E^2MaC: an Energy Efficient MAC protocol for multimedia traffic

Paul J.M. Havinga, Gerard J.M. Smit

University of Twente
department of Computer Science
the Netherlands
{havinga, smit}@cs.utwente.nl

Abstract

Energy efficiency is an important issue for mobile computers since they must rely on their batteries. We present a novel MAC protocol that achieves a good energy efficiency of wireless interface of the mobile and provides support for diverse traffic types and QoS. The scheduler of the base station is responsible to provide the required QoS to connections on the wireless link and to minimise the amount of energy spend by the *mobile*. The main principles of the E^2MaC protocol are to avoid unsuccessful actions, minimise the number of transitions, and synchronise the mobile and the base-station. We will show that considerable amounts of energy can be saved using these principles. In the protocol the actions of the mobile are minimised. The base-station with plenty of energy performs actions in courtesy of the mobile. We have paid much attention in reducing the cost of a mobile for just being connected. The protocol is able to provide near-optimal energy efficiency (i.e. energy is only spent for the actual transfer) for a mobile within the constraints of the QoS of all connections in a cell, and only requires a small overhead.

1 Introduction

Portable computers like PDAs and laptops that use wireless communication to interact with the environment rely on their limited battery energy for their operation. Energy consumption is becoming the limiting factor in the amount of functionality that can be placed in these devices. More extensive and continuous use of network services will only aggravate this problem. However, even today, research is still focussed on performance and (low power) circuit design. We believe that it is more effective to save energy by a carefully designed architecture of the mobile and wireless communication protocols that consider judicious use of the available energy. Energy reduction should be considered in the whole system of the mobile and through all layers of the protocol stack, including the application layer [5]. In this paper we address the issue of energy efficiency in medium access control (MAC) protocols for wireless multimedia networks. Access protocols typically address network performance metrics such as throughput, efficiency, fairness and packet delay. This paper addresses the additional goal of efficient energy usage of the mobiles. The assumption is that mobiles will always have limited power, whereas the wired base-stations will have virtually unlimited power.

It is expected that the new generation of wireless networks will carry diverse types of multimedia traffic. In these systems Quality of Service (QoS) is used to control and manage the available resources for multimedia traffic [2]. These communications will include video, audio, images, and bulk data transfer. In multimedia traffic important parameters are jitter, delay, reliability, and throughput [6]. To provide the various service requirements of multimedia traffic (like CBR, real time and non-real time VBR, and ABR services), the access protocol must be able to provide the required QoS on demand.

We consider an ATM based infrastructure network where a base-station co-ordinates access to one or more channels for mobiles in its cell. The channels can be individual frequencies in

FDMA, time slots in TDMA, or orthogonal codes or hopping patterns in case of spread-spectrum.

The paper first presents in section 2 the main sources of energy consumption on wireless interfaces, which provide us the main principles of energy efficient MAC design. Then we give in section 3 an overview of a new MAC protocol E^2MaC whose design is driven by energy consumption, diverse traffic type support, and QoS support considerations. Section 4 then provides an overview of the performance of the E^2MaC protocol. Related work is presented in section 5, and we will finish with some conclusions.

2 Main principles

In the context as described in the introduction the objective of an energy efficient MAC protocol design is to maximise the performance while minimising the energy consumption of the *mobile*. These requirements often conflict, and a trade-off has to be made. For example, in the IEEE 802.11 standard [8] the base-station can collect data destined for a mobile and periodically broadcasts a beacon about its buffered data. When the mobile wakes up, it informs the base that it is ready to receive. Although this approach saves energy, it results in significant delay and jitter.

2.1 Sources of unessential energy consumption

The focus of this work is on minimising the energy consumption of a mobile and in particular the wireless interface, the transceiver. Typically, the transceiver can be in five modes; in order of increasing energy consumption, these are off, sleep, idle, receive, and transmit. In many protocols the overhead involved to receive or transmit an amount of data can be very large, and may depend on the load of the network.

The unessential energy consumption can be caused by several reasons.

First of all, most applications have no very high demanding traffic needs, and the transceiver is *idling* most of the time. Measurements show that on typical applications like a web-browser or e-mail, the energy consumed when the interface is on and idle is more than the cost of actually receiving packets [11][14]. Secondly, the typical *inactivity threshold*, which is the time before a transceiver will go in the off or standby state after a period of no activity, causes the receiver to be needless in an energy consuming mode for a significant time. Third, in a typical wireless broadcast environment, the receiver has to be powered on at all times to be able to *receive messages* from the base station, resulting in significant energy consumption. The receiver subsystem typically receives all packets and forwards only the packets destined for this mobile. Even in a scheme in which the base transmits a traffic schedule to a mobile, the mobile has to receive the traffic control information regularly to check for waiting downlink traffic. When the mobile is not synchronised with the base-station, then it might have to wait some time before it receives the traffic control. Fourthly, significant time and energy is further spent by the mobile in switching from transmit to receive modes, and vice-versa. The *turnaround time* between these modes typically takes between 6 to 30 microseconds. The transition from sleep to transmit or receive generally takes even more time (250 μs). A protocol that assigns the channel per slot will cause significant overhead due to turnaround. Fifthly, another main cause of energy consumption is due to *collisions* that occur mainly at high load situations. Collision causes all parties involved to retransmit the original data, leading to unnecessary energy consumption and also to possible unbounded delays. Finally, the high *error rate* that is typical for wireless links might cause another source of energy consumption. When the data is not correctly received the energy that was needed is spoiled. On the data link layer level error correction is generally used to reduce the impact of errors on the wireless link. The residual errors occur as burst errors covering a period of up to a few hundred milliseconds. To overcome these errors retransmission techniques or error correction techniques are used. Energy is also consumed for the calculation and transfer of redundant data packets and an error detection code (e.g. a CRC).

