Preleiminary version: accepted for the 6th WINLAB Workshop on Third Generation Wireless Systems , New Brunswick, NJ, USA, March 1997
also to appear in „Advances in Wireless Communication“, Kluwer Academic Publichers, Boston / Dordrecht / London, 1998

A DISTRIBUTED MEDIA ACCESS CONTROL FOR WIRELESS ATM ENVIRONMENTS[1]

Jean-Pierre Ebert*, Ralf Holtkamp*, Adam Wolisz*†, Louis Ramel#

* Technical University Berlin – Telecommunication Network Group
# Thomson-CSF – Communication, Navigation, Identification Division
{ebert, holtkamp, wolisz}@ee.tu-berlin.de, louis.ramel@cni.thomson.fr

Abstract: In this article we present **RNET MAC** - a novel **MAC** protocol to be used in **wireless ATM** (**WATM**) environments. The **MAC** protocol features a **distributed control** for media access. Therefore **RNET MAC** fits well for spontaneous network setups and frequent network configuration changes. We discuss some design options and show basic performance results of **RNET MAC** in different network scenarios such as **fully meshed**, **hidden terminal** and **client server**.

1 INTRODUCTION

Evolving **WATM** systems (e.g. [9, 11]) often require a **media access control** and data link control due to multiple access to the communication channel and the error prone medium respectively. For the sake of minimum changes of the ATM protocol functions and **QoS** enforcement, the **MAC** protocol has to provide dedicated mechanisms for bandwidth reservation, priority handling, low access delays as well as jitter. One can find several proposals of **MAC** protocols in the recent literature (e.g. [6, 7, 9, 11, 12, 13]), which all provide in some way **QoS** enforcement. However, all of them are based on the assumption

†also with GMD Fokus

that there is a dedicated entity (BS)[2] controls the access to the communication channel. Furthermore, the traffic has to be transmitted via this BS. The first assumption leads to mobiles with a small **MAC** functionality and therefore potentially smaller battery capacity requirements of the mobiles. On the other side, a relative expensive BS is always required and spontaneous networking events (for instance conferences or meetings), where no pre-installed infrastructure is given, are not possible. The latter assumption leads to a waste of bandwidth in case two mobiles (M) want to communicate with each other, since the data has to be transmitted twice (from M to BS and vice versa). We present in this paper **RNET MAC** (section II) that features a **distributed MAC** and point-to-point communication. In the next section (III) some performance results are given showing the capability of **RNET MAC** to work in ATM environments and different network scenarios. In section IV we conclude with a rough comparison to **MAC**s with centralized control.

2 RNET MAC PROTOCOL

The basic idea of the **MAC** is derived from the **RNET** proposal (Radio Network, see [10]) as it has been proposed for **HIPERLAN /2**[3] systems to the **ETSI**[4] in 1995 by Thomson. **RNET** is characterized by free physically separated channels (FDMA). Two of these channels are used for signaling purposes and one is for data transmission. The signaling channels are slotted (**TDMA**). Data will be transmitted by an ascending **ramp** (see Figure 1) .

2.1 RNET MAC channel structure

We consider an arbitrary amount of buffered mobiles in a wireless micro-cell. As outlined above there are two signaling (header and feedback) and one data

