A Real-Time Multimedia Streaming Protocol for Wireless Networks

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*— This paper describes a new token-based medium access protocol for real-time networks and its implementation on a wireless network. Originally, the protocol is developed for use in low cost domestic or home networks that are based on Ethernet hardware. In contrast to existing protocols the token is assigned to network nodes on basis of a pre-emptive earliest deadline first (PEDF) schedule of the multimedia streams. The scheduler is distributed over all active nodes in the network. Although other schedulers could be used, PEDF is chosen because it has a theoretical bandwidth utilization of one hundred percent and feasibility analysis is very simple, so even simple devices can participate in such a network. This is confirmed by simulation experiments and a prototype based on Ethernet hardware. The protocol is successfully adapted to and implemented on an IEEE 802.11b wireless LAN, even though this type of network has some unpredictable properties, such as bandwidth switching.**

***Index Terms*— Real-time, token network, EDF, wireless, medium access protocol, streaming media.**

I. Real-Time Network Scheduling

The application of a token to achieve real-time behaviour in a network is not new. Examples are IEEE802.4 token bus, IEEE802.5 token ring and FDDI. The main properties are described by Malcom and Zhao [1] and by Sevcik and Johnson [2]. In these timed token networks every node in the network is visited once during one rotation of the token, which in worst case wastes a lot of bandwidth. The protocol we propose also uses a token, however it does not follow a simple round robin schedule. Instead the token is scheduled to visit only those nodes that need servicing. The token is not simply passed on to the node next to the node holding the token, but the token is passed on based on a pre-emptive earliest deadline first (PEDF) schedule. The network is stream-based, which means that a node that wants to send data has to add a stream (or channel). This stream is specified by passing a period and a required bandwidth to the network socket. The scheduler then decides whether the new network configuration is feasible and if it is, the scheduler calculates a new PEDF schedule for all the streams in the network. The most important aspects of the real-time network protocol will be described next. For a more detailed explanation see [3] or [4].

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

A. The Scheduler

The scheduler is a pre-emptive earliest deadline first scheduler. As the name already indicates, this scheduler schedules the stream with the earliest deadline first. It is pre-emptive because an instance of a periodical stream can be interrupted by another stream if this stream has an earlier deadline, but arrives later. This is shown in

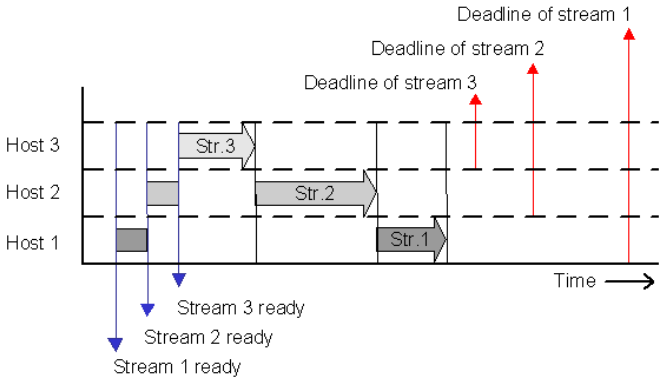


Fig. 1. Pre-emptive EDF scheduling

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. The PEDF scheduler has some nice properties. The most important one is that it can utilise the network at one hundred percent of its bandwidth. Other advantages over alternative scheduling algorithms are that streams can be dynamically added and removed, it can handle periodic and a-periodic data and it has a fairly simple feasibility analysis formula. A

disadvantage is that PEDF scheduling performs badly in the presence of an overload. But because overloads are avoided by the feasibility analyses this is not a problem. For more information see [5]. 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. Because in the real-time phase only those nodes with real-time streams are visited by the token, the non-real-time phase is also used for network management. This is the phase in which nodes can add new real-time streams to the PEDF schedule, or new nodes may announce their arrival in the network.

B. Feasibility Analysis

In a real-time system a task should never miss its deadline. 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 will be rejected if it can not be scheduled. According to [5], 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
- T_i: Period of task i

The feasibility analysis of the network is derived from the standard PEDF feasibility analysis [3]

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

Where

- B_i: Bandwidth of stream i
- B: Maximum available bandwidth of the network

When the streams in the network meet this requirement, the PEDF scheduler will find a schedule.

