A Request-TDMA Multiple-Access Scheme for Wireless Multimedia Networks

G.R.J. Linnenbank, P. Venkataram*, P.J.M. Havinga, S.J. Mullender, G.J.M. Smit

University of Twente, Dept. of Computer Science, P.O.box 217, 7500 AE Enschede, the Netherlands

* Dept. of ECE, Indian Institute of Science, Bangalore 560012, India

Abstract

This paper describes a cellular multiple-access scheme based on TDMA for multimedia communication networks. The scheme proposes an admission control of two different multimedia application stream types: real-time and non-real-time.
We do not consider interference between cells[1]. The proposed protocol, that is based on TDMA, exploits the available bandwidth fully. The throughput per mobile[2] station is higher compared to other multiple-access protocols, it offers low latency for both real-time and non-real-time communication and the unused reserved bandwidth is reallocated for non-real-time communication. Furthermore, the throughput and latency remain stable under high loads.

This research is supported by the Moby Dick project. The Moby Dick project is a joint european project (Esprit Long Term Research 20422) to develop and define the architecture of a new generation of mobile hand-held computers.

1 Introduction

In current wireless communication networks, bandwidth is a scarce resource and has to be exploited as efficiently as possible. Considerations of cost and saving power dictate that mobile wireless stations must have as few electronic components as possible. Therefore, mechanisms to exploit the bandwidth efficiently must be of low complexity.

Many multiple-access protocols have been suggested to access the available network bandwidth. In [1] we argued that most of these protocols are not suitable for our intended system. Contention based multiple access protocols such as Aloha [2], ISMA [3] [4], R-ISMA [5] [6] and PRMA [7] [8] are unsuitable due to their performance degradation under heavy loads. Source division multiple access protocols such as FDMA [9], CDMA [10] have the drawback that it is difficult to access multiple channels. In TDMA [9] channels are sequential, multiple channels can be used to obtain more bandwidth. However, once a channel has been allocated to a mobile station the bandwidth is lost when that station has an inactive period.

In our system we will have small cells (3-5 m radius) in which only a few mobile stations need to be served. In this paper, we ignore interference between cells; we will address this issue in a forthcoming paper. We use near-field radio [11] as the communication medium. The available network bandwidth is limited (1-10 Mbps) and must support a mixture of real-time and non-real-time communication for multimedia applications. When a single mobile station is active, it should be able to obtain the full link bandwidth. When more mobile stations are active, the aggregate obtainable bandwidth of these mobile stations must also approach the available network bandwidth as close as possible.

Section 2 briefly describes the protocol requirements imposed by multimedia applications. In Section 3 we will describe the near-field wireless network system for which we intend to use the multiple-access protocol. Then we will describe the multiple-access protocol in detail in Section 4. Throughput and latency analysis of the protocol is given in Section 5 for both real-time and non-real-time communication. In Section 6 we will present the simulation configurations followed by the results in Section 7. The conclusions are given in Section 8.

2 Multimedia application requirements

Multimedia applications have properties that cannot be supported by many of the known multiple-access protocols. Some applications require real-time behaviour. Therefore the protocol must have low-latency characteristics.

In a multimedia environment, the information of one application can have a higher priority than that of another application. For example, delaying a file transfer will not be hazardous but it is desirable that the transfer of an audio packet is not delayed. Few wireless multiple-access protocols (like the one described in [13]) have successfully included priorities. We must include priorities in the multiple-access protocol to support multimedia applications successfully.

Multimedia applications may also generate variable amounts of data in time. For bursty real-time applications, such as variable-bit-rate video, reservations can be made for the peak bandwidth. However, most of the time the bit rate is much lower. Therefore, a significant portion of the reserved bandwidth will not be used and the utilization of the network bandwidth is not optimal. Since the available bandwidth in our wireless network is scarce (1-10 Mbps), we must be able to reallocate the reserved but unused bandwidth to non-real-time applications.

Summarising, a protocol is needed that is stable regarding throughput and latency under high loads. It must support priorities (at least real-time and non-real-time communication) and it must be able to dynamically allocate bandwidth to utilize the bandwidth optimally.

3 The System Architecture

Our goal is to design and implement a wireless network for an office building environment with one base station per room and without inter-cell interference, see Figure 1. Base stations are connected to a backbone B-ISDN network. In each room a

1. When we use the word cell, we refer to a cellular network cell. A B-ISDN ATM cell, is referred to as ATM cell to avoid confusion.