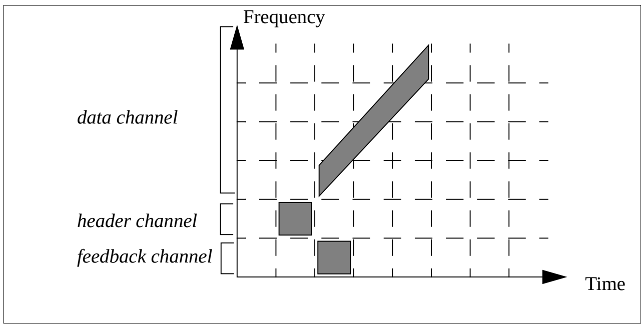

Figure 1: RNET channel structure

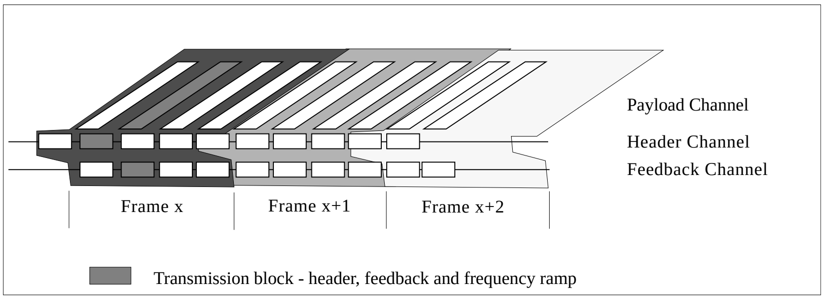

Figure 2: Channel structure with ramp spanning over three slots

(payload) channel. The payload channel consumes considerably more bandwidth than the signaling channels together. Header and feedback channels are slotted. About 56 bits fits in every slot for signaling and paging purposes. There is no fixed defined format yet. The payload channel is accessed by several ascending but independent **frequency ramps** at a certain point in time. One **frequency ramp** can carry one or more **ATM cells** as outlined in the following sections.

2.1.1 Framing and transmission blocks

A frame is defined as a group of consecutive header slots, associated feedback slots and payload ramps. The length of a frame is determined by the spanning width of a ramp. If a ramp spans over x (e.g. 3) slots then a frame consists of x+1 (e.g. 4) slots (see Figure 2). A header slot at time x, a feedback slot at time x+1 and a ramp, starting immediately after the header slot, are associated with each other and form a **transmission block**. There are guard times between the slots to encompass signal delays and clock variances.

2.1.2 Frame synchronization

Following the idea of **distributed control**, there is no common frame in **RNET MAC**. Instead, a **synchronization** of all mobiles with respect to frame start and frame end is not required. This simplifies the protocol (e.g. registration procedures). Rather than having a superframe to which all mobiles are synchronized, framing is locally. Every mobile maintains its own frame. Therefore, there may be offsets between the start point of frames. A local frame starts as soon as a mobile is switched on.

2.1.3 Frame size and ramp capacity

For the sake of achieving an optimal bandwidth utilization at a certain bandwidth and modulation scheme, the ascend of the ramp has to be chosen carefully. If the ascend is too low then two consecutive ramps will overlap each other. In the other case the payload channel is used only for a short time[5]. Therefore an optimum ascend m is

$$m = B_B / L_T$$

where B_B (e.g. 5 MHz)[6] is the base bandwidth and L_T is the slot time. L_T is defined as

$$L_T = L_L / T_R$$

where L_L is the slot length (e.g. 56 bit) and T_R (e.g. 5 Mbit/s) is the transmission rate. With the given values the optimum ascend m is equal to 446428.57 Hz/msec. To compute the number of slots per frame K the following formula has to be applied

$$K = 1 + R_T / (L_T + I_T)$$

where R_T is the ramp time and I_T is the intergap time (e.g. 3.8 msec). I_T is determined by signal delays and **synchronization** clock variances among mobiles. RT is determined by the ascend of the ramp and the bandwidth for the payload channel B_P (e.g. 190 MHz)

$$R_T = B_P / m$$

With the given values the number of slots per frame K results in 29 approximately. We can also compute the capacity of the ramp

$$C_R = R_T * T_R$$

which results in 2128 bit or about 5 **ATM cells**. If one considers a higher transmission rate (T_R) and/or a higher base bandwidth we will get an higher ascend of the ramp which can result in a smaller number of slots per frame and a smaller capacity of a ramp. The same is valid for a larger intergap time between the slots.

2.2 RNET MAC access control

The **access control** is based on a completely **distributed** mechanism. We assume that every mobile's **MAC** is associated with a buffer.

A mobile which wants to transmit data by means of radio packets must have sensed the header and feedback channel at least for one frame duration. Thus, the mobile is able to gather information about the receiving state of other mobiles and the availability of transmission blocks. The check of receiver state is necessary since a mobile can neither synchronize to two **frequency ramps** (see Figure 3) nor receive packets while sending.

A **transmission block** is considered to be available (not reserved), if the same block was not used in the last frame. The decision of availability can depend on the received header information, the feedback information or on both. This information differentiation for reservation evaluation purposes very much affects the channel efficiency in hidden terminal scenarios as discussed later.