C. Ethernet Simulation and Prototype

The network and its PEDF token mechanism are simulated and a first prototype based on Ethernet is built. Figure 2 shows dynamic graphical simulator output for the PEDF scheduling of a set of periodic streams. The graph shows for every stream the remaining number of bytes to be sent during that period. When the stream is transmitted this line decreases linearly with the number of bytes sent. A horizontal line shows where the stream is pre-empted by another stream, so no collisions occur in the network.

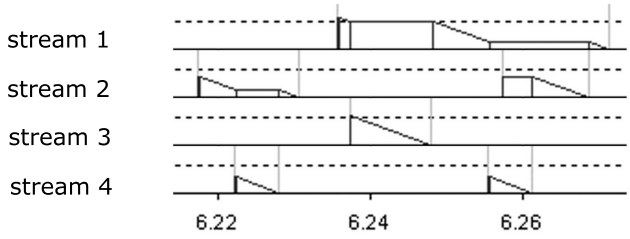


Fig. 2. PEDF schedule of a set of periodic streams

Measurements taken in the prototype, based on the Linux operating system and Ethernet hardware confirm the validity of the simulation and its parameters. More information on the Ethernet implementation can be found in [6].

II. Introduction Wireless LAN

In the previous section a real-time protocol (RT protocol) based on Ethernet hardware is introduced for domestic networks. The choice of Ethernet hardware is dictated by the requirement that the network must be cheap, and even small and cheap devices should be equipped with a network connection. However, there is a tendency to use wireless means for communication at home. In this section we will give a short description of those aspects of a WLAN that are of interest for the mapping of the real-time protocol.

WLAN is a layered protocol and its layers originate from the IEEE 802.x standards. The complete specification is defined in [7]. The architecture of an IEEE 802.11 wireless network is in some respects 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 WLAN is different from the datalink layer of (wired) Ethernet.

A. Topology

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.

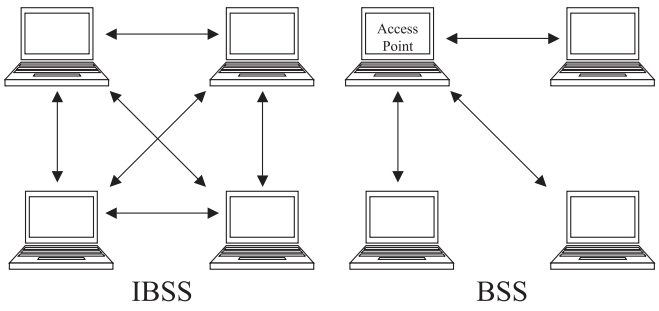


Fig. 3. Difference in topologies of BSS and IBSS

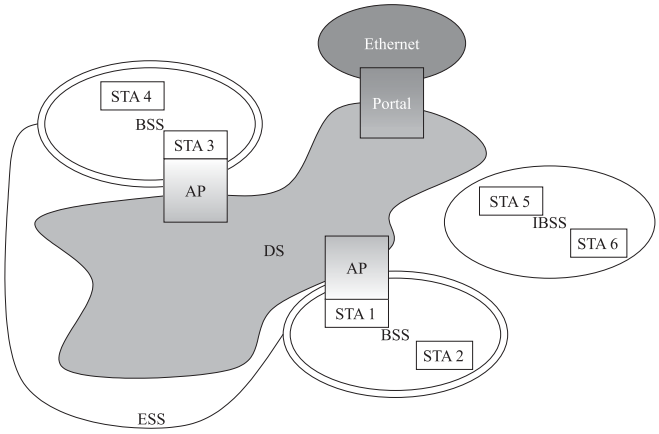


Fig. 4. Example of an 802.11 topology

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 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 3. 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 4 gives an overview of the 802.11 topology.

B. Communication

WLAN 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.