2. In this paper we consider a mobile station to be a wireless station that is not a base station. Since we only consider single cell environments, mobility itself is not an issue here.

small number of mobile multimedia stations can be present.

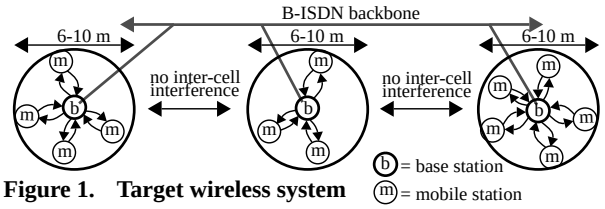

Figure 1. Target wireless system

We use near-field radio as the transmission medium. As the power of a near-field radio signal drops an order of magnitude more rapidly than a high-frequency radio signal, it offers a good separation between cells [11]. Therefore, cells can be placed close together without introducing inter-cell interference. In an office building, each room can be provided with the full bandwidth of a base station as no channel reuse schemes are needed, giving a high bandwidth density. In our prototype system we will use a radio link providing a bandwidth of 1 Mbps using binary-PSK.

Since we intend to avoid overlapping cells, continuous mobile communication is not supported. The communication of a mobile station is suspended when it moves from one room to another and continued when a new connection has been made with the new base station. In this paper, the functionality requirements for connect/disconnect, hand-over, etc., is not an issue since these functions must be implemented at a higher level in the protocol stack.

4 The MAC layer Protocol

4.1 Design assumptions

Because the B-ISDN network connecting the base stations is an ATM network, we use ATM cell transmissions over the wireless link. This keeps the base station architecture simple since no message and protocol conversion is needed between the wireless channel and the B-ISDN network.

The prototype transceiver runs at 1 Mbps. The basic unit of information exchange is a byte, allowing the exchange of very short messages between base and mobile stations. Normal messages will be ATM cells which are 53 bytes large. All messages are preceded by 2 synchronisation bytes.

The error rates of the near-field wireless link still have to be investigated. When the channel error rates are too high, extra error detection/correction coding must be used to avoid unnecessary processing of corrupted packets. For this paper, we assume an errorless transmission channel.

Each mobile station can have multiple connections with a base station. In our protocol, a ***connection*** is a communication agreement between base station and mobile station. For each connection an amount of bandwidth can be reserved during connection set-up. A connection that reserves bandwidth is called a ***real-time connection*** and a connection with no bandwidth guarantees is called a ***non-real-time connection***.

4.2 Frame structure

In the prototype R-TDMA protocol, time is divided into fixed size frames. A *frame* contains *S* time slots for communication. The frame structure is shown in Figure 2. There are two special slot types in each frame, the *Clear To Send* (CTS) and the *Request To Send* (RTS) slots. The first *C* slots in the frame are the CTS slots. The base station uses these slots to inform the mobile stations to which connections the data slots are allocated in the *current frame*. Somewhere in the frame are the *R* RTS slots, where mobile stations can request data slots for the *next frame*. The remaining *S-C-R=D* slots are the *data slots* which can be used for data communication.

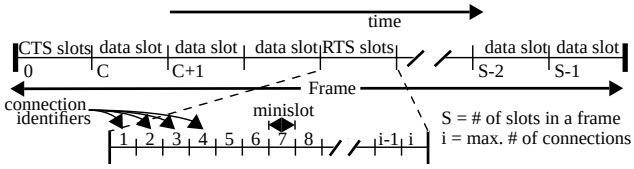

Figure 2. The R-TDMA frame structure

In a normal system, both uplink and downlink data traffic will be present. This paper focuses on the uplink traffic which is more difficult to control. Downlink traffic is not included in the analysis and simulations.

4.2.1 Request To Send slots

The Request To Send slots are divided into ***minislots*** that allow the transmission of single byte messages preceded by two synchronisation bytes. Minislots are used to inform the base station how many ATM cells the mobile stations want to send over their connections in the next frame. The base station uses this information and the Quality of Service (QoS) agreements to allocate data slots.

During connection set-up, the base station will assign the n^{th} minislot to the n^{th} connection. There can be exactly as many connections as there are minislots available. Minislots can be used in at least two different ways. First, a connection can send a request for up to *D* slots in each frame. New requests must be made in every following frame until all the messages have been transmitted. When a minislot is lost (for example due to noise), the connection will not obtain slots in the next frame and it will loose bandwidth.

