one. In particular, this means that packet collisions are already prevented and not subject to further considerations. The main reason for violation of QoS requirements in MANETs are congestion effects and packet losses. While packet losses are compensated by local retransmissions in the link layer, congestion must be prevented by the routing. It is not necessary to create an exact schedule to prevent concurrent transmissions, because this is already done by the MAC layer. Instead, it must ensure that the medium time consumed by a station, its neighbors, and its two-hop neighbors does not exceed the maximum available one.

3 PRINCIPLE OF OPERATION

The protocol consists of two sublayers, the cluster layer and the global routing layer. The aim of the cluster layer is to provide a QoS-capable medium access. It allows local reservation of bandwidth, is responsible for local retransmissions, and has a predictable packet delay. In order to achieve this, it creates local domains of control, called clusters, which are maintained by a local coordinator, the cluster head. The cluster head initiates medium access in the cluster by polling other stations. This is very similar to the point coordination function (PCF) of the WLAN standard. However, we operate in the ad hoc mode, where such functionality is not available. Efficient retransmissions and local two-hop routing are provided, based on the corresponding microprotocols presented in (Nett and Schemmer, 2003).

Above the clustering sublayer a routing sublayer is used, which is tailored to the cluster structure. We use a proactive link state routing. Three classes of traffic are available, control traffic, best effort traffic, and QoS traffic. Because of the strict scheduling of control messages with the lowest priority, QoS streams are never affected by control traffic. The routing sublayer uses a hop-by-hop reservation of routes. Bandwidth is reserved per cluster and the deterministic delay in the clusters allows an approximation of the worst-case packet delay of the path. An appropriate path is discovered by an optimistic path search, which tries to use the shortest path. If it fails in case of insufficient resources on a hop, alternatives are tried by local, backtracking-like rerouting of the reservation requests. If a path with sufficient resources exists, it is often found by the first reservation. If the fist try of the shortest path is not usable, a different usable path will be discovered. This helps to reduce the number of required control messages.

4 CLUSTERING

In a typical WLAN infrastructure, two classes of stations exist – Access Points and Clients. In an ad hoc mesh network all stations have the same role. They act as router and communication endpoint at the same time. We create a spatial partitioning of the network into local domains (clusters) by assigning roles to the stations. A station can either be a *cluster head* or a *cluster client*. Every cluster head has a set of clients in communication range. A client can belong to multiple clusters. In such a case it can act as gateway and allows routing between clusters. A cluster head never has other heads in communication range. An example of such a structure is depicted in figure 1.

The clustering subprotocol is responsible for assigning these roles. It uses a periodic cluster an-

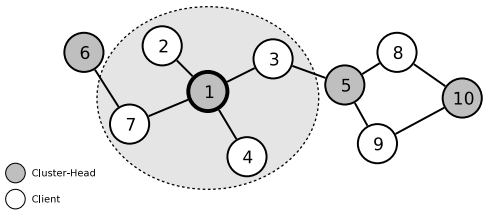

Figure 1: Cluster Structure.

nouncement packet, which is transmitted periodically by every head and contains the list of all clients. The following rules are applied:

1. A client that receives a cluster announcement from a head, it is not associated with, adds the cluster to its set and replies with a join packet. Typically this is received by the head and the client is included in the cluster.
2. A client that missed a defined number of announcements removes the head from its set of clusters.
3. A client that has no associated clusters changes its role and becomes a cluster head.
4. A cluster head that receives announcements from another head, changes its role to a client.

The resulting structure has the following properties: A client node has only one-hop links to head nodes, and vice versa. Every two-hop link between two nodes has endpoints of the same role. A two-hop neighbor of a client is always a client, which is connected by a head node. Thus, both client nodes belong to a common cluster defined by the intermediate head node. Two clusters can have common client nodes that serve as gateway.

