IEEE 802.11 Wireless Access Methods and Physical Layer Specifications

Title: Analyzing the RTS/CTS Mechanism

Date: July 1995

Authors: J.P. Ebert, J. Weinmiller, H. Woesner, A. Wolisz[1]
Institut für Fernmeldetechnik
Technische Universität Berlin
Einsteinufer 25
10587 Berlin, Germany

Abstract

The distributed coordination function that is used as the access scheme for asynchronous traffic optionally applies the RTS/CTS message exchange to solve the 'hidden terminal' scenario. In this paper we show the effectiveness of this mechanism in various cases like hidden terminals and fully meshed, optional (distributed) use of RTS/CTS and the dependencies from various parameters like physical preambles and packet sizes. We conclude this paper with a discussion of the results drawn from the simulations and an outline of problems that have yet to be investigated.

[1] also with GMD FOKUS - STEP, Hardenbergplatz 2, Berlin 10587, Germany

1. Introduction

The distributed coordination function (DCF) of the IEEE 802.11 draft standard shares access to the medium based on a CSMA/CA scheme that is extended with a RTS/CTS message exchange to improve performance in two aspects. If collisions occur, they occur with smaller packets, therefore the loss bandwidth is smaller compared to a collision of long packets. The main purpose for the integration of the mechanism into DFWMAC however is the solution of the hidden terminal problem: The successful exchange of small messages - RTS (Request To Send) sent by the sender and CTS (Clear To Send) - reserves the area within range of the receiver and the sender for the intended transmission guaranteeing undisturbed media for the longer data packet. According to the draft standard [1] this mechanism is optionally applied whereas the rules of application of this are set on a per-station-basis.

In this paper we present the results of our simulative analysis of the RTS/CTS message exchange, the effects in the hidden terminal case and in the fully meshed case, defining the dependencies and areas of useful application and showing its effects on the overall network performance when its application strategies are not harmonized throughout the network. Section 2 presents simulations of the RTS/CTS mechanism and discusses the problems arising from it and section 3 finishes this paper with conclusions and an outlook on further problems that have to be addressed.

To sum up the effects of the RTS/CTS mechanism, we can say, that it

- ❍ increases bandwidth efficiency by its reduced collision probability since the ongoing transmission has been made known everywhere within the range of it
- ❍ increases bandwidth efficiency since if collisions occur they do not occur with the long data packets but with the relative small control packets
- ❍ decreases bandwidth efficiency since it transmits two additional packets without any payload
- ❍ decreases bandwidth efficiency since it reserves geographical space for its transmission where or when it might actually not need it.

Due to the above listed trade-offs of the RTS/CTS mechanism, the draft standard allows its usage but does not demand it. Usage policy is set on a per-station-basis with the help of a manageable object *RTS_Threshold* that indicates the MPDU length under which the data frames should be sent without the RTS/CTS prefix. This parameter is not fixed in the draft standard and has to be set seperatly by each station. The packet size is the only parameter that is used to decide whether the mechanism is applied, however we will show in the simulations presented below, that there are several more factors relevant to guarantee efficiency gain. We identified the following elements: configuration and geometry and the physical preamble length, but there might be further dependencies.

2. Discussing the RTS/CTS Mechanism in DFWMAC

In order to analyze the behavior of the DFWMAC we simulated a possible network cell using the simulation tool PTOLEMY[6].

We simulated a WLAN consisting of 8 stations using DFWMAC's distributed coordination function as the access scheme with a raw physical throughput of 2 Mbit/s. The channel and packet source model is basically the same as it is described in [7], i.e. Poisson distributed packet sources in a wireless channel model. In addition to that we simulated the signal run time, which can be left out if one only considers a range of a microcell, but which becomes considerable at distances of 300 m and more. Several parameters had to be set to a certain value for our simulations since they were not (yet) defined in the draft standard: The contention window size was set to 32 slots, lasting 4µs each, the physical preamble was set to 30 bytes unless other values are explicitly mentioned, 16 RTS packets transmission attempts with missing CTS response and 4 data packet transmissions with missing ACK response are sent out before the transmission is cancelled and the packet is dropped.

