Investigation of MAC protocol design for wireless ATM systems

Jürgen Rapp
Aachen Univ. of Technology, Communication Networks
Kopernikusstr. 16, D-52074 Aachen
E-Mail: rapp@comnets.rwth-aachen.de
WWW: http://www.comnets.rwth-aachen.de/˜rapp

Hui Li
Ericsson Eurolab Deutschland GmbH, Abteilung RA
Nordost Park 12, D-90411 Nuernberg
E-Mail: Hui.Li@eedn.ericsson.se

Abstract — **Growing demands for bandwidth as well as mobility led to several research projects which are investigating high speed wireless ATM systems. The design of *Medium Access Control* (MAC) protocols is a key part in the research activities because of the scarce transmission capacity of radio channels. In this paper, a new MAC protocol structure is introduced, and it is shown how the performance of a wireless ATM system can be optimized by using flexible protocol data units with a low protocol overhead. The MAC protocol is compared with the DSA++protocol, and the comparison results are calculated first and used as cross check for computer simulations. The simulations were performed in a wireless ATM system which consists of various types of sources, complete *Go-back-N* (GbN) ARQ scheme and channel model with bursty error characteristic. Furthermore, possible drawbacks of the new proposal are investigated.**

I. Introduction

Growing demands for bandwidth as well as mobility led to several research projects which are investigating high speed wireless ATM systems [1]. In Europe, a wireless LAN system, called HIPERLAN/2, will be standardized. It can be used in combination with e.g. ATM or TCP/IP and as part of UMTS. High data rates (up to 25 Mbit/s net rate) should be provided on an error prone radio link while guaranteeing the requested *Quality of Service* (QoS) like maximum allowed delay and *Cell Loss Ratio* (CLR). Mostly responsible for that task is the *Data Link Control* (DLC) layer which lays between physical layer and ATM layer. The DLC layer is further subdivided into *Logical Link Control* (LLC) layer and MAC layer. In the LLC layer, error control functions are performed, e.g. standard ARQ protocols like GbN or Selective Repeat, while the MAC layer contains functionality to distribute the available channel capacity upon the connections. The MAC protocols which organize the data transmission over the air have to be designed in a most efficient way to maximize the utilization of the scarce resource radio link.

It has been decided that the TDMA/TDD transmission scheme will be adopted in the HIPERLAN/2 system. Several protocol variants for the DLC layer of the HIPERLAN/2 system have been proposed and one of them is DSA++. The DSA++protocol was first proposed as DSA in 1994 [2] and an enhanced version was presented 1995 as DSA++[3], which is currently being implemented in the german research project ATM*mobil*. In this paper the DSA++protocol is compared with a new MAC protocol which is called *Cell Train Based Transmission of Signalling and Data* (CTBTSD) protocol.

This paper is written with the following structure. The scheduling algorithm that is used for the performance simulation of the DSA++and CTBTSD is described first. Then, the protocol structures of the DSA++and CTBTSD are shown and the throughput of both protocols in the worst case and best case is calculated. Based on the simulated performance for a multimedia scenario, a realistic comparison of the DSA++and CTBTSD is performed. Finally, the benefits and the possible drawbacks of the CTBTSD approach are concluded.

II. Scheduling Algorithm

The scheduling algorithm used in the following investigation is given as follows.

The base station serves as central controller and determines how much capacity is given to the different connections. As criteria for the scheduling exist static and dynamic values. First of all, every connection is associated to one of four service categories which are CBR, VBR, ABR and UBR, with CBR having the highest and UBR having the lowest static priority. Their names are chosen with respect to the definitions in the Traffic Management Specification 4.0 of the ATM Forum. The scheduler provides capacity to the connections first according to their static priority. This means that a request with CBR priority is always preferred before a request with VBR priority. Only if there are requests with same static priority are the dynamic priority values taken into account. These dynamic values consist of a due date and some values indicating the amount of data packets that are waiting. The due date is derived from the maximum allowed delay of a connection. The priority of a connection becomes the higher, the nearer it is to its maximum allowed delay. How much capacity is scheduled depends on how many data packets are waiting. The base station has perfect knowledge about the status of its own queues but has to be informed about the status of the wireless terminals. This is done by so called dynamic parameters which are transmitted from the wireless terminals to the base station.