We define *energy efficiency* as the quotient between the intrinsic amount of energy needed to transfer a certain quantity of data and the actually needed amount of energy (thus including all overheads). We will use this metric to quantify how well a MAC protocol behaves with respect to its energy consumption.

2.2 Main principles of energy efficient MAC design

The observations above are just some of the possible sources of unessential energy consumption related to the medium access control protocol. We have no intention to provide a complete list. We can, however, deduce the following three main principles that can be used to design a MAC protocol that is energy efficient for the mobile.

- *Avoid unsuccessful actions of the transceiver.*
  Two main topics cause unsuccessful actions: collisions and errors.
  Every time a collision occurs energy is spoiled because the same transfer has to be tried again after a backoff period. A protocol that does not suffer from collisions can have a good throughput even under high load conditions. These protocols generally also have good energy consumption characteristics. However, if it requires the receiver to be turned on for long periods of time, the advantage will be diminished.
  A protocol, in which a base-station broadcasts a traffic control for all mobiles in range with information about when a mobile is allowed to transmit or is supposed to receive data, reduces the occurrence of collisions significant. Collisions can only occur when new requests have to be made. New requests can be made per packet in a communication stream, per application of a mobile, or even per mobile. The trade-off between efficient use of resources and QoS determines the size to which a request applies. Note that this might spoil bandwidth (but not energy) when slots are reserved for a request, but not actually used all the time. In such a reservation mechanism, energy consumption is further reduced because there is no need for a handshake to acknowledge the transfer.
  Errors on the wireless link can be overcome by mechanisms like retransmissions or error correcting codes. Both mechanisms induce extra energy consumption. A strategy to reduce the effect of errors is to avoid traffic during periods of bad error conditions. This however is not always possible for all traffic types as it influences the QoS. A solution might be to use some error correcting codes for this kind of traffic [6].
- *Minimise the number of transitions*
  Scheduling traffic into bursts in which a mobile can continuously transmit or receive data, - possibly even bundled for different applications - can reduce the number of transitions. Notice however that this trade-offs with QoS parameters like delay and jitter. When the traffic is continuous and can be scheduled for a longer period ahead, then the mobile does not even have to listen to the traffic control since it knows when it can expect data or may transmit. The number of transitions needed can also be reduced by collecting multiple requests of multiple applications on a mobile, and piggy-backing new requests on current data streams.
- *Synchronise the mobile and the base-station.*
  Synchronisation is beneficial for both uplink as downlink traffic. When the base-station and mobile are synchronised in time, the mobile can go in standby or off mode, and wake up just in time to communicate with the base-station. The energy consumption needed for downlink traffic can be reduced when the time that the receiver has to be on - just to listen whether the base-station has some data for the mobile - can be minimised. The premise is that the base has plenty of energy and can broadcast its beacon frequently. The application of a mobile with the least tolerable delay determines the frequency of which a mobile needs to turn its receiver on.

There are many ways in which these principles can be implemented. We will consider an environment suitable for multimedia applications in which the MAC protocol also has other requirements like provisions for QoS of real-time traffic, and to provide a high throughput for bulk data.

3 The E^2MaC protocol

3.1 Assumptions and overview

The structure of the wireless ATM network is composed of base-stations that each handles a single radio cell covering mobile stations. We consider an office environment in which the cells are small and have the size of one or several rooms. This not only saves energy because the transmitters can be low powered, it also provides a high aggregate bandwidth since it needs to be shared with only several mobiles. The backbone of the base-stations is a wired ATM network. Uplink and downlink traffic is Time Division Multiplexed (TDM).

In the protocol the base station controls the access of mobiles to the transmission slots. The premise is that the base station has virtually no processing and energy limitations, and will perform actions in courtesy of the mobile. The protocol uses fixed-length frames of multiple slots. The base-station and mobile are completely synchronised, which allows the mobile to power-on precisely when needed. A frame is divided in slots that can have three basic types: *traffic control*, *shared*, and *data*. The base-station controls the traffic for all mobiles in range of the cell and broadcasts the schedule in the traffic control slot. The data slots are used to send the actual data. New reservations or updates of current reservations for the next frames can be piggy-backed onto a data-slot. Only new connections may encounter collisions, the data slots are collision-less. All remaining slots are used half by the base-station to transmit *downlink calls*, and the mobile can use the tail to make *connection requests*. In this way mobiles that wake-up or enter the cell can synchronise with the base-station quickly. The downlink traffic information allows the mobile to power-on the receiver for a short time, just enough to scan if there is downlink traffic waiting at the base-station.