4.1 Timing

The medium access in a cluster is initiated by the cluster head. It polls its clients with packet `Poll[x]`, whereupon it must be answered by a new data packet `D[x]`. The data packet is not guaranteed to be received by the other clients in the cluster. Therefore, it is repeated by the cluster head. As this packets is followed by the poll for the next client, both packets are combined. The head is allowed to communicate by polling itself (`PollH`). An example is shown in figure 2. Every such packet pair consisting of the repetition by the head including the poll and the data packet from the client defines one *slot*. The slots have a fixed duration l. The number of stations per cluster is limited by a predefined value C. The duration of $C+1$ slots is one round. The head chooses a schedule that guarantees every station to be polled at least after C slots, once per round. Clients can request more slots per round, by sending a corresponding request

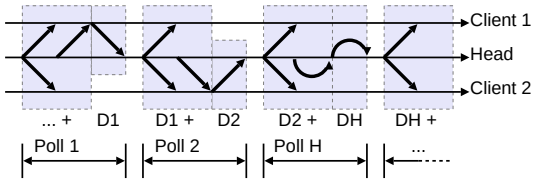

Figure 2: Polling scheme of a cluster.

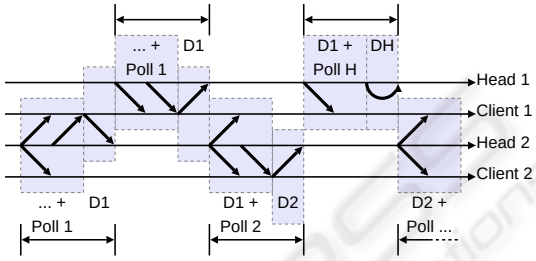

Figure 3: Example schedule with $H = 2$.

in response to the heads cluster announcement packet. One slot per round is reserved for this packet.

Clients can belong to multiple clusters and receive all packets from each of them. Therefore, it is not possible for a cluster head to use the whole medium time. This is prevented by the carrier signal from adjacent cluster heads, which are at most in two-hop range. Its transmissions of poll packets would be deferred and the output queue would fill up and overflow. In order to prevent this effect, the slots are widened by a factor of H, which corresponds to the maximum number of clusters with a common client. It can be easily shown that for stations with equal communication ranges arranged in a plane, a value of $H = 7$ is sufficient. An example schedule of two clusters with a common client (client 1) and $H = 2$ is shown in figure 3.

The strict timing allows to determine the delay of packets and further the detection of topology changes in a bounded time. The polling period of a client is defined as CHl. When allowing a maximum number of d lost equal packets, a client detects the loss of a cluster head in $dCHl$ and a cluster can detect the loss of a client in $2dCHl$.

4.2 Preventing Exponential Backoff

The exponential increase of the backoff in the WLAN MAC layer is used to reduce the collision probability during the contention based medium access. However, it also adds randomness to the transmission start, which leads to unpredictable latencies. Our protocol uses only broadcast packets that are never acknowledged by the receiver and thus never increase the backoff window. As a consequence, packet loss cannot be detected by the MAC layer anymore and no local retransmission are done. The *Atomic Multicast* in

(Nett and Schemmer, 2003) provides a *Dynamic Time Redundancy microprotocol* that allows fault tolerant and time bounded delivery to a group of stations. It uses an omission degree that corresponds to the previously defined parameter d. We use this microprotocol for fault tolerant packet delivery to all stations in the cluster. It requires a maximum duration of $2dCH$ slots lengths per packet.

4.3 Properties

The clustering protocol has the following properties:

(1) The medium time is divided into slots that respect the two-hop range of carrier sense: All one-hop neighbors belong to a common cluster. Two-hop neighbors are either both clients in a common cluster or both heads of adjacent clusters. In a common cluster the polling scheme controls the medium access. For adjacent clusters the additional time between cluster slots prevents that the available medium time is exceeded.

(2) Medium time can be reserved by requesting additional slots from the cluster head. Consensus about the assigned slots in a cluster is guaranteed because it is coordinated by a single station, the head.