The result of the scheduling is broadcasted to the wireless terminals to inform them about when the base station will send to them and when they are allowed to send. Such a broadcast PDU is called Period Control (PCTRL) PDU. A period, or MAC Frame, is the interval from one broadcast PDU to the next broadcast PDU and its structure is shown in fig. 1.

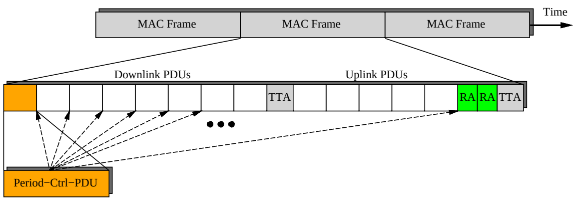

Fig. 1: Structure of a MAC period in a TDMA/TDD based system

After the PCTRL PDU comes the downlink transmission phase in which the base station sends to the wireless terminals. After a *Transceiver Turn Around time* (TTA) follows the uplink transmission phase in which the wireless terminals are allowed to send. Thereafter come one or more short time slots which can be used for random access. After another transceiver turn around time begins the next period with the next PCTRL-PDU. The length of the downlink and uplink transmission phase is variable and depends on the capacity requests of the connections. The maximum length of a period is set by a system parameter and can be chosen freely.

The downlink and uplink transmission phases consist of PDUs that can carry data and signalling. The structure of the PDUs is an important design issue of MAC protocols.

III. DSA++ and CTBTSD MAC protocols

In the DSA++ protocol the following PDU types are used (cf. fig. 2).

Down- and UpInfo PDUs are able to carry one ATM cell payload. To identify the connection this data packet belongs to, an LLC header is included. Furthermore, there is capacity to transmit an acknowledgement. In uplink direction, in addition to the acknowledgement, dynamic parameters have to be transmitted. There exists also a short PDU which is called UpSig-PDU. It is used for random access or in uplink direction if there is no data packet to transmit but only an acknowledgement or dynamic parameters. These fixed PDU types can lead to a waste of capacity if a terminal has more than one data packet to transmit. Let's consider the following case that a terminal has N data packets to transmit. With these fixed PDU types this leads to transmitting

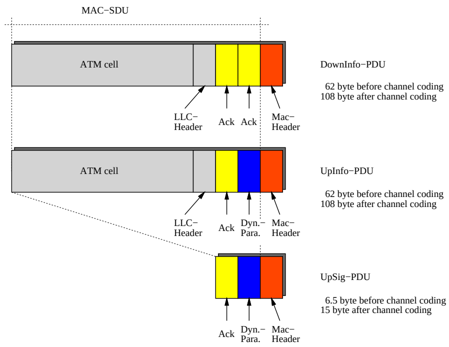

Fig. 2: PDU types in the DSA++ system

N MAC header, $(2\cdot N)$ acknowledgements and $(2\cdot N)$ LLC header. Since all data packets are addressed to the same terminal, only 1 MAC header would have been necessary. Assuming furthermore that a GbN ARQ scheme is used, only 1 acknowledgement per connection is needed. It has to be given the connection identifier of the data packet, so every data packet must have its LLC header and N LLC header are needed. Making the final calculation there is an overhead of $(N-1)$ MAC header $+$ $(2\cdot N-1)$ acknowledgements, with MAC header and acknowledgement being around 2.5 bytes. This consideration led to the following proposal [4] with a more flexible PDU structure. It is called *Cell Train Based Transmission of Signalling and Data* (CTBTSD).

Despite having fixed PDUs, certain parts of the PDUs are available separately in CTBTSD (cf. fig. 3).

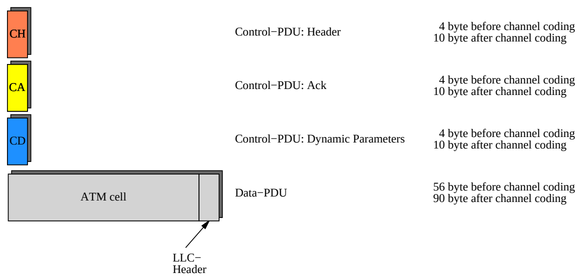

Fig. 3: PDU types for a more flexible PDU concept

First of all there is the data part which is able to carry one ATM data packet and the LLC header. In addition to the data part exist three different control PDUs which are all of the same length. They are used as MAC header, acknowledgement or for the transmission of dynamic parameters. Like when using a kit, so called cell trains can be built which only include the information that is actually needed. Examples for possible cell trains are given in fig. 4.

