5

Energy-efficient wireless communication

In this chapter we present an energy-efficient highly adaptive network interface architecture and a novel data link layer protocol for wireless networks that provides Quality of Service (QoS) support for diverse traffic types[1]. Due to the dynamic nature of wireless networks, adaptations in bandwidth scheduling and error control are necessary to achieve energy efficiency and an acceptable quality of service.

In our approach we apply adaptability through all layers of the protocol stack, and provide feedback to the applications. In this way the applications can adapt the data streams, and the network protocols can adapt the communication parameters.

5.1 Introduction

As already observed before in the previous chapters, the energy consumption of portable computers like PDAs and laptops is the limiting factor in the amount of functionality that can be placed in these devices. More extensive and continuous use of wireless network services will only aggravate this problem. However, even today, research is still focused on performance and (low power) circuit design. There has been substantial research in the hardware aspects of mobile communications energy-efficiency, such as low-power electronics, power-down modes, and energy efficient modulation. However, due to fundamental physical limitations, progress towards further energy-efficiency will become mostly an architectural and software-level issue.

We have shown in Chapter 2 that it is more effective to save energy by a carefully designed architecture of the mobile, the communications device and wireless communication protocols that consider judicious use of the available energy [37].

[1] Major parts of this chapter have been presented at the Second *IEEE International Workshop on Wireless Mobile ATM Implementations (wmATM'99)*, 1999 [32], and will appear in the *Journal on Mobile Networks and Applications (MONET),* 2000 [33].

Energy reduction should be considered in the whole system of the mobile and through all layers of the protocol stack, including the application layer. In this chapter we address the issue of *energy efficiency* in the data link network layer protocols for wireless networks. These protocols typically address network performance metrics such as throughput, efficiency, fairness and packet delay. This chapter addresses the additional goal of efficient energy usage of the mobiles. Considerations of energy efficiency are fundamentally influenced by the trade-off between energy consumption and achievable Quality of Service (QoS). The aim is to meet the required QoS, while minimising the required amount of energy.

The objective of this chapter is to present the design and analysis of a network interface and a medium access protocol, referred to as E^2MaC. The design is driven by two major factors. The first factor is that the design should be energy-efficient since the mobiles typically have limited energy capacity. The second factor is that it should provide support for multiple traffic types, with appropriate Quality of Service levels for each type.

Service model

Traditional communication networks provide a single service model that delivers packets on a best effort basis. The available bandwidth is shared by competing senders on a per packet basis. As a consequence, the packets experience an unpredictable – and possibly very long – delay in getting to their destination. For many traditional applications this is not a real problem as long as the overall delays are not excessive. Other applications, however, that e.g. transfer digitised voice, require a predictable service model. Circuit switched networks can offer such a service with a fixed slot of bandwidth allocated for use by a sender in each time period, and with equal delivery time for each slot.

It is expected that the new generation of wireless networks will carry diverse types of multimedia traffic. Multimedia services, like packet audio and video, and real-time services, e.g. for process control, have strict communication constraints. Multimedia services are typically sensitive to delay and jitter (variations in delay) and demand high bandwidths, but may be prepared to tolerate some data loss [36]. For example, dropping several pixels in a high-resolution image may not be noticeable. Even dropping one frame now and then from a video sequence at 25 frames per second can be tolerated. Hard-real-time applications usually have lower bandwidth requirements, but demand predictable delay and cannot tolerate any errors.

Quality of Service (QoS) is an attractive model for resource allocation and sharing, and is applied in communication networks like ATM [8]. QoS guarantees provide the basis for modern high-bandwidth and real-time multimedia applications like teleteaching and video conferencing. All the multimedia service types and the specific requirements can be expressed in terms of the QoS expected by the application. The notion of QoS service originally stems from communication, but because of its potential in the allocation of all scarce resources, it has found its way into other domains, e.g. operating systems [40]. QoS then involves all layers that are below the application. QoS based resource allocation is based on services or users requesting a resource on some level of quality from a service provider.