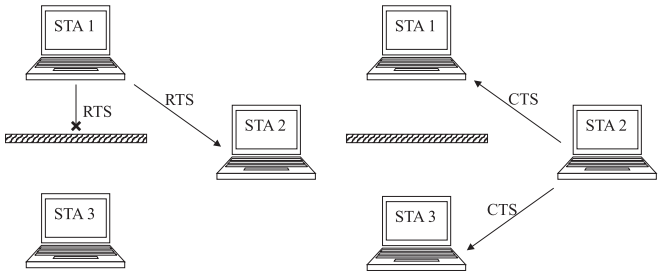


Fig. 5. Working of the RTS/CTS mechanism

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 5. 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.

C. Beacon Frames

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.

III. Mapping of the RT protocol on Wireless LAN

Trying to combine the IEEE 802.11 standard for WLAN and the proposed real-time network protocol is not a trivial matter. Although there are strong similarities between Ethernet and WLAN, there are mapping issues that need to be solved. In the following we will address these issues.

A. Topology and Network Mode

The proposed real-time network protocol on Ethernet is based on a fully connected network topology. When a node has finished its time slot, it will send the token to the following node in the schedule. This means that every node must be able to reach every other node in the network since there is no token forwarding mechanism present in the system. WLAN 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.

From the previous the following solutions present themselves: (1) a token forwarding mechanism is introduced which guarantees that every node can send to every other node; (2) the managed network mode, which presents a fully connected network, is used; and (3) an environment is assumed in which there are no hidden nodes, so the ad hoc mode can be used. This may seem improbable, but measurements have shown that this is likely the case in a home environment.

In order to implement the RT protocol on WLAN, 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, the RT protocol 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. The RT protocol 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 the RT protocol. 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 and the hardware is not available. 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.

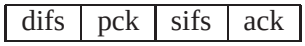
B. Beacon Frames

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. These collisions can only occur when the node with the token starts sending at the same moment the Access Point decides to send a beacon frame. Sending beacon frames adheres completely to the rules to which all the other frames have to obey, which ensures that beacon frames are only sent when the medium is clear. These rules include the random back off algorithm, which reduces the chance of a collision, since there are only two possible stations that want to send, the Access Point and the node holding the token. There are two possible solutions to tackle this problem. The first one is not really a solution: maybe it is sufficient to ignore beacon frames. Beacon frames obey the normal frame rules, which reduce the chances of a collision significantly. Beacon frames are not sent very often and consist of only one frame, which causes them to use very little bandwidth indeed. Perhaps this loss of bandwidth is so little that it can be ignored. A far more elegant 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.

C. Bandwidth

Considering the previous points, an estimate can be made for the effective available bandwidth in the network. It is important to know the effective available bandwidth, because it is an essential element in the feasibility analysis of the RT protocol. 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). A major performance difference between both modes is not yet mentioned. Ad hoc is at least two times faster than a managed network. This has to do with the relaying of the frames 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 approximately 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 6(a).



(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. 6. Communication without RTS/CTS and back off

The duration of this sequence can be calculated ([7] and [8]) and the effective bandwidth at different packet

sizes is summarised in figure 6(b).

Figures 7(a) and 7(b) summarise managed networks without RTS/CTS, but with back off, while figures 7(c) and 7(d) summarise managed networks with RTS/CTS and back off.

difs	backoff	difs	pck	sifs	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. 7. Communication in varied modes

Ad hoc mode has no relaying access point, which means that in theory the maximum throughput is doubled when compared to managed mode. Because the RT protocol guarantees that only one STA is sending at one time and since there is no relaying AP, there are no problems with back off. Taking beacon frames into account the estimated throughput of ad hoc networks is summarized in figure 8.

	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. 8. Effective bandwidth in ad hoc mode

A comparison of available bandwidth in managed and ad hoc networks is shown in figure 9. It shows that ad hoc has a far better throughput than managed mode.

