On-line Monitoring for Model-based QoS Management in IEEE 802.11 Wireless Networks

Johannes Semmler
Humboldt-Universität zu Berlin
Unter den Linden 6
10099 Berlin, Germany
semmler@informatik.hu-berlin.de

Katinka Wolter
Humboldt-Universität zu Berlin
Unter den Linden 6
10099 Berlin, Germany
wolter@informatik.hu-berlin.de

Philipp Reinecke
Humboldt-Universität zu Berlin
Unter den Linden 6
10099 Berlin, Germany
preineck@informatik.hu-berlin.de

ABSTRACT

Ensuring Quality of Service (QoS) in wireless networks poses an open problem in many application domains. We propose an automatic on-line QoS monitoring and management infrastructure that can be incorporated into existing network setups. Based on model-based assessment of current and future QoS conditions, our solution will control traffic in the network through a combination of admission control, enforced hand-over, traffic shaping and transmission parameter adjustments. Correctness of the model is evaluated through experimental evaluation and simulations. We implement a prototype of the proposed system using open-source components.

1. INTRODUCTION

As wireless networks are becoming a matter of course service quality in those networks is gaining importance. One may want to provide selected users with higher quality of service (QoS) than others or account for special service requirements of different applications. It is well-known that data transfer has strict requirements with respect to data loss and can tolerate some delay. For video and voice communication almost the opposite holds. In wireless networks QoS poses many challenges [17, 26]. Over the past years channel reservation methods [2] as well as prioritised contention algorithms [9, 10, 25] have been proposed. The standard IEEE 802.11e [8] for QoS in wireless LANs includes both types of channel allocation. The protocol assigns priorities to the different types of traffic aiming at suitable service quality for all applications. In this paper we only use the contention mechanism EDCA in IEEE 802.11e as it is the more commonly used algorithm.

However, despite prioritised traffic in a congested network service quality may still be suboptimal. As stations cannot be blocked from sending this situation can only be improved to a certain extent and even statically assigned priorities are not able to resolve all problems. But priorities can be the basis of network management as we will show in this paper.

We propose a management architecture that is based on an analytical formulation of expected frame transmission time [5] and monitoring of the actual QoS characteristics of the network. If the monitored service quality does not meet the QoS requirements, different actions can be taken by the management component. Either admission of a new station can be prohibited, hand-over to a neighbouring access point (AP) can be enforced, or priorities of stations connected to the AP can be manipulated in order to improve QoS.

We have implemented a prototype strictly using open source software and demonstrate the capability of the monitoring component as well as some basic management action. Management as of now only supports either rejecting a newly entering station or automatic shifting of priorities if the management component identifies that QoS requirements of the station cannot be met. The framework allows for an easy integration of more sophisticated management actions.

The paper is organised as follows. In the next section we review related work. Section 3 presents our framework for QoS management in IEEE 802.11e wireless networks, discussing the considered metrics of the QoS model as well as the monitoring component in more detail. In Section 4 we evaluate important parts of the QoS model and the management component. In Section 5 we describe our prototype implementation of the framework, and Section 6 concludes the paper.

2. RELATED WORK

Service quality in wireless networks has been studied extensively in the last years, but to the best of our knowledge, there does not exist a flexible open-source on-line QoS monitoring and management system that utilises the full power provided by the IEEE 802.11e standard. Existing approaches are often limited to the IEEE 802.11b standard (which does not support access categories), or utilise technologies other than wireless LAN.

The framework proposed in [3] provides an example for the former. Here, different QoS levels are supported over IEEE 802.11b, by applying IP traffic shaping on the clients, and by access control. The general idea is similar to that of our approach, however, we utilise model-based assessment of future QoS levels in the management decisions. Our management component also supports the newer IEEE 802.11e acccess categories, which provide priorities on the MAC layer, and does thus not depend on the special software that im-

MASCOTS 2009 September 21–23, 2009, London, UK

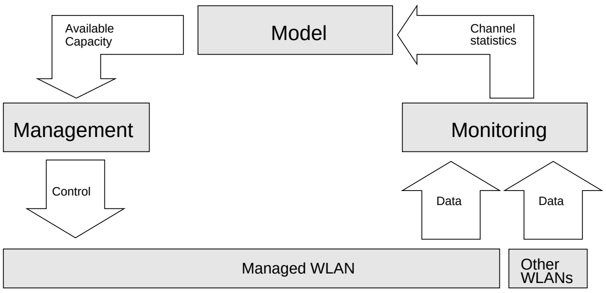

Figure 1: System Overview

plements DiffServ support in the clients.