Each mobile can have multiple unidirectional connections with different Quality of Service. They can have the following service classes: *deterministic* for (hard) real-time applications, *statistical* for soft real-time applications, and *best effort* for everything else without guarantees. More formally, the ATM Forum traffic management group has defined service classes by: constant bit rate (CBR), real time VBR, non-real time VBR, unspecified bit rate (UBR), and available bit rate (ABR). The required resources at both the base-station and the mobile are reserved at connection setup time.

3.2 Connection setup

When a new connection has to be made the service class and the required QoS of the wireless connection is passed to the base-station. The required QoS is determined by classical parameters like *throughput*, *reliability*, *jitter* and *delay*. The quantitative QoS parameters used in the E^2MaC protocol are:

- d: the number of data slots required in a frame.
- j: jitter, the allowable variation in delay in a frame.
- f: the frequency that a connection will use slots in a frame.

It is the task of the system to translate the original QoS parameters into these MAC level parameters. It thereby can also incorporate the expected error rate of the wireless link. For time-critical traffic it might use an error correcting code [6]. The base-station contains the central scheduler for the traffic of all mobiles in its range. The mobiles send requests for new connections or update information to the base-station. The base-station determines according to the current traffic in the cell whether it can allow the new connection. When the request is granted, the base-station assigns a connection-ID (of one byte) to the new connection and notifies this ID to the mobile in a dedicated field in the traffic control slot. The mobile will then create a queue for that connection. When data arrives in the queue that has to be sent to the base-station, it can reserve slots in the next frame. When the connection has continuous traffic, slots can be reserved in the following frames as well. Non-real time data can be requested with a best-effort indication. The scheduler of the base-station can then assign unreserved slots to these connections to optimise the throughput on the wireless link.

The base-station initiates *downlink traffic* with a connection request to the mobile. This request will be forwarded to the system of the mobile. When the system or application grants the downlink connection, it initiates this new connection with a connection request to the base. In this way the base-station's functionality is kept simple since it only needs to handle uplink connection requests.

3.3 Frame structure

The frame is divided in slots that can have three basic types: *traffic control*, *shared*, and *data*. Only the traffic control type has a fixed position at the start of the frame. The number of slots needed for traffic control depends on the size of the frame and is thus implementation dependent. Typically one slot is sufficient. The other types are dynamic, have no fixed size and can be anywhere in the rest of the frame. The base-station controls the traffic for all mobiles in range of the cell and broadcasts the schedule in the traffic control slot. Only new connections may encounter collisions, the data slots are collision-less.

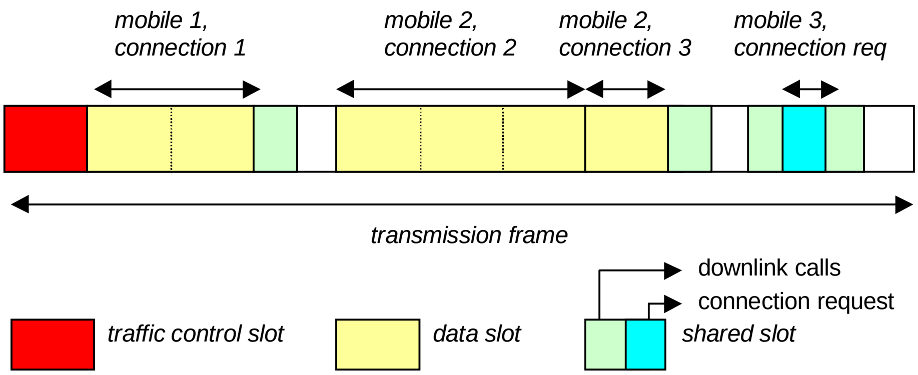

Figure 1: Example of a transmission frame.

The *traffic control* slot contains information about the type and direction of each slot in the current frame, and the connection-ID that may use the slot. Since a corrupted traffic control can influence the QoS of all connections, these slots are protected with an error correction mechanism. The traffic control contains 1) the schedule of the slots for the connections assigned in the frame (connection-ID, slot number, length), 2) the downlink queue status for a connection (connection-ID, number of messages in the queue), and 3) two fields used for connection set-up for uplink and downlink connections (mobile-ID, connection-ID). This data-structure allows for 15 connections to be registered into one ATM cell when we assume that a frame has maximal 256 slots and a mobile-ID is 16 bits. These numbers seems suitable in a micro-cellular network in which the cells have the size of one or several rooms.

The *shared slots* are used by the base station to transmit *downlink calls*, and by the mobile to make *connection requests*. The first half of this slot is used by the base-station, and mobiles can use the second half. All slots of a frame that are not used for a connection are shared slots that can be utilised by a mobile to request a new connection. Connection requests allow mobiles that have entered the cell to register at the base-station. In it the mobile indicates its identity and the requested service class and QoS. The *downlink calls* part of the shared slot indicates for which mobiles it has downlink traffic and contains information about the location of the next traffic control. In this way it allows mobiles that wake-up or enter the cell to synchronise with the base-station quickly since it does not have to wait for the traffic control slot. The downlink traffic information allows the mobile to power-on the receiver for a short time, just enough to scan if there is downlink traffic waiting at the base-station. When the base has downlink traffic in its queue, then the mobile can find the exact connection information in the traffic control.