Originally, the RTS/CTS mechanism was introduced to solve the hidden terminal problem. In result our first simulations intended to show the positive effect of this mechanism in this case. In further simulations we evaluated the use of RTS/CTS in a fully meshed network (i.e. without hidden terminals) and special values for certain parameters.

2.1. RTS/CTS in the hidden terminal scenario

To show the effect of the RTS/CTS mechanism, we simulated the following scenario: There are 8 stations in a cell, where station 1 is "hidden" to stations 2 and 3, respectively. All of the outer stations (1,2 and 3) try to send data to the inner, i.e. audible to all, stations. The destinations of data traffic are shown in **Figure 1**.

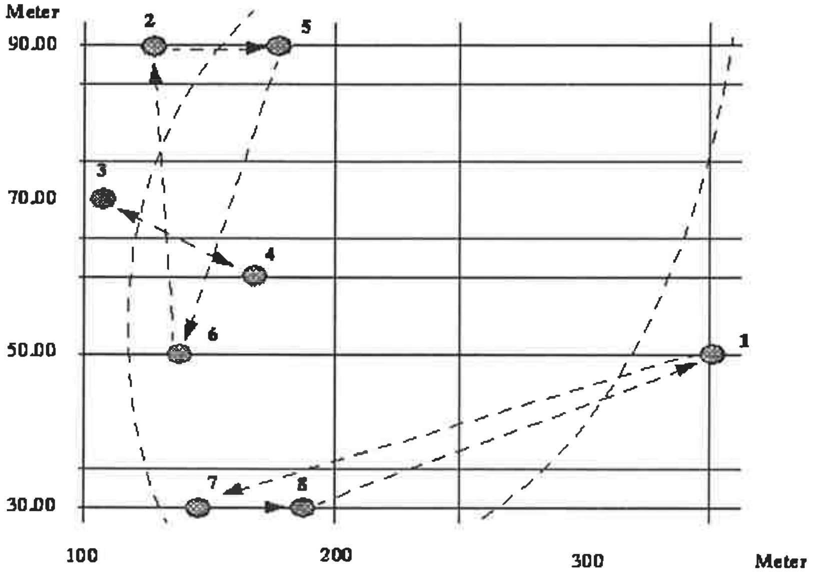

Figure 1: Map of the stations in the hidden terminal simulation

The result of these simulations was that using the RTS/CTS does not completely solve the hidden terminal problem, even though significant improvements can be achieved. The breakdown of inbound data traffic in the case of higher load stems from the fact that the mutually hidden stations become synchronized by an earlier data exchange in the area between them. In result, they start their backoff counters at the same time but they are unable to detect the begin of transmission of the other station. **Figure 2** shows the throughput achieved by the different stations when RTS/CTS is permanently switched off. The relevant differences are clearly visible in the higher load range. The stations that are hidden to other stations hardly get any packets through due to the above mentioned synchronization effect. The stations that attempt to send towards the 'hidden stations' have significantly lower throughput than the two stations 5 and 7 that only send to non-hidden stations. The first group successfully gets packets through, however many acknowledgment packets are destroyed by traffic from the hidden stations.

Figure 3 shows the same setup when RTS/CTS is permanently used. This figure shows clearly that still the hidden terminal scenario is not solved: Station 1 still hardly gets any packets through, but its throughput is improved compared to the figure without RTS/CTS. The same goes for stations 2 and 3 - all of the hidden stations benefit from the captured CTS packets. The non-hidden terminals all achieve the same (high) throughput due to the fact that outbound traffic is protected by the RTS packets.