If a **transmission block** is not reserved, a mobile can transmit a header on the header channel. Immediately after the header transmission follows the packet transmission on a frequency ramp (payload channel). If a mobile receives correctly a header with its own address it responds immediately with an acknowledgment in the next time slot on the feedback channel and starts packet reception. Note, the feedback is not the acknowledgment for the correct packet transmission in the payload channel. It serves as an packet sending cut-off criteria if there is no feedback. However, we assume, that a correct header reception implies the correct reception of the packet in the payload channel, because the transmission quality is considered to be stable for sufficient short time intervals.

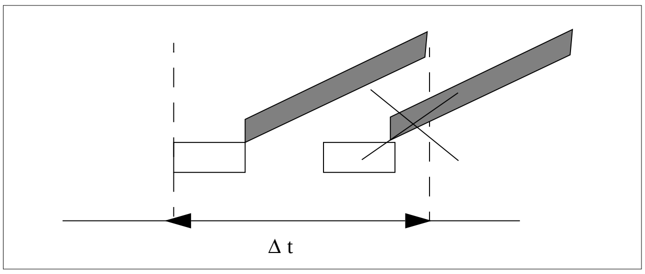

Figure 3: Single receiver station

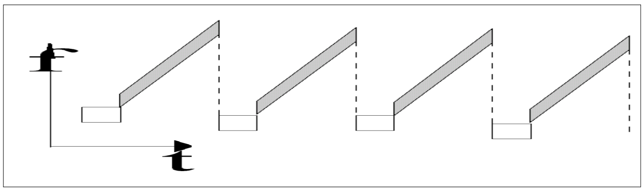

Figure 4: Temporal Channel

2.2.1 Header channels access

Header slots of non-reserved transmission blocks are accessed in a **Slotted Aloha** principle. In any error case (e.g. collision of header, header/feedback get lost ...) the mobile switches to a backlog state and has to wait k free slots whereby k is uniformly **distributed** in a certain interval. The error case is determined if the feedback is not received. The mobile will chose the first non-reserved header slot (1 persistent) for transmission.

2.2.2 Temporal channel concept

After a successful header transmission determined by the reception of the feedback a mobile has reserved automatically the same **transmission block** in the next frame. This extends the header channel access to an **Reservation Aloha** based scheme. Thus, a mobile is able to transmit constantly at a certain data rate (see Figure 4). In other words, if a mobile has gained access for one **transmission block**, it has **contention free** access to the same **transmission block** in all consecutive frames as long as there is data to transmit. This reservation is lost if the mobile has nothing to send and the header slot remains empty.

As an design option, an extended reservation scheme is possible. For instance, only in every second (third ...) frame the same slot is reserved. This would fit very well for low bit rate CBR traffic.

The concept of temporal channels can provide a very good bandwidth utilization in high load cases. However, an increasing load per mobiles can result in a very unfair bandwidth sharing!

2.2.3 Channel Holding Time (CHT)

Since the temporal channel concept causes unfairness under high system load conditions, a **Channel Holding Time**is introduced. The **Channel Holding Time** defines the number of packets, which may be sent consecutively by a mobile. With respect to the header channel access, the CHT defines the maximum number of **transmission block** reservations without disruption.

As an design option the channel holding time could be determined in a **distributed** dynamic fashion by every station. For instance CHT could be set with respect to **QoS** requirements of certain applications or due to the evaluation of network load condition. The latter could be done on the feedback channel information.

2.2.4 STOP Bit

A header slot plus the ramp in a frame can be used if no reservation exists. That is, the header slot has to be free (not used) at least in the last frame. This leads to waste of bandwidth. To eliminate this bandwidth waste, a stop or a follow bit in the header can be used to declare the next header as non-reserved. The stop/follow bit has to be switched on/off if the temporal channel holding time is over.

2.3 Access control in hidden terminal scenarios

In a scenario, where each mobile is in coverage distance of another mobile the evaluation of the slot reservation is quite simple, since all the header and feedback information is globally available. In a hidden terminal scenario it makes a difference as to which information (header, feedback or both) is taken in order to decide whether a **transmission block** is reserved. As known from the literature [2, 5] an evaluation based on receiver information should be preferred for the following reasons. The slot reservation evaluation is only done at the sending mobile. A receiving mobile responds in every case if the received header was correct. In case the header is not correct, as the header slot would be in use, the receiving mobile will not respond (see Figure 5 c).