(3) Packets are reliably transmitted to all nodes in a cluster, as long as the cluster is not disbanded. A transmission reaches all nodes of a cluster, even the clients with a distance of two hops. This requires at most $2dCH$ slot lengths, which corresponds to the worst case latency in cluster. Furthermore, the guaranteed bandwidth of a client can be derived from the maximum packet size, the worst-case transmit duration, and the number of assigned slots in a cluster. This is used for reservation.

(4) The communication is time-triggered. Every station in the network has its exclusive slots that must be used for communication. Only the announcement slot of a cluster can be used by multiple stations. The cluster head polls all members of a cluster and thereby controls when a station has to transmit.

5 ROUTING

The purpose of a wireless mesh network is to allow nodes without a direct link to exchange data. To achieve this, the nodes need to obtain knowledge about the network structure, either as *distance vectors* to other nodes or as *link state* data about the topology. This information can either be gathered *on demand*, when a connection is requested or in a *proactive* fashion during regular operation. This paper presents a proactive link state based approach, which allows faster path searches and the calculation of alternative routes around congested areas.

The following section describes a topology propagation protocol, which exchanges global link state data between interconnected nodes. This information is used by a best effort routing protocol described in section 5.2 to allow data exchange between nodes not directly connected. The routing capability is used in combination with the topology data for a directed search of suitable QoS paths and for their monitoring by a protocol described in 5.3.

5.1 Topology Propagation

Because every node only has data about its own direct neighborhood, a cooperative approach is needed to obtain information about the whole network, which then can be used for routing purposes. To achieve this, the neighborhood data, which is provided by the clustering module on every node, must be propagated to other stations and stored there. For storage a local link graph is used, which contains representations of nodes and lists of their respective neighbors.

Information about the network topology is transmitted via broadcast packets to all neighbors. Every such packet contains a list of elements, each consisting of a node's ID, a sequence number and that node's list of neighbors. When a node is not listed in the local link graph or the transmitted sequence number is higher than the last known one, the receiver merges the element from the datagram into the local topology database.

The propagation scheme is similar to the Fisheye State Routing protocol (Pei et al., 2000). Every node has up-to-date information about its neighborhood, while its knowledge about distant links is older, and thus less accurate, because topology transmissions from these nodes take more time. On the other hand, specific knowledge is only needed for routing when the target is near, for distant targets it is sufficient to have the approximate direction, because the following gateway nodes have more current link state data and are able to direct the packet.

A node has to send an answer every time it is polled by the head, even when no application data is available for transmission. These *idle* time slots are used for topology propagation. Thus, the world models are most accurate when the network is not heavily used, but the effect of traffic load on the propagation speed is only moderate, as will be shown in section 6.1.

Because a single packet can only carry information about approx. 30 nodes and their neighborhood, a sophisticated algorithm is used to select parts of the world model for transmission. This algorithm prefers updated information while still preventing starvation of older data. It allows a newly connected node to be informed about the whole network topology after a relatively short time while tolerating packet loss,

which makes the protocol a reliable base for routing.

5.2 Best Effort Routing

To allow communication between nodes not directly connected, a multi-hop routing mechanism must be provided. This mechanism can be used not only for best effort communication, it is also required for the reservation and monitoring of QoS links.

Because every head reliably retransmits data packets to all clients in its cluster, only gateway nodes (which are member of two or more clusters) can be used for routing. Thus, before transmitting a packet, a node has to determine the cluster and the corresponding local gateway leading to the destination. To achieve this, the Dijkstra algorithm is applied to the network topology stored in the link graph, and the packet is sent to the gateway on the shortest path. On the next node the same procedure is applied, where the packet is routed to the next cluster according to its local link graph.

Because a gateway node can only send a packet into a cluster after being polled by the head, packets crossing cluster boundaries must be cached by the gateway until they are requested. On highly utilized links this can lead to congestion, when several different packets are sent over one gateway to one target cluster. This congestion is acceptable for best effort traffic, but must be avoided for QoS connections. The next section describes in detail how this is accomplished.