Second, a connection can request more than *D* slots and the base station will allocate data slots to this connection in the following frames until enough slots have been allocated to that connection. The base station keeps track of the number of remaining slots that have to be allocated using a counter for each connection. A new request will replace the current counter value. The loss of a request will not decrease the throughput of a connection much, provided that the related counter has not already reached zero. The base station still can use the older counter value to allocate slots to the connection who's request was lost. As the loss of a request has no severe consequences, it will not be necessary to add error detection to the requests messages. All the bits can be used for request information.

Since we assume an error-free channel we will use the first scheme that is less complex.

4.2.2 CTS slot

The base stations use the Clear To Send slots to send CTS messages. A CTS message has two primary functions. First it informs every mobile station in its reach that a base station is present and what its name is. This information is needed since the mobile stations must know that they are in reach of a certain base station in order to set up a connection.

The other function is to broadcast a table containing the slot allocation for the current frame to the active mobile stations that reside in the base station's cell. The table contains a row for each slot in a frame indicating to which connection it is allocated. Slots that have not been allocated to a connection may be used for connection set-up. Usually free slots are available during low loads. Under high loads a slot must be kept free regularly by the base station to enable new mobile stations to set up a connection.

For short frames a single CTS slot is sufficient. More CTS slots are needed for larger frames. Although, in that case, more slots are used for CTS transmissions, the relative contribution is linear with the frame size.

4.2.3 Data slots

The data slots are used to send packets containing application data. We use ATM cells in order to simplify the translation to the fixed ATM network.

In B-ISDN networks the ATM header requires 5 bytes to detect header errors and to identify all types of communication from all connections on a single link. In our system, the base station already knows the source connection of the packet. Hence, less header information is needed. A header of one byte is sufficient to rebuild the ATM header at the base station for a single connection (5 byte ATM UNI header - 2 bytes VCI - 1 byte VPI - 1 byte HEC = 1 byte).

ATM was designed for fixed networks with very low bit error rates. Error checking was only applied to the header in order to avoid unexpected side effects or misdelivery. In wireless networks the bit error rates are much higher. Therefore, the ATM header CRC should be replaced by a full-packet CRC to avoid unnecessary processing of corrupted packets. The total cell overhead can be reduced from 9.4% (5 bytes/53 bytes) to 5.8% (3 bytes/51 bytes) by using a one byte header and a two byte CRC. This reduction is significant. But for simplicity and because an error-free channel is assumed, we will use full ATM cells in our prototype system.

4.3 Transmission Policy

Transmissions are performed in two stages. First, a station sets up a connection and, when succeeded, successive transmissions are performed. When the connection is no longer needed it is closed in order to free the reserved bandwidth and the associated minislot for other communication.

4.3.1 Connection Setup

The connection setup is performed on a contention basis. It is the only stage of the protocol where collisions can occur. A mobile station that wants to setup a connection waits for a CTS message from the base station. The CTS message indicates which data slots are unallocated in the current frame. The mobile station selects a data slot from the unallocated slots according to a Slotted Aloha mechanism with probability *p*. It is possible that no free slot is selected. In that case, the mobile station will wait for the next CTS and restart the connection setup procedure. When a slot is selected the mobile station sends a connection request message using that data slot and waits for a reply in the next frame. When no reply is received, there may have been a collision or there was no data slot available for the base station to send the reply. In either case, the mobile station will reduce its free-slot access probability *p* and try again. When a negative reply is received, the connection was refused due to the unavailability of a minislot or the QoS requirements could not be met. In the first case the mobile station can try again later and in the latter case it can try again with lesser QoS requirements. When a positive reply is received, the mobile station goes to the transaction phase. A formal description of the connection setup algorithm performed by the mobile station is given below.

Mobile Station Connection Setup Policy

1 Set free-slot access probability p to an initial value ($0<p<1$).
2 From CTS message of frame F_j extract ordered set V containing the unassigned slots U_i in frame F_j.
3 With probability $P(i) = p(1-p)^{i-1}$ select U_i from V.
When no slot is selected proceed at step 2 in frame F_{j+1}.
4 Send a request using the selected free slot U_i of frame F_j.
5 If no reply received in frame F_{j+1}
then reduce p and proceed at step 2 in frame F_{j+2}.
6 If a reply is received in frame F_{j+1}
then the connection has been accepted or refused.