	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
mngd	55 kB/s	103 kB/s	238 kB/s	379 kB/s

Fig. 9. Comparison ad hoc and managed mode

D. Dynamic Bandwidth and Feasibility Analysis

Part of the RT protocol is the feasibility analysis to check whether the network can handle the requested load or not. The original feasibility analysis is based on the assumption that the network bandwidth is constant, which is not the case in a WLAN. The available bandwidth between two nodes can deteriorate within the duration of one time frame, due to interference or other circumstances. In addition, available bandwidth can differ between different points in the same network. So in one and the same network bandwidth can vary from one to several tenths of Mbit/s. The standard feasibility analysis is not able to cope with these changes in bandwidth and must be adapted to the new network [9]. A solution is to base the feasibility analysis on the lowest possible bandwidth. The lowest possible bandwidth is defined in the standard at 1 Mbit/s. When even this speed cannot be sustained, there is no connection at all. The RT protocol was designed to be able to cope with a broken connection, so this is not a problem. If the actual bandwidth is higher than the minimum bandwidth the surplus bandwidth could be used for non-real-time communications. A slight variation would be a variable minimum bandwidth, which depends on the actual situation.

When a node needs to decrease its bandwidth due to interference, bandwidth shortage could appear. When a stream requires 8 Mbit of available bandwidth each second and the node switches back from 11 Mbit/s to 5.5 Mbit/s, this requirement cannot be met. The node will be notified that the stream cannot be sustained. This looks pretty straightforward. But, when multiple streams are involved, priorities must be established. The scheduler itself can assign priorities, based on the data in the token, or the user requests a priority from the network.

1) Token-Data Based Priorities: There are lots of possibilities of prioritising based on data about a certain stream in the token. This section will examine some options. The main problem with this approach is that the data in the token has no information on the semantic priority of a thread.

Creation Time: The streams are registered in the token in order of their creation. This can be used for prioritising the streams. The oldest stream can be stopped first or the newest can. Both approaches have merely one advantage:

they can be easily implemented. The downside is that they are purely random mechanisms, since the creation order is in no way related to the semantic priority a stream could have.
Periodicity: The streams could be prioritised based on their periodicity. One of the most efficient fixed priority schedulers, Rate Monotonic, uses this mechanism for assigning priorities. However, Rate Monotonic has nothing to do with semantic priorities. This approach has the same properties as prioritising on creation time.
Total Load: Prioritising on total load per second can be a good choice. Since the semantic priorities cannot be deduced from the data in the token, the chance that a stream with a large load has a high semantic priority is the same as a stream with a small load. Therefore, when a stream with a large load is shut down, multiple streams with a smaller load can possibly be sustained, which otherwise would have been stopped. This method provides a bigger chance of a semantically important stream being sustained.

Prioritising on load seems the most efficient solution. Although it cannot really decide which stream has a larger semantic priority, it has the biggest chance of retaining the higher priority streams.

2) User-Defined Priorities: In this context, 'user-defined priorities' does not mean that the human user selects a priority at which the stream should run (although a program could offer that kind of functionality), but that the program that initiates the stream decides which priority is needed for that stream. Two mechanisms can be applied in this case. One mechanism gives guarantees about a bandwidth up to a certain point. The other one does not give such guarantees.

Without Guarantees: This mechanism is nearly trivial. The user assigns a priority to the stream and when the bandwidth drops, first the streams with the lowest priorities are terminated. When multiple streams have the same priority and only some of them need to be terminated, token-based priorities can be used, such as the total load prioritising scheme.
With Guarantees: Priorities are mapped on standard bandwidth values. When a node requests bandwidth for a stream at high priority, this bandwidth will be guaranteed at a low standard bandwidth value, e.g. in 802.11b 1 Mbit/s. To be able to guarantee the requested bandwidths at different speeds, multiple feasibility analyses have to be done, beginning with a feasibility analysis of the current state of the network at the current speed with the new stream. When this scheme is not feasible, the new stream is rejected. When this schedule is feasible the stream has to be tested with lower standard bandwidth values, together with the other streams that are guaranteed at those bandwidths. When the bandwidth drops, the streams with a lower priority than the priority that is associated with this bandwidth are terminated when the original stream-set is not feasible at this bandwidth. When the lower priority streams fit in the schedule, there is no reason to terminate them. It could happen that there are multiple streams with the same priority, but not every stream needs to be terminated. In that case a token-based priority can be used, for example the total load prioritising mechanism.