5.3 QoS Routing

Several requirements must be met to allow QoS communication between distant nodes. First, the application must be enabled to specify the target and its QoS requirements. Then, a path must be found through the network, which fulfils the requested parameters. A bandwidth reservation must be performed on this path and a monitoring mechanism has to continuously check if it is still valid.

To create a QoS path, the application specifies the required bandwidth and latency. These values are mapped to the number of slots to reserve (bandwidth) and maximum hop count (latency). The QoS protocol then uses the link graph to calculate a shortest path to the destination and requests the specified number of slots in the corresponding cluster. If the request is granted, a packet is forwarded to the next gateway, which again makes a path calculation and a bandwidth request. Thus, the request packet is forwarded from the requester to the target and the required bandwidth is reserved on the whole path. If the request cannot be granted in one cluster (because all time slots are already reserved), the corresponding gateway performs the following steps: first, it puts the cluster into a *black list* in the request packet. Then, it makes a new path search with the blacklisted cluster(s) excluded from the link graph and reserves the bandwidth in the newly calculated target cluster. If no alternative route can be found, the gateway signals its predecessor to blacklist the current cluster and to dereserve the previous bandwidth request.

This optimistic reactive approach results in a distributed backtracking algorithm that finds a relatively short route through clusters with enough bandwidth to allow the requested communication channel.

Every node maintains a table of all QoS paths that are routed through it. This table contains everything needed for routing and monitoring the path, plus its current status. When the reservation request reaches the target, it is acknowledged on the exact same path it was sent, and every participating gateway sets the according entry to *reserved*.

Because the connectivity in a mobile ad-hoc network changes steadily, it is not possible to guarantee the consistency of a path. Thus, to meet the QoS requirements, all links on a path must be monitored, every failure must be detected and reported to both ends of the QoS stream. Every node participates in the monitoring process by several means. First, it forwards monitoring notifications (such as *broken path* packets) from other nodes to the originator or destination of a QoS path. Second, it checks the data flow on the reserved path. When no data is transmitted for a certain time, the connection is treated as closed and silently dropped. The third part of the monitoring process is a steady inspection of the local link graph. When a neighbor disappears, the QoS path table is checked if it was a source or gateway for one or several of the links. In this case, monitoring notifications are generated in the opposite direction, i.e. a notification is sent to the originator when a gateway disappears, and a notification is generated for the target, when the previous hop node is disconnected. Because of the strict clustering scheme and the bidirectionality of every link, a disconnect is detected on both sides of the broken connection, and both originator and target are informed, while the reserved bandwidth is freed at the same time.

5.4 Summary

Based on the reliable packet transmission in a cluster, the combination of proactive world model propagation, best effort routing and optimistic reactive search allows the fast and directed reservation of QoS paths with application specified parameters. After the paths are reserved, a sophisticated monitoring mechanism observes their reliability and notifies the application when the given guarantees cannot be met because of node failure, topology change or other reasons. This combination allows the usage of the protocol in criti-

Table 1: Simulation Parameters.

simulation parameter	*value*
area	1000 m × 1000 m
number of stations	100
initial positions	uniform random
propagation model	two-ray ground
transmission range	250 m
interference range	500 m
transmission rate	54 Mbit/s
packet loss rate	5 %
protocol parameter	*value*
max. clients per cluster C	16
max heads per client H	7
slot length l	4 ms
omission degree d	3
maximum packet size	1000 byte
mobility paramter	*value*
mobility model	random waypoint
speed of stations	$3-5\,\frac{m}{s}$

cal domains like automation control.

6 EVALUATION

The described protocol was implemented based on an abstraction layer (Herms and Mahrenholz, 2005) that allows to use the same binary code under Linux with real WLAN devices and in the network simulator NS−2(NS2, 2007). In this paper we present results from the simulation. The software was also successfully used in a real wireless mesh network, but an extensive evaluation has not been done, yet. The relevant simulation parameters are summarized in table 1.

6.1 Topology Propagation

The first critical aspect of the protocol is the quality of the link graph on the participating nodes. Inaccurate information can be tolerated by the routing but may lead to suboptimal path selections. If the divergence between the local link graph and the real topology grows too big, routing can even become impossible. Because the topology information is propagated in the idle slots of the clustering mechanism, it does not affect application performance. However, the propagation speed depends on the medium usage. Here we measure the dependency of propagation speed on the application network load.