The processing of the requests by the base station is straight forward. The following algorithm describes the way each request is handled, where B_c is the bandwidth guarantee requested for the new connection, B_r is the bandwidth reserved currently and B is the total link bandwidth that may be reserved.

Base Station Connection Setup Policy

1 Get the required bandwidth B_c in the connection request in frame F_j.
2 When $B_r+B_c > B$ or no minislot is available
then a connection-refuse reply message is created.
3 When $B_r+B_c \le B$ and a minislot is available
then select minislot and create a connection-accept reply message.
4 Request a slot from the slot scheduler for the reply in frame F_{j+1}.
5 If the reply is not allocated in frame F_{j+1}, discard the reply.

4.3.2 Transactions

After a connection is set up successfully, mobile stations request bandwidth on demand. When the queue at the mobile station contains one or more messages, the mobile station invokes the transaction procedure. The system makes sure that it provides each connection with at least the minimum bandwidth agreed upon. After sending a request, the mobile station monitors the next CTS message to see which slots it may use for data transmission.

Mobile Station Transaction Policy

1 At RTS time of connection c in frame F_j compute $n_{cj}=m_{cj}-r_{cj}$.
m_{cj} is the number of queued messages of connection c in frame F_j.
r_{cj} is the number of remaining slots of connection c in frame F_j.
2 Transmit a message requesting n_{cj} data slots for frame F_{j+1}.
3 From the CTS message of frame F_{j+1}, r_{cj+1} slots are allocated to connection c ($r_{cj+1} \le n_{cj}$) and ($n_{cj} \ge g_c \Leftrightarrow r_{cj+1} \ge g_c$).
g_c is the guarantee for connection c.
4 Data transmissions of connection c for frame F_{j+1} are scheduled.
5 The connection proceeds at step 1.

In each frame, the base station will receive all the slot requests and allocate the data slots based on the current need and the QoS parameters. The allocation algorithm first assigns data slots to the real-time connections up to the guaranteed number of slots. Then the remaining data slots are allocated until the frame is filled or all requested slots have been allocated. A formal description of this transaction policy is as follows:

Base Station Transaction Policy

1 At RTS time in frame F_j collect requests n_{1j}, .., n_{mj} for frame $F_{j+1.}$
m is the number of active connections.
2 For each connection c_i: $s_{ij} = \min(n_{ij}, g_i)$ and $n_{ij} = n_{ij} - s_{ij}$
3 For each connection c_i allocate 1 slot and reduce n_{ij} by 1
until $n_{1j}+n_{2j}+..+n_{mj} = 0$ or $s_{1j}+s_{2j}+..+s_{mj} = D$.
4 Create CTS message containing the computed allocation
5 Send CTS message at start of frame F_{j+1} and continue at step 1.

An idle mobile station only needs to receive the CTS messages to see if it will receive new messages. The mobile station does not need to be active between two CTS messages. Therefore, we consider the protocol to be energy efficient.

5 Analysis

When the communication requirements of the users in the system are all different, the channel allocation and access control schemes of R-TDMA are too complex to be modelled. Therefore, we use a model where each connection of the same type generates Poisson traffic with equal intensity.

5.1 Throughput

The aggregate throughput depends on the offered load in a very simple way. When the average total number of packets generated in a frame does not exceed the number of data slots in a frame, all requests can be honoured. Therefore, the throughput is equal to the load when the load does not exceed the network capacity.

When the average total number of requested slots is more than the number of data slots in a frame, some requests will not be honoured but all data slots in the next frame will be allocated. Since some connections will not get enough slots, they will send new requests in the next frame.

For a frame of size *S*, *D* data slots and an average load of *G*, the throughput η is given by

$$\eta(G) = \begin{cases} G & G \le \frac{D}{S} \\ \frac{D}{S} & G \ge \frac{D}{S} \end{cases} \quad \textbf{Eq. 1}$$

As *D = S-C-R*, *D/S* will grow with larger frame sizes when *C* and *R* remain fixed. Therefore, a larger frame size allows a higher throughput. When *C* is linear with *S* (which it is partially since the slot allocation table is linear with *S*), the available bandwidth still grows with *S* when *R* is fixed.

Note, that in this and all further analysis, we neglect the influence of having the base station leaving a slot free for connection setup every now and then.