Because the part for channel coding can't fall below certain limits, the relation between user data and coding overhead becomes the worse the smaller the PDU becomes. That's why the length of a cell train with the same contents as a DownInfo, UpInfo or UpSig PDU is somewhat bigger than the fixed PDU type. Due to this, there exist a best and a worst case for the CTBTSD. The best case is that only one cell train for one connection is sent within the whole

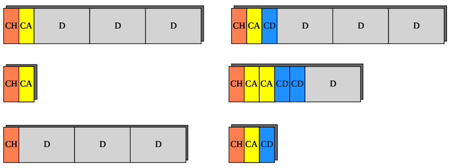

Fig. 4: Possible cell trains in CTBTSD

period. Than, only one MAC header is needed and in case a GbN-ARQ scheme is used only one acknowledgement is needed. This case is depicted in fig. 5 for both DSA++ and CTBTSD.

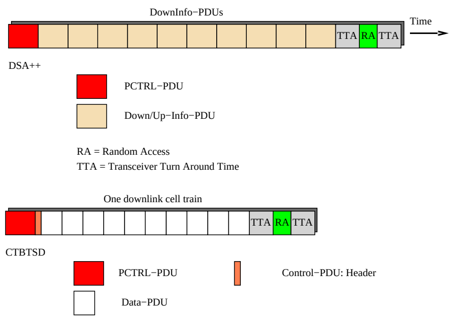

Fig. 5: Best case for CTBTSD

It is visible that less capacity is needed for CTBTSD than for DSA++. Assuming a maximum period length of 20 times the length of a PCTRL-PDU, the maximum net data throughput for DSA++ can be calculated as follows (eq. 1 to 7). Please notice that for DSA++ worst case and best case and all other cases lead to the same maximum throughput.

$$Gross\,Transmission\,Rate = 50 Mbit/s \tag{1}$$

$$Length\,of\,Period = 3.456 \cdot 10^{-4}\,s \tag{2}$$

$$\begin{aligned} & Bits\,for\,data\,transmission\,in\,DSA{++} \\ = \; & Bits_{Data\,Tr\,DSA{++}} \\ = \; & 20 \cdot Bits_{PCTRL} - Bits_{PCTRL} - 2 \cdot Bits_{TTA} \\ & - Bits_{RA\,in\,DSA{++}} \\ = \; & (20 \cdot 864 - 864 - 2 \cdot 62 - 90)\,bits = 16202\,bits \end{aligned} \tag{3}$$

$$Gross\,bits\,for\,DSA{++}\,data\,PDU = 108 \cdot 8\,bit \tag{4}$$

$$Net\,bits\,in\,DSA{++}\,data\,PDU = 53 \cdot 8\,bit \tag{5}$$

$$\begin{aligned} & Net\,bits\,for\,DSA{++}\,data\,transmission \\ = \; & Bits_{Data\,Tr\,DSA{++}} \cdot \\ & \frac{Net\,bits\,in\,DSA{++}\,data\,PDU}{Gross\,bits\,for\,DSA{++}\,data\,PDU} \\ = \; & 16202\,bit \cdot \frac{53 \cdot 8\,bit}{108 \cdot 8\,bit} = 7950\,bit \end{aligned} \tag{6}$$

$$\begin{aligned} & \mathbf{Maximum\,net\,throughput\,for\,DSA{++}} \\ = \; & \frac{Net\,Bits\,for\,DSA{++}\,data\,transmission}{Length\,of\,Period} \\ = \; & \frac{7950\,bit}{3.456 \cdot 10^{-4}} = \mathbf{23\,Mbit/s} \end{aligned} \tag{7}$$

For CTBTSD the maximum net throughput for the best case can be calculated as shown in eq. 8 to 13.