2.3.1 Slot reservation evaluation based on header information

The main drawback of this method is waste of bandwidth and higher collision risk. The reason for the waste of bandwidth is, that the bandwidth is reserved around the sender. No other mobile hearing a header signal in a certain slot is allowed to use the slot although there would not be any contention at the receiver (see Figure 5 b). The higher collision risk results from the lack of information about slots already in use, if a mobile can hear the receiver but not the sender (see Figure 5 d).

2.3.2 Slot reservation evaluation based on header and feedback information

This method will reduce the problem of a higher collision risk, but the problem of wasting bandwidth still remains.

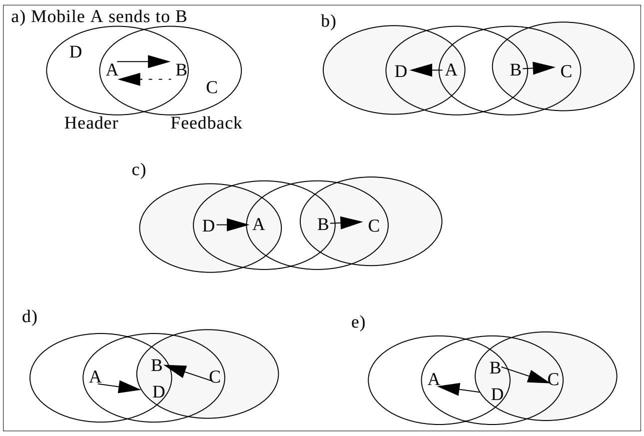

Figure 5: a) Coverage of header and feedback signals, b) c) d) hidden terminal scenarios

2.3.3 Slot reservation evaluation based on feedback information

If the feedback information is taken to evaluate the slot reservation then only a bandwidth reservation around the receiver will happen. This results in an optimal use of bandwidth since the sender does not care about contention in his coverage area if other mobiles transmit to mobiles outside its area (compare Figure 5 b, d). The problem which arises from this method is the chance of feedback contention. The sending mobile is not able to receive a correct feedback signal (see Figure 5 d, e).

Therefore, we assume the reception of any feedback signal (e.g. a burst signal) is a sufficient assumption of reception of the header signal at the receiver. In case the feedback signal contention is caused by more than 3 receivers[3], there is still a good chance of a proper header signal reception at the mobiles which justify the assumption made above. Furthermore, the corresponding header slot in the next frame is considered to be free if there is no feedback.

2.4 Time synchronization

Since **RNET MAC** uses a **TDMA** scheme, a basic working condition is the time **synchronization** of all mobiles. There is no explicit **synchronization** signal defined in **RNET MAC**. For **synchronization** purposes header and/or feedback signals are taken into account. Local timers are adjusted with the reception of every header or feedback transmission. To encompass signal delays and variances of clocks, an appropriate intergap time between two consecutive slots must be found. There are well known problems in **distributed** timing control related with time shift and erroneous timing of mobiles. Even the **hidden terminal** scenario's may have influences on timing control because of distance and possible different **synchronisation** signals for a mobile. For some of these problems there are several **distributed** time control solutions proposed in the literature (e.g. [1, 3]). A further evaluation of that topic is necessary.

3 PERFORMANCE RESULTS

The performance of **RNET MAC** was evaluated by **simulation**. For the sake of simplicity we assume no link errors and synchronicity of mobiles. Further a frame consists of 5 slots and a message consists of a fixed number of packets, each filling up a complete ramp (about 300 bits).

The **simulation** scenario are shown in Figure 6. First, we took a **fully meshed** scenario, where all mobiles cover each other. There are 20 stations whereby mobiles 1,2 ... 10 send to mobiles 11,12 ... 20. This has the effect that the mobile compete only for channel access. Contention at the receiver is excluded (see section II B.). A sender has not to wait for the end of another ongoing transmission to the chosen receiver, since there is always only one sender for a certain receiver. We took this scenario to exploit the potential of the access mechanism.

Second, we took a **hidden terminal** scenario, where mobile 1 is out of reception distance of mobile 10 and vice versa, but the rest of mobiles are able to receive signals from all other mobiles including 1 and 10. We made again the assumptions of single sender/receiver pairs.