For measuring the accuracy of the link graph we define a quality metric κ_i that corresponds to the correctness of the local link graph of node i. It depends on the number of really existing links L, the number of false positives l_i^+ (links in the link graph without a corresponding real link) and the number of false negatives l_i^- (real links not represented in the link graph). For allowing comparison regardless of the number of links, the value is normalized by dividing it by the number of really existing links:

$$\kappa_i = \frac{L - l_i^+ - l_i^-}{L} \tag{1}$$

The overall accuracy of the topology propagation is represented by averaging all local accuracies: $K = \frac{1}{n}\sum\kappa_i$ with $K \in [0,1]$. A value of $K = 1$ indicates a perfect propagation while smaller values mean lower accuracy.

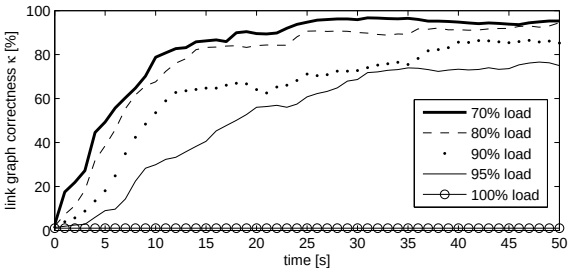

Figure 4: Topology propagation with varying network load.

In figure 4 the accuracy of the link graph propagation over time is depicted with varying network load. It can be seen that the accuracy increases after the start and that higher traffic leads to slower propagation of the link graph. If all slots are used for data traffic (load 100%) no link graph data is propagated because no idle slots are available. This case though is only theoretical, because it never will be met with normal data traffic where additional slots are reserved for retransmissions and never entirely used. Here, only high loads starting from 70 % are considered and even under this conditions sufficient accuracies are reached after a few seconds. It can also be seen that the perfect propagation of 100 % is never completely reached. This is caused by the mobility, which changes the topology while older link state data is still propagated. Due to the approach similar to FSR (Pei et al., 2000), a completely correct link graph is not actually needed for routing, because a node only needs to have exact information about its neighbors and approximate data about the direction to remote nodes. Thus, we regard a link graph propagation of $K = 60\%$ to be sufficient for proper routing and QoS path reservation, which is achieved after less than 12 seconds on a 90% saturated medium, starting with no prior topology knowledge.

6.2 QoS Routing

Our QoS routing aims to prevent unpredictable timings in the network that are caused by congestion. An

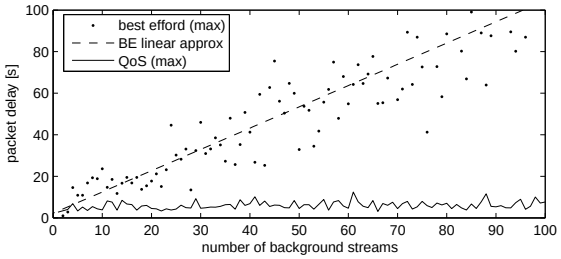

Figure 5: Comparison of QoS and Best effort traffic.

admission control is used that requires prior reservation of QoS connections. This section shows the compliance of the QoS protocol to the timing restrictions, compared to data transmission times using the best effort channel. We use a simulation setup with the same parameters as in the previous experiments. The traffic was generated using different numbers of data streams created from random source nodes to random destinations. The number of streams was varied from 0 to 100. By using the QoS routing, data packets were only sent after a successful path reservation. Best effort data was sent by the nodes in the same intervals as if a path were reserved.