5.1.1 Real-time throughput

Real-time connections reserve bandwidth that is sufficient for their average load. For frame size *S* and real-time load of G_{RT}, the real-time throughput η_{RT} is

$$\eta_{RT}\left(G_{RT}\right) = \begin{cases} G_{RT} & G_{RT} \le \frac{D}{S} \\ \frac{D}{S} & G_{RT} \ge \frac{D}{S} \end{cases} \quad \textbf{Eq. 2}$$

When we have *N* real-time connections each generating a load *p*, then, with assuming fairness, the throughput per real-time connection η_{cRT} is given by

$$\eta_{cRT}(p) = \begin{cases} p & Np \le \frac{D}{S} \\ \frac{D}{S \cdot N} & Np \ge \frac{D}{S} \end{cases} \quad \textbf{Eq. 3}$$

5.1.2 Non-real-time throughput

Non-real-time connections do not reserve bandwidth. Their throughput depends on the load of all other connections. The activity of the real-time connections determines the amount of bandwidth that remains and can be shared by the non-real-time connections. For frame size *S*, *D* data slots, aggregate real-time load G_{RT} and aggregate non-real-time load G_{NRT}, the non-real-time throughput η_{NRT} can be described as

$$\eta_{NRT}\left(G_{NRT}, G_{RT}\right) = \begin{cases} G_{NRT} & G_{NRT} + G_{RT} \le \frac{D}{S} \\ \frac{D}{S} - G_{RT} & \frac{D}{S} - G_{NRT} \le G_{RT} \le \frac{D}{S} \\ 0 & G_{RT} \ge \frac{D}{S} \end{cases} \quad \textbf{Eq. 4}$$

When we have *N* real-time connections each generating load *p* and *M* non-real-time connections each generating load *q*, then the throughput per non-real-time connection η_{cNRT} is

$$\eta_{cNRT}(p, N, q, M) = \begin{cases} p & Np + Mq \le \frac{D}{S} \\ \frac{D - NpS}{S \cdot M} & Np + Mq \ge \frac{D}{S} \wedge Np < \frac{D}{S} \\ 0 & Np \ge \frac{D}{S} \end{cases} \quad \textbf{Eq. 5}$$

Note that the non-real-time throughput depends on the average real-time load rather than on the number of reserved slots.

5.2 Latency

5.2.1 Real-time latency

First we will discuss the real-time latency under low loads, i.e. the total number of packets generated in a frame by all connections does not exceed the number of slots in a frame.

As a packet can be generated in any slot in a frame, it takes half a frame on average until the next RTS slots in which the real-time connection can request bandwidth for the next frame. Then the connection must wait S_{RC} slot times to until the next CTS period that contains the slot allocation for that frame. Receiving the CTS message takes *C* slot times. Since all the connections have a low load, enough slots will be obtained in the next frame. When there are *N* active real-time connections each generating a load of *p*, the frame size is *S*, then the average number of real-time packets that is created in a frame is described as *NpS*. Since the scheduler uses a round-robin scheduling scheme, the average data slot number allocated to a real-time packet is the same for each connection and can be formulated as *NpS*/2. The total average real-time access latency AL_{RT} under low loads is described as

$$AL_{RT} = \frac{S}{2} + S_{RC} + C + \frac{NpS}{2} \quad \textbf{Eq. 6}$$

From Equation 6 it is obvious that the access latency can be reduced by decreasing the frame size and/or the time between the RTS slots and the CTS slots. However, decreasing S_{RC} is limited since during this time the slot allocations are calculated.

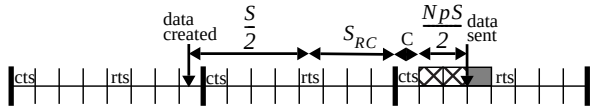

Figure 3. Real-time latency components

Under high loads, it is possible that more real-time slots are created than the number of slots that can be obtained in the next frame. In that case real-time packets will remain in the queue. Therefore, queuing delay must be taken into account. We used the queuing analysis described in [9] and [12].