Third, we simulated a **client server** scenario. As outlined in section III B., a mobile can only send data to another mobile if this mobile is not receiving or sending data. Since the receiving phase is exactly one frame, the mobile can again receive data only when this phase is over. As this phase can be longer than one frame (CHT > 1) and/or other mobiles get access first, the waiting time can be very long. To evaluate such a scenario we took a fully meshed scenario, where 9 mobiles (clients) requesting access to 1 mobile (server).

3.1 Fully meshed working scenario

3.1.1 RNET MAC with/without stop bit

To investigate the influence of the stop bit, the channel holding time (CHT) was varied. Every mobile worked under full load, that is, at any point in time there was a message in the **MAC** buffer. A message consists of one packet.

[3] If there is a feedback contention caused by 2 mobiles then clearly both response to different but correct header receptions. If two or more mobiles are involved in feedback contention then at least 2 mobiles are responding to a correct header reception.

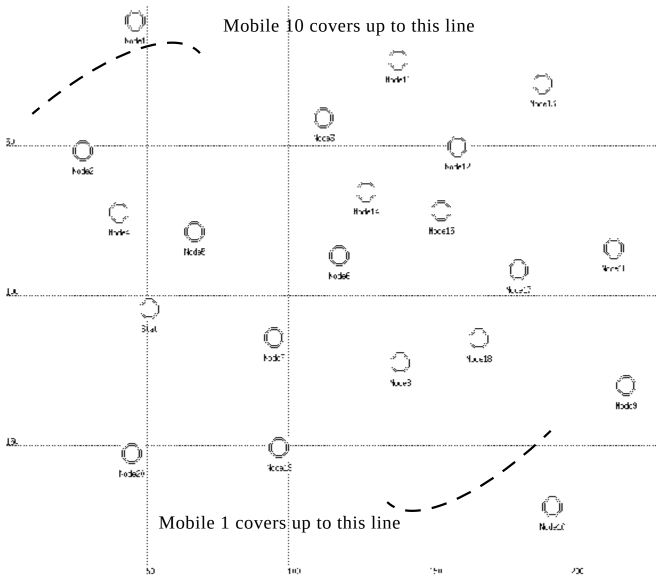

Figure 6: Simulation scenario: 20 mobiles in a 200 x 250 meter cell

As anticipated, the use of the stop bit has advantages (Figure A.1). The advantage is lowered as the CHT becomes larger. If the CHT is infinite, the throughput of both curves would be equal. But as the CHT increases the access fairness decreases. In a more realistic region of 1 to 20 CHT the gain of the stop bit is around 10%. Furthermore, at CHT 1 the performance of **RNET MAC** is comparable to **slotted Aloha** under full load. In case no stop bit is used, the performance is lower because one slot is wasted after every transmission. In all following simulations the stop bit is used and CHT is set to 5.

3.1.2 RNET MAC burst handling (handling of short and long messages)

Beside CHT, the length of the messages have to be considered. This has consequences on the throughput, queueing delay and collision rate. To evaluate the influence of the message length, two constant message sizes are chosen (1 packet per message and 5 packets per message (burst case)) and the load is varied. That is, the interfamily time between to consecutive messages is larger for messages consisting of 5 packets than for messages consisting of one packet at the same load.

At a first glimpse, the throughput curves (Figure A.2) are very similar even though the curves of collisions (Figure A.3) show a completely different behavior. There are only some marginal throughput differences in the load region from 0.75 till 1.0.

The collision rate grows rapidly for short messages and stabilizes at a load of one. For long messages, the collision rate increases slower than for short messages and stabilizes at the same points as for short messages. The reason for the similar throughput curves are based on the following facts: In lower load regions there is enough capacity to resolve collisions of short (as well as long) messages. Therefore, there is no performance degradation for short messages. As the load increases, the short messages are treated by **RNET MAC** as long messages since the probability, that the buffer is not empty and a temporal channel is built up is higher. This effect reduces the collision rate for short messages, if the load becomes more than 0.5.

As anticipated, the queueing delays (Figure A.4) are slightly shorter for short messages in lower load regions (<0.5) since packets do not have to wait very long for transmission. For messages consisting of 5 packets, the 5th packet has to wait at least 4 slots before it is transmitted. There is a break point at a throughput of around 50%. The delay here is around 27 slots. After this point the delay for short messages will grow exponentially because of retries resulting from a large number of collisions and later on through the infinity of buffer space. Delays for long messages remains stable for loads up to 65% and also grows exponentially afterwards.