An example on an 802.11b network to illustrate the mechanism: There are three streams in the network. Stream one has a period of a half second and a load of a half Mbit each period. This stream has the highest priority, which maps on 1 Mbit/s. The second stream has the same properties, only its priority is medium, which maps on 5.5 Mbit/s. The last stream has a period of one second, a load of 3 Mbit each period and the lowest priority, which maps on 11 Mbit/s.
This example will use a slightly simplified model of the 802.11b standard; the bandwidths can be completely efficiently used and the streams do not run through an Access Point. In order to check whether the schedule is feasible, a feasibility analysis has to be performed at each used priority level. This also has to be done each time a stream has to be added, but in that case only the priority level of that stream and lower need to be done, the higher levels remain unchanged.

- Lowest priority at 11 Mbit/s.
$$\frac{0.5}{0.5 \cdot 11}+\frac{0.5}{0.5 \cdot 11}+\frac{3}{1 \cdot 11}=\frac{1}{11}+\frac{1}{11}+\frac{3}{11}=\frac{5}{11} \leq 1$$
- Low priority at 5.5 Mbit/s. Only stream 1 and 2 have to fit in this schedule.
$$\frac{0.5}{0.5 \cdot 5.5}+\frac{0.5}{0.5 \cdot 5.5}=\frac{2}{11}+\frac{2}{11}=\frac{4}{11} \leq 1$$
- Medium priority at 2 Mbit/s. Stream 1 and 2 have to fit in this schedule.
$$\frac{0.5}{0.5 \cdot 2}+\frac{0.5}{0.5 \cdot 2}=\frac{1}{2}+\frac{1}{2}=1 \leq 1$$
- High priority at 1 Mbit/s. Only stream 1 has to fit in this schedule.
$$\frac{0.5}{0.5 \cdot 1}=1 \leq 1$$

These results show that the system is schedulable observing the restraints of the prioritising method. As shown in the first equation the system is schedulable on 11 Mbit/s with all three streams, the resulting schedule is shown in figure 10.

When the bandwidth drops, the scheduler can drop all the streams that are scheduled at a lower priority.

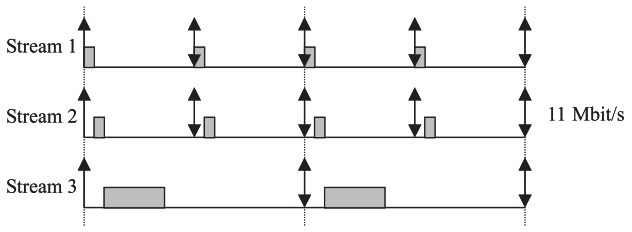


Fig. 10. User defined priorities with guarantees example at 11 Mbit/s

But when there is no need to drop (some of) the lower priority streams, there is no reason to not keep them in the system. In order to check whether these lower priority streams fit in the system, a feasibility analysis at that speed has to be done.

Imagine that in the example system the bandwidth drops from 11 Mbit/s to 5.5 Mbit/s. To check whether streams (stream 3 in this case) have to be dropped, a feasibility analysis is done:

$$\frac{0.5}{0.5 \cdot 5.5} + \frac{0.5}{0.5 \cdot 5.5} + \frac{3}{1 \cdot 5.5} = \frac{2}{11} + \frac{2}{11} + \frac{6}{11} = \frac{10}{11} \leq 1$$

The schedule is still feasible, so there is no need to drop stream 3. The resulting schedule is depicted in figure 11. When the bandwidth drops further to 2 Mbit/s, stream 3 has to be dropped:

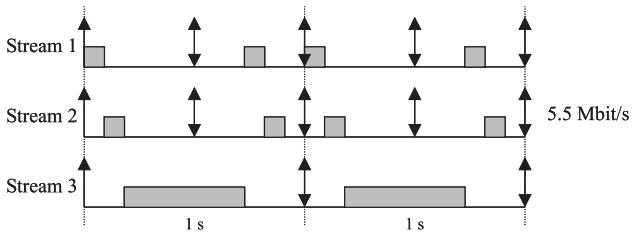


Fig. 11. User defined priorities with guarantees example at 5.5 Mbit/s

$$\frac{0.5}{0.5 \cdot 2} + \frac{0.5}{0.5 \cdot 2} + \frac{3}{1 \cdot 2} = \frac{1}{2} + \frac{1}{2} + \frac{3}{2} = \frac{5}{2} \not\leq 1$$

Only the streams with a priority higher than or equal to the priority corresponding to this bandwidth, in this case the medium priority can be sustained. The feasibility analysis has already been done in the initial feasibility analysis. The new schedule is drawn in figure 12.

