Benchmarking for RTN, a Real-Time Variation on IEEE802.11

Hans Scholten, Pierre Jansen, Ferdy Hanssen, Wietse Mank and Arjan Zwikker
University of Twente, Department of Computer Science (EEMCS), Enschede, the Netherlands
Email: {scholten, jansen, hanssen}@cs.utwente.nl

***Abstract*— RTN is a network access protocol that uses a token scheduling mechanism to enable real-time multimedia streaming on Ethernet-like networks. Unlike other token based protocols, RTN's token mechanism is based on pre-emptive earliest deadline first (PEDF) scheduling. PEDF has interesting characteristics, such as one hundred percent bandwidth utilization and a uncomplicated feasibility analysis. RTN is simulated and implemented on Ethernet and IEEE802.11. The latter implementation is not straightforward, because some for RTN important details are not defined in the standard. As a consequence the standard had to be evaluated before adaptation and implementation of RTN could start. This paper briefly describes the RTN protocol. The focus, however, is on benchmarking the IEEE802.11 standard.**

***Index Terms*— Network measurements, experimentation with real networks/testbeds**

I. RTN Real-Time Token Protocol

Examples of networks that use tokens are IEEE802.4 token bus, IEEE802.5 token ring and FDDI. Main properties of these networks are described in [1] and [2]. The RTN protocol also uses a token, however the token does not follow a static or simple round robin schedule, instead the token is scheduled to visit only nodes with active streams. The token is passed on from node to node following a dynamically calculated pre-emptive earliest deadline first (PEDF) schedule. This schedule is based on characteristics of the active streams in the network: period and requested bandwidth. Before a new stream is admitted to the network a feasibility analysis is made. Only if the set of streams is feasible, a schedule is made and the new stream is permitted.

The scheduler resides in every node of the network and the schedule travels around in the network with the token. When a node gets the token, it can perform network management, and change, add or remove streams. When the set of streams changes, a new schedule has to be calculated by the scheduler residing in the active node. The scheduler is an earliest deadline first scheduler.

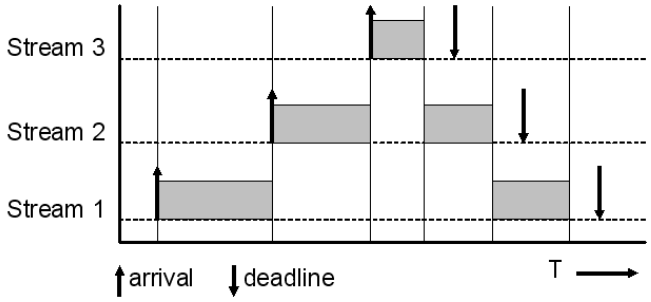


Fig. 1. Pre-emptive EDF scheduling

It is pre-emptive, so stream can be interrupted by another stream if this stream has an earlier deadline, but arrives later. This is illustrated in figure 1: stream 1 arrives first, but is pre-empted by stream 2, which arrives later, but has an earlier deadline. In its turn stream 2 is pre-empted by stream 3. After stream 3 is finished in the current period, stream 2 will be finished and stream 1 will be last.

Because of its properties [3], pre-emptive earliest deadline first scheduling is well fitted for RTN:

- maximal bandwidth utilization is one hundred percent;
- suitable for both periodic and aperiodic data;
- scheduling is dynamic and in real-time;
- feasibility analysis is simple and straightforward.

PEDF scheduling performs badly in the presence of an overload, but because this is avoided by the feasibility analyses it is not a problem. When the network is idle, i.e. no real-time streams are to be transmitted in the current period, the rest of the cycle is used for non-real-time traffic. During this phase a round-robin schedule is used and the token visits every node in the network.

Before a set of tasks can execute the scheduler must verify that the task set will never cause a deadline miss. When the task set changes because a new task is added or the characteristics of a task changes (different period, different deadline) the feasibility analysis must be performed on the new task set. The new task set

This work is sponsored by the Netherlands Organization for Scientific Research (NWO) under grant number 612.060.111, and by the IBM Equinox programme.

will be rejected if it can not be scheduled. Under the assumption that a task's period is equal to its deadline, a set of periodic tasks is schedulable with EDF if and only if

$$\sum_{i=1}^{n} \frac{C_i}{T_i} \leq 1$$

Where C_i: Computation time of task i; and T_i: Period of task i. The RTN feasibility analysis for a set of streams is derived from the standard PEDF feasibility analysis for a set of tasks (see [4])

$$\sum_{i=1}^{n} \frac{B_i}{B} \leq 1$$

Where B_i: Bandwidth of stream i; and B: Maximum bandwidth of the network. When the streams in the network meet this requirement, the PEDF scheduler will find a schedule.

A. Simulation and Ethernet Prototype

The network and its PEDF token mechanism are simulated and prototypes based on Ethernet and IEEE802.11b are built. Figure 2 shows dynamic graphical simulator

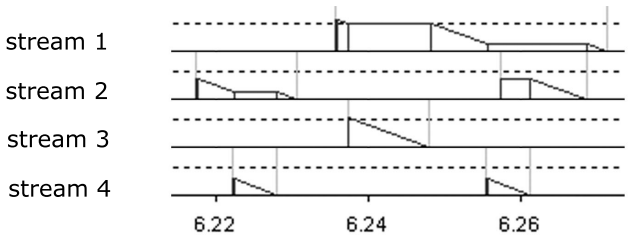


Fig. 2. PEDF schedule of a set of periodic streams

output for the PEDF scheduling of a set of periodic streams in an Ethernet based network. The streams in this simulation all have different periods. The deadline of a stream during a period corresponds with the end of the same period and is equal to the start of the next period. The arrival time of a stream during a period corresponds with the start of that period. Or in other words, when a video frame becomes ready to be sent, the previous one must have been sent. So during each period of a stream one packet of that stream, e.g. a video frame, arrives at the receiver. Because this is guaranteed, only a buffer the size of the packet is needed and latency is limited to the time it needs to play a packet. Figure 2 shows the remaining number of bytes to be sent by a stream during a period. When the stream is transmitted this line decreases. A horizontal black line shows where the stream is pre-empted by another stream. So at any time no more than one stream is allowed to send and thus no collisions occur. The arrival time of a stream is the start of a period. An alternative representation of the

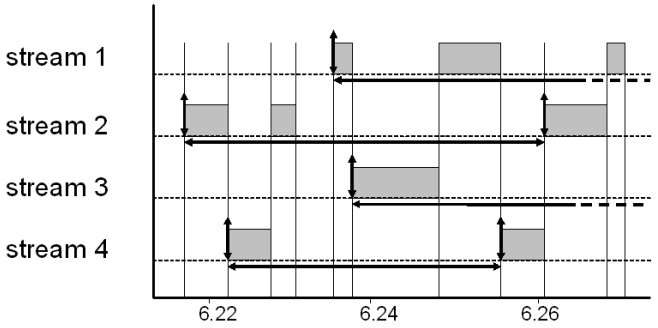


Fig. 3. Alternative representation of figure 2

simulator output as shown in figure 2 which is more in line with the representation in figure 1 is shown in figure 3. Here, the horizontal arrows denote the duration of periods of the individual streams. The periods of streams 1 and 3 are too long to be shown in the figure. A stream can be pre-empted by multiple streams: stream 1 is pre-empted by stream 3, 4 and 2 respectively. It will be clear that stream 1 has the longest period of this set of streams.

Measurements taken in the Ethernet prototype, based on the Linux operating system and Ethernet hardware confirm the validity of the simulation and its parameters.