From the curves above, one can conclude that **RNET MAC** shows a nearly equal handling of short and long messages. Long messages only have advantages in terms of delays in higher load regions.

3.2 Hidden terminal scenario

For the sake of comparability the fully meshed is chosen. There is only one change: mobile 1 (left upper edge) and mobile 10 (right lower edge) are hidden from each other. The objective of the **simulation** is to show how **RNET MAC** behaves in a **hidden terminal** scenario; header information or feedback information is used to evaluate reservation of slots. As shown in [14] for the **IEEE 802.11 MAC** protocol[4] with a similar network configuration, hidden terminals can lead to an excessive degradation of the performance.

As anticipated in section II C, header slot reservation based only on feedback (HSRF) information outperforms header slot reservation, which is only based on header information (HSRH) (Figure A.5). The reason for that is two-fold.

[4]It should be mentioned, that the **IEEE 802.11 MAC** protocol has an optional feature to avoid collision through a per packet bandwidth reservation mechanism. This mechanism is based on two additional signalling messages: RTS (ready to send - bandwidth reservation around sender) and CTS (clear to send - bandwidth reservation around the receiver). RTS is comparable to the header information and CTS is comparable to the feedback information in RNET.

First, HSRH is not able to resolve collisions between the two hidden mobiles because the feedback information is not evaluated. Clearly this results in a higher collision rate (Figure A.6). Second, if two mobiles start the transmission at the same time even in the full coverage area (mobile 2-9, 11-20) they cannot recognize them. As outlined before, one can find more basic scenarios which show the superiority of HSRF.

If the performance results from section III A are compared with the results of HSRF one can only find marginal differences even though there are hidden terminals. **RNET** behaves nearly optimally in **hidden terminal** environments.

3.3 Client server scenario

As shown in the results, the performance of **RNET MAC** in this scenario (9 mobiles sends to one server) is relatively poor. Therefore other concepts like ramp sharing or cutting will be investigated. It should be mentioned that this effect is improved, if the server sends data to the mobiles as there is no concurrence (collision) and the clients could be served one after the other. However, data would still have to wait in the transmit queue.

A comparison with the throughput curve of the fully meshed scenario from section III A (compare Figure A.2 and A.8) shows a throughput degradation of about 70%. Less than 20% of the available bandwidth is used. But there is still a positive point; the maximum achievable throughput in this specific configuration is about 20% which is nearly achieved.

In contrast to the scenario chosen in section III A and III B, the collision rate goes up very early and to a high level (compare to Figure A.3 and A.6). Also the queueing delay (Figure A.10) becomes infinite at a low load level. This is because **RNET MAC** is only able to serve at maximum 1 slot per frame (about 20% throughput) in this network configuration. If the load increase above this level the queueing delay grows infinitely.

The simple idea of multiple transmitter/receiver at the server can improve the performance of **RNET MAC** in the **client server** scenario substantially. This is justified by the assumption, that a server is always more expensive than "simple" mobiles.

4 CONCLUSIONS

We presented a novel **MAC** protocol, which features point-to-point communication and **distributed access control**. The features of this **MAC** as well as performance results have shown, that **RNET MAC** is applicable in principle in **WATM** environments.

The **distributed access control** of **RNET MAC** results in a good applicability for environments without pre-installed infrastructure as for instance base stations, which are the more expensive part wireless networks with centralized control. However, centralized control reduce the functionality of mobiles, which makes them less expensive. Also, the implementation of **QoS** control is easier. The point-to-point communication paradigm improves the bandwidth efficiency, because communication via a BS divides the available bandwidth by the factor of two. **RNET MAC** is flexible with respect to packet sizes. According to the chosen parameter (for instant ascend of ramp), a ramp can carry a packet with an arbitrary number of bits. For the sake of transparency the ramp size should be a multiple of an ATM size. As well as **WATM MAC** based on centralized control, **RNET MAC** supports ATM **QoS** enforcement by means of reservation and the channel holding time. In particular, CBR services as well as efficient support of best effort services (UBR) are possible. **RNET MAC** gives also indirect means for error control. In case of collisions, a retransmission will be executed. Note, that **RNET MAC** bases on the assumption, that correct header reception implies a correct reception of data. That is valid for sufficient short time intervals only.