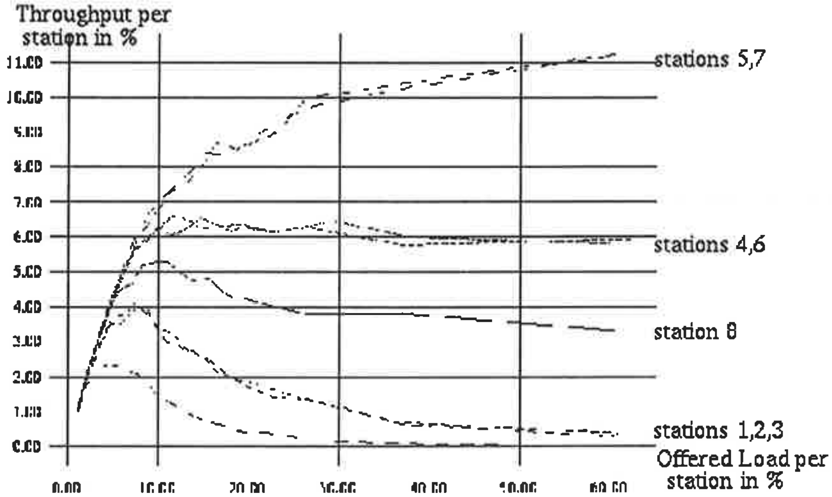

Figure 2: Throughput per Station, Hidden Terminals, RTS/CTS off

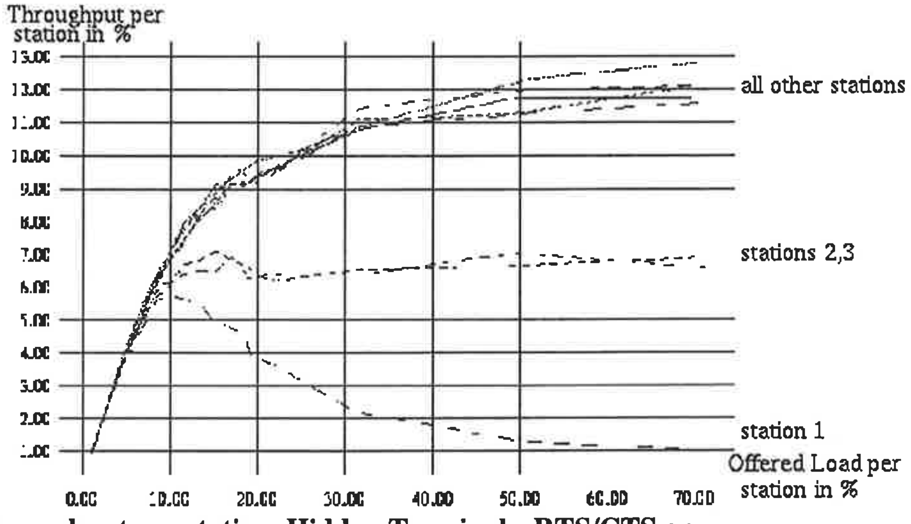

Figure 3: Throughput per station, Hidden Terminals, RTS/CTS on

The synchronisation effect is due to the a CSMA based media access in a hidden terminal situation. One will always find situations like this in a real network. This fact is an argument against the use of RTS/CTS being dependent on the MPDU size, because in a hidden terminal situation it should be used all the time to at least protect the outbound traffic.

2.2. RTS/CTS in a fully meshed network

In the above paragraph we showed the performance gain that can be achieved by applying the RTS/CTS mechanism in the case with hidden terminals. For the fully meshed case we followed several different goals in our simulations:

- ❍ we showed that there are situations where application of the mechanism has significant positive effects as well as there are situations with significant negative effects thus proving that there cannot be an overall same strategy on the usage policy
- ❍ we then tried to analyze the relevant factors that determine the efficiency of the mechanism
- ❍ under fixed non-extreme circumstances we defined a range of values for useful application of RTS/CTS (RTS_Threshold parameter)
- ❍ we investigated asymmetric usage policies of the mechanism.

Our first two simulations were meant to show extreme situations, one resulting in significant negative effect in using RTS/CTS, one resulting in significant negative effects when NOT using it, both compared to a realistic scenario. The first situation was determined by a short packet length of only 64 byte payload, while the PHY preamble was set to 30 bytes. Since observations of Ethernet traffic show that a very large number of packets is shorter than 100 byte, this situation can actually be considered realistic. The figure shows decreasing performance - lower throughput due to the higher overhead of the RTS/CTS extension.