An instance of the latter is [1]. In this approach, the performance of WLAN is improved not by carefully sharing a scarce resource according to priorities but by combining the wireless local area network with a wireless Metropolitan Area Network (MAN). A Markov model is presented, and blocking probabilities are studied under varying load. The architecture WATCHMAN [12] aims at providing reliable service and charges the user for quality. This is achieved by extending the routing protocol OLSR. A model-based QoS management is proposed in [6], where the authors use three priority classes to handle real-time and non-real-time traffic at the same time without using the priority classes of the 802.11e standard.

Considerable work exists regarding the individual building blocks of a system as proposed in the present paper, however, no attempt has been made to combine these into one model-based monitoring and management system. In the following we review work on QoS management and QoS monitoring, before briefly touching upon the issue of implementation.

Approaches to managing QoS in wireless network tend to focus on the access categories defined in the IEEE 802.11e standard [8]. Many simulation studies (e.g. [4, 15]) demonstrate the ability of the standard to provide different QoS levels to different priority classes. [25, 18], on the other hand, show that in some situations the standard access categories may result in suboptimal performance.

There has been some work on monitoring QoS in wireless networks. For instance, [11] shows that wireless measuring allows to infer more detailed information about the medium characteristics than is possible with measurements made in wired networks. In particular, data about faulty transmissions and transmission duration may be obtained. Another important aspect of monitoring relates to the available bandwith. In [14] IdleGap, a method for estimating available bandwidth in wirless network is proposed. However, the method relies on input data from a lower layer, which is not accessible by monitoring software on most devices.

Several open-source projects provide the software required for the implementation of a QoS management system. Based on the linux distribution OpenWRT [22], software can be developed for wireless routers. The MadWifi chipset driver [21] and the PCAP library [23] allow monitoring the wireless channel. Management through access control can be implemented using the Hostapd [16] IEEE 802.11i authenticator and the FreeRADIUS RADIUS server [20].

3. QOS MANAGEMENT SYSTEM

An abstract overview of the proposed system is given in Figure 1: Based on estimates of available capacity (e.g. throughput) a management component controls the network through a variety of measures, in order to guarantee Service-Level Agreements. Available capacity is derived based on a model of the system that translates channel statistics, obtained by a monitoring component, into capacity estimates. The monitoring component observes channel conditions. Note that the monitoring component observes both frames sent within the managed network and within other networks not under control of the management component. This is necessary to obtain a correct representation of medium usage, since traffic in neighbouring networks may negatively affect the managed WLAN. In the following we describe these components in more detail.

3.1 QoS Management

QoS management has the goal of ensuring that required SLAs are indeed met. In our discussion we first distinguish between management *methods* and management *models*. Management methods refer to what is done to control the system, while management models describe the manner in which the methods are applied.

3.1.1 Management Methods

In general, management methods can be divided into modifications of the network topology and adjustments to transmission parameters.

Network topology can be controlled by access control methods and by enforced hand-over. If the network cannot sustain any more clients without breaking an SLA, the management component may simply limit the load by rejecting additional clients. Enforcing hand-over between access points (APs) may also ensure QoS. In this approach, clients are redirected to other access points if the AP they are currently connected to is overloaded.

Transmission parameters in wireless networks can be adjusted on the IP level of the network, and by directly changing low-level parameters of frame transmissions. The former is commonly referred to as traffic shaping, and there are implementations available that support various traffic shaping methods such as the ‘leaky bucket’ approach and token bucket queueing, see e.g. [7]. These usually have a broader focus on general network traffic control. We focus on direct control of low-level traffic parameters using the access categories defined in IEEE 802.11e [8].

IEEE 802.11e access categories are implemented in terms of parameters that influence opportunities for frame transmissions. The standard distinguishes between four priorities (Background, Best Effort, Video, and Voice) by the choice of parameter values for the different access categories. Medium access is primarily controlled by the values of the arbitration inter-frame space (AIFS), the time a station needs to wait before transmitting on a free channel, and the contention window size, from which the back-off timer, used to avoid collisions, is randomly drawn [8].

Access categories can be employed for the management of throughput in several ways. First, lower-priority stations may be assigned lower access categories, in order to ensure throughput for high-priority stations. Second, access categories may be shifted *downwards*. In [25], it was observed that in situations with many clients in the same high access category overall throughput suffers due to a high collision probability. By shifting categories down, i.e. assigning lower ACs to all stations while keeping their relative ordering constant (e.g. Best Effort instead of Video for all Video stations, and Background instead of Best Effort), throughput can be improved in this situation. If, on the other hand, there are few stations of low priority in the network, and the higher priorities are not occupied, the network is underutilised. Then, performance can be improved by shifting these low-priority stations *up* into the unused higher categories [25, 18].

3.1.2 Management Model

We further distinguish between management models, that is, we consider *how* the above methods are employed. First, we can apply management in a 'gate-keeper' fashion, i.e. when stations enter or leave the network. Second, we may use management methods to control traffic from stations already in the network, for example, if traffic characteristics change.