In statically connected systems, the service provider will try to reserve resources (end-to-end) upon a request from a user. If the service provider grants the request (possibly after negotiation), the two parts have a QoS contract that gives some notion of guarantee that the service level in the contract shall be sustained. The service user will often rely on the availability of the resources specified in the QoS contract. However, in dynamically connected systems like wireless networks, the availability and quality of resources are generally unpredictable. Therefore, a service provider generally cannot issue a QoS contract that the service user can rely on. QoS based resource management in mobile systems therefore must take this fundamental difference into account.

Wireless system architecture

Wireless LANs can be classified as distributed (*ad-hoc*) or *centralised* systems. Essentially, the existence or lack of fixed wired infrastructure differentiates them.

- In *ad-hoc networks* the infrastructure is build up of mobiles which establish wireless links between them and build a network topology allowing multihop connectivity. Its key characteristics are that there is no fixed infrastructure, and that there is wireless multihop communication, dynamically set up and reconfigurable as mobiles move around.
- *Centralised systems* consist of base stations and mobiles. Its key characteristics are that there is some fixed wired infrastructure, which is always accessible through a single hop wireless link. The base stations are connected to the fixed network and support the communication of the mobiles in range of the base station's radio.

Ad-hoc networks provide more flexibility than centralised systems. However, in ad-hoc networks the data possibly has to pass multiple hops before it reaches its final destination. This leads to a waste of bandwidth as well as an increased risk of data corruption, and thus potentially higher energy consumption (due to the required error control mechanism). Only if the source and destination mobile are in each others reach, ad-hoc networking can be more efficient. However, the use of ad-hoc networks is limited because in general there is not much mobile-to-mobile communication, and in many situations a fixed network is still required.

Although ad-hoc networks are more flexible than centralised systems, they are less suitable for the design of low energy consuming mobiles. The assumption is that mobiles will always have a limited amount of energy, whereas the wired base-stations will have virtually unlimited energy. In a centralised system the base station can therefore be equipped with more intelligent and sophisticated hardware, that probably has a significantly higher energy consumption than the hardware required in the mobile. Portables can then be offloaded with some functionality that will be handled by the base station.

In centralised systems it is further much easier to provide a certain quality of service for applications or users. Since both energy requirements and QoS are our main targets, we will only consider a centralised system here.

Overview of the chapter

In this chapter we will 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. Hybrid TDMA/CDMA schemes benefit from both the capacity of TDMA schemes to handle high bit-rate packet-switched services, and the flexibility of CDMA techniques that allow smooth coexistence of different types of traffic [5]. In this chapter we will deal with three main aspects involved with energy-efficient wireless communication: Medium Access Control (MAC) design, error control, and network interface architecture.

Section 5.2 first presents the basics of wireless data link layer design issues are discussed, i.e. the wireless link limitations, the basic wireless networking functions needed, and introduces the concept of QoS renegotiations. Section 5.3 determines the main sources of energy consumption on wireless interfaces, which provide us the main principles of energy efficient MAC design. Then, Section 5.4 presents a short introduction to ATM and the peculiarities when applied to a wireless system. Section 5.5 presents various error-control alternatives and their consequences on energy consumption. Then, Section 5.6 describes the basic principles and mechanisms of the network interface architecture, and a new MAC protocol E^{2}MaC whose design is driven by energy consumption, diverse traffic type support, and QoS support considerations. Section 5.7 provides an evaluation of the performance of the E^{2}MaC protocol. Related work is presented in Section 5.8, and we will finish with some conclusions.

5.2 Wireless data link layer network design issues

The context in this section is data link-level communication protocols for wireless networks that provide multimedia services to mobile users. As mentioned before, portable devices have severe constraints on the size, the energy consumption, and the communication bandwidth available, and are required to handle many classes of data transfer over a limited bandwidth wireless connection, including delay sensitive, real-time traffic such as speech and video. This combination of limited bandwidth, high error rates, and delay-sensitive data requires tight integration of all subsystems in the device, including aggressive optimisation of the protocols to suit the intended application. The protocols must be robust in the presence of errors; they must be able to differentiate between classes of data, giving each class the exact service it requires; and they must have an implementation suitable for low-power portable electronic devices.