II. Real-Time on IEEE802.11 Wireless LAN

The architecture of an IEEE 802.11 wireless network [5] is in some important aspects different from that of a wired Ethernet. In a wireless network the mobility of the nodes (named station or STA) must be taken into account. STAs can go into power-saving mode to save batteries and the communication is less reliable. In order to cope with these conditions, the datalink layer of a IEEE802.11 is different from the datalink layer of (wired) Ethernet. This is one of the main reasons why the RTN protocol, as implemented on an Ethernet based network, can not be mapped on IEEE802.11 directly. It has to be adapted to the peculiarities of a wireless network.

A. IEEE802.11 Architecture

A minimal 802.11 network consists of two stations. These STAs can only communicate with each other within a limited radius. This radius is called a Basic Service Set (BSS). STAs can dynamically enter and leave a BSS and can move around freely within a BSS.

A STA can communicate with STAs from another BSS in the presence of an Access Point (AP). The AP functions within the BSS like a normal STA, but it also

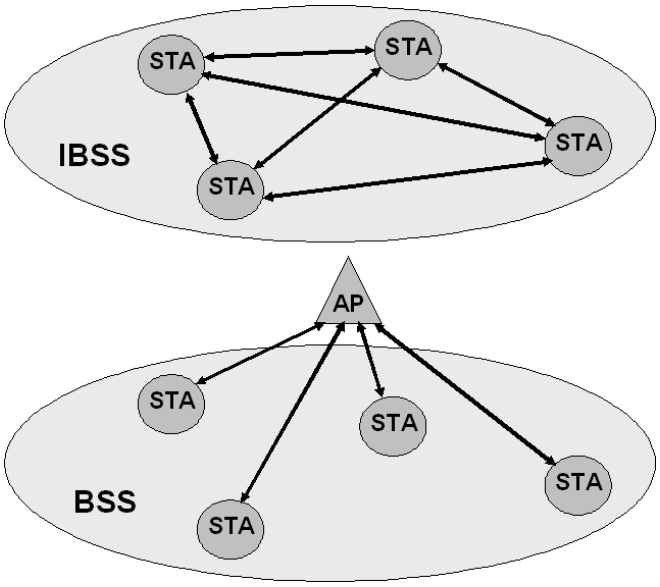


Fig. 4. Difference in topologies of BSS and IBSS

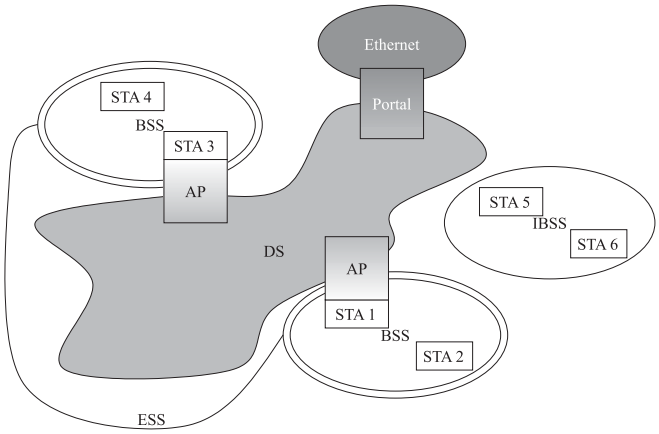


Fig. 5. Example of an 802.11 topology

acts as a gateway to the outside world. The AP gives access to a Distribution System (DS) and is used by STAs from different BSSs to communicate with each other. How the DS must be implemented is not specified in the 802.11 standard. When, in the absence of an AP, two or more STAs are in each other's proximity they can initiate an ad hoc network, called an Independent Basic Service Set (IBSS). An ad hoc network is different from a BSS, because it has no AP and does not have the ability to communicate with the outside world. A BSS is sometimes called an infrastructure network or managed network. See figure 4. DSs and BSSs together offer the possibilities for an infinite large network. The 802.11 standard calls a combination of DSs and BSSs an Extended Service Set (ESS). Every STA can communicate with every other STA and can move from one BSS to another, as long as this takes place in the context of the same ESS. Figure 5 gives an overview of the 802.11 topology.

IEEE802.11 knows two basic communication modes, the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF). DCF is the standard way of communication and is used in both BSS and IBSS.

Collisions can occur easily on a wireless medium. The protocol to avoid collisions as used by Ethernet (CSMA/CD) is not usable, because it depends on the fact that every network card can observe collisions. This is not possible with wireless communication, as every station is deaf when it is sending. Therefore DCF uses a modified version of CSMA/CD, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). CSMA/CA is more complicated than CSMA/CD. The carrier sense mechanism has been split in two parts, the physical carrier sense and the virtual carrier sense. The physical carrier sense is nothing more than listening whether the wireless medium is occupied. For the virtual carrier sense every STA has its own Network Allocation Vector (NAV). This vector is used to keep track of how long the medium will be occupied. If the medium is in use the value of NAV will be decreased. If the NAV is zero and the physical carrier sense senses no signal, the medium is free.

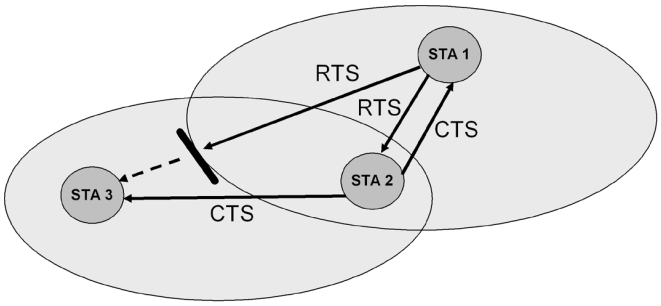


Fig. 6. Working of the RTS/CTS mechanism

The NAV can be initialized in two ways. The first one is that a node that sends, puts the duration of the frame in the frame itself. Every STA that can receive this frame thus knows when the frame will end. The second method is the use of a special Request to Send frame (RTS) followed by a Clear to Send frame (CTS). Both frames contain the initial value for the NAV. Suppose there are three stations. STA 1 will send to STA 2, while STA 3 is within range of STA 2 but not within range of STA 1. Without the RTS/CTS protocol STA 3 may cause collisions, because it does not know the NAV value.

The use of RTS/CTS minimizes the risk of interference from partial hidden nodes in a wireless network. This is illustrated in figure 6. If the physical carrier sense senses a signal when NAV is zero, a random back off algorithm is started to avoid collisions.

The Point Coordination Function PCF) is a layer on top of DCF and is used to send information contention free. Contention free means that STAs do not 'fight' for the right to send. The difference between PCF and DCF concerning implementation is noticeable in the AP where a Point Coordinator (PC) runs. For normal STAs the changes are less noticeable. In a Contention Free Period (CFP) STAs associated and authenticated to an AP will be polled one by one. This information will be sent to the PC, which in turn will send it to the STA the message was meant for. Every technique and mechanism incorporated in DCF is present in PCF. However, if a STA has a special PCF implementation, no RTS/CTS exchange will take place.

The AP sends beacon frames at a specified interval for management functions. In IBSS mode beacon frames are also sent, not by the (non-present AP), but by the nodes in the network. Beacon frames contain data about the beacon interval, the clock of the AP, the supported speeds, frequency hopping, the Distributed Service (DS), Contention Free Period (CFP) and, if appropriate, data concerning the IBSS. One use of the beacon frames is network management, another is synchronisation of the clocks of the nodes. The interval in which beacon frames are sent depends on the physical layer, but usually its value is around 100 ms.