The first approach supports all management methods. These methods are applied at the time a station enters or leaves the network, and can be used to ensure that QoS requirements for all stations already within the network are not violated by the arrival of a new station. In this regard, the 'gate-keeper' model is conservative. Unfortunately, it relies on the rather limiting assumption that stations' QoS requirements and traffic characteristics will be stationary throughout the time of their presence in the network. The second approach is much more flexible in this respect, since it can also react to changing traffic characteristics and network conditions. However, we cannot apply all proposed management methods with the second approach. In particular, due to the nature of the wireless medium, the access point cannot force stations to leave the network, i.e., topology control cannot be exerted.

We therefore focus on the 'gate-keeper' management model. Then, for every station that wishes to enter the network, we must predict the QoS levels that will be attained once the new station starts transmitting data. If the new QoS levels are not sufficient, management methods must be applied: Either the current network topology and transmission parameters have to be adjusted (e.g. by enforced hand-over or priority shifting), or, if this is not feasible (e.g. because the network is already overloaded, or because higher-priority stations would suffer a QoS decrease), access must be denied to the new station.

3.2 On-line QoS Model

Service-Level Agreements (SLAs) specify limits on several parameters of the medium that must not be exceeded. In wireless networks, delay, jitter, loss rate, and throughput of transmissions are of particular importance.

As has been discussed in the previous section, we focus on the 'gate-keeper' approach to management. This requires that the management component be able to assess whether the network can sustain the additional station without violating SLAs. In the following we describe methods to derive the required parameters using data observed in the monitoring component.

3.2.1 Delay

Frame delay can be computed as follows: Prior to transmission, each frame is assigned a time-stamp that is stored in the frame's MAC header. The frame delay is obtained by comparing this time-stamp with the local time at which the frame was received. In order to compute correct delays, all stations in the network must be synchronised to the same time.

	BK	BE	VI	VO
$\min(t_{\text{Frame}})$ in μs	339	303	294	294
$\max(t_{\text{Frame}})$ in μs	474	438	357	321

Table 1: Minimum and maximum channel busy times for frames of 1546 bytes length.

3.2.2 Jitter

Jitter is computed based on the difference $J(i)$ of the inter-arrival times I_{i-1}, I_i of subsequent frames $i-1, i$ [19]:

$$J(i) = |I_i - I_{i-1}|.$$

The jitter J is then the moving average of the last n observations:

$$J = \frac{i}{n-1} \sum_{i=2}^{n} J(i).$$

3.2.3 Loss Rate

The loss rate is derived from the amount of damaged frames. Damaged frames can be identified by mismatches between the MAC checksum and the frame contents.

3.2.4 Throughput

Available additional throughput is computed as the product of medium idle time and the data rate expected to be offered by the additional client. That is, we assume that the newly connecting client will use the currently unused capacity of the network.

Unfortunately, medium idle time cannot generally be observed directly without special-purpose hardware. For this reason, we compute idle time based on the *busy time*, which can be derived from medium usage. Thus in contrast to delay, jitter and loss rate, we can only estimate throughput.

We compute the medium busy time as the sum of all times spent for frame transmissions. The time required for the transmission of a single frame is is computed using the following expression derived by [5]:

$$t_{\text{Frame}} = t_{\text{DIFS}} + t_{\text{SIFS}} + t_{\text{Contention}} + 2t_{\text{PH}} + t_{\text{Transmission}} + t_{\text{Ack}}, \tag{1}$$

where t_{DIFS} and t_{SIFS} are the Distributed and Short Inter Frame Space, respectively, $t_{\text{Contention}}$ is an approximate for the time spent in the contention phase, and $t_{\text{PH}}, t_{\text{Transmission}}$ and t_{Ack} indicate the times required for transmission of the PLCP preamble and the PLCP header, sending of the data frame by the sender, and transmission of the acknowledgement by the receiver [5].

As can be seen in (1), the length of the frame transmission time t_{Frame} depends on the time spent in the contention phase, $t_{\text{Contention}}$. When there are no collisions, this time consists just of the waiting time chosen randomly in each station prior to transmission. The waiting time is drawn from the contention window, which differs between access categories. Table 1 displays the frame transmission times that result from (1) when we assume that the lower or upper boundary of the contention window was chosen. In the