The real-time packet arrival rate λ in packets per slot is given by *Np*. The real-time packet consumption rate μ in packets per slot is given by the number of reserved slots in a frame and the number of remaining data slots equally divided among *N* real-time and *M* non-real-time connections and is expressed as

$$\mu = \frac{N\lceil pS\rceil}{S} + \frac{N}{(N+M)}\frac{D - N\lceil pS\rceil}{S} = \frac{ND + NM\lceil pS\rceil}{S(N+M)} \quad \textbf{Eq. 7}$$

The average number of real-time packets waiting in the real-time queues is then given by

$$\frac{\lambda}{\mu - \lambda} = \frac{(N+M)pS}{D + M\lceil pS\rceil - (N+M)pS} \quad \textbf{Eq. 8}$$

Since the consumption rate is μ, we can now compute the additional real-time queuing delay, i.e. the time in slots that the real-time packets remain in the queue until transmission starts. This is expressed as the number of real-time packets in the queue divided by the consumption rate of real-time packets. This real-time queuing latency QL_{RT} then is given by

$$QL_{RT} = \frac{\lambda}{\mu - \lambda} \cdot \frac{1}{\mu} = \frac{(N+M)\,pS}{D + M\lceil pS \rceil - (N+M)\,pS} \cdot \frac{S\,(N+M)}{ND + NM\lceil pS \rceil} \quad \textbf{Eq. 9}$$

This expression shows that QL_{RT} will grow infinitely when the real-time load saturates the channel capacity.

Adding Equation 6 and Equation 9, we obtain the overall average real-time transmission latency TL_{RT}

$$TL_{RT} = AL_{RT} + QL_{RT} \quad \textbf{Eq. 10}$$

5.2.2 Non-real-time latency

Again, we consider the low load latency first. When a non-real-time connection creates a packet, it has to wait half a frame on average for the next RTS slots, another S_{RC} slots to the next CTS slots and C slots to receive the CTS packet. Then first NpS real-time packets are transmitted followed by half of MqS non-real-time packets.

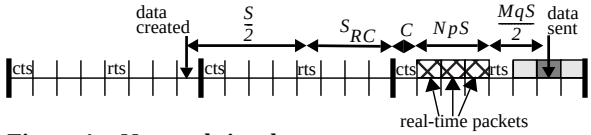

Figure 4. Non-real-time latency components

Analogous to the real-time access latency, we can describe the non-real-time access latency AL_{NRT} as

$$AL_{NRT} = \frac{S}{2} + S_{RC} + C + NpS + \frac{MqS}{2} \quad \textbf{Eq. 11}$$

Again this is not sufficient under higher loads. Under high loads queuing delay is needed in the analysis. The non-real-time packet arrival rate λ is given by Mq and the non-real-time packet consumption rate μ in packets per slot is given by the number of available slots after allocating the average number of real-time slots NpS and is given by $\mu = \frac{D - NpS}{S}$.

The average number of non-real-time packets in the non-real-time queues therefore can be expressed as

$$\frac{\lambda}{\mu - \lambda} = \frac{SMq}{D - NpS - MqS} \quad \textbf{Eq. 12}$$

The time required to transfer these non-real-time packets is

$$QL_{NRT} = \frac{\lambda}{\mu - \lambda} \cdot \frac{1}{\mu} = \frac{SMq}{D - NpS - MqS} \cdot \frac{S}{D - NpS} \quad \textbf{Eq. 13}$$

This expression shows that the queuing latency will grow be large when either the real-time load or the sum of the real-time and non-real-time loads approaches the data capacity.

Adding Equation 13 to Equation 11, gives the average non-real-time transmission latency TL_{NRT} as

$$TL_{NRT} = AL_{NRT} + QL_{NRT} \quad \textbf{Eq. 14}$$

6 Protocol simulation

In this section we describe the simulation of the protocol and the results we obtained. We used a MAC protocol simulator that can simulate the R-TDMA MAC protocol for multiple mobile stations and multiple base stations. Its basic transmission unit is a message of variable size. Signal effects such as multi-path, fading, noise are not included in the simulator. Collisions are simulated by corrupting packet contents. We assume that corruptions can be identified and corrupted packets are discarded by the receiving station.

6.1 Frame implementation

In our simulations we used frames of 20 slots with a single CTS and a single RTS slot, giving 18 data slots per frame. The first slot in the frame is the CTS slot and the 11th slot is used for the RTS messages. Packets are 53*8=424 bits large and preceded by 2 synchronisation bytes. Therefore, the slot size is 440 bit times. Note that, in this case, the bandwidth reduction caused by introducing synchronisation bytes is

$$1 - \frac{424}{424 + 16} \times \frac{18}{20} = 0.13 \quad \textbf{Eq. 15}$$

A minislot is 24 bits long and contains a single information byte preceded by 2 synchronisation bytes. A slot size of 440 gives room for 18 minislots, but in the simulation we used 16 connections. Mobile stations can inform the base station that they have up to 255 ATM cells to transmit.