The comparison of best effort and QoS routing is depicted in figure 5, where the maximum delays of data packets are presented. It can be seen that with an increasing number of streams the delay of best effort streams rises. Without admission control the network becomes congested and packets spend a lot of time in the outgoing queues on gateway nodes or even can be dropped. Using QoS routing with prior reservation solves the problem by limiting the number of concurrently admitted streams. The maximum transmission delays remain below a certain level. Compared to other kinds of network, the latency of the routing is still rather high. On the one hand, this results from the quantity used for the experiments – the maximum delay among all streams, which corresponds to the maximum delay of the longest route and can be very long due to the random selection of the paths destination. On the other hand the polling scheme of the clusters is not optimal for short delays. Stations are only allowed to transmit when they are polled and thus a packet cannot immediately be forwarded. This means a delay of up to one cluster round on every gateway, accumulating on paths with a high number of hops and resulting in long transmission times.

Still, it can be seen that the usage of a QoS reservation mechanism completely evades congestion related latency. While the best effort transmission times grow with the number of participating nodes, QoS traffic is not delayed when more nodes try to communicate. The only impact of a congested network is either rerouting around saturated clusters or, when no alternative paths can be found, rejection of reservation requests.

7 RELATED WORK

The most similar structure to our approach can be found in CGSR (Chiang et al., 1997). It also uses scheduling for controlling the medium access. The intracluster schedule is done by the head polling its clients (TDMA). For avoiding collisions between adjacent clusters, every cluster uses its own CDMA code. Because CDMA cannot be realized on typical IEEE 802.11 devices, this can only be implemented with special hardware. Furthermore, the routing does not consider QoS, although this seems possible. In (Dong et al., 2002) a *supernode-based reverse labeling algorithm* is presented that can give guarantees on bandwidth and delay of links. Supernodes are elected that act as coordinators for the routing and coordinate the search for routes that satisfy the QoS requirements. The routing assumes that the MAC provides information about available bandwidth and delay. However, it is not discussed how this is accomplished. CEDAR(Sinha et al., 1999) is a link state based routing that creates a logical structure based on so called *core nodes* that are very similar to our cluster heads. For providing bandwidth guarantees, the corresponding information is flooded in the network. The core nodes are responsible for finding routes that meet the bandwidth requirements. It is also not explained, how the available bandwidth is calculated, but assumed that the MAC layer can provide this information.

Besides of cluster based routings, some flat routing protocols exist that aim to provide QoS. Here, the problem of managing the bandwidth also exists and there are no spatial structures (like clusters) that correspond to the groups of neighbor nodes. In (Liao et al., 2002) and (Zhu and Corson, 2002) slot-based bandwidth reservation mechanisms for MANETs are presented that use coordination with 1-hop neighbors. However, the protocols are not applicable to CSMA/CA-based MAC layers. In (Chakeres and Belding-Royer, 2004) and (de Renesse et al., 2005) admission control for bandwidth reservation is done by measuring the ratio of busy and idle state of the MAC layer. This gives an exact view of the current usage of the local bandwidth, including the effects of physical carrier sense. However, it requires modifications in the MAC layer to export this information to the upper layers and can only report the previously used medium time, which does not consider ongoing reservations. The admission control in (Kuo et al., 2005) uses one-hop beaconing for exchanging information about the available bandwidth with neighbor nodes. It is designed for the IEEE 802.11 MAC layer and considers the effects of CSMA/CA on the bandwidth, but the extended physical carrier sense of the MAC layer is not regarded. In (Chen and Heinzelman, 2005) the effects of carrier sense are correctly

considered. A two-hop beaconing is used for reaching all affected nodes and calculating the available local bandwidth. This is used for verifying the suitability of a path returned by an AODV routing. If the first guessed path is not suitable, no alternatives are used in contrast to our backtracking based protocol.

8 CONCLUSION AND OUTLOOK

In this paper we addressed the problem of providing QoS in terms of bandwidth and latency in wireless mesh networks based on standard compliant WLAN hardware. We have shown that it is essential to consider the MAC layer with its extended carrier sense range. A cluster-based local coordination protocol is presented that does two-hop coordination between nodes and allows reservation of medium time. On top of this a routing and a path reservation mechanism are used that allow bandwidth reservation and latency limitation per stream. Because the consumed medium time must be coordinated even with two-hop neighbors, a relatively expensive mechanism is applied. The resulting routing gives strict guarantees for individual streams, but these are paid with the high overhead, which results in low usable bandwidth. Therefore, it is not suitable for applications like multimedia streaming or Internet connections. Instead it can be used in automation scenarios where bounded latencies with strict guarantees are required.