B. Mapping RTN on IEEE802.11

Although there are strong similarities between Ethernet and IEEE802.11, mapping RTN on the wireless network is not a trivial matter. In the following we will address some of the issues.

1) Topology and Network Mode: IEEE802.11 is not a fully connected network at the MAC level. It is possible that a network contains hidden nodes. In an ad hoc network communication takes place between nodes directly. Although a hidden node can take part in the protocol, in an ad hoc network it is only able to send the token to a subset of all nodes in the network. In a managed network (BSS) however, all communication takes places via an Access Point. Even if the network contains hidden nodes, i.e. not every node is able to communicate with every other node directly, all nodes are within reach of the Access Point. The Access Point is used as a relay for communication between nodes and the network behaves as a fully connected network in a star topology.

In order to implement RTN on IEEE802.11, a choice has to be made between the three different MAC layer implementations: the Distributed Coordination Function (DCF), the Point Coordinator Function (PCF) and ad hoc mode. DCF has a built-in random back off algorithm that cannot be turned off. Since this algorithm is activated only when the medium is being used, it will not be activated when the proposed protocol is active, because it prevents two stations from sending at the same time. DCF as well as PCF is centralised around an Access Point (AP). Every message that is being sent, is relayed via the AP to the target node. In order to guarantee real-time properties, RTN must control each packet. Using these modes this is not entirely the case. Another problem with the use of PCF is that the order in which the nodes are being polled is fixed according to the order in which the stations have associated themselves with the AP. Due to this fixed order, it is impossible to work with priorities efficiently. In the ad hoc mode the wireless network is completely decentralised, and in essence, is the same as Ethernet. Unfortunately, with ad hoc networking, there is the danger of hidden nodes. An advantage of ad hoc is that it is lacks a single point of failure. Centralisation of a managed network is hard to avoid. Actually, this is not a serious problem. RTN prevents two stations from sending at the same time and this is still true when a store-and-forward mechanism within the same network is being used. PCF is useable if the order in which it polls the stations would be customisable. This is the most efficient solution (shorter waiting times) in combination with an adapted version of RTN. PCF could be used in its current form. However, this would spill a lot of bandwidth since nodes that do not possess the token are still being polled. Ad hoc mode is as useful as DCF, only the hidden node problem is a disadvantage.

In conclusion, DCF is the preferred MAC layer mode, as it is impossible to handle priorities with PCF without modification of this protocol. The more efficient approach of PCF is not used because the polling order is not customisable. This leaves a choice between BSS and IBSS (ad hoc) mode. Since both modes are very similar and both have their advantages and disadvantages, both types are implemented in the prototype.

2) Beacon Frames: Beacon frames are sent by the Access Point at a fixed interval. The size of this interval depends on the physical layer. The MAC layer autonomously sends these beacon frames and higher layers cannot control these transmissions. This can result in a delay for real-time traffic when it has to wait for a beacon frame, or far worse, when it collides with a beacon frame. A solution is to incorporate this 'waste'

of bandwidth into the feasibility analysis. This ensures that the real-time streams will keep their deadlines. Unfortunately, a missed deadline can still occur, because it is not known exactly when the beacon frame will be sent. This depends on the interval and the size of the beacon frame. To cope with beacon frames, the feasibility analysis can use fixed maximum values.

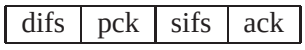
3) Bandwidth: An estimate can be made for the effective available bandwidth in the network. Effective bandwidth is the bandwidth that remains after all overhead has been subtracted, where overhead is a function of multiple factors, like the mode the network works in, or whether RTS/CTS or back off is used. As described earlier only DCF is considered, both managed (BBS) and ad hoc (IBBS). Ad hoc is about two times faster than a managed network, because frames are relayed by the AP. When two frames are sent from one STA to another using ad hoc mode, both messages will be sent immediately after each other. For a managed network this simple procedure is quite different. Consider two frames that have to be transmitted, first the sending STA sends frame 1 to the AP. The AP wants to send this frame to the receiving STA, but the sending STA wants to send frame 2 to the AP. Since they use one shared medium, it is impossible that this happens at the same time. Therefore the maximum bandwidth using a managed network is half that of an ad hoc network.

First we will consider managed networks, starting without RTS/CTS and back off, which is the least complex situation. Two STAs are considered. STA 1 is sending and STA 2 receiving. STA 1 checks if the medium is free. If this is the case the STA has to wait a Distributed (Coordination Function) Interframe Space (DIFS). If the medium is still free after the DIFS, STA 1 sends its packet to STA 2. STA 2 waits a short interframe space (SIFS) before it returns an acknowledgement frame (ACK) to STA 1. This sequence is illustrated in figure 7(a).

The duration of this sequence can be calculated ([5] and [6]). Effective bandwidth at different packet sizes is summarised in figure 7(b).

Figures 8(a) and 8(b) summarise managed networks without RTS/CTS, but with back off, while figures 8(c) and 8(d) summarise managed networks with RTS/CTS and back off.

Ad hoc mode has a number of advantages over managed mode. There is no relaying access point, which means that in theory the maximum throughput is doubled when compared to managed mode. Because RTN guarantees that only one STA is sending at one time and since there is no relaying AP, there are no problems with back off.



(a) Sending a packet without RTS/CTS and back off

	1 Mbit/s	2 Mbit/s	5.5 Mbit/s	11 Mbit/s
500	80.97%	74.29%	56.07%	40.47%
1500	92.74%	89.66%	79.29%	67.10%
2296	95.13%	92.99%	85.42%	75.74%

(b) Effective bandwidth in managed networks without RTS/CTS and back off

Fig. 7. Communication without RTS/CTS and back off

Beacon frames are handled differently when using ad hoc mode, because there is no central station and there may be hidden STAs. Therefore every STA periodically sends a beacon frame. The total calculated time for a beacon frame is 2190 μs for 1 Mbit/s and 1790 μs for 2 Mbit/s and above. A beacon frame is sent every 0.1 seconds, so the bandwidth efficiency is:

$$1 - 10 \cdot 2190 \cdot 10^{-6} = 0.9781$$

And for rates higher than 1 Mbit/s:

$$1 - 10 \cdot 1790 \cdot 10^{-6} = 0.9821$$

This gives the calculated estimated throughput of ad hoc networks, see figure 9.

The calculated effective bandwidth of the ad hoc mode is compared to the bandwidth of a managed network. The case with beacon frames and back off is considered. The results are shown in figure 10. As can be seen, ad hoc has a far better throughput than managed mode.

III. Measuring the 802.11b Protocol

The feasibility analysis in the RTN protocol presumes values for e.g. latency and effective throughput in the network. From the standard most of these values are calculated in the previous section. In this section we will describe the tests performed to check the calculations. Latency will be considered first, after which throughput will be analysed. First the test set-up used is described. Finally, the test results will be compared with the theoretical estimations and calculations.