6.2 Configurations

We simulated the protocol in various simple configurations. In each configuration there was a single base station. The number of mobile stations (N=M) and the generated Poisson traffic loads p=q, were varied. Each mobile station has a real-time and a non-real-time connection. During connection set-up each of the real-time connections requests the base station to reserve $\lceil Sp \rceil$ slots, which is the average number of slots that is generated per frame by a real-time connection. The connection is refused when the base station cannot reserve this number of slots. No attempts for reserving fewer slots are made.

7 Results

The results of the simulations are given in Figure 5 through Figure 8. We have not plotted the analysis graphs in these figures because they are so similar that it leads to confusion if we plotted the analysis and the simulation results in the same figure. Due to space limitation we cannot show the analysis graphs in separate figures. We will briefly discuss the figures obtained by simulation.

7.1 Throughput

From Figure 5 we conclude that the bandwidth is used efficiently. The throughput is identical to the load as long as the network is not saturated. When the load is higher than the available network bandwidth, all bandwidth is used.

Figure 6 shows that non-real-time throughput decreases when the aggregated load exceeds the network capacity. Due to bandwidth reservation real-time connections can send data at the cost of non-real-time data. The figure also shows that the real-time throughput remains stable after network saturation.

7.2 Latency

From Figure 7 and Figure 8, we conclude that the average latency is very limited. For real-time traffic the average latency exceeds two frames (40 slots) only when the total real-time load approaches the network capacity very closely. The non-real-time latency behaves roughly the same, exceeding two frames when the aggregated load of real-time and non-real-time traffic approached the network capacity.

8 Conclusions

In R-TDMA the bandwidth is optimally used. Real-time connections have a guaranteed throughput with limited latency. Non-real-time connections share the bandwidth in a best effort way. All the bandwidth that is not used by real-time connections can be used by the non-real-time connections. When the network bandwidth is not saturated, non-real-time latency remains low.

The frame size S can be increased to increase the throughput. Decreasing the S leads to lower latency for real-time traffic. A fixed part of the latency is the time between the RTS and the CTS period. The faster the scheduler, the smaller this time can be, reducing the total latency. Optimally, this time should be zero, giving the scheduler virtually 0 time to schedule the slots

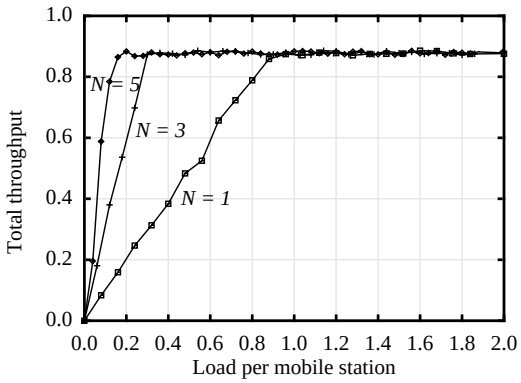

Figure 5. Simulated throughput as a function of the load of 1, 2 and 5 mobile stations

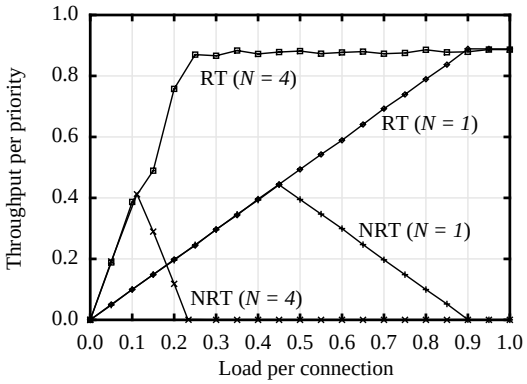

Figure 6. Simulated throughput per priority as a function of connection load with 1 and 4 mobile stations

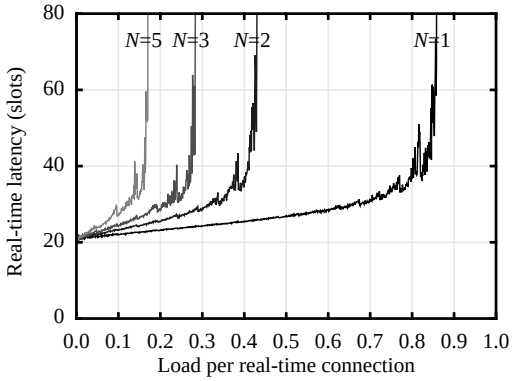

Figure 7. Simulated real-time latency as a function of load for 1, 2, 3 & 5 real-time connections