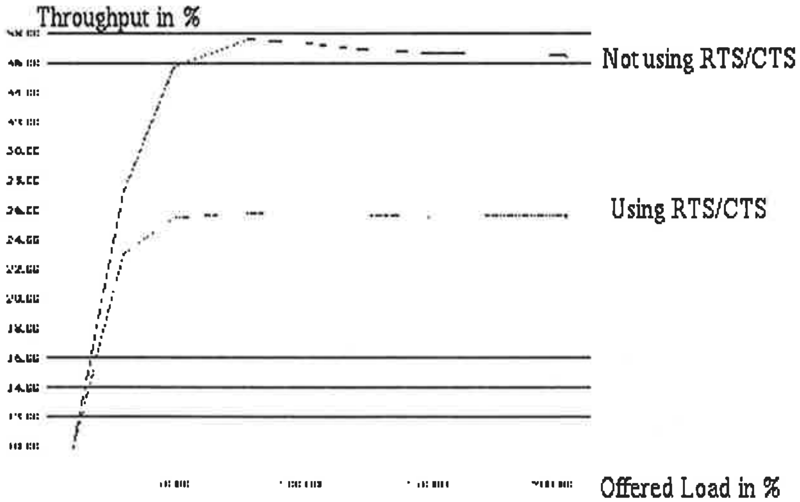

Figure 4: Negative effect on throughput, if packet size is small

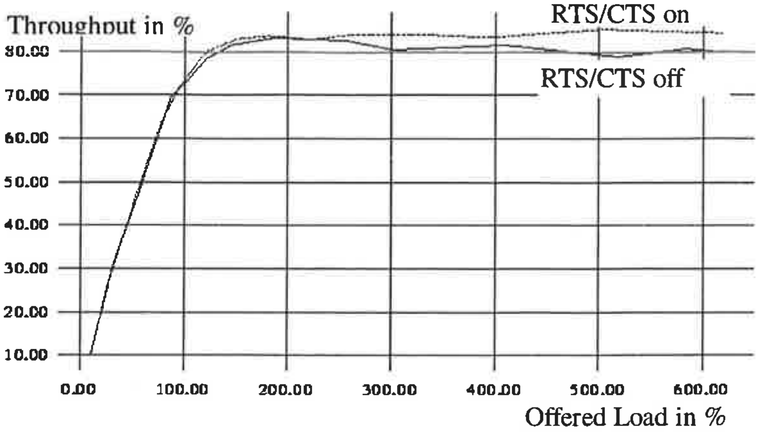

Figure 6: Positive effect on throughput, if packet size is larger (1518 byte)

For the following simulations we used packet sizes from an Ethernet trace (Leland trace, [8]) i.e. packet sizes are distributed as in real ethernet traffic with an average packet size of 682 byte. Since the packet size on the wireless channel consists of the MAC-layer packet size and the physical layers preamble, the length of this latter part has to be taken into account when determining the *RTS_Threshold* parameter. Our simulations gave larger “best“ values for *RTS_Treshold* for larger physical preambles. Since this physical preamble has different lengths depending on the channel used - for the defined physical layers in the 802.11 specification those values are 128 bits with a variable number of stuff bits for 2.4GHz FHSS channel, 192 bits for the 2.4GHz DSSS channel and between 96 and 112 slots of 250ns length plus 32 bits for the baseband infrared channel - therefore this value has to be taken into account when configuring the *RTS_Threshold* parameter. This gets even more difficult in the case of the 2.4GHz FHSS channel as well as for the infrared channel, where no fixed length is added but a variable number of bits.