When the bandwidth is lowered to its minimum, only the high priority threads should be sustained. To test whether the medium priority thread can be sustained the following feasibility analysis is done:

$$\frac{0.5}{0.5 \cdot 1} + \frac{0.5}{0.5 \cdot 1} = 1 + 1 = 2 \not\leq 1$$

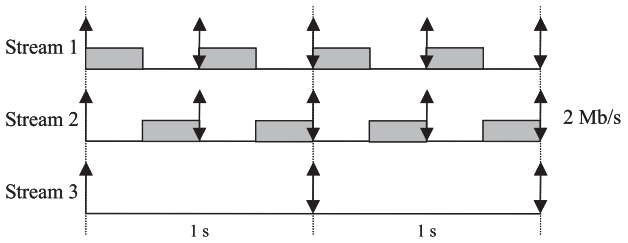


Fig. 12. User defined priorities with guarantees example at 2 Mbit/s

According to this analysis stream 2 has to be dropped. This results in a scheme with only one stream (see figure 13).

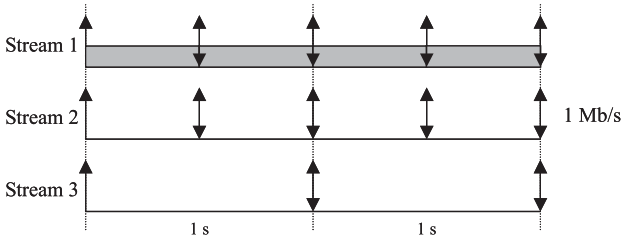


Fig. 13. User defined priorities with guarantees example at 1 Mbit/s

The ‘User defined priorities with guarantees’-scheme is chosen as the main prioritising scheme. It has the great advantage that hard real-time streams can be sent with the maximal guarantee, while soft real-time streams, such as video can use the full advantages of the speed of WLAN. As a sub-scheme the ‘Total load’-scheme is chosen, it is easy to implement and has the highest chance of retaining important streams.

IV. Testing and Measurements

The RT protocol is tested with standard PCs fitted with Orinoco Silver WaveLan Cards with firmware 8.10, and a Compaq WaveLan Card with firmware 0.7.4. The test setup is completed by an Orinoco AP-1000 Access Point. All tests are performed in a computer lab. There are no microwaves or other sources of interference. The frequency exclusively used for testing is the 2.462 GHz band (channel 11). All the computers run the Linux operating system with a modified 2.4.18 kernel, using the pcmcia-cs 3.1.34 package with the Orinoco drivers version 0.13.

A. Basic Test: two nodes, one data stream

This test examines sending and receiving streams using the RT protocol. Both nodes send a stream to the

other node. These streams have a period of 1 second with a load of 100 kB/s and a running time of 120 seconds. One stream has the lowest priority. See figure 14. Every period node 2 receives 100 kB effective data. After this 100 kB has been delivered node 1 will stop sending data, this way the ‘stairs’-effect is obtained. Node 1 generates the nearly vertical line when it sends its data. When node 1 is not sending data, the horizontal line is generated. The horizontal line is about 0.887 s long, so node 1 is sending for 0.113 s, delivering a throughput of 884 kB/s. Figure 15 shows the duration between

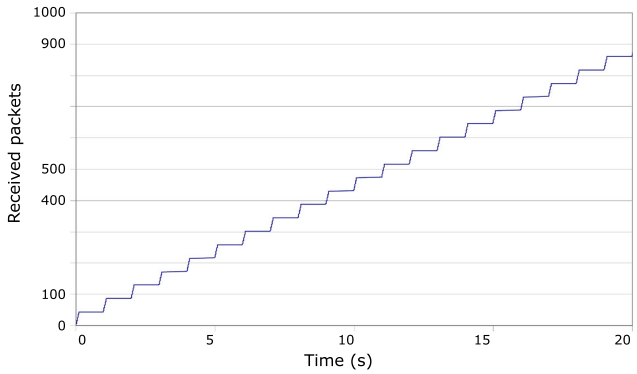


Fig. 14. Two nodes one stream - plot

receiving two packets. Each time node 1 is done with its load, it will wait until its period starts again. These waiting times are shown by the points in the figure with a latency of about 0.89 s. The other points are generated when node 1 is busy sending its 100 kB. The packets are sent right after each other, generating a latency of almost zero. The last graph generated from the data of