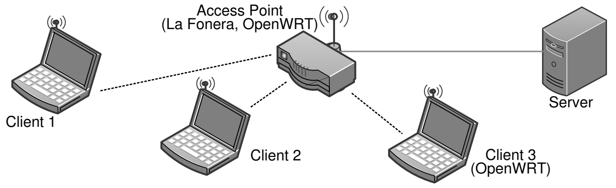

Figure 2: Experiment Setup

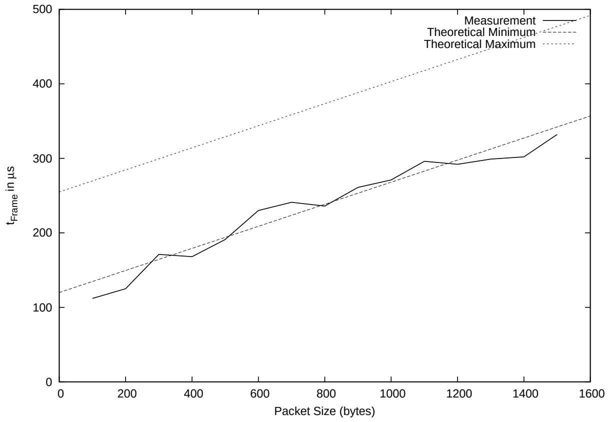

Figure 3: Comparison of predicted and measured transmission times t_{Frame}.

absence of collisions, the expected value of t_{Frame} transmitted in the respective category stays within these intervals.

3.3 Monitoring Component

The monitoring component observes transmissions performed within the network. Additionally, transmissions by other networks that affect medium availability are also monitored. Monitoring is performed in a distributed fashion on both access points and clients. This improves coverage, since clients may be able to observe traffic from neighbouring networks that is not visible to the access point, but still affects channel conditions.

For each frame they receive, access points store the MAC address, the access category, the transmission time, the payload length, the transmission rate and the send and receive time-stamps. The MAC address and the QoS class are used to identify the client and its access category. Transmission time, payload length and transmission rate allow computation of channel load, throughput and data rate. Send and receive time-stamps are used in the computation of delay and jitter. Additionally, from damaged frames (identified by checksum errors) the loss rate can be derived.

Client-side monitoring is performed in a similar fashion. However, the client only stores data rates and channel load, which are available for all observed frames. Delay, jitter and loss, on the other hand, can only be observed for frames transmitted in the managed network. These are already accounted for by monitoring in the access point and thus do not need to be observed in the client.

In addition to monitoring traffic, clients also maintain a list of visible access points. Using this list, the management component can adjust the network topology by directing the client to reconnect to another access point from its list.

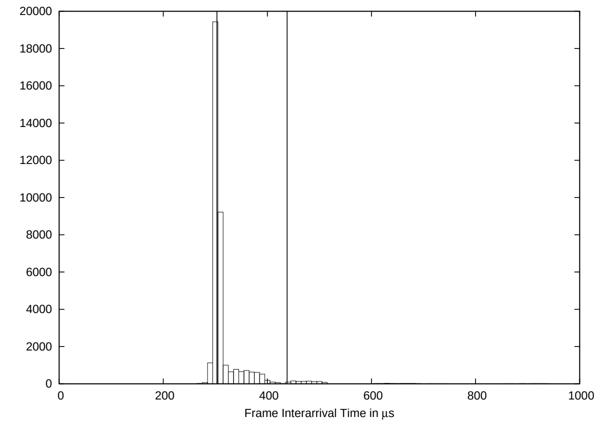

Figure 4: Distribution of frame inter-arrival times for Best Effort (BE), and predicted transmission time t_{Frame} (vertical lines).

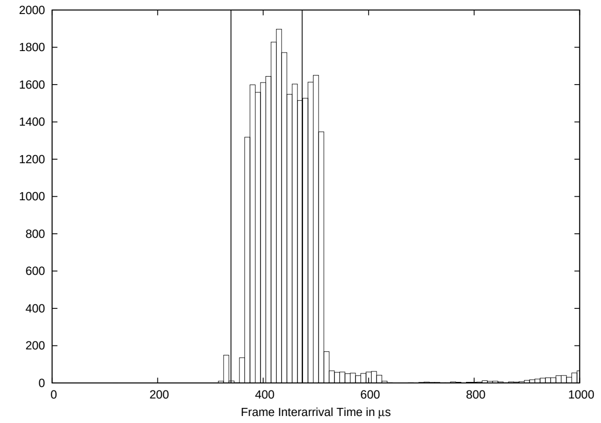

Figure 5: Frame inter-arrival time distribution for Background (BK).

4. EVALUATION

In Figure 1 we can identify two important building blocks of our approach. First, it relies on correct capacity estimation, particularly on the estimation of available throughput. We therefore check whether the model presented in Section 3.2 indeed captures medium conditions. Second, the effect of the management methods described in Section 3.1.1 must be studied. Theoretical results on upwards and downwards priority shifting were presented in [18] and [25], respectively. Here, we only quantify the effect of prioritised channel access.

4.1 Evaluation of the Model

While the computation of delay, jitter and loss rate are straightforward, busy intervals must be estimated using (1) (Section 3.2), based on observed data.

In order to validate the approach, we study busy times in a simplified scenario with only one client sending data. We can then approximate true busy times using the frame interarrival time, i.e. time elapsed between subsequent frames. Given these data, we can assess how well (1) estimates transmission times.