Both the mobile and the base-station use the *data slots* to send the actual data. New reservations or updates of current reservations for the next frames can be piggy-backed onto a data-slot. Data slots have a control field of two bytes and a data field with the size of typically one ATM cell. The control contains the ID of the connection, the type of the data, and a field for additional data.

The base-station initiates downlink traffic. First it broadcasts that it has downlink traffic for a mobile in each connection request slot. When the mobile acknowledges the reception of this message, the base-station uses a single data-slot to request the connection. This slot contains the service class and QoS requirements for the new connection and is forwarded to the mobile's system.

3.4 Slot-scheduler

The base-station schedules all reservation requests according to the QoS requirements and tries to minimise the energy consumption of the mobiles by minimising the number of transitions the mobile has to make. It therefor schedules the traffic of a mobile such that all downlink and uplink connections are grouped as much as possible within the limitations imposed by the QoS of the connections. If the QoS of a connection allows jitter (like non-real-time bulk data transfer), then the scheduler has more flexibility to group the traffic. When a mobile requests a connection and indicates that it does not allow any jitter, then the scheduler is forced to assign the same data slots in each frame for that connection. Only at connection setup, the scheduler is free to assign the slots. In this way the mobile can minimise its energy consumption, since it knows precisely when it is allowed to transmit data, or when it can expect data. It does not even need to listen to the traffic control. Only the drift of the clock might force the mobile once in a while to synchronise with the base-station.

Operation

The mobile initially requests with a connection request in a shared slot a certain service class and QoS for a connection. If no collision occurred with other mobiles, the base-station receives the request and determines whether it can fulfil the request. If so, it initially assigns one data slot per frame for that connection. In the traffic control it indicates using the ID of the mobile that it has acknowledged the connection and assigns a connection-ID. If the connection becomes active, the mobile can ask for more slots, up to the requested maximum. Updates to the current reservation can be piggy-backed in each data-slot mobiles. Possible updates are:

- **request repeat (d)**. This requests *d* slots in the next frame. The number *d* is an indication of the amount of data the connection has, or expects to have, in the queue for the next frame. If the scheduler receives no update, it reserves the current number of slots in the next frame.
- **release connection**. To release the current connection and free all reservations.
- **sleep (s)**. This informs the base-station that the mobile will sleep for *s* frames. This allows the scheduler to re-assign the slots for other connections during *s* frames. A mobile can be forced to send a keep-alive message by indicating a sleep (*s*). In this way the base-station will know when the mobile has left the cell or is turned off, and can free the reserved resources.

The data-slot can also be utilised as a control slot for the base-station to setup new connections or acknowledgement from the mobile that is ready to receive downlink traffic for another connection. The data-field might be needed to transfer the required parameters.

- **new connection**. The current connection is used to request a new connection by the base-station as well as by the mobile. In this way the mobile does not have to compete with other mobiles to access a connection request slot which reduces the occurrence of collisions. The data field is used to indicate the required service class and QoS of the new connection.

- **activate downlink connection (id)**. This allows a mobile to inform the base-station that it is ready to receive downlink traffic for a certain connection (*id*). This message might be needed when the mobile wakes up and notices that the base station has traffic waiting in its queue.

This mechanism allows a mobile when it enters a cell to register itself to the base-station and setup a dedicated bi-directional *control connection*. This connection is collision-less and can be used by the mobile to request new connections and acknowledge downlink traffic. The base-station can use this connection to request new downlink connections.

Adaptation to error conditions

The slot-scheduler makes the MAC protocol very flexible. It can not only use a different scheduling policy when needed, but it can also adapt its scheduling policy to the current conditions in the cell dynamically. The error-rate of a wireless channel is a main cause of changing conditions.

The base-station will keep track of the current error condition of a wireless connection by counting the number of corrupted slots using the CRC of each connection. The scheduler can try to avoid periods of bad error conditions by not scheduling non-time critical traffic during these periods. Note that the size of an error-burst may be up to 100 milliseconds, which will cause on a 2 Mbit/s wireless link that more than 400 ATM sized slots can be affected. Hard-real time traffic remains scheduled, although it has a higher chance of being corrupted. The base station uses this traffic to probe whether the channel is good again. When the mobile has no real-time connections, it will use a statistical backoff period. Note that the error conditions perceived by each mobile in a cell can be different. Since the base station keeps track of the error conditions per connection (and thus also per mobile), it can give other mobiles that have better conditions more bandwidth. This can lead to a throughput that may even exceed the average rate on the channel, due to the introduced dependence between admitted connections and channel quality [4].

3.5 Future design issues

As discussed in section 2.1, a main source of unessential energy consumption is due to the costs of just being connected to the network. In E^2MaC we have tried to minimise this overhead, but it still requires the receiver to be switched on from time to time, just to listen whether the base-station has some messages waiting. Therefor we consider using a very low power receiver for the signalling only. This receiver will be used to wake-up a mobile by the base-station. It uses the same synchronisation mechanism between mobile and base-station, but now uses a simple, low performance, low power receiver.