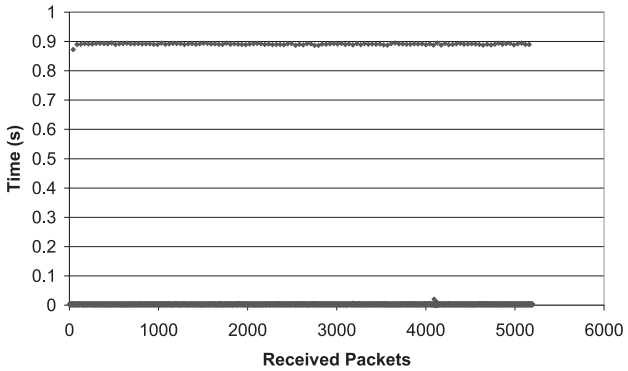


Fig. 15. Two nodes one stream - latency

this test is shown in figure 16 and shows the effective throughput. The throughput fluctuates heavily in the beginning. First node 1 sends packets at its maximum speed until it has sent 100 kB and then waits until its next period. Therefore the throughput is higher in the beginning of each second. However, when more and more measurements are obtained, this effect becomes less and throughput is a sustained 100 kB/s.

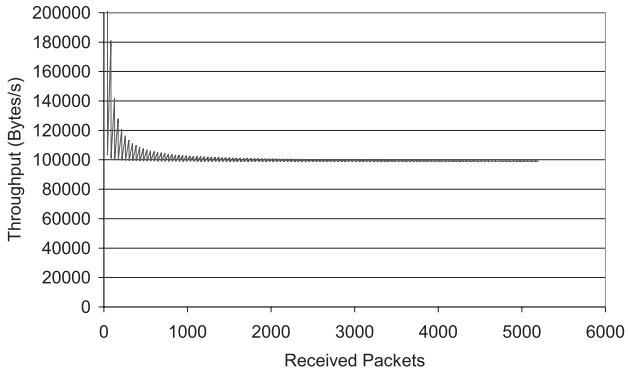


Fig. 16. Two nodes one stream - throughput

B. Testing Pre-emptive Scheduler

This test examines whether the pre-emptive scheduler works properly. This test uses two nodes and two streams. The first stream has a load of 550 kB per period of 1 s sending from node 1 to the node 2. The second stream has a load of only 10 kB per period of 0.5 s sending from the node 2 to node 1. With this test both streams 1 and 2 are in the system at the same time. Stream 2 will pre-empt stream 1 every time. This test will be compared to the results obtained when stream 1 is the only stream in the system. As can be seen in figure 17, the stream that is getting pre-empted has a slightly better throughput than the one that is not pre-empted, while the same results were expected. This has to do with the rounding off to whole time slices. The arrival time has a very high precision, which means that an arrival can occur in the middle of a time slice. This means that the stream that is being pre-empted gets more time to send, because it may send until the current time slice is over. This effect is also noticeable in the throughput figure 18. The stream with pre-emption has a higher throughput than the one without pre-emption.

C. Changing Bandwidth

This paragraph describes a test where bandwidth is changed during the course of communications. Two nodes will run the RT protocol without any extra streams. The rate of the WLAN card of node 2 will be lowered from 11 Mbit/s to 1 Mbit/s, and increased to 11 Mbit/s and lowered to 5.5 Mbit/s, 2 Mbit/s and finally 1 Mbit/s. The rate will be changed with the help of the ‘iwconfig’ command. Unfortunately this test did not go as planned.