Notes

1. THIS WORK HAS BEEN SUPPORTED BY A GRANT FROM THE BMBF (GERMAN MINISTRY FOR SCIENCE AND TECHNOLOGY) WITHIN THE PRIORITY PROGRAM ATM*MOBIL*.

2. Normally, this entity is called bases station (BS) or access point (AP).

3. HIgh PERformance LAN type 2 is a Wireless LAN with ATM capabilities

4. European Telecommunications Standards Institute

5. An extreme case would be an infinite ascend (TDM) which would require cost intensive broadband transmitter/receiver. An ascend with the value zero would result in an FDM scheme.

6. The values are hypothetical and subject to change.

References

[1] Arvind, K.: "Probabilistic Clock Synchronization in Distributed Systems", IEEE Transactions on Parallel and Distributed Systems, Vol. 5, No. 5, pp. 474-487.

[2] Bharghavan, V et al.: "**MAC**AW - A Media Access Protocol for Wireless LAN's", Proceedings of SIGCOM'94, 1994

[4]For a more detailed version see [4]

[3] Chuang, Justin C.-I.: "Autonomous Time Synchronization Among Radio Ports in Wireless Personal Communications", IEEE Transactions on vehicular technology, Vol. 43, No. 1, February 1994, pp. 27-32.

[4] Holtkamp, R.,Ebert, J.-P., Wolisz, A., Ramel, L.: "A Distributed Media Access Control (D**MAC**) for Wireless ATM Networks", accepted at 5th Intl. Conf. on Telecommunication Systems, Modeling and Analysis, Nashville, March 1997

[5] Phil Karn: " **MAC**A - A New Channel Access Method for Packet Radio", AHHL/CHHL Amateur Radio 9th Computer Networking Conference, Sept. 22, 1990

[6] M. J. Karol, Z. Liu and K.Y. Eng: "An efficient demand-assignment multiple access protocol for wireless packet (ATM) networks", Wireless Networks, Baltzer Science Publishers, vol. 1, no. III, 1995

[7] D. Petras: "Medium Access Control Protocol for wireless, transparent ATM access", In IEEE Wireless Communication Systems Symposium, Long Island, NY, November 1995

[8] John Porter, Andy Hopper: "An ATM based protocol for Wireless LANs", ORL Technical Report 94.2, 1994

[9] John Porter et al.: "The ORL Radio ATM System, Architecture and Implementation", ORL Technical Report, 1996

[10] Ramel, Louis: "Contribution to an Hiperlan (17,2 GHz)", ETSI RES 10 Conference-Paper, Thomson / CSF-CNI, Nizza, 6.-9. June 1995

[11] Raychaudhuri, D. et al.. (1996): "WATMnet: A Prototype Wireless ATM System for Multimedia Personal Communication", Proc. of ICC96, pp. 469-477

[12] Smulders, P.f.M., Blondia, C.: Application of the Asynchronous transfer Mode in Indoor Radio Networks", Proc. of Joint PIMRC and WCN conference 94, The Hague, pp. 839, Netherlands,1994

[13] IEEE Personal Communications Magazine: Special Issue on "Wireless ATM", vol.3, no.4, August 1996

[14] Jost Weinmiller, Hagen Woesner, Jean-Pierre Ebert, Adam Wolisz: "Analyzing the RTS/CTS Mechanism in the DFW**MAC** Media Access Protocol for Wireless LAN's", IFIP TC6 Workshop Personal Wireless Communications (Wireless Local Access), Praque, April '95