$$\begin{aligned} & Bits\,for\,data\,transmission\,in\,CTB \\ = \; & Bits_{Data\,Tr\,CTB} \\ = \; & 20 \cdot Bits_{PCTRL} - Bits_{PCTRL} - 2 \cdot Bits_{TTA} \\ & - Bits_{RA\,in\,CTB} \\ = \; & (19 \cdot 864 - 2 \cdot 62 - 240)\,bits = 16052\,bits \end{aligned} \tag{8}$$

$$\begin{aligned} & Gross\,bits\,for\,data\,transmission\,in\,CTB \\ & Best\,case \\ = \; & Bits_{Data\,Tr\,CTB\,Best\,case} \\ = \; & Bits_{Data\,Tr\,CTB} - Bits_{Header} \\ = \; & (16052 - 80)\,bit = 15972\,bit \end{aligned} \tag{9}$$

$$Gross\,bits\,for\,CTB\,data\,PDU = 90 \cdot 8\,bit \tag{10}$$

$$Net\,bits\,in\,CTB\,data\,PDU = 53 \cdot 8\,bit \tag{11}$$

$$\begin{aligned} & Net\,Bits\,for\,CTB\,data\,transmission \\ & Best\,case \\ = \; & Bits_{Data\,Tr\,CTB\,Best\,case} \\ & \cdot \frac{Net\,bits\,in\,CTB\,data\,PDU}{Gross\,bits\,for\,CTB\,data\,PDU} \\ = \; & 15972\,bit \cdot \frac{53 \cdot 8\,bit}{90 \cdot 8\,bit} = 9405\,bit \end{aligned} \tag{12}$$

$$\begin{aligned} & \mathbf{Maximum\,net\,throughput\,for\,CTBTSD} \\ & \mathbf{Best\,case} \\ = \; & \frac{Net\,Bits\,for\,CTB\,data\,transmission_{Best\,case}}{Length\,of\,Period} \\ = \; & \frac{9405\,bit}{3.456 \cdot 10^{-4}} = \mathbf{27.2\,Mbit/s} \end{aligned} \tag{13}$$

The worst case for CTBTSD is that only cell trains with length one are sent (cf. Fig. 6).

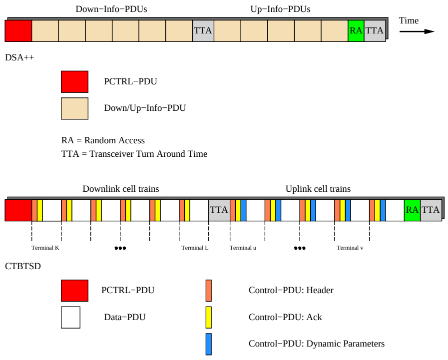

Fig. 6: Worst case for CTBTSD

This means that a little bit more capacity is needed for CTBTSD than for DSA++, as explained above. The maximum net data throughput for CTBTSD in worst case is calculated in eq. 14 to 20.

$$\begin{aligned}
& Bits\ for\ data\,transmission\,in\,CTB \qquad (14)\\
= \ & Bits_{Data\,Tr\,CTB}\\
= \ & 20 \cdot Bits_{PCTRL} - Bits_{PCTRL} - 2 \cdot Bits_{TTA}\\
& - Bits_{RA\,in\,CTB}\\
= \ & (20 \cdot 864 - 864 - 2 \cdot 62 - 240)\, bits\\
= \ & 16052\, bits
\end{aligned}$$

$$\begin{aligned}
& Gross\,bits\ for\ CTB\,uplink\,train \qquad (15)\\
& Worst\,case\\
= \ & Bits_{Uplink\,train}\\
= \ & Bits_{Header} + Bits_{Ack} + Bits_{DynP} + Bits_{Data\,PDU}\\
= \ & (80 + 80 + 80 + 90 \cdot 8)\, bit = 960\, bit
\end{aligned}$$

$$\begin{aligned}
& Gross\,bits\ for\ CTB\,downlink\,train \qquad (16)\\
& Worst\,case\\
= \ & Bits_{Downlink\,train}\\
= \ & Bits_{Header} + Bits_{Ack} + Bits_{Data\,PDU}\\
= \ & (80 + 80 + 90 \cdot 8)\, bit = 880\, bit
\end{aligned}$$

$$\begin{aligned}
& Average\,gross\,bits\ for\ CTB\,cell\,train \qquad (17)\\
& Worst\,case\\
= \ & Gross\,Bits_{Cell\,train\,Worst\,case}\\
= \ & \frac{Bits_{Uplink\,train} + Bits_{Downlink\,train}}{2}\\
= \ & \frac{960\,bit + 880\,bit}{2} = 920\,bit
\end{aligned}$$