5.2.1 The ISO/OSI network design model

Data communication protocols govern the way in which electronic systems exchange information by specifying a set of rules that, when followed, provide a consistent, repeatable, and well-understood data transfer service. In designing communication protocols and the systems that implement them, one would like to ensure that the protocol is correct and efficient. The ISO/OSI model is a design guide for how network software in general should be built. In this model, protocols are conceptually organised as a series of layers, each one built upon its predecessor. Most network architectures use some kind of layering model, although the specific layers may not be an exact match with the layers defined in the ISO/OSI model.

The rationale behind this layering approach is that it makes in principle possible to replace the implementation of a particular layer with another implementation, requiring only that each implementation provide a consistent interface that offers the same services and service access points to the upper layer. Thus, the goal of service abstraction is modularity and freedom to choose the implementation that is best suited for a particular environment. However, while this model provides an excellent starting point for conceptually partitioning a set of protocol services, it has two implicit assumptions that fail to hold in many practical contexts [78]. First, there is the assumption that cost of abstraction and separation is negligible compared to the gained modularity and flexibility. Second, there is the assumption that interchanging layers that provide the same logical services – for example, a wired physical layer and a wireless physical layer – provide equivalent service.

These assumptions are in general not valid for mobile systems and can impose severe limitations. For example, although the TCP specification contains no explicit reference to the characteristics of the lower layers, implicitly in the timeout and retransmission mechanisms there are the assumption that the error rate is low, and that lost packets occur due to network congestion. TCP has no way of distinguishing between a packet corrupted by bit errors in the wireless channel from packets that are lost due to congestion in the network. The applied measures result on a wireless channel in unnecessary increases in energy consumption and deterioration of QoS. This example attests the need to tailor protocols to the environment they operate in. Separating the design of the protocol from the context in which it exists leads to penalties in performance and energy consumption that are unacceptable for wireless, multimedia applications.

The context of this section is mainly the data link layer. Data link protocols are usually divided into two main functional components: the *Logical Link Control* (LLC) and the *Medium Access Control* (MAC), that are responsible for providing a point-to-point packet transfer service to the network, and a means by which multiple users can share the same medium. The main task of the Data Link layer protocols on a wireless network is to provide access to the radio channel. Wireless link particularities, such as high error rate and scarce resources like bandwidth and energy, and the requirements to provide access for different connection classes with a variety of traffic characteristics and QoS requirements, makes this a non-trivial task. It requires a flexible, yet simple scheme that should be able to adjust itself to different operating conditions in order to satisfy all connections and overall requirements like efficient use of resources like energy and radio bandwidth. The protocols have to support traffic allocation according an agreed traffic contract of a connection, but must also be flexible enough to adapt to the dynamic environment and provide support for QoS renegotiations. It further has to provide error control and mobility related services.

5.2.2 Wireless link restrictions

The characteristics of the wireless channel the Data Link protocol has to deal with are basically high bit error rate (BER), limited bandwidth, broadcast transmission, high energy consumption and half duplex links.

Wireless networks have a much higher *error rate* than the normal wired networks. The errors that occur on the physical channel are caused by phenomena such as signal fading, transmission interference, user mobility and multi-path effects. Typically, the bit error rates observed may be as bad as 10^{-3} or 10^{-4}, which is far more worse than assumed by networks with wired connections. Additionally, the errors show a dynamic nature due to movement of the mobile. In indoor environments propagation mechanisms caused by the interactions between electromagnetic fields and various objects can increase error rates considerably. Especially in the outer regions of the radio cell, the low signal-to noise ratio (SNR) makes wireless link errors a norm rather than an exception in the system.

The *available bandwidth* on a wireless channel is usually much less than offered by wired networks. Consequently, an important design consideration in the design of a protocol, is the efficient use of the available bandwidth.