Further work will include a more detailed evaluation of the protocol, especially by using a real wireless mesh network instead of simulations. This will help to estimate the properties of the routing in real scenarios. The current approach still has points for possible improvements. The latency of multihop connections can be reduced by synchronizing the polling times of gateway nodes. The usable bandwidth can be increased by overcoming the strict partitioning of medium time between adjacent clusters or maybe by using a solution without clusters. Some approaches mentioned in section 7 seem promising, but would require some rework for being applicable to standard hardware.

REFERENCES

Chakeres, I. D. and Belding-Royer, E. M. (2004). Pac: Perceptive admission control for mobile wireless networks. In *First International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QSHINE'04)*, pages 18 – 26.

Chen, L. and Heinzelman, W. (2005). Qos-aware routing based on bandwidth estimation for mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(3):561 – 572.

Chiang, C.-C., Wu, H.-K., Liu, W., and Gerla, M. (1997). Routing in clustered multihop, mobile wireless networks with fading channel. In *The Next Millenium*. The IEEE SICON.

de Renesse, R., Ghassemian, M., Friderikos, V., and Aghvami, A. H. (2005). Adaptive Admission Control for Ad Hoc and Sensor Networks Providing Quality of Service. Technical report, Kings College London.

Dong, Y., Yang, T., Makrakis, D., and Lambadaris, I. (2002). Supernode-based reverse labeling algorithm: Qos support on mobile ad hoc networks. In *Canadian Conference on Electrical and Computer Engineering*, volume 3, pages 1368 – 1373.

Herms, A. and Mahrenholz, D. (2005). Unified development and deployment of network protocols. In *Meshnets '05*, Budapest, Hungary.

IEEE-802.11 (1999). *ANSI/IEEE Std 802.11, 1999 Edition*. IEEE-SA Standards Board, 1999 edition. http://standards.ieee.org/getieee802/download/802.11-1999.pdf.

Kuo, Y.-L., Wu, E. H.-K., and Chen, G.-H. (2005). Admission control for differentiated services in wireless mesh networks. In *Meshnets '05*.

Liao, W.-H., Tseng, Y.-C., and Shih, K.-P. (2002). A TDMA-based bandwidth reservation protocol for QoS routing in a wireless mobile ad hoc network. In *IEEE International Conference on Communications*, volume 5, pages 3186 – 3190.

Liu, C. and Kaiser, J. (2003). Survey of mobile ad hoc routing protocols. Technical Report 2003-08, Department of Computer Structures, University of Ulm, Germany.

Mahrenholz, D. (2006). *Providing QoS for Publish/Subscribe Communication in Dynamic Ad-hoc Networks*. Doctoral dissertation, University of Magdeburg. online `http://diglib.uni-magdeburg.de/Dissertationen/2006/danmahrenholz.pdf`.

Nett, E. and Schemmer, S. (2003). Reliable real-time communication in cooperative mobile applications. *IEEE Transactions on Computers*, 52(2):166–180.

NS2 (2007). The network simulator - ns-2. Homepage. http://nsnam.sf.net.

Pei, G., Gerla, M., and Chen, T.-W. (2000). Fisheye state routing: A routing scheme for ad hoc wireless networks. In *ICC (1)*, pages 70–74.

Sinha, P., Sivakumar, R., and Bharghavan, V. (1999). CEDAR: a core-extraction distributed ad hoc routing algorithm. In *INFOCOM (1)*, pages 202–209.

Tanenbaum, A. S. (2002). *Computer Networks*, chapter Wireless LANs. Prentice Hall PTR, 4 edition.

Zhu, C. and Corson, M. (2002). Qos routing for mobile ad hoc networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 958–967. IEEE.