A. Latency

1) Test Set-Up: The tests are performed with two Orinoco wireless LAN cards and an Access Point with

difs	b'off	difs	pck	sift	ack

(a) Sending a packet without RTS/CTS, but with back off

	1 Mbit/s	2 Mbit/s	5.5 Mbit/s	11 Mbit/s
500	71.30%	59.49%	36.97%	23.18%
1500	88.17%	81.50%	63.77%	47.52%
2296	91.94%	87.09%	72.93%	58.09%

(b) Effective bandwidth in managed networks without RTS/CTS, but with backoff

difs	rts	sifs	cts	sifs	pck	sifs	ack

(c) Sending a packet with RTS/CTS and backoff

	1 Mbit/s	2 Mbit/s	5.5 Mbit/s	11 Mbit/s
500	63.63%	51.15%	29.01%	17.25%
1500	84.00%	75.93%	55.07%	38.47%
2296	88.93%	82.84%	66.23%	48.90%

(d) Effective bandwidth in managed networks with CTS/RTS and back off

Fig. 8. Communication in varied modes

	1 Mbit/s	2 Mbit/s	5.5 Mbit/s	11 Mbit/s
500	79.20%	72.90%	55.02%	39.71%
1000	90.70%	87.97%	77.80%	65.84%
2296	93.05%	91.24%	83.82%	74.31%

Fig. 9. Effective bandwidth in ad hoc mode

another Orinoco wireless LAN card. The frequency for the tests is channel 11 (2.462 MHz). This frequency is exclusively available for testing purposes. The Linux driver used for the tests is pcmcia-cs 3.1.34 in combination with Orinoco driver 0.13.

2) Test Method: The software to measure the latency is 'sendlat' and 'receivelat'. 'Sendlat' should be run on the client side and 'receivelat' on the server side. 'Sendlat' sends packets as fast as possible. 'Receivelat' logs the difference between the times messages are received. This is depicted in figure 11 for ad hoc. The

	1 Mbit/s	2 Mbit/s	5.5 Mbit/s	11 Mbit/s
ah	116 kB/s	228 kB/s	576 kB/s	1022 kB/s
mg	55 kB/s	103 kB/s	238 kB/s	379 kB/s

Fig. 10. Comparison ad hoc and managed mode

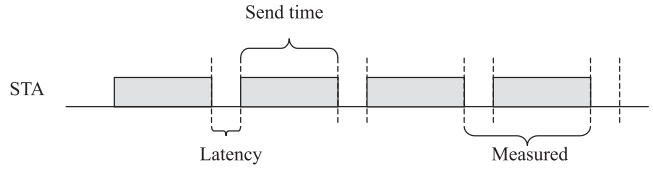


Fig. 11. Measurement of latency in ad hoc mode

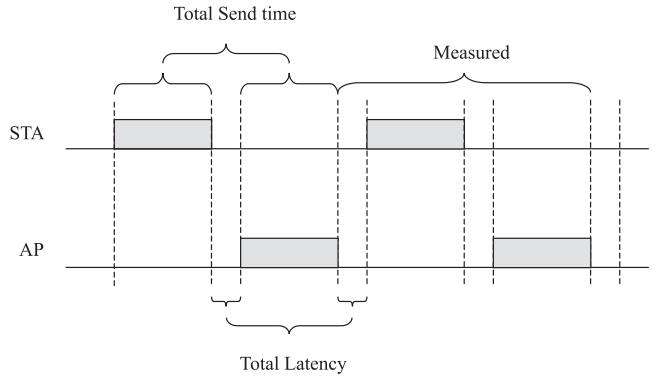


Fig. 12. Measurement of latency in managed mode

values returned by 'receivelat' are the latencies plus the time it takes to send a message from one STA to another. Since this time can be calculated, the latency generated by the drivers and the OS can also be calculated. This is also possible using the managed network mode (see figure 12). The total send time is the time it takes to send the packet to the AP plus the time it takes the AP to relay this packet. It is not possible to discern the latency created by the drivers and the OS and that of the access point. Because these two latencies are always combined, this is not a problem.

3) Tests:

a) Managed Network Mode: The results of the four different bandwidth tests are presented in figure 13 to 16 for 1 Mbit/s to 11 Mbit/s. The latencies are grouped in different horizontal levels and show a 'stairs' pattern; they have a constant value over a certain period, but abruptly leap to a higher level and again stay constant for a while.

The horizontal levels can be explained by the backoff of the AP. In an ideal situation, the relaying of messages happens according to situation 1 in figure 17; STA 1 sends a message to the AP and the AP relays the message to STA 2 before STA 1 sends the next message. This way, on 1 Mbit/s, a value of 40 ms seconds should be measured, which can be seen as the line near the 40 ms on the right side of the figure 13. Unfortunately, in a real situation, the sending STA does not wait until the AP has relayed the message, but tries to send the next message immediately. In this case (depicted as situation 2 in figure

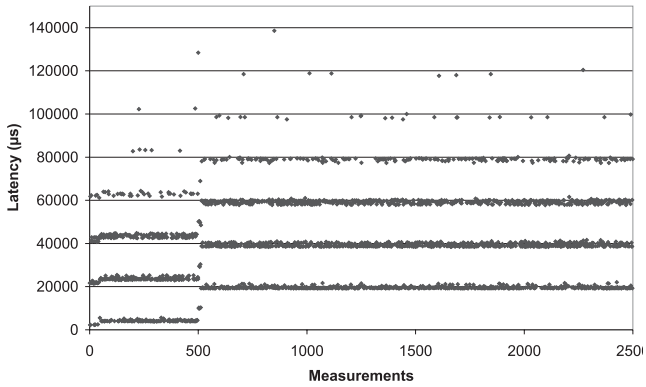


Fig. 13. Latency test using the Access Point at 1 Mbit/s

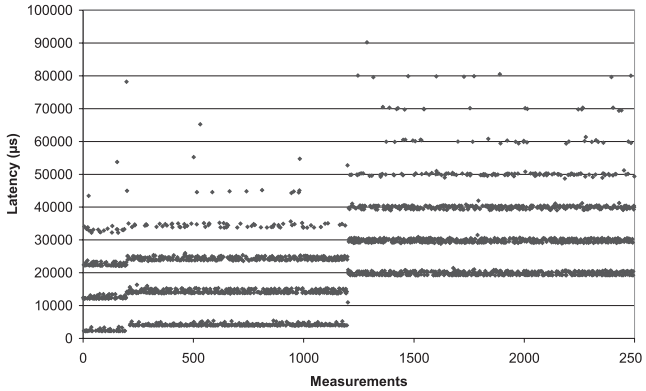


Fig. 14. Latency test using the Access Point at 2 Mbit/s

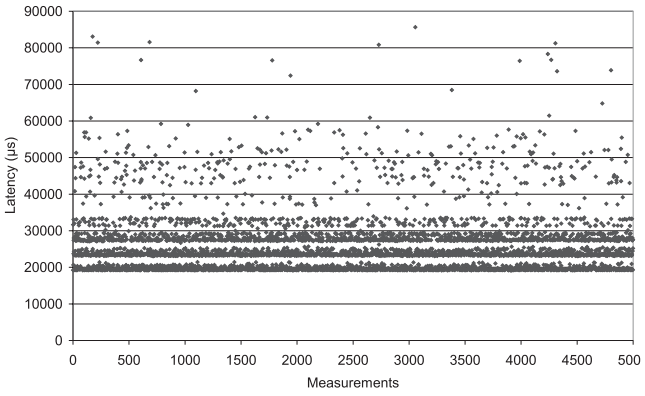


Fig. 15. Latency test using the Access Point at 5.5 Mbit/s

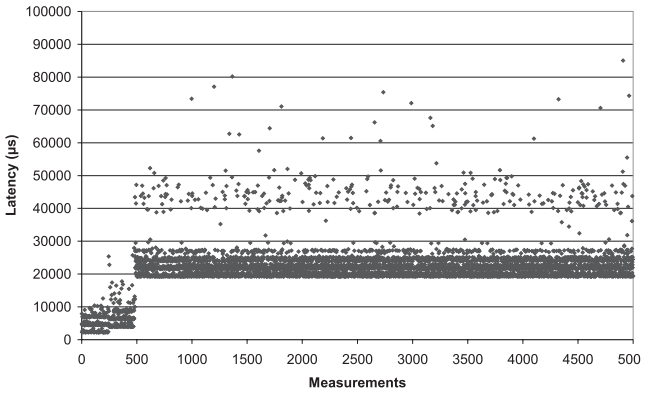


Fig. 16. Latency test using the Access Point at 11 Mbit/s

17), both the AP and the STA want to send, which results in the use of the random backoff mechanism. This means that when the sending STA ‘wins’, the message at the AP is delayed by the time it takes to send the second message from the sending STA to the AP. For each time the sending STA ‘wins’ the random backoff mechanism, this time is added. When the AP ‘wins’ the message is relayed.