Closely related to this is the amount of *energy* that is needed to transmit or receive data. The required amount of energy is high, and typically depends on the distance that the radio signal has to propagate between sender and receiver. Since wireless networks for mobile systems will be used more widely and more intensively, the energy consumption that is required to communicate will take a large part of the available energy resources (batteries) of the mobile. So energy consumption will be another main design constraint for the wireless data link protocol of the mobile. In general, saving energy for the base station is not really an issue, as it is part of the fixed infrastructure and typically obtains energy from a mains outlet. However, since the current trend is to have ever smaller area cell sizes, and the complexity of the base station is increasing, this issue might become more important in the future mainly because of economical and thermal reasons.

By their nature wireless radio transmission is a *broadcast medium* to all receivers within the range of a transmitter. This characteristic gives rise to several problems in a wireless environment with multiple cells and mobiles. A mobile that is in reach of more than one base station and communicates with only one of them can cause errors on the communication in the neighbouring cell. Even if the mobiles are just in reach of one base station, interference between mobiles in different cells can also cause errors. Solutions on the physical layer are possible (colouring schemes with multiple frequencies, spread spectrum technologies, near field radio [72], etc.) but are out of the scope of our research. However, provisions for handoff when a mobile moves from one area cell to another, are important and have consequences for the design of a data link protocol.

A radio modem transceiver typically has one part dedicated to transmission, and the other part to reception. Consequently, the radio channel is generally used in *half duplex mode*. The only way to allow full duplex operation over the radio channel is to duplicate transceiver hardware and use two sub-bands in the frequency band, each of them being used for one-way transmission. Because such a solution is not economically viable, and also raises some technical problems, the data link protocol should be designed in such a way that connections in both directions are treated fairly.

5.2.3 Basic wireless networking functions

The challenge of a wireless data link protocol is to overcome the harsh reality of wireless transmission and to provide mobility and multimedia services. The data link layer of a wireless network has to provide assistance to several basic functions: *QoS management* when a connection is initiated or when the operating conditions have changed; *traffic and resource allocation* according to a traffic contract; *error control* to overcome the effect of errors on the wireless link, *flow control* to avoid buffer overflow and also to discard cells of which the maximum allowed delay is exceeded due to retransmissions; *security and privacy* for the mobile user, and *mobility features* to allow handover when a mobile moves to another area cell. In this section we will discuss these items briefly and describe the consequences for the data link layer.

QoS management

To support diverse traffic over a wireless channel, the notion of QoS of a connection is useful. Setting up a connection involves negotiation along a path from sender to receiver in order to reserve the required resources to fulfil the QoS needed. Due to the dynamic nature of wireless channels and the movement of the mobile the agreed QoS level in one or more contracts generally cannot be sustained for a longer period. These situations are not errors, but are modus operandi for mobile computers. Therefore, these situations must be handled efficiently, and QoS renegotiations will occur frequently. Multimedia applications can show a more dynamic range of acceptable performance parameters depending on the user's quality expectations, application usage modes, and application's tolerance to degradation.

Traffic and resource allocation

Each accepted connection has a certain traffic contract that describes the traffic type and required QoS parameters. A slot-scheduler is responsible to assign slots in a transmission frame according to the various traffic contracts. At the same time it must attain a high utilisation of the scarce radio bandwidth and minimise the energy consumption for the mobile.

Error control

Due to the high bit error rate (BER) that is typical for a wireless link, many packets can be corrupted during transmission. If this rate exceeds the allowable cell loss rate of a connection, an effective and efficient error control scheme must be implemented to handle such situations. At the radio physical level redundancy for detecting symbols reduces the bit error rate for the first time. However, it is usually inefficient to provide a very high degree of error correction, and some residual errors pass through. The residual channel characteristic is based on *erases*, i.e. missing packets in a stream. Erasures are easier to deal with than errors, since the exact location of the missing data is known. Then, integrated into the MAC layer (and possibly also into the higher layers), an error control scheme further enhances transmission quality by applying error correction and/or retransmission schemes.

Since different connections do not have the same requirements concerning cell loss rate and cell transfer delay, different error control schemes must be applied for different connection types [60]. The alternatives are Forward Error Correction (FEC), retransmission techniques like automatic repeat request (ARQ), or hybrid FEC/ARQ schemes. To reduce the overhead and energy involved the error control scheme can also be adapted to the current error condition of the wireless connection. The error control mechanisms should trade off complexity, buffering requirements and energy requirements (taking into account the required energy for both computation and communication) for throughput and delay.