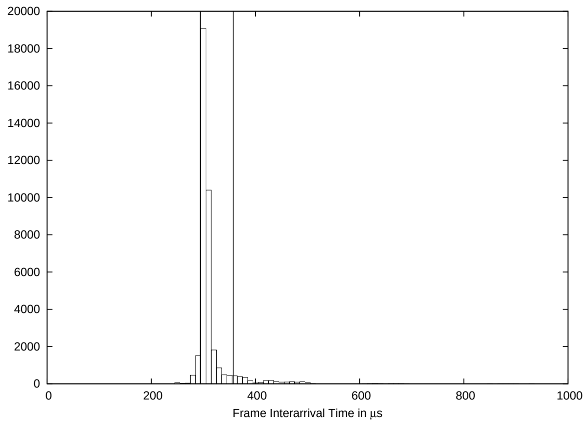

Figure 6: Frame inter-arrival time distribution for Video (VI).

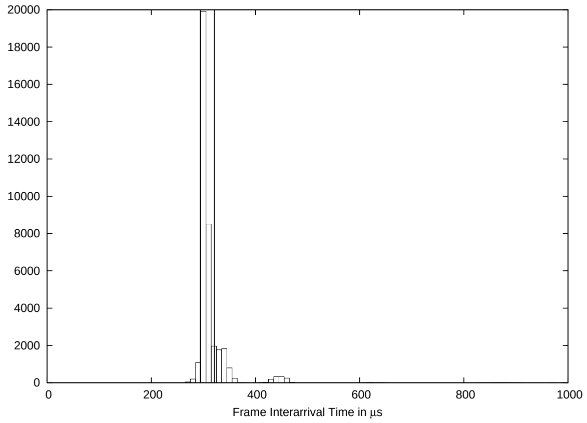

Figure 7: Frame inter-arrival time distribution for Voice (VO).

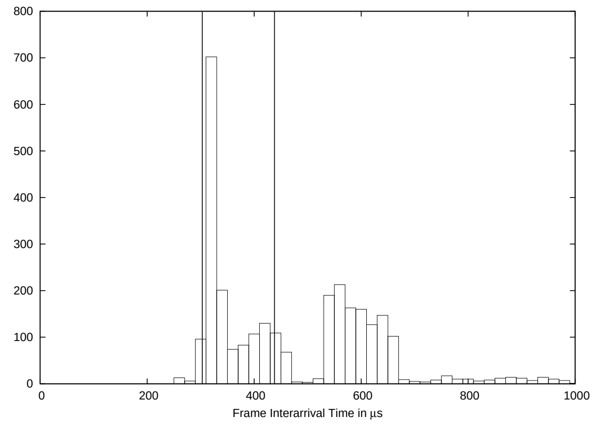

Figure 8: Frame inter-arrival times for Best Effort with disturbances.

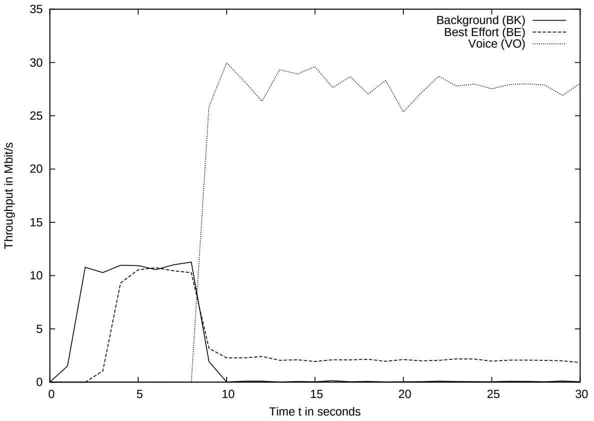

Figure 9: Measured throughput for parallel transmissions.

We set up the experiment as shown in Figure 2: A 100 Mbit UDP data stream is sent over an access point to a client. Transmissions are monitored by a second notebook.

Note that the wireless network can only sustain a maximum load of 54 Mbit. The high data rate was chosen to ensure that there are no idle times on the medium other than IFS times and contention intervals. However, beacons, transmissions of other networks, and frames missed by the monitor may still result in a disrupted stream. We address this issue by only using times from frames with subsequent sequence numbers. Furthermore, we also exclude times that differ from the observed average by more than 10%.

Figure 3 shows results for a stream sent in the Best Effort (BE) access category and frame sizes varying from 100 to 1500 bytes. We observe that transmission times (approximated by frame inter-arrival times) generally stay within the interval computed using (1). Since there are no other senders in our setup, and thus no collisions within the network, transmission times tend to stay at the lower bound of the interval. In Figure 4 we explore the distribution of transmission times for frames of 1546 bytes length in more detail. Here, we also observe that transmission times tend to lie within the computed intervals.