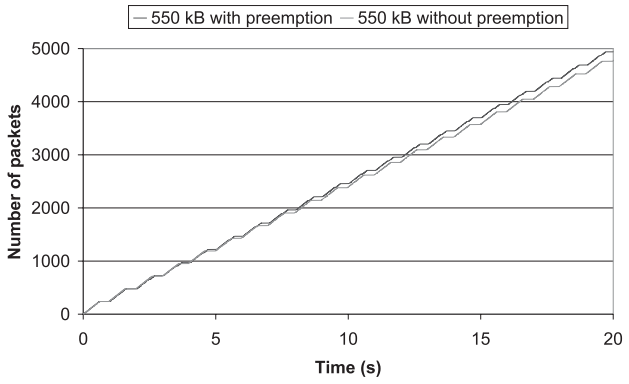


Fig. 17. Stream 1 (550 kB/s) with and without pre-emption - plot

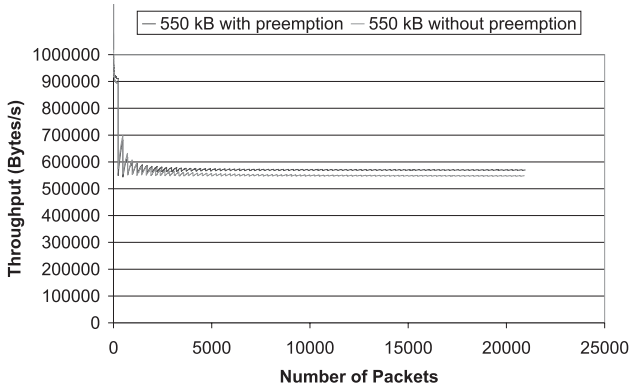


Fig. 18. Stream 1 (550 kB/s) with and without pre-emption - throughput

At a certain time one of the two nodes misses deadlines when the bandwidth is reduced by hand. However, this does not occur when the bandwidth is increased. One explanation for this unfortunate phenomenon is latency. If the WLAN card takes too long to switch to a lower bandwidth, this could take more time than the token holding time. This explains deadline misses. The other node will decide that the node with the token has crashed (because it has not received the token in time) and will start its own network. The result is that there are two nodes in the network with each their own real-time network (the original token holder did not crash at all, it just was delayed). When the network itself switches to a higher or lower bandwidth then everything works correctly. Therefore it is probable that the influence of 'iwconfig' is larger than only switching back the bandwidth. It probably has to lock the network card to switch rates, when this lock takes too long, the RT protocol can miss its deadlines.

V. Conclusion

In this paper we have described a new token-based medium access protocol for real-time (home) networks. A simulation and an implementation on Ethernet hardware of the protocol have been presented and showed that the protocol behaves as predicted. We also presented a prototype based on IEEE 802.11b hardware. Most of the tests that are executed deliver expected results, and confirm that our protocol enables real-time multimedia communication through a wireless network.

The feasibility analysis works perfectly, as does sending and receiving real-time streams. Earliest deadline first scheduling and pre-emption of streams work as expected.

Unfortunately, the prototype does not handle bandwidth switching very well. This action is not specified in the IEEE standard. Manufacturers can implement this feature at will. Consequently it is not possible to predict precisely enough the impact on the connection when using commercially available access points and network cards with unmodified firmware.

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

[4] J. H. Wijnberg, "Prototyping the real-time 'HOTnet' network protocol on ethernet," Master's thesis, University of Twente, May 2002.

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

[6] J. Scholten, P. G. Jansen, F. T. Y. Hanssen, and T. Hattink, "An In-Home network architecture for Real-Time and Non-Real-Time communication," in *IEEE Region 10 International Conference on Computers, Communications, Control and Power Engineering (TENCON)*. Beijing, China: IEEE Computer Society Press, Los Alamitos, California, Oct 2002, pp. 728–731, `http:// www.ub.utwente.nl/ webdocs/ ctit/ 1/ 000000ae.pdf`. [Online]. Available: http://www.ub.utwente.nl/webdocs/ctit/1/000000ae.pdf

[7] *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.

[8] *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.

[9] W. H. Mank and A. Zwikker, "Designing and prototyping the real-time rtnet protocol on wireless lan," Master's thesis, University of Twente, November 2002.