A short example will clarify this. Four messages in a row are being sent from STA A to STA B with a speed of 1 Mbit/s. The first message is sent from STA A to the AP. This takes about 20 ms. The AP tries to relay this message, but STA A ‘wins’ the random backoff and starts sending the second message. This also takes about 20 ms, thus the total time that has elapsed since the first message was sent is now 40 ms. Next, when the AP ‘wins’ the random backoff, it sends the first message to STA B. This means that the message was sent 60 ms ago. This case can also be seen on the right side of figure 13.

This still does not explain the 20 ms line in figure 13, which is faster than theoretically feasible. This can be made clear by using the example from the previous paragraph. The receiving STA saves the time between different messages that arrive on that station. When the example is continued and the AP immediately ‘wins’ the random backoff again, the second message (which was already at the AP) is relayed to the receiving STA right after the first. This means that between the arrival of the first and second message is exactly the time it takes to send a message from the AP to the STA, in this case 20 ms.

The ‘stairs’ pattern in these figures are harder to explain. When figure 13 is observed, in the right side of the graph (from about measurement 500), the time it takes to send a message is what is expected from a one Mbit/s network. But the left side shows a ‘stairs’ pattern. A more suitable figure to explain this is figure 18. This figure is obtained from the same test on 1 Mbit/s, but with 5000 measurement points. This figure shows more clearly each step. In the first three steps the network sends messages faster than theoretically possible on 1 Mbit/s. The only possibility is that the AP relays the messages at a higher rate than the sending STA does. This would indicate that in the first case, the STA sends

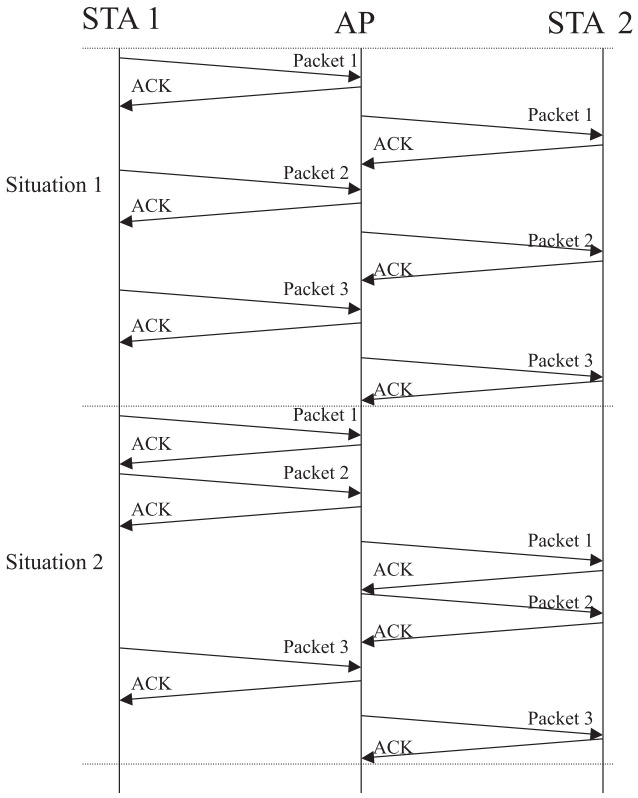


Fig. 17. Different relaying situations

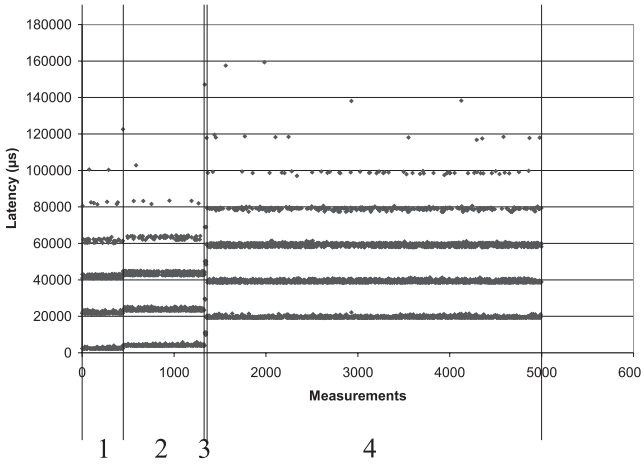


Fig. 18. Stairs pattern

to the AP at 1 Mbit/s while the AP sends to the receiving STA at a rate of 11 Mbit/s. In stairs 2 and 3, this will be 2 and 5.5 Mbit/s respectively. These values seem to match with corresponding calculations. This means that the AP switches back from relaying at 11 Mbit/s to 1 Mbit/s. When the figures of the higher speeds are observed, the same conclusion can be drawn. They all end up at the same latencies as the 1 Mbit/s test, only the horizontal levels are closer to each other, since it takes less time to send a message from a STA to the AP, which means

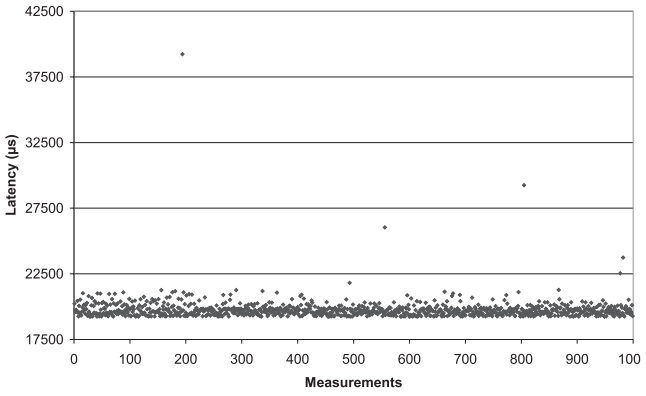


Fig. 19. Latency test using ad hoc at 1 Mbit/s

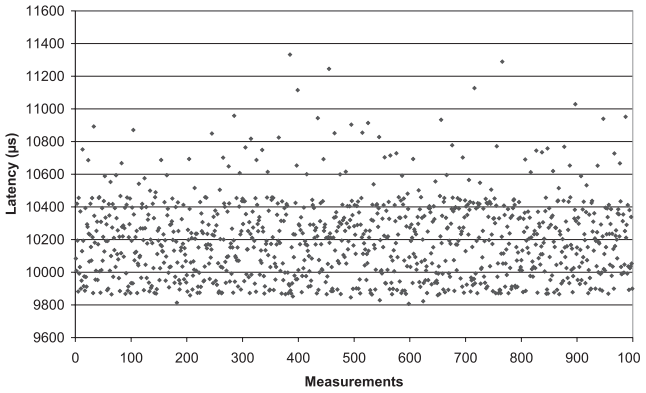


Fig. 20. Latency test using ad hoc at 2 Mbit/s

shorter waiting times for relaying messages. Why the AP decreases its throughput is not entirely clear. Probably, it cannot handle the large amount of messages that is used in these tests. When it cannot handle these messages at 11 Mbit/s, it switches back to 5.5 Mbit/s and so on. When the message stream decreases, the AP switches back up. This also explains why the 5.5 Mbit/s figure (figure 15) does not show a ‘stairs’ pattern. In this test the AP has not switched up again from a previous test.

b) Ad Hoc Network Mode: The results of the four different bandwidth tests are presented in figure 19 to 22 for 1 Mbit to 11 Mbit. The values in these figures are not corrected for the time it takes to send a message from one STA to another. The peaks displayed in the figures are attributed to the Linux operating system, which is not real-time and has a non pre-emptable kernel.