So far, we only studied times for the Best Effort access category. In Figures 5–7 we show the distribution of transmission times for frames of 1546 bytes length in the other three access categories defined in the IEEE 802.11e standard. In all histograms, the predicted minimum and maximum values of the frame transmission time t_{Frame} are indicated (Table 1). For the Voice and Video categories we see the same trend as before: Observations are clustered at the lower boundary of the estimated interval, since there are no collisions in this setup. For Background, however, we note that transmission times appear to follow a uniform distribution over the whole interval. This can be attributed to the fact that with Best Effort stations always wait a random time before sending.

In Section 3 we noted that the monitoring component observes both the managed network and neighbouring networks. Figure 8 illustrates why this is necessary: Here, we show the frame inter-arrival time distribution for Best Effort observed at another instant in time than that of Figure 4. Note that there is a second bulge of observations just outside the predicted interval of channel busy times. This indicates disturbances caused by another network in the neighbourhood of our measurement setup.

4.2 Evaluation of the Management Component

We now study the effectiveness of prioritised traffic. Since

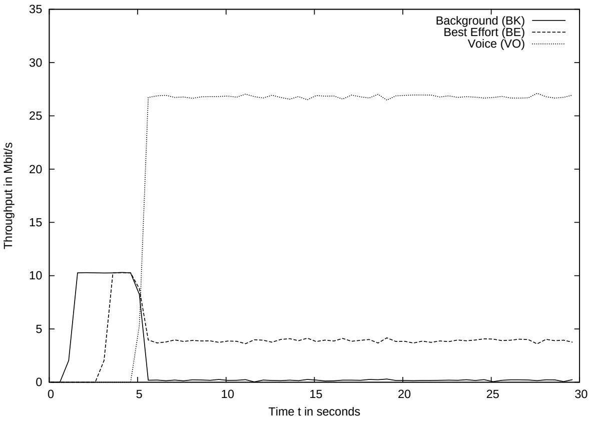

Figure 10: Throughput according to simulation for parallel transmissions.

	Throughput per client							
	All clients at VO				One client at BE			
$l\backslash n$	2	3	4	5	2	3	4	5
5	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
10	10.0	9.2	6.5	5.0	10.0	10.0	8.9	6.4
15	15.0	9.5	6.5	5.0	15.0	14.9	9.1	6.5
20	16.1	9.6	6.9	5.1	20.0	15.4	9.3	6.9

Table 2: Throughput improvement by re-assigning one client to the Best Effort access category

the wireless medium cannot be controlled, we prefer to use a simulation for these experiments. However, we must first evaluate whether such a simulation is appropriate. We do this by direct comparison between simulation and measurement data. Figure 9 shows throughput achieved by parallel transmissions in a real scenario, while Figure 10 shows the results for an ns-2 [24] simulation of the same scenario. We observe that the simulation captures the data very well, and will therefore use the simulation in the following evaluation.

In our simulation we measure the achieved throughput per client for $n = 2, \ldots, 5$ clients and an offered load $l = 5, 10, 15, 20\,\mathrm{Mbit}$ per client. Table 2 summarises the results. The left-hand side of the table displays the throughput achieved by a single client when all clients are in the Voice access category. We note that throughput drops as the number of clients grows, with the exception of the situation with $l = 5\,\mathrm{Mbit}$, where it stays constant. The latter illustrates that the network can comfortably handle such a low load. Furthermore, throughput also drops when we increase the offered load, due to a reduced number of idle times that would be available for transmissions.

We then re-assign one client to a lower access category (Best Effort). The results on the right-hand of the table side show the throughput now obtained by each of the remaining clients in the Voice category. Again with the exception of the low load situation ($l = 5\,\mathrm{Mbit}$), we observe a stark throughput increase. The improvement is most obvious in the case with 3 clients and an offered load of $l = 20\,\mathrm{Mbit}$ per client.

5. IMPLEMENTATION

We implemented the monitoring and management components using avaiable open-source components. The overall structure of the system is shown in Figure 11: We use OpenWRT [22] as the operating system on the access points. Client authentication is implemented using Hostapd [16] and the Freeradius RADIUS server [20]. We use the MadWifi chip-set driver [21] on the access points. In particular, MadWifi supports IEEE 802.11e access categories. We set the access category of the frame used to transport an IP packet based on the Differentiated Services field in the IP header. In combination with the PCAP library [23], MadWifi allows us to monitor all network traffic, including traffic belonging to other networks.

The main functionality is implemented in the SLAM component, whose FMC block diagram is shown in Figure 12:[1] The *Client Cache* is the central data structure. The cache stores the following data about connected clients: The client's MAC and IP addresses, the client's classification and the connection status (connected/disconnected). Furthermore, for each client the cache contains transmission parameters, viz. the channel, the channel busy time, average transmission rate, jitter, and signal strength for the last 1, 5, 10 and 30 seconds.