A further extension might be to use a dedicated *bi-directional* signalling network that could be used for the MAC protocol only and operates in parallel with the actual data-stream with another transceiver on the same interface. This data-stream transceiver has more bandwidth and consumes more energy, but will be turned on only when there is actually data to be transmitted, and is not used for ‘useless’ signalling.

4 Performance of the E^2MaC protocol

The E^2MaC protocol is designed to provide QoS to various service classes with a low energy consumption of the mobile. The base-station with plenty of energy performs actions in courtesy of the mobile. In the protocol the actions of the mobile are minimised. In the remainder we will thus only consider the energy efficiency of the mobile, and not of the base-station. The main restriction comes from the required QoS of the applications on the mobiles.

The achieved energy efficiency depends on the implementation of the scheduler, the error rate, and also on the applications. The application, and also the user, must provide proper QoS requirements to the system. The E^2MaC protocol then offers the tools to the system to reduce the energy consumption that is needed for the wireless interface.

4.1 Energy efficiency principles applied on the E^2MaC protocol

All three main principles are used in the design of the protocol. Note that some principles interact, for example the synchronisation between base-station and mobile is not only used to power the transceiver just in time, but also to avoid collisions.

A. Synchronise the mobile and the base-station

When a mobile has a connection then it is fully synchronised with the base-station and can - when it is idle - enter a minimal energy consuming mode, just enough to update its clock. The synchronisation is used for uplink and downlink connections. When the mobile wants to send data it first has to receive the traffic control to find the assigned slots to use in the frame. Since the mobile and base-station are synchronised in time, the mobile can power up the receiver just in time. Once a connection has been setup and the mobile has no data to send, it can simply tell the base-station that it will sleep for some time. The base-station will then release the slot and use it for other connections until the sleep period is over. When the mobile does not use the slot, then the base will let the connection sleep again for the same period. This mechanism allows the mobile to sleep for a long period, and still is certain that it can acquire a slot within a bounded time. In this way the mobile reserves a slot periodically in a frame, and the bandwidth spend depends on the tolerable delay.

A mobile that just has to listen if there is *downlink traffic* waiting at the base-station spoils much energy. The E^2MaC protocol therefor tries to minimise the amount of energy needed by broadcasting such information in any free slot (the ‘downlink calls’ in shared slots). This allows a mobile to sleep when a connection is not used for some time. Synchronisation in time between mobile and base-station can occur also fast since the location of the traffic control is sent in each shared slot as well. The costs of just being connected is determined by the application of the mobile with the least tolerable delay or the drift in clocks between mobile and base-station.

In reservation schemes like the E^2MaC protocol there is always an inevitable overhead due to the *traffic control*. In the E^2MaC protocol the required overhead for a mobile to receive the traffic control can be reduced when the traffic can be scheduled in advance. The mobile can request a static connection with no jitter in the frame. In this way the mobile has near-optimum energy efficiency since it does not need to listen to the traffic control: it knows when to expect the slot(s) assigned to the connection. This however may be a bit egoistic since it reduces the freedom of the slot-scheduler. Only when the load on the wireless channel is not high, then the scheduler is able to assign such connections. When the load is too high, the scheduler cannot fulfil all wishes. A strategy of a mobile can be to ask a best-fit connection with no jitter. The scheduler can then decide depending on the current load of the cell to honour the request or not.

B. Avoid unsuccessful actions

Unsuccessful transfers are minimised because the chance of a collision is small and the base station tries to avoid periods of bad error conditions. The chance of a *collision* in the E^2MaC protocol is small since 1) it can only occur when a mobile enters the cell and requests a connection, and 2) because many slots (i.e. all not used slots in a frame) can be used to request the connection. When a mobile has a connection, then it has reserved slots, and no collisions occur. In a next paragraph (4.2) we will show the advantages of a reservation scheme compared with a protocol that suffers from collisions.

How profitable it is to *avoid periods of bad error conditions* for non-real time traffic depends on factors like the typical size of an error burst and how fast the slot-scheduler can react (which also depends on the frame-length). Energy will be saved in any case (and will be maximal the energy that otherwise would have been spoiled during the bad-error period), but the consequence for the throughput in a cell is more complicated, because other - error free - connections will use the bandwidth instead. As already stated, this can lead to a throughput that may even exceed the average rate on the channel.

C. Minimise the number of transitions

The number of *transitions* between transmitting, receiving, idle, sleep, and off is also small. The slot-scheduler tries to group the transmissions and receptions of a mobile as much as possible according to the service classes, QoS and current load.