$$\begin{aligned}
& Net\,Bits\,in\,CTB\,cell\,train_{Worst\,case} \qquad (18)\\
= \ & 53 \cdot 8\,bit = 424\,bit
\end{aligned}$$

$$\begin{aligned}
& Net\,bits\ for\ data\,transmission\,in\,CTB \qquad (19)\\
& Worst\,case\\
= \ & Bits_{Data\,Tr\,CTB}\\
& \cdot \frac{Net\,Bits\,in\,CTB\,cell\,train_{Worst\,case}}{Gross\,Bits_{Cell\,train\,Worst\,case}}\\
= \ & 16052\,bit \cdot \frac{424\,bit}{920\,bit} = 7398\,bit
\end{aligned}$$

$$\begin{aligned}
& \mathbf{Maximum\ net\ throughput\ for\ CTBTSD} \qquad (20)\\
& \mathbf{Worst\ case}\\
= \ & \frac{Net\,Bits\ for\ CTB\,data\,transmission_{Worstcase}}{Length\ of\ Period}\\
= \ & \frac{7398\,bit}{3.456 \cdot 10^{-4}} = \mathbf{21.4\,Mbit/s}
\end{aligned}$$

These calculation are checked against computer simulation in the following and the behaviour of the two concepts is investigated for more realistic scenarios.

IV. Simulative investigation

A. Maximum net data throughput

A number of simulation has been carried out to verify the theoretical considerations and to analyse the proposed system in realistic scenarios [5]. The scheduling is as described above and a GbN ARQ scheme is used.

At first the theoretical considerations with regard to best and worst case were checked against computer simulations. In the best case scenario it was assumed that one terminal fills the whole period with one cell train and is thereby able to reach a maximum net data throughput of 27.2 Mbit/s. This scenario is modelled by one CBR source which is connected to the base station and a wireless terminal without any source. This means that the whole load in the system is produced by this one source. The channel error rate was set to zero to make the results comparable to the theoretical considerations.

In the worst case, only cell trains with a length of one are transmitted. As approximation to this, a scenario with 20 CBR sources was used. 10 sources were connected to the base station and 10 sources to the wireless terminals so that there are 10 bidirectional connections. As simulative results show, the mean cell train length on the uplink is around 1.01 and on the downlink around 0.87 which seems well enough as approximation to the worst case. As can be seen in fig. 7 and table I the results fit fine with the theoretical values.

To investigate to performance of CTBTSD under a more realistic scenario the following setting was used to model a multimedia scenario (cf. fig. 8 and table II). The variable F is used to vary the load in the simulations. A gilbert channel with bursty error characteristic was used [7] and the channel error rate was set to 2.5%. The load was varied as shown in the diagrams.

TABLE II: Parameter setting of multimedia scenario

Downlink				Uplink		
Typ	#	Data rate	Max. allowed delay	Data rate	#	Typ
Video	2	2 MBit/s $\cdot F$	10 ms	2 Mbit/s $\cdot F$	2	Video
Poisson	3	1.7 MBit/s $\cdot F$	n/a	1.7 Mbit/s $\cdot F$	3	Poisson
Poisson	1	1.7 MBit/s $\cdot F$	n/a	0.064 Mbit/s $\cdot F$	1	Poisson
Poisson	1	0.064 MBit/s $\cdot F$	n/a	1.7 Mbit/s $\cdot F$	1	Poisson
CBR	3	0.064 MBit/s $\cdot F$	5 ms	0.064 Mbit/s $\cdot F$	3	CBR

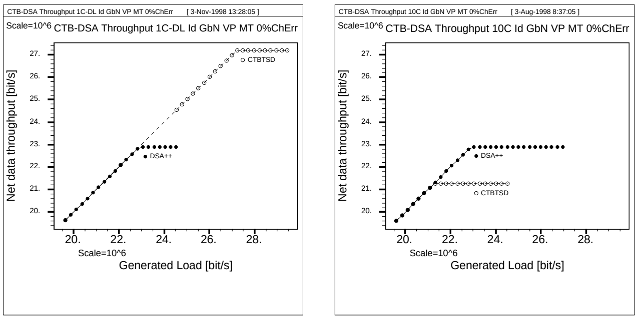

Fig. 7: Maximum net data throughput for best and worst case

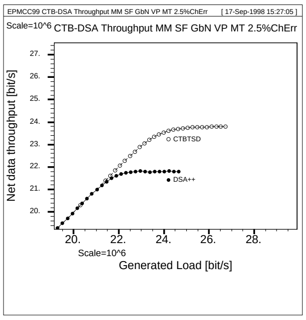

Fig. 9: Maximum net data throughput for multimedia scenario

TABLE I
Comparison of throuphput values

Maximum net data throughput [Mbit/s]				
	Best case		Worst case	
	DSA++	CTBTSD	DSA++	CTBTSD
Calculated	23	27.2	23	21.4
Simulated	22.9	27.2	22.9	21.3