These data are updated by the Server Thread and the Monitor Thread. The *Server Thread* receives a notification from the Hostapd daemon whenever a client connects to or disconnects from the network. This notification contains the client's addresses, classification and connection status. The Server Thread then updates the client's entry in the cache accordingly.

If client-based monitoring is enabled, the Server Thread may also receive notifications from a *SLAM Client*. These messages contain the MAC address, data rate, channel busy time, channel, and signal strength of a client observed by the station on which the SLAM client is running. In contrast to the notifications from the Hostapd daemon, these data may also refer to clients that are not part of our network.

The *Monitor Thread* monitors the network through the PCAP library. It analyses packets received on the wireless network, and, for each sender, computes throughput, delay, jitter and loss rate. These data for the last 1, 5, 10 and 30 seconds are then stored in the client cache.

QoS management is performed based on the contents of the client cache and on given SLAs. In the current implementation we only perform management using IEEE 802.11e access categories. This functionality is implemented in the *Priority Scheduler*: For each data frame, the associated access category is derived from the client class and the application class. The access category is then set using suitable iptables rules. These rules are regenerated as needed whenever the client cache changes.

Management methods currently supported are access control and priority shifting. Access control works as follows: Suppose that n clients are already in the network, and a new client wants to join. Then, it is first checked whether either the network has enough available throughput to support the requirements of all $n + 1$ clients combined, or whether the new client has a higher priority than clients in the network. In the first case, the client is admitted to the network, in the second, the client is admitted at its nominal category, and all clients in lower categories are shifted downwards, in or-

[1] The accompanying legend explains the main components of the diagram; for more detail we refer the reader unfamiliar with FMC block diagrams to [13].

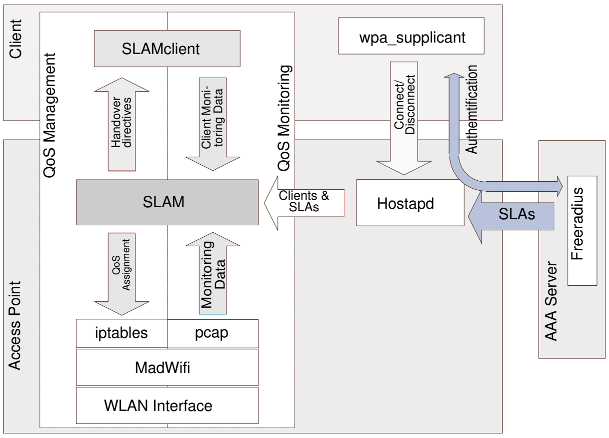

Figure 11: Implementation overview.

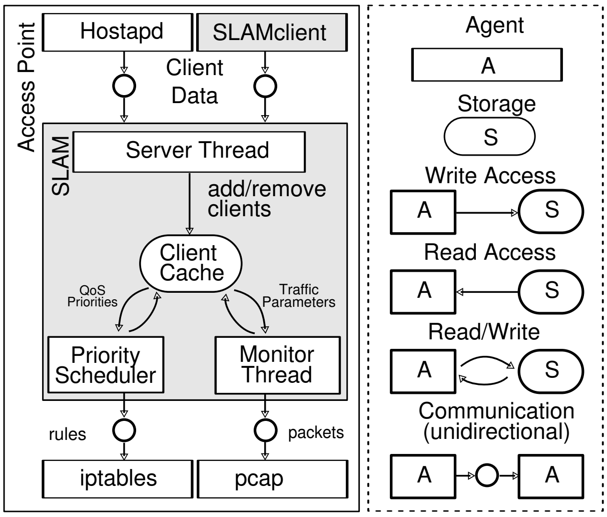

Figure 12: Architecture of the SLAM Component (FMC block diagram).

der to ensure that the client receives its allotted throughput. Additionally, SLAM supports shifting of clients into unused upper access categories, in order to improve QoS by better resource utilisation.

6. CONCLUSION

We have presented a framework for dynamic QoS management in IEEE 802.11 wireless networks. The framework is based on an analytical formulation of frame transmission times. It includes as main building blocks a monitoring and a management component. The framework has been implemented using open source tools. The output of the monitoring component has been found to match the analytical model. The management component shifts priorities down if a newly arriving station otherwise can not be served at its assigned priority. Measurements confirm the feasibility of the proposed approach. In future work we will implement admission control and enforced hand-over as management actions of the SLAM component and evaluate their impact in experimental analysis.

7. ACKNOWLEDGEMENTS

The work presented in this paper was supported by the German Science Foundation (DFG), grant number Wo 898/1-2.

8. REFERENCES

[1] T. Ali-Yahiya, H. Chaouchi, A.-L. Beylot, and G. Pujolle. Threshold based wimax resource reservation. In *Mobility '06: Proceedings of the 3rd international conference on Mobile technology, applications & systems*, page 40, New York, NY, USA, 2006. ACM.