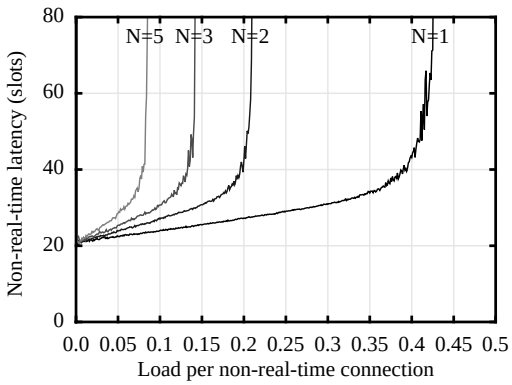

Figure 8. Simulated non-real-time latency as a function of load for 1, 2, 3 & 5 non-real-time connections

for the next frame. Perhaps a scheduler can be found that schedules the slot during the transmission of the CTS header.

The slot scheduler makes our multiple access protocol very flexible. Besides real-time and non-real-time connections, the scheduler can also take more priorities into account. In our system we use a round robin scheduler with two priorities, but other scheduling mechanisms can be used as well. However, the more complex the scheduling algorithm, the more computation time is needed and the longer the RTS - CTS time will become, thus increasing the average latency.

Finally, the analysis we give is appropriate for our purposes. The simulation results confirm that the analysis is accurate.

9 References

[1] Linnenbank, G.R.J., et al., Request-TDMA: A Multiple-Access Protocol for Wireless Multimedia Networks, Proceedings IEEE 3rd Symposium on Communications and Vehicular Technology in the Benelux, pp. 20-27.

[2] Abramson, N., Development of the ALOHANET, IEEE Trans. on Information Theory, vol. IT-31, pp. 119-123, March 1995.

[3] Mukumoto, K., Fukuda, A., Idle-Signal Multiple-Access (ISMA) Scheme for Terrestrial Packet Radio Networks, Electronics and Communications in Japan, Vol. 64-B, No. 10, pp. 66 - 74, 1981.

[4] Murase, A., Imamura, K., Idle-Signal Casting Multiple Access with Collision Detection (ICMA-CD) for Land Mobile Radio, IEEE Trans. on Vehicular Technology, Vol. VT-36, No. 2, pp. 45 - 50.

[5] Murase, A., Imamura, K., Idle-Signal Casting Multiple Access with Data Slot Reservation (ICMA-DR) for Packet Radio Communications, IEEE Trans. on Vehicular Technology, Vol. 38, No. 2, pp. 45 - 50.

[6] Wu, G., et. al., Performance Evaluation of Reserved Idle Signal Multiple-Access Scheme for Wireless Communication Networks, IEEE Trans. on Vehicular Technology, Vol. 43, No. 3, pp. 653 - 658.

[7] Goodman, D.J., Wei, S.X., Efficiency of Packet Reservation Multiple Access, IEEE Trans. on Vehicular Technology, Vol. 49, No. 1, pp. 170 - 176.

[8] Wen, J.-H., Wang, J.-W., Throughput Analysis of Packet Reservation Multiple Access Protocol for Wireless Communications, Proceedings PIMRC'94/WCN94, pp. 1242 - 1246, Sept. 1994.

[9] Tanenbaum, A.S., Computer Networks, Second Edition, Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[10] Lee, W.C.Y., Overview of Cellular CDMA, IEEE Trans. on Vehicular Technology, Vol. 40, No. 2, pp. 291 - 302.

[11] Demers, A., et. al., A Nano-Cellular Local Area Network Using Near-Field RF Coupling, Virginia Tech's Fourth Symp. on Wireless Personal Communications, 1994.

[12] Hammond, J.L., O'Reilly, P.J.P., Performance Analysis of Local Area Networks, Addison-Wesley Publishing Company, pp. 125 - 128, 1986.

[13] Raychaudhuri, D., Wilson, N.D., ATM-Based Transport Architecture for Multiservices Wireless Personal Communication Networks, IEEE Journal on Selected Areas in Communications, Vol. 12, No. 8, pp. 1401 - 1414.