Appendix: Performance figures

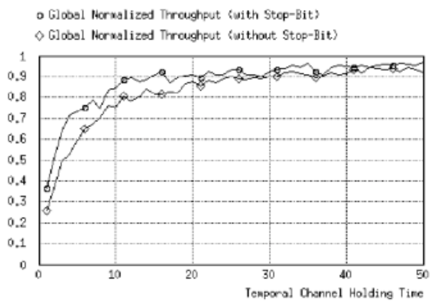

Figure A.1: Throughput vs. CHT under full load - Use of stop bit

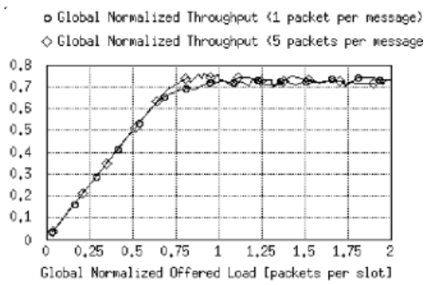

Figure A.2: Throughput vs. load for long and short messages

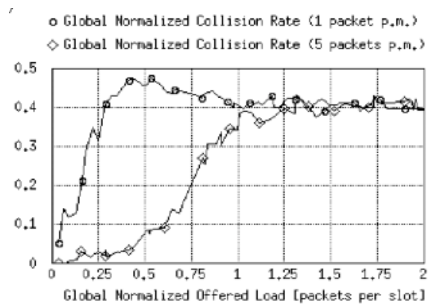

Figure A.3: Collision rate vs. load for long and short messages

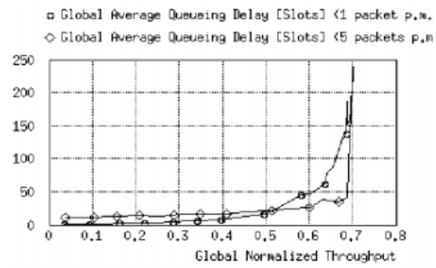

Figure A.4: Queueing delay vs. load for long and short messages

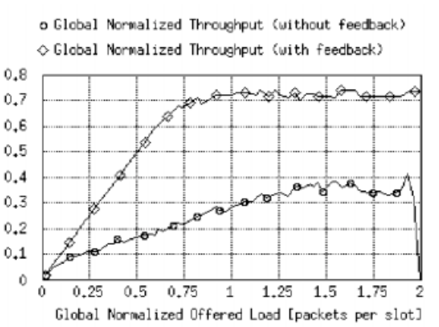

Figure A.5: Throughput vs. load - feedback or header use for slot reservation evaluation

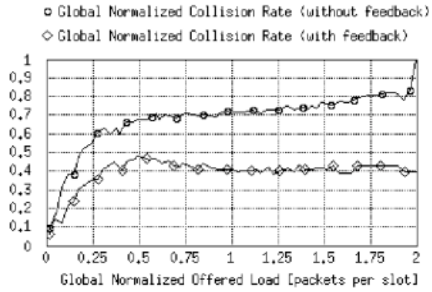

Figure A.6: Collision rate vs. load - feedback or header use for slot reservation evaluation

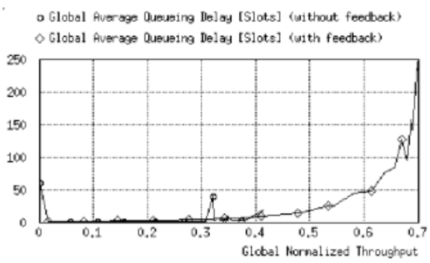

Figure A.7: Queueing delay vs. load - feedback or header use for slot reservation evaluation

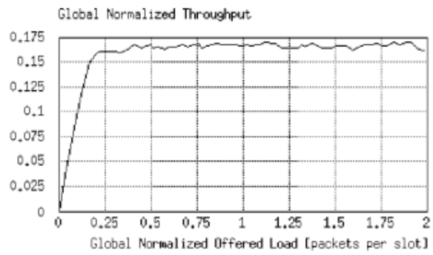

Figure A.8: Throughput vs. load in a client server scenario

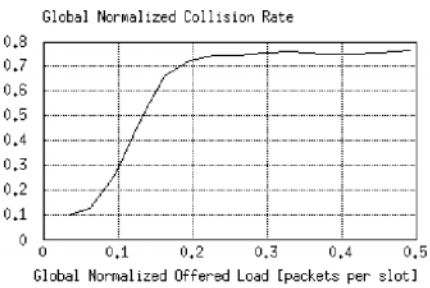

Figure A.9: Collision rate vs. load in a client server scenario

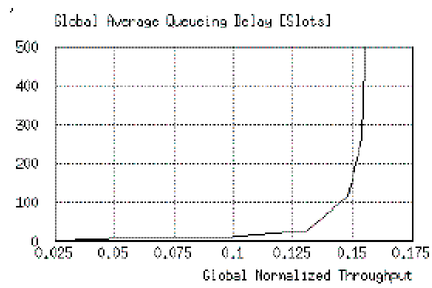

Figure A.10: Queueing delay vs. load in a client server scenario