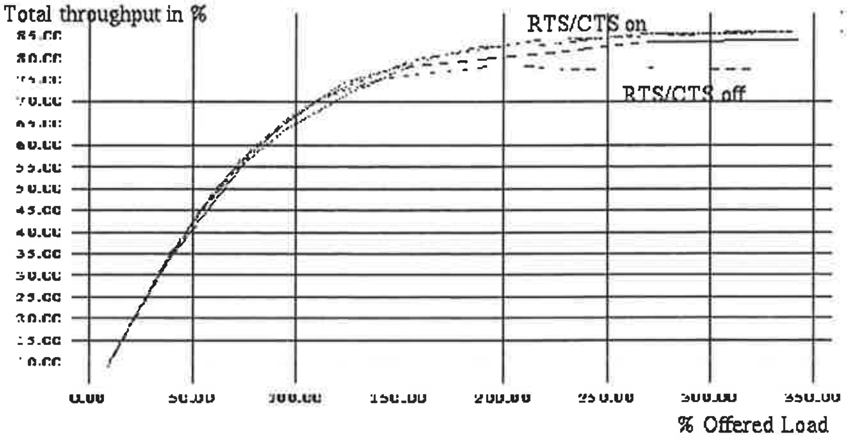

Figure 8: Throughput vs. offered load, PHY_preamble =0

If we assume the physical preamble to be non-existent we get the highest total throughput for *RTS_Threshold*=0, i.e. RTS/CTS is switched on permanently.

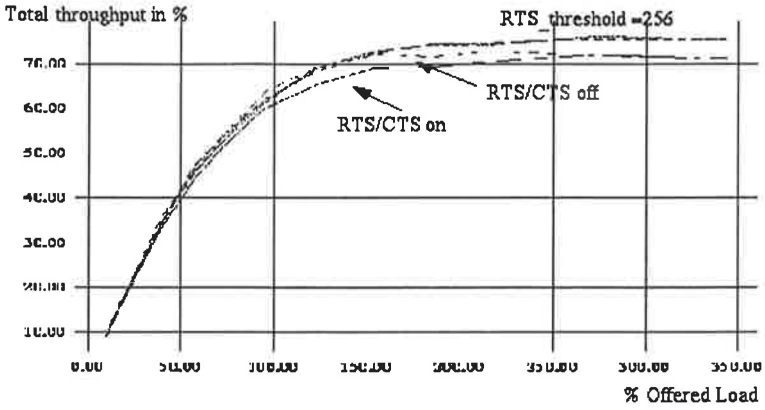

Figure 9: Throughput vs. Load, PHY_preamble=30 bytes

For a physical preamble length of 30 bytes, we achieve best results for *RTS_Threshold* around 256 bytes.

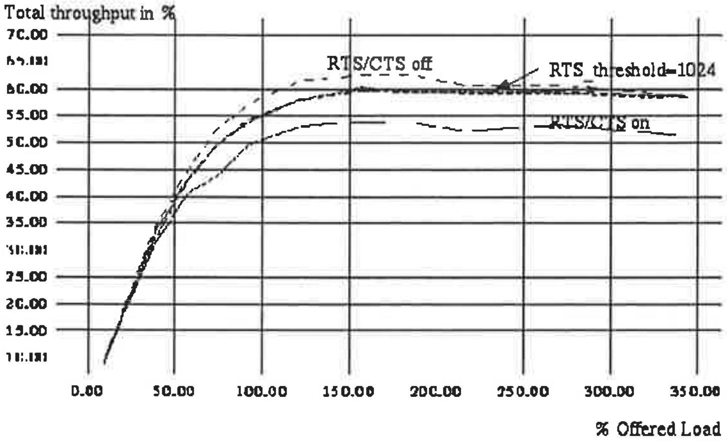

Figure 10: Throughput vs. Load, PHY_preamble=100 bytes

For the extreme (and slightly unrealistic) case of the PHY_preamble having 100 bytes we get optimal values for RTS_threshold above 512 bytes.

Summing up we can say that for the proposed PHY_preamble sizes of the draft standard we recommend, based on our simulations, a value of around 200 byte MPDU size for the *RTS_Threshold.* This of course depends on the source model of our simulations. Since traffic on a wireless network can be expected similar to Ethernet traffic we consider our applied source model to be reasonable.

The use of RTS/CTS in the DFWMAC draft standard is managed on a per-station basis. This can result in asymmetric configurations in the network, e.g. one station does never use the mechanism, all the others do. We simulated some of those asymmetric configurations to determine whether the stations behaving different than the rest might win performance at the cost of the others, whether they will loose or whether they will cause degradation of the overall network performance. It showed that there is no individual gain for a station which does for some reason not behave like the others, but that there is a small decrease in the overall performance