It can be seen that CTBTSD is able to achieve a significantly higher throughput under a more realistic scenario, too. The load values are given in Mbit/s and amount to 23.6 Mbit/s for CTBTSD and 22 Mbit/s for DSA++(cf. fig. 9).

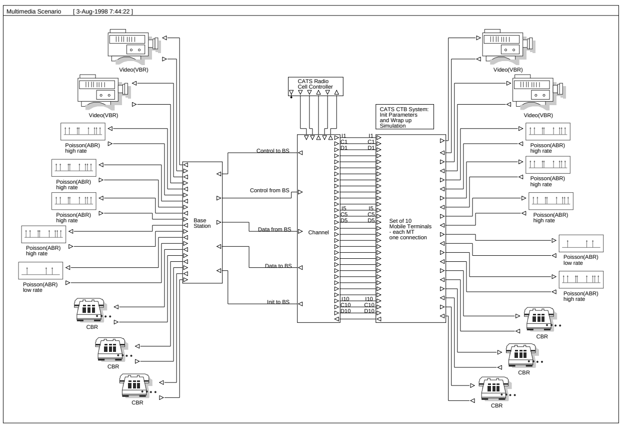

Fig. 8: Multimedia scenario

B. Comparison of delays

Investigating not only the maximum achievable throughput but also the performance with respect to delays, multimedia simulations over a wide range of load values have been carried out. They show that CTBTSD has slightly better delays for all service categories as well in uplink as in downlink direction. Together with the results for maximum throughput it can be seen that CTBTSD does not only achieve a higher maximum throughput but even is able to achieve better delays. For reasons of limited space, only uplink delays are shown but results for downlink are similar. Furthermore, to make comparisons between the different service categories possible, the same scale for abscissa and ordinate is chosen for all diagrams. Please note that delays are given in ms and the generated load in Mbit/s.

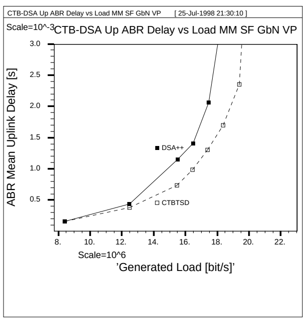

Fig. 10: ABR mean uplink delay

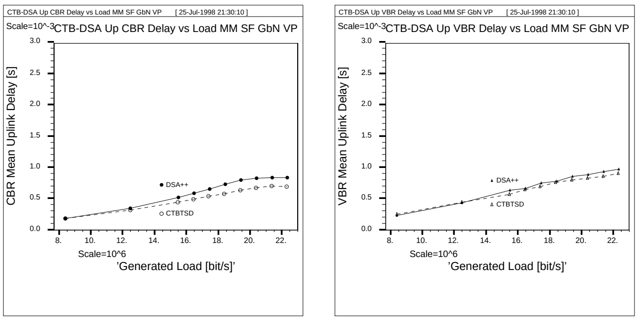

Fig. 11: CBR and VBR mean uplink delay

C. Resilience against channel errors

Another point that is investigated is how sensitive the two variants react on channel errors.

In long cell trains of CTBTSD exists only one MAC header. If the MAC header gets lost due to channel errors the whole train gets lost.

Also important for the performance of the systems is the reception of acknowledgements. In DSA++there is at least one acknowledgement in every Up- or DownInfo PDU. If, for example, three PDUs are sent for one connection and one of them gets lost, there are still two acknowledgements that reach the receiver. In CTBTSD only one acknowledgement per GbN connection is transmitted and no acknowledgement is received if this acknowledgement gets lost.