The number of transitions is further reduced since control messages (like connection updates or new requests) to the base since can be piggy-backed in the control header of data-slots. Such a mechanism of bundling of transmission and reception can give a large reduction in energy consumption when compared with a scheme that requires two transitions per ATM cell (i.e. change the mode from idle to transmission, and back to idle). When we assume a wireless interface that has a throughput of 2 Mbit/s, then a transition time from sleep to idle of 250 µs already takes virtually 62.5 bytes. The overhead is thus more than the transmission of the actual cell. The two-byte control field of a data slot reduces the throughput, but this is only maximal 4% when the size of a slot is one ATM cell. However, when the average size of a burst of traffic for one connection in a frame is large, then it seems that this energy and bandwidth is spoiled, because it is only actually used once per burst in a frame. An alternative would be send this control information in (part of) a dedicated slot, which would save the two bytes overhead per data slot. This however would cause the transceiver to make two additional transitions, which would induce needless energy consumption as well. When a transition needs the time equivalent to transmitting 8 bytes of data, which is a realistic assumption, then the minimal size of a control slot would be virtually 18 bytes. The point where it might be profitable to use a dedicated control slot is thus expected to be with average bursts-sizes of greater than nine slots when the control slot is two bytes large. When the control slot has the size of one ATM cell (which is much simpler to implement and more energy efficient since the frame-length can be a multiple of a ATM cell) then the average burst-size must be even larger than 34.5. We expect that due to latency and delay requirements, and implementation restrictions the average size of a burst will be not that large for most traffic types. It is not likely that the transition times will become much smaller, since when the radio has to react faster, the energy consumption will increase as well. When transmission speeds increase the required virtual size may even become larger.

Overhead

The maximal throughput of the network is determined by the overhead in transmitting control information, the required gab between slots, and by error control. The overhead caused by the transmission of the traffic control depends on the frame-length, which is implementation dependent. The length is restricted by the amount of buffer-space in the base-station and the mobile, but also by the introduced latency. Figure 2 shows the effect of the frame size on the energy efficiency versus the load per mobile. The two bytes control on each data slot only contributes in a reduction of maximal 4%, but can save considerable energy. The overhead to transmit the CRC also reduces the throughput. However, when it is omitted, because error detection and error control are also used in higher levels of the communication protocol, then the slot-scheduler cannot detect a period of bad error conditions. The required gab between slots influences the throughput, but has no effect on the energy consumption. The size of the gab depends on the hardware of the transceiver.

4.2 Energy efficiency of uplink traffic on E^2MaC and Slotted Aloha

In this section we will compare the energy efficiency of a mobile with *uplink traffic* using the E^2MaC protocol and Slotted Aloha, a collision based protocol that is often used as a reference. Downlink traffic is not considered since Slotted Aloha (just like many other protocols) does not care about the energy consumption, and just assumes that mobiles turn their receiver on to find out about downlink traffic. We will not incorporate insignificant details, but will concentrate on the main issue to show the difference in energy efficiency between a reservation and a collision protocol. Energy saving properties like avoiding periods of bad error conditions are not incorporated.

Energy efficiency of Slotted Aloha

In Slotted Aloha, time is divided into slots [1]. Each slot is accessed with probability p by each mobile. When we assume that the aggregate network load does not change when a single station goes in backoff, then we can state that whether or not backoff is used, the probability of success of each data transmission does not change. Therefor, a backoff procedure is of no concern in the analysis of energy consumption [9]. The energy dissipated is determined by the time that the transmitter and receiver must have been on. We neglect the energy needed to receive the identification message from the base station since this happens only once when entering a cell. When the mobile sends a message, the probability that it is successful received by the base station is:

$$P_{tx} = (1 - p)^{n-1} \qquad (1)$$

where p is the probability that a station sends a message in a slot and n is the number of active mobiles in a cell. Every time the mobile attempts to send a message, the receiver is switched on to receive possible positive acknowledgements. The energy efficiency e_{sa} is thus determined by:

$$e_{sa} = (E_{tx} P_{tx} + E_{ack} P_{tx})^{-1} \qquad (2)$$

in which E_{tx} is the energy efficiency factor needed to transmit a message and E_{ack} the efficiency factor to receive the acknowledge.

Energy efficiency of E^2MaC

Since in the E^2MaC protocol collisions can only occur when the mobile enters a cell, their contribution to the average energy consumption per message can be neglected. When a connection has been set-up, then the overhead is determined by the control message and by the reception of the traffic control. The transmitting a control message piggy-backed on a data-slot will be neglected because it only contributes maximal 4%.

When a mobile indicates that it has continuous traffic and does not allow any jitter, then the slot-scheduler will reserve the same slots in each frame for that connection. The mobile thus only needs to receive the traffic control once. This situation has almost optimal energy efficiency and will not be analysed further. In other cases a mobile needs to receive the traffic control once per frame (when we assume that each frame will be used). Since no collisions can occur, acknowledgements are not needed on this level. The energy efficiency e_{e2mac} is thus:

$$e_{e2mac} = E_{tx}\, p \,/\, (E_{tx}\, p + E_{rx} / N) \qquad (3)$$

where N is the number of slots in a frame, p the probability that a mobile sends a message in a frame, E_{tx} is the energy efficiency factor needed to transmit a message and E_{rx} the efficiency factor to receive the traffic control.

Comparison

In our analysis we will assume that the energy consumption for transmission is equal to reception, thus $E_{tx} = E_{rx}$. This approximates the power consumption characteristics of the WaveLAN interface [15]. In our analysis of Slotted Aloha we will further assume that the acknowledgement uses the same channel as used for data transfer and that the receiver needs to be on for 1/8 of the time to transmit one data message (which is optimistic when the size of a slot is one ATM cell). So we can use E_{ack} = 1/8 E_{tx} in equation (2).