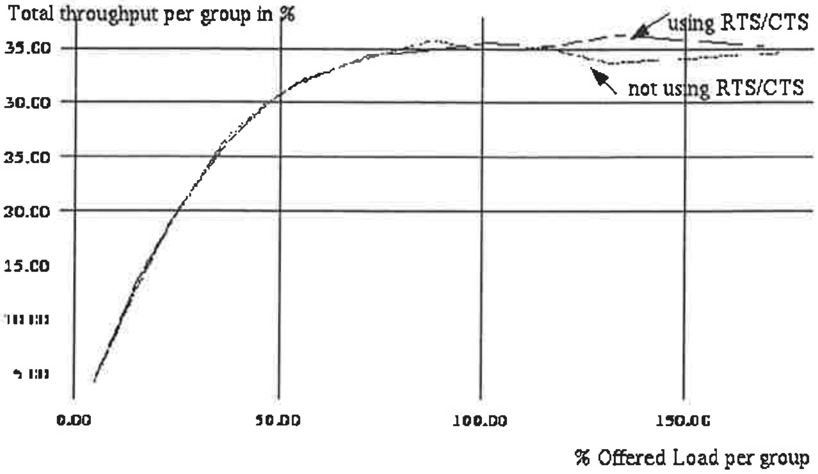

Figure 11: Throughput vs. Load 4 Stations use RTS/CTS, 4 Stations don't

3. Conclusions

The strategy based on setting a fixed value for *RTS_Threshold* obviously does not optimize the optional use of the mechanism. We will experiment with a different approach trying to design it as a load adaptive scheme: Considering the fact that under low load condition in the network the few collisions caused by the hidden terminal scenario do not harm the overall performance as much as under high load condition, it seems to be reasonable only to use RTS/CTS when load is high. However it is not trivial to get the knowledge about current load to the stations. One possible switch-on criteria could be an experienced collision: Once a data packet collided the next attempt would have to be preceded by a RTS/CTS exchange (possibly this step to more secure transmission should only take place after two or more collided attempts). This strategy however leaves it open when the station should switch back to non-RTS/CTS mode - after each successful transmission of a data packet, after a certain silence period, after a certain time period or other.

As we have shown several factors have significant influence on the efficiency with which the RTS/CTS mechanism can be applied in the context of DFWMAC. In general, the optimal *RTS_Threshold* increases with an increasing PHY_preamble and we would advise for a value around 512 byte MPDU size for the PHY-preambles under discussion.

Another topic for further work could be the exploitation of the hidden terminal scenario - since the collision can only occur at the receiving side of the communication it is not necessary to reserve space around the sending side. A station only hearing the RTS packet and the data packets but not the CTS packet can assume that it is out of range of the receiving station and thus not disturbing the ongoing communication. It could therefore start communication with another station if that station is able to receive the signal as well undisturbed by the other ongoing communication. These problems will be the subject for our future activities in this area.

References

[1] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Draft Standard IEEE 802.11, P802.11/D1; The editors of IEEE 802.11

[2] HIPERLAN Functional Specification Section 6 MAC Layer, Draft Version 0.6, ETSI Secretariat, December 1994

[3] MACA - A new Channel Access Method for Packet Radio, P. Karn, ARRL/CRRL Amateur Radio 9th Computer Networking Conference, Sept. 22 1990

[4] MACAW: A Media Access Protocol for Wireless LAN's; Bharghavan V., Demers A., Shenker S., Zhang L., SIGCOM 94; http://beta.xerox.com/pub/net-research/macaw-cr.ps

[5] Modified Backoff Algorithms for DFWMAC`s Distributed Coordination Function; Woesner H., Weinmiller J., Ebert J-P., Wolisz A.; submitted to 2. ITG-Fachtagung Mobile Kommunikation '95; http://ftsu10.ee.tu-berlin.de/bibl/ours/backoff-ITG.ps.Z

[6] PTOLEMY, anonymous ftp site: ftp.ptolemy.eecs.berkeley.edu, www: http://ptolemy.eecs.berkeley.edu, Copyright © 1990-1995 The Regents of the University of California

[7] A Wireless MAC Protocol comparison, W. Diepstraten; IEEE 802.11 working paper P802.11-92/51, May 1992

[8] W. Leland et al.: On the Self-similar nature of Ethernet traffic (extended version), IEEE Transactions on Networking, Vol 2, No 1, Feb 1994