Flow control

A connection involves buffering at several places on the path between sender and receiver. Traffic type requirements concerning delay, and implementation restrictions on the buffer capacity generally limit the amount of buffer space available to a connection. Due to the dynamic character of wireless networks and user mobility, the stream of data might be hindered on the way from source to destination. Therefore, flow control mechanisms are needed to prevent buffer overflow, but also to discard packets that have exceeded the allowable transfer time. Depending on the service class and QoS of a connection a different flow control can be applied. For instance, in a video application it is useless to transmit images that are already outdated. It is more important to have the ’fresh’ images. For such traffic the buffer is probably small, and when the connection is hindered somewhere, the oldest data will be discarded and the fresh data will be shifted into the fifo. Flow control can cover several hierarchical layers, but in the context of link access protocols we mainly deal with the buffering required directly at both sides of the wireless link.

Security and privacy

Since eavesdropping of the data bits is a real threat because they will be transmitted over the wireless air interface, security and privacy are important issues in wireless systems. These items are important on two levels: protection of the data on the wireless link, and end-to-end application security. The MAC layer is only capable to provide some basic protection of the data on the wireless link. Since it is hard to make this very secure, end-to-end security will be the most attractive and secure solution.

Mobility features

In a wireless environment the mobility of the mobile will enforce handover procedures when the mobile moves from one area cell to another. As the current trend is that the radius of an area cell decreases (because of the higher bandwidth density and lower energy requirements) handover situations will be encountered frequently.

The task of the link layer is to provide the higher layers of the mobile with information about which area cells are in range, and provide services to actually handle the handover. The radio link quality will be the first parameter to be taken into account for the handover initiation procedure. In the new area-cell a new connection has to be prepared and bandwidth reserved. When a mobile is being handovered to a new area cell, the connection will be dropped if there is insufficient bandwidth to support the connection. Since dropping connections is more undesirable than blocking new connection requests, some bandwidth can be reserved in neighbouring area cells in advance, before the mobile reaches that area cell. It is possible to provide a general pool of bandwidth that can be used for new connections. If it is possible to predict the movement of mobiles, then bandwidth can be saved since not in all neighbouring area cells bandwidth has to be reserved [75].

5.2.4 QoS renegotiation

In a wired network, QoS is usually guaranteed for the lifetime of a connection. In a wireless environment these guarantees are not realistic due to the movement of mobiles and the frequent occurrence of errors on the wireless link.

To prevent service interruptions in a proactive fashion, QoS renegotiations may be required to assure a lower, but deliverable level of service. The difficulty is to provide a mechanism with which QoS parameters of an active connection can be changed dynamically.

QoS control is important during the handover procedure as the mobile moves into a cell and places demand on resources presently allocated to connections from other mobiles. If a mobile faces a significant drop in bandwidth availability as it moves from one cell to another, rather than dropping the connection, the QoS manager might be able to reallocate bandwidth among selected active connections in the new cell. The QoS manager of the new cell selects a set of connections, called *donors*, and changes their bandwidth reservations to attempt satisfactory service for all. To quickly process handover requests, the QoS manager can use cached bandwidth reserves. This cache can then be replenished after the QoS manager has obtained the required bandwidth from the donor connections.

When the mobile moves to a cell where the traffic on the wireless link is much higher, it is not just the current connection that needs renegotiations, even other connections of applications in that area cell may become subject to the QoS renegotiations to allow the new mobile access. The movement of the mobile also influences the (already poor) quality of wireless channels and can introduce dynamic changes in error rate. Especially an indoor environment with small rooms and corridors can cause interactions between the electromagnetic fields and various objects. These interactions can increase error rates considerably. To be able to guarantee an agreed QoS for – especially error sensitive – connections, error recovery techniques using error correcting codes or retransmission is required. In addition to this, before completely closing a connection on a faulty link, the link errors can be gracefully tolerated by renegotiations of QoS. Many multimedia applications can deal with varying bandwidth availability once provided with sufficient information about the operating conditions. For instance, video transmission schemes may adjust their resolution, their frame rates, and encoding mechanism to match the available bandwidth or deal with the current error conditions.