Figure 2 shows the energy efficiency characteristics of Slotted Aloha for a various number of mobiles, and for E^2MaC for two frame-sizes.

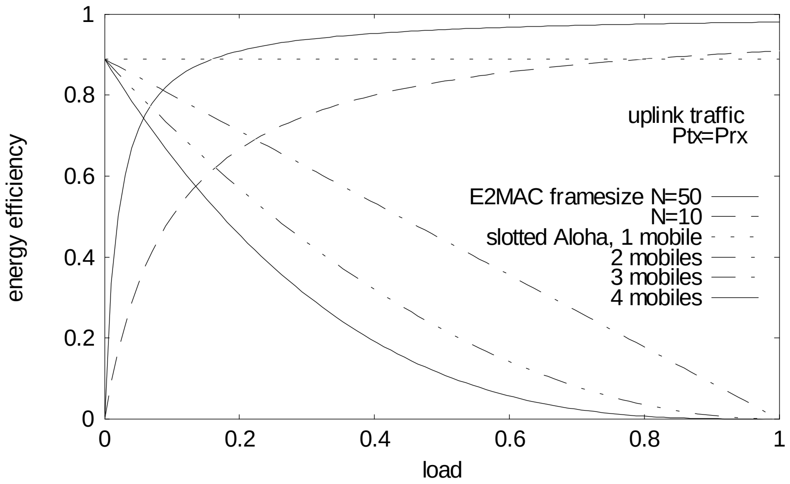

Figure 2: Energy efficiency vs. load for uplink traffic on Slotted Aloha and E^2MaC.

First of all we must note that the energy efficiency of the E^2MaC protocol is independent of the activity of the users, in contrast with Slotted Aloha where the efficiency strongly depends on the activity of other users. Therefor the indicated load for Slotted Aloha is the total load in a cell, and for E^2MaC it is the load per mobile. The energy efficiency of the E^2MaC protocol is much better than the Slotted Aloha protocol. Only when the load in the cell is very low (e.g. when there is only one mobile in the cell), then the energy consumed by the E^2MaC protocol is more than with Slotted Aloha that does not have this overhead. However, when a connection was requested with a best-fit option, or when the load in the cell is low, then the scheduler could have decided to establish a connection that uses the same slots in all frames. This gives the mobile a near-optimal efficiency because it does not even have to receive traffic control. Furthermore, when with Slotted Aloha the load is higher the chance of collisions grow, leading to retries and unpredictable delays. QoS provisions are thus not possible using Slotted Aloha.

The figure also shows the consequences for the energy efficiency of the E^2MaC protocol of various frame-sizes. When the frame-size is larger the energy efficiency is better because the traffic control will be averaged over more data.

In many cases the receiver hardware consumes less energy than the transmitter, thus $E_{tx} > E_{rx}$. This however has little influence on the characteristics, and the conclusions remain the same.

5 Related work

In the recent years much research has been done in providing QoS for the wireless link. Access protocols for these systems typically address network performance metrics such as throughput, efficiency, and packet delay. However, thus far little attention is given to energy conserving protocols, and mainly focuses their effort on energy reduction by circuit design. For example current designs for cellular phones have set aggressive goals for standby time, though all of their efforts are focussed on supply voltage and circuit design [11]. The few that showed some attention to low power protocol design uses one or few principles to minimise the energy consumption and cannot provide QoS to end-users.

In [10] is shown that there is much to be gained from variable frame length in terms of user seen throughput, effective transmission range, and transmitter power for wireless links. This is interesting, since we have shown that using a *fixed* frame size can save energy. Their point of view however was caused by the high error rate on wireless links, where a high error rate on a large frame might not be efficient. A similar point of view to reduce energy consumption is used in [4] that tries to avoid transmission during bad channel periods. Both protocols however lack QoS provisions. In E^2MaC the scheduler of the base station tries to avoid only non-time critical traffic during these periods, thereby not affecting traffic with demanding QoS.

The 802.11 protocol [6] addresses energy consumption explicitly. In this approach the mobile is allowed to turn off and the base station buffers data destined for the mobile meanwhile. When the mobile wakes up, the base station transmits the buffered data for the mobile. This saves energy but also influences the QoS for the connections drastically. It also uses a traffic control for inter-frame synchronisation, but does not guarantee that it will not be delayed.

The R-TDMA protocol was shown to be energy efficient [9], but this is mainly due to the reservation scheme that was used to provide QoS for real-time connections. Other energy saving techniques are not applied.

The protocol design of the energy-conserving medium access control (EC-MAC) protocol [13] is related to the E^2MaC protocol in the sense that it provides QoS and uses a fixed frame length to allow transceivers to turn their radio on exactly in time. Their protocol however does not provide the close QoS relationship as E^2MaC has with its **sleep** command. It further does not apply some energy saving mechanisms such as piggy-backing of control information and adaptations to varying error conditions.