[2] E. M. E. Carlson. *QoS in distributed wireless 802.11-based multi-hop networks*. PhD thesis, TU Berlin, 2007.

[3] H. Chaouchi and A. Munaretto. Adaptive qos management for ieee 802.11 future wireless isps. *Wirel. Netw.*, 10(4):413–421, 2004.

[4] P. Garg, R. Doshi, R. Greene, M. Baker, M. Malek, and X. Cheng. Using IEEE 802.11e MAC for QoS over Wireless. In *Conference Proceedings of the 2003 IEEE IPCCC*, pages 537–542, 2003.

[5] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11b. In *IEEE Infocom*, pages 836–843, 2003.

[6] L. Huang, S. Kumar, and C.-C. Kuo. Adaptive resource allocation for multimedia qos management in wireless networks. *IEEE Transactions on Vehicular Technology*, 53:547–558, March 2004.

[7] B. Hubert, T. Graf, G. Maxwell, R. van Mook, M. van Oosterhoutand P. Schroeder, and P. Larroy. Linux Advanced Routing and Traffic Control. http://lartc.org. Last visited April 10th, 2009.

[8] IEEE 802.11 Working Group. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements (802.11e). http://www.ieee802.org/11/. last seen June 6, 2008.

[9] A. Iera, A. Molinaro, G. Ruggeri, and D. Tripodi. Improving QoS and throughput in single- and multihop WLANs through dynamic traffic prioritization. *IEEE Network*, 19(4):35–44, 2005.

[10] A. Iera, G. Ruggeri, and D. Tripodi. Providing Throughput Guarantees in 802.11e WLAN Through a Dynamic Priority Assignment Mechanism. *Wireless Personal Communications*, 34:109–125, 2005.

[11] A. A. Jihwang Yeo, Suman Banerjee. Measuring Traffic on the Wireless Medium: Experience and Pitfalls, 2002.

[12] A. R. Khakpour, M. Laurent-Maknavicius, and H. Chaouchi. Watchman: An overlay distributed aaa architecture for mobile ad hoc networks. In *ARES '08: Proceedings of the 2008 Third International Conference on Availability, Reliability and Security*, pages 144–152, Washington, DC, USA, 2008. IEEE Computer Society.

[13] A. Knöpfel, B. Gröne, and P. Tabeling. *Fundamental Modeling Concepts. Effective Communication of IT Systems*. John Wiley, 2006.

[14] H. K. Lee, V. Hall, K. H. Yum, K. I. Kim, and E. J. Kim. Bandwidth estimation in wireless LANs for

multimedia streaming services. In *in Proc. IEEE International Conference on Multimedia and Expo*, 2006.

[15] S. O. Lindgren A., Almquist A. Evaluation of Quality of Service Schemes for IEEE 802.11 Wireless LANs. In *Mobile Networks and Applications 8*, pages 223–235, 2003.

[16] J. Malinen. hostapd: Ieee 802.11 ap, ieee 802.1x/wpa/wpa2/eap/radius authenticator. http://hostap.epitest.fi/hostapd/. (last seen April 9th, 2009).

[17] Q. Ni, L. Romdhani, and T. Turletti. A survey of QoS enhancements for IEEE 802.11 wireless LAN. *J. of Wireless Communications and Mobile Computing*, 4(5):547–566, 2004.

[18] P. Reinecke, K. Wolter, and J. Zapotoczky. Performance analysis of dynamic priority shifting. In N. Thomas and C. Juiz, editors, *Proceedings of the 5th European Performance Engineering Workshop (EPEW2008)*, volume 5261 of *LNCS*. Springer, September 2008.

[19] R. Steinmetz and K. Nahrstedt. *Multimedia Systems.* Springer-Verlag, 2004.

[20] Various authors. Freeradius. http://freeradius.org. (last seen April 9th, 2009).

[21] Various authors. Madwifi. http://madwifi.org. (last seen April 9th, 2009).

[22] Various authors. Openwrt. http://openwrt.org. (last seen April 9th, 2009).

[23] Various authors. tcpdump/libpcap public repository. http://www.tcpdump.org. (last seen April 9th, 2009).

[24] Various authors. The Network Simulator ns-2. http://www.isi.edu/nsnam/ns/. (last seen June 6, 2008).

[25] J. Zapotoczky and K. Wolter. Increasing Performance of the 802.11e Protocol through Adaptive Shifting of Priorities. In P. Buchholz, editor, *Proc.14th GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB 08)*, Dortmund, March 2008. to appear.

[26] H. Zhu, M. Li, I. Chlamtac, and B. Prabhakaran. A Survey of Quality of Service in IEEE 802.11 Networks. *IEEE Wireless Communications*, 11(4):6–14, 2004.