The resiliance against channel errors was investigated for several scenarios and the multimedia scenario is shown exemplarily. The load is set to 17.8 Mbit/s and the channel error rate is varied from 0% to 15%. As before a gilbert channel model with bursty error characteristic is used.The mean time between two fades is set to 20ms and the mean length of a fade equals *mean time between fades* $\cdot$ *channel error rate*.

For a gilbert channel model the resilience of CTBTSD against channel errors is equal to DSA++or even better (cf. fig. 12).

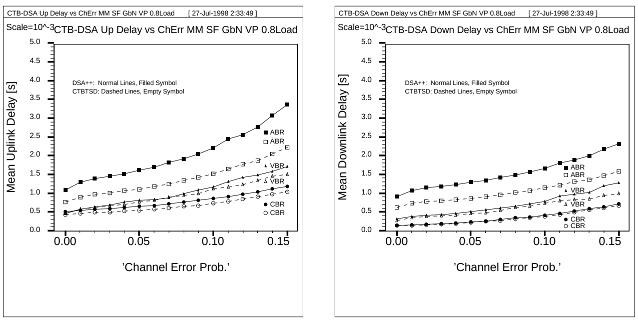

Fig. 12: Up-and downlink delays vs channel error

This is due to the fact that in this channel model not single independent PDUs are corrupted but all PDUs within an error burst. Therefore, the critical case that only a header or an acknowledgement is corrupted does hardly happen. Furthermore, after the channel is again in state good, CTBTSD is able to achieve higher maximum throughput than DSA++ and thereby is able to do the necessary retransmissions more easily.

V. Summary and conclusions

Exemplarily for a variety of proposals for a wireless ATM system, it was shown how the maximum net data throughput can be significantly increased. This is done by introducing a flexible PDU concept that helps minimizing protocol overhead. The concept was analysed analytically and the calculations were used as cross check for computer simulations which fit fine with the theoretical results. As simulation results showed for a multimedia scenario with a variety of sources, GbN ARQ scheme and 2.5% channel error rate, the new proposal increases throughput in a more realistic scenario also. Not only is the throughput increased but delays are improved, too. Only if in a scenario there are exclusively terminals that have at maximum one ATM cell for transportation does the new proposal have a worse throughput than the previous proposal. But this case seems rather unlikely in reality since an ATM cell is only 48 bytes of payload and many applications will have more data to transmit when their time for transmission has come. It was also investigated if the resilience against channel errors is a critical factor but this was not observed for a channel with bursty error characteristic. This, of course, has to be checked for more realistic channel models.

VI. Acknowledgements

This work has been done within the german ATM*mobil* project which is partly funded by the German Federal Ministry of Education, Science, Research and Technology. The authors would like to thank Prof. Walke and all colleagues for their suggestions. Furthermore, this work would not have been possible without the hard work and enthusiasm of the students working on their diploma thesis.

VII. References

[1] W. Franz, K. Ben Younes, C. Sinner, and R. Sigle, “Drahtlose ATM-Netze auf dem Weg zum Standard,” *NTZ*, pp. 52–56, July 97.

[2] D. Petras, “Performance Evaluation of Medium Access Control Schemes for Mobile Broadband Systems,” in *DMR VI*, (Stockholm, S), pp. 255 – 261, June 1994. available at http://www.comnets.rwth-aachen.de/˜petras.

[3] D. Petras, “Medium Access Control Protocol for wireless, transparent ATM access,” in *IEEE Wireless Communication Systems Symposium*, (Long Island, NY), pp. 79–84, Nov. 1995. available at http://www.comnets.rwth-aachen.de/˜petras.

[4] H. Li, J. Khun-Jush, J. Thielecke, and G. Bautz, “A MAC Protocol Based on Three PDU Types,” tech. rep., ETSI BRAN, WG3, Temporary Document 72, 1997.

[5] P. Laing, “Implementation and Examination of Methods for Guaranteeing the QoS in Wireless ATM Systems,” Master’s thesis, Communication Networks, Aachen University of Technology, July 1998.

[6] M. Nomura et al., “Basic Characteristics of Variable Rate Video Coding in ATM Environment,” *IEEE J. Select. Areas Commun.*, vol. 7, pp. 752–760, June 1989.

[7] L. Kanal and A. Sastry, “Models for Channels with Memory and their Applications to Error Control,” *Proc. IEEE vol.66*, pp. 724–744, July 78.

[8] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1987.