B. Throughput

1) Test Set-Up: The test set-up used to measure throughput is the same as used for measuring latency.

2) Test Method: The software used for the tests is ‘tpsend’ and ‘tpreceive’. The packets made by ‘tpsend’

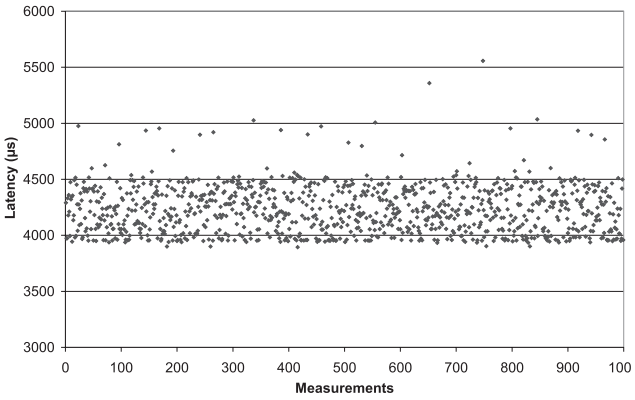


Fig. 21. Latency test using ad hoc at 5.5 Mbit/s

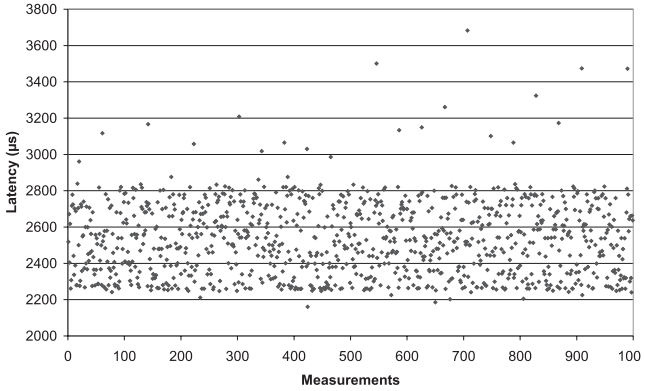


Fig. 22. Latency test using ad hoc at 11 Mbit/s

are directly delivered to the MAC layer of the wireless LAN card. One of the features of 'tpsend' is that it can generate packets that are variable in size. Because 'tpsend' is designed to test the throughput it floods the MAC layer with packets for 25 seconds. The receiving side runs the program 'tpreceive' which checks if all the packets sent are correctly received. The test range runs from a packet size of 500 to 2250 bytes with a granularity of 50 bytes.

3) Tests: For the first test a PC and the Access Point are used, where the PC sends data to the AP. This test is conducted using different drivers. After a driver evaluation the rest of the tests is conducted with the best driver. The second test send from PC1 to PC2, using the access point. The third test sends from PC1 to PC2 in an ad hoc set-up.

a) Test One: From PC1 to AP with Different Driver Sets: During this test it is expected that a high throughput close to the calculated maximum will be obtained. No other traffic on the used channel is allowed, so there is no backoff. The initial tests use pcmcia-cs version 3.1.33 with version 0.09b of the Orinoco drivers. This

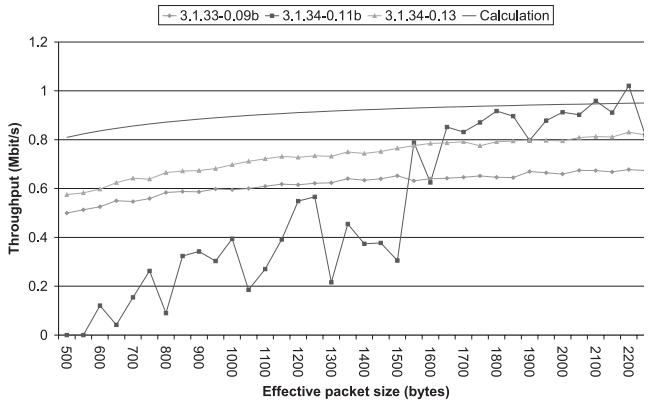


Fig. 23. Results test one using the various drivers at 1 Mbit/s

will be repeated using pcmcia-cs 3.1.34 with versions 0.11b and 0.13 of the Orinoco drivers. The last tests use the official Orinoco drivers.

The differences between the driver sets are illustrated in figure 23 for 1 Mbit/s, figure 24 for 2 Mbit/s, figure 25 for 5.5 Mbit/s and figure 26 for 11 Mbit/s. The graph shows that 3.1.34 with 0.11b are the worst drivers by far in terms of consistency; 3.1.33 (with 0.09b) are the slowest drivers. And 3.1.34 with 0.13 are the best non-official drivers. The last tested driver is the latest Orinoco/Agere driver, version 6.20. There are a few problems with this driver, like the inability to change the bandwidth rate manually on the fly. Furthermore it is not possible to send packets larger than 1500 bytes, the maximum MTU size is 1500. The driver itself, however, is fairly fast, and very consistent in its performance. However, because of its limitations only 11 Mbit/s is tested.

The results vary greatly from driver set to driver set, and it is safe to conclude that the used Linux pcmcia drivers are still very immature. The pcmcia-cs 3.1.34 drivers in combination with Orinoco driver 0.13 give the best results, and is used for the remaining tests.

b) Test Two: From PC1 to PC2 using the AP: This test involves an environment that is a candidate for wireless RTN. It should measure only half the throughput of test one. However, since the AP is only sending at 1 Mbit/s instead of 11 Mbit/s, the results are even lower. These results are depicted in the figures 27 to 30. Unfortunately, comparing these measurements to the calculations, the measurements are far too low.

c) Test Three: From PC1 to PC2 using Ad Hoc: The results of the tests are shown in figure 31 to 34 for 1 Mbit/s to 11 Mbit/s.