The principle of synchronisation between mobile and base station has been used for some time in paging systems [11]. Paging systems increase battery life by allowing the receiver to be turned off for a relatively long time, while still maintaining contact with the paging infrastructure using a well designed synchronous protocol using various forms of TDM.

The LPMAC protocol [12] uses a similar approach, but it requires that the mobile always receives the traffic control. It also allows bulk data transfers, but provides no QoS guarantees.

Our approach is different. We have moved the main complexity and processing needs from the mobile to the base-station, thereby reducing the energy consumption of the mobile. We have first determined the possible sources of unessential energy consumption on the wireless link for a mobile, and determined the main principles to reduce its energy consumption. Then we have applied *all these principles* to design a wireless MAC protocol that is energy efficient and provides QoS for various traffic characteristics.

6 Conclusions

We have presented a novel MAC protocol that provides support for diverse traffic types and QoS while achieving a good energy efficiency of the wireless interface of the mobile. The main complexity is moved from the mobile to the base station with plenty of energy. The scheduler of the base station is responsible to provide the connections on the wireless link the required QoS and tries to minimise the amount of energy spend by the mobile. The main principles of the E^2MaC protocol are avoid unsuccessful actions, minimise the number of transitions, and synchronise the mobile and the base-station. We have shown in the paper that considerable amounts of energy can be saved using these principles. The protocol is able to provide near-optimal energy efficiency (i.e. only energy is spent for the actual transfer) for a mobile within the constraints of the QoS of all connections, and only requires a small overhead.

In the protocol collisions can only occur when a mobile enters a cell. The mobile and base-station are synchronised in time, which allows the mobile to power up its transceiver precisely when needed. The number of transitions is reduced to the minimum possible under the constraints of the QoS of all connections in the cell. Special attention is given for the signalling of downlink traffic. Every free downlink slot is used to broadcast waiting downlink traffic at the base station, and to provide synchronisation timing information. This allows a mobile to sleep, and once in a while quickly sense whether there is traffic waiting for it.

This protocol is not suited for ad-hoc networks with multiple mobiles, since much of the complexity and energy requirements is moved to a base station to provide a high energy efficiency for the mobile. Furthermore, the typical traffic on an ad-hoc network is quite different from a network with a base station. Therefor, a hybrid MAC protocol that can operate in two modes and that is optimised for both network types will probably be the most efficient.

7 References

[1] Abramson, N.: "Development of the ALOHANET", IEEE transactions on Information Theory, vol. IT-31, pp. 119-123, March 1985.

[2] Borriss, M. "QoS support in ATM and selected protocol implementations", technical report TU Dresden, IBDR, http://www.inf.tu-dresden.de/~mb14/atm.html, Oct. 1995

[3] Chen, et al. "Comparison of MAC Protocols for Wireless Local Networks Based on Battery Power Consumption", IEEE Infocom'98, San Francisco, USA, pp. 150-157, March 1998.

[4] Chockalingam, A., Zorzi, M.: "Energy consumption performance of a class of access protocols for mobile data networks", VTC'98, Ottawa, Canada, May 1998.

[5] Havinga, P.J.M., Smit, G.J.M.: "Minimizing energy consumption for wireless computers in Moby Dick", proceedings IEEE International Conference on Personal Wireless Communication ICPWC'97, Dec. 1997.

[6] Havinga, P.J.M., "Energy efficiency of error correcting mechanisms for wireless communication", CTIT memorandum 1998, the Netherlands, to be published.

[7] Ferrari, D.: "Real-Time Communication in an Internetwork", Journal of High Speed Networks, Vol. 1, n. 1, pp. 79-103, 1992

[8] IEEE, "Wireless LAN medium access control (MAC) and physical layer (PHY) Spec." P802.1VD5, Draft Standard IEEE 802.11, May 1996.

[9] Linnenbank, G.R.J.: "A power dissipation comparison of the R-TDMA and the Slotted-Aloha wireless MAC protocols", Moby Dick technical report, http://www.huygens.org/~linnenba/papers/MobyDick/energy.html, 1997.

[10] Lettieri, P., Srivastava, M.B.: "Adaptive Frame Length Control for Improving Wireless Link Throughput, Range, and Energy Efficiency", IEEE Infocom'98, San Francisco, USA, pp. 307-314, March 1998.

[11] Mangione-Smith, B.: "Low power communications protocols: paging and beyond", Low power symposium 1995, http://www.icsl.ucla.edu/~billms/Publications/pagingprotocols.pdf.

[12] Mangione-Smith, B. et al.: "A low power architecture for wireless multimedia systems: lessons learned from building a power hog", proceedings ISLPED 1996, Monterey CA, USA, pp. 23-28, 1996.

[13] Sivalingam, K.M., Srivastava, M.B. Agrawal, P.: "Low power link and access protocols for wireless multimedia networks", Proceedings IEEE Vehicular Technology Conference, Phoenix, AZ, pp. 1331-1335, May 1997.

[14] Stemm, M. et al.: "Reducing power consumption of network interfaces for hand-held devices", Proceedings MoMuc-3, 1996.

[15] "WaveLAN/PCMCIA network adapter card", http://www.wavelan.com/support/libpdf/fs-pcm.pdf.