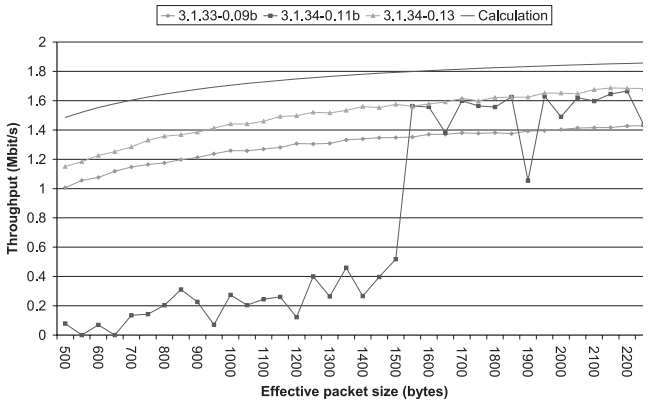


Fig. 24. Results test one using the various drivers at 2 Mbit/s

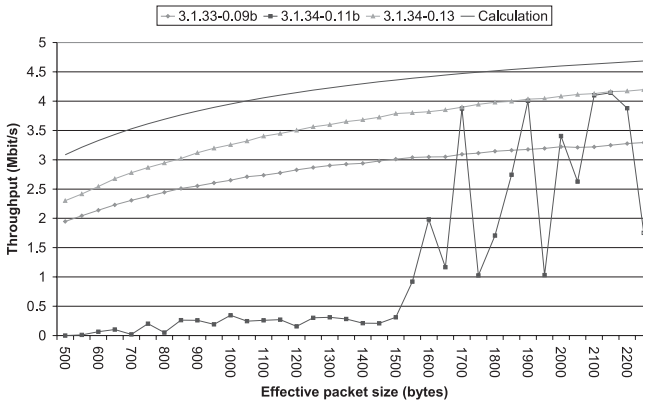


Fig. 25. Results test one using the various drivers at 5.5 Mbit/s

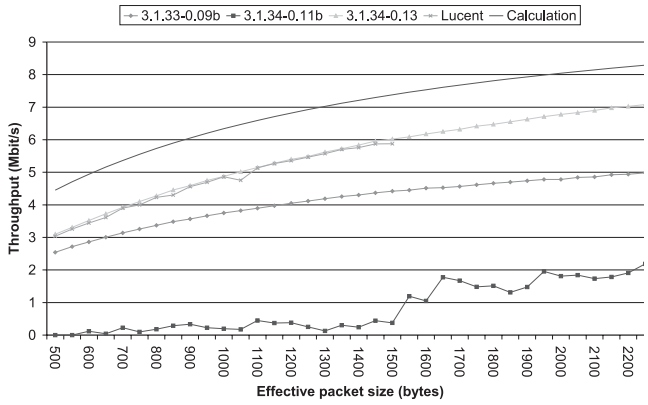


Fig. 26. Results test one using the various drivers at 11 Mbit/s

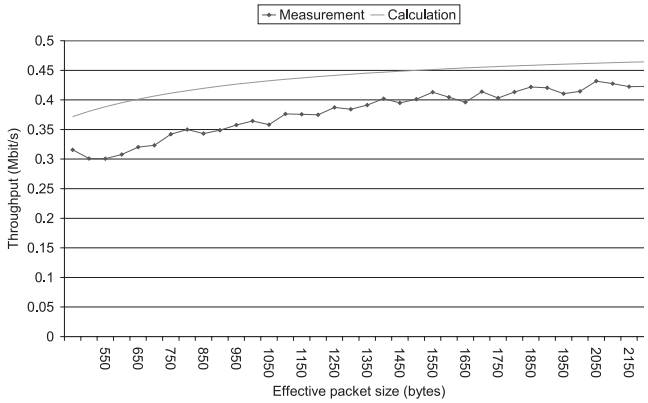


Fig. 27. Results test two at 1 Mbit/s

C. Test Results Interpretations

1) Comparison Ad Hoc and Managed: We may conclude that latencies when using ad hoc network mode are consistent and quite low. However, when the managed network mode is used, latencies are rather unpredictable and can be infinitely long. Therefore, the ad hoc network mode is a far better choice for a real-time network protocol like RTN. The theoretical differences between ad hoc and managed mode are already explained. In spite of the hidden node problem, ad hoc should be considered, since the test results are good. As can be seen in the figures, an ad hoc network is almost as fast as predicted. Furthermore it is much faster than a managed network. This makes ad hoc a very good choice.

2) Comparison Measurements and Calculations: A comparison between calculations and measurements makes it possible to check whether presumptions and calculations are correct. This is important since the calculations are the basis for the real-time protocols.

The figures show that the measurements differ from the calculations. Why this discrepancy occurs will be investigated in the following sections. There can be several reasons:

a) Theoretical Calculations are Wrong: After extensive research, another article on maximum throughput calculations of 802.11b was found. This document describes the exact same formulae as used for this paper. The measurements and calculations in that article match up perfectly. Based on this paper [7] and the 802.11 standard, it is highly probable the calculations are correct.

b) Hardware: Both PC1 and PC2 have the same inconsistencies in the calculations, however, their hardware is very different. The only common aspect in both computers is the Orinoco network card. To make sure the problem is not in the network card, a test is performed with a Compaq Wireless LAN card using the Prism II chipset as the sending card. The test reveales that the Compaq card is slower than the Orinoco card, however, the discrepancy with the calculated values stays.

c) Drivers: As described earlier, differences exist between driver revisions. To see if there are problems inherent to Linux or the drivers, the throughput speed under Microsoft Windows XP is tested. The test shows

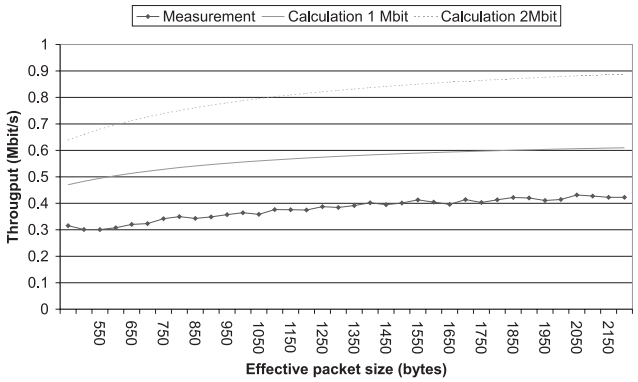


Fig. 28. Results test two at 2 Mbit/s

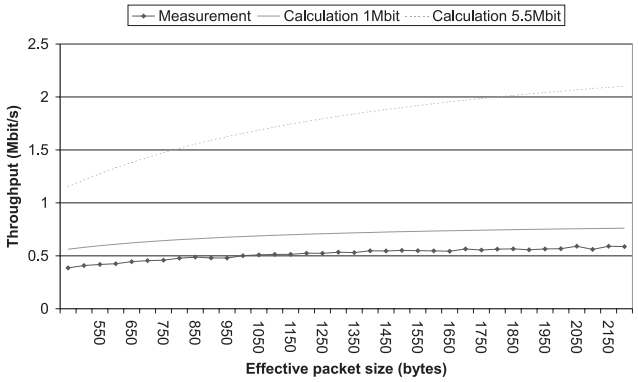


Fig. 29. Results test two at 5.5 Mbit/s

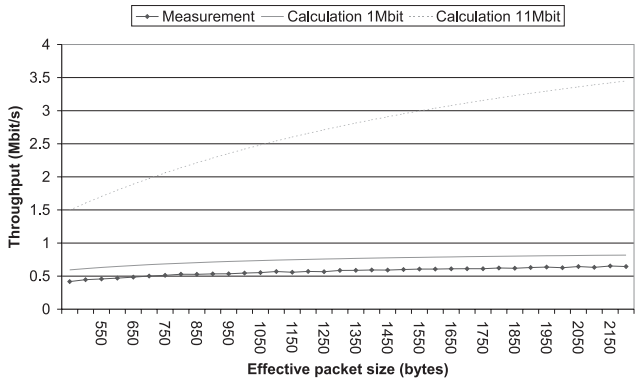


Fig. 30. Results test two at 11 Mbit/s

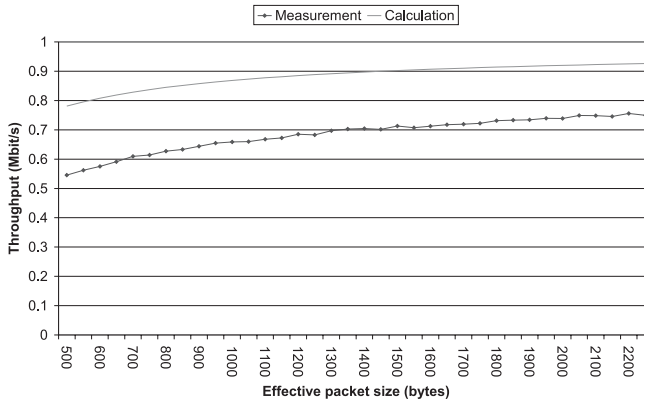


Fig. 31. Results test three at 1 Mbit/s

that the Orinoco driver version 0.13 performs better than the Microsoft Windows XP driver. Probably there is no problem inherent to Linux or the PCMCIA subsystem.

d) Lost Packets: The differences between the measurements and the calculations can be explained by lost packets. Packets can get lost without being detected. The internal retry limit for lost packets on the Orinoco cards is four. This means a packet can be resent four times by the card before it informs the driver. There is no way of checking the number of times a packet is lost efficiently. Unfortunately, on Orinoco cards it is impossible to change the retry limit. The results are very consistent. Each time the test is run it gives the same results. If this is because of lost packets, packets must be lost at a regular interval. It is possible to explain this with lost packets, however, it is highly unlikely.

e) Rate Switching: The preamble and header for every packet are sent at 1 Mbit/s. This means the sending card has to change its rate two times, first to 1 Mbit for the preamble and the header, then to 11 Mbit/s for the packet. The receiving card also has to change its rate settings two times when sending the Acknowledgement. First to 1 Mbit for the preamble and the header, then to 2 Mbit/s for the ACK. Thus the throughput rate is changed a total of four times. However, the exact extend of these rate changes is unclear, since the maximum time it takes to switch rates is not specified in the 802.11 standard.

IV. Conclusion

Besides the discrepancy in throughput, which can be corrected for, the calculations and the measurements presented in the previous sections are consistent. As a result a model is made to predict the maximum throughput in an ad hoc network:

$$\frac{8 \cdot P}{514 + \frac{(P+48)\cdot 8}{B} + \frac{14\cdot 8}{B_{Ack}}} \cdot \left(1 - \frac{\frac{BS\cdot 1390\cdot 100\cdot 8}{B_{Ack}}}{1000000}\right) \cdot Per$$

Where P is the effective size of the send packet; B is the rate at which the packets are sent; B_{Ack} is the rate at which the management frames are sent; BS is the number of beacons send per second; and Per is the fault percentage, which describes the inconsistency between the tests and the measurements.

This model is important, because it is the basis for the feasibility analysis in the RTN real-time network protocol. The formula and all the measurement graphs

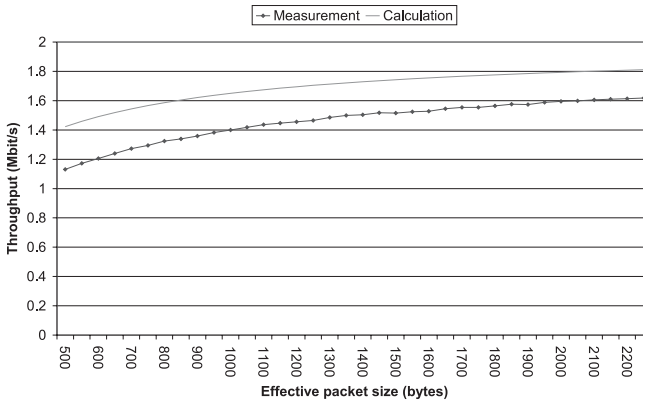


Fig. 32. Results test three using at 2 Mbit/s

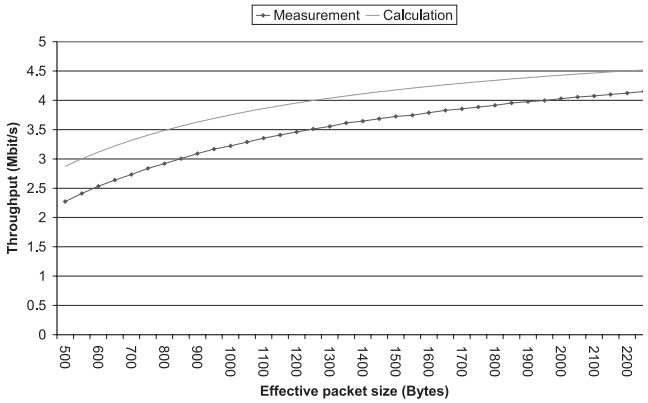


Fig. 33. Results test three using at 5.5 Mbit/s

Rate	Bandwidth
1 Mbit/s	0,74 Mbit/s
2 Mbit/s	1.5 Mbit/s
5.5 Mbit/s	3.9 Mbit/s
11 Mbit/s	6.97 Mbit/s

TABLE I
EFFECTIVE BANDWIDTH IN AN AD HOC NETWORK

show that the throughput is higher when larger packets are sent. To get the highest possible throughput, the packet size must be as large as possible. The value for Per will be different for each rate. However, 2, 5.5 and 11 Mbit/s are sufficiently close to each other for the same percentage. For 1 Mbit/s Per is 80% and for the other three rates Per is 88%.

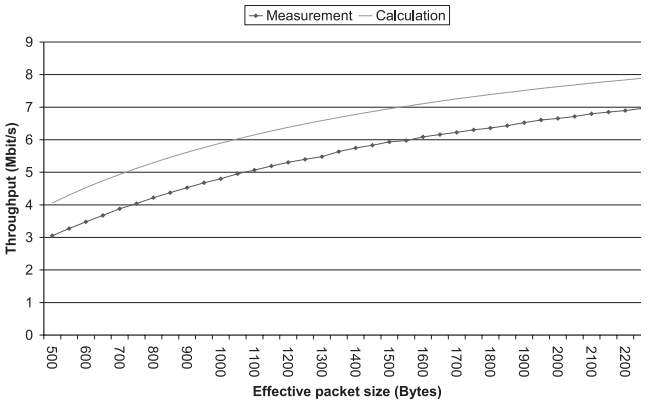


Fig. 34. Results test three using at 11 Mbit/s

It is not possible to retrieve the number of beacons per second BS from the drivers, but during testing, an average of 10 beacons per second was measured. Therefore BS will be 10. Using these values the rates shown in table I are calculated.

A prototype of RTN is being build and research is in progress to avoid the problem of hidden nodes.

REFERENCES

[1] N. Malcom and W. Zhao, "The timed token protocol for real-time communications," *IEEE Computers*, vol. 10, no. 1, p. 3541, Jan. 1994.

[2] K. Sevcik and M. Johnson, "Cycle time properties of the fddi token ring protocol," *IEEE Transactions on Software Engineering*, vol. 13, no. 3, pp. 376–385, Mar. 1987.

[3] G. C. Buttazzo, *Hard Real-Time Computing Systems – Predictable Scheduling Algorithms and Applications*. Kluwer Academic Publishers, 1997, iSBN 0-7923-9994-3.

[4] T. Hattink, "HOTnet, a real-time network protocol," Master's thesis, University of Twente, August 2001.

[5] *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE std. 802.11-1999 ed., Institute of Electrical and Electronics Engineers, 1999, ISO/IEC 8802-11:1999.

[6] *Supplement to IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band*, IEEE std. 802.11b-1999 ed., Institute of Electrical and Electronics Engineers, 1999.

[7] S. Choi, J. del Prado, and M. Sherman, "802.11a and 802.11b maximum throughput for simulation model conformance," IEEE, Tech. Rep., January 2001.