5.3 Energy-efficient wireless MAC design

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.

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 transmit mode, the device is transmitting data; in receive mode, the receiver is receiving data; in idle mode, it is doing neither, but the transceiver is still powered and ready to receive or transmit; in sleep mode, the transceiver circuitry is powered down, except sometimes for a small amount of circuitry listening for incoming transmissions [50].

Several causes for unessential energy consumption exist. We will review in this section some of the most relevant sources of unessential energy consumption.

- First of all, most applications have low traffic needs, and hence the transceiver is *idling* most of the time. Measurements show that on typical applications like a web-browser or e-mail, the energy consumed while the interface is on and idle is more than the cost of actually receiving packets [54][74].
- Second, the typical *inactivity threshold*, which is the time before a transceiver will go in the off or standby state after a period of inactivity, causes the receiver to be in a too high energy consuming mode needlessly 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 receive 'useless' data before it receives the traffic control.

- Fourth, 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 (e.g. 250 µs for WaveLAN). A protocol that assigns the channel per slot will cause significant overhead due to turnaround.
- Fifth, in broadcast networks *collisions* may occur (happens mainly at high load situations). This causes the data to become useless and the energy needed to transport that data to be lost.
- Sixth, the *overhead of a protocol* also influences the energy requirements due to the amount of 'useless' control data and the required computation for protocol handling. The overhead can be caused by long headers (e.g. for addressing, mobility control, etc), by long trailers (e.g. for error detection and correction), and by the number of required control messages (e.g. acknowledgements). In many protocols the overhead involved to receive or transmit an amount of data can be large, and may depend on the load of the network. In general, simple protocols need relatively less energy than complex protocols.
- Finally, the high *error rate* that is typical for wireless links is another source of energy consumption. First, when the data is not correctly received the energy that was needed to transport and process that data is wasted. Secondly, energy is used for error control mechanisms. 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. Furthermore, energy is consumed for the calculation and transfer of redundant data packets and an error detection code (e.g. a CRC). Finally, because in wireless communication the error rate and the channel's signal-to-noise ratio (SNR) vary widely over time and space, a fixed-point error control mechanism that is designed to be able to correct errors that rarely occur, wastes energy and bandwidth. If the application is error-resilient, trying to withstand all possible errors wastes even more energy in needless error control.

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

Main principles of energy-efficient MAC design

The above observations 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 main principles that can be used to design a MAC protocol that is energy efficient for the mobile.

- *Avoid unsuccessful actions of the transceiver.* *(P1)*

  Two main topics cause unsuccessful actions: collisions and errors.

  Every time a *collision* occurs energy is wasted because the same transfer has to be repeated again after a backoff period. A protocol that does not suffer from collisions can have 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 diminishes.

  A protocol, in which a base-station broadcasts 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 significantly. 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 waste bandwidth (but not energy) when slots are reserved for a request, but not used always. In such a reservation mechanism, energy consumption is further reduced because there is less 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. The error control mechanisms can be adapted to the current error condition in such a way that it minimises the energy consumption needed and still provides (just) enough fault tolerance for a certain connection. Due to the dynamic nature of wireless networks, *adaptive error control* can give significant gains in bandwidth and energy efficiency [23][82]. This avoids applying error control overhead to connections that do not need it, and allows the possibility to apply it selectively to match the required QoS and the conditions of the radio link. Note that this introduces a trade-off between communication and computation [36]. Section 5.5 goes into more detail on this issue. A different 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.

- *Minimise the number of transitions.* *(P2)*

  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 there is a trade-off 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 by piggy-backing new requests on current data streams. Simple protocols can further reduce the required number of transitions due to the low amount of control messages needed.

- *Synchronise the mobile and the base station.* *(P3)*

  Synchronisation is beneficial for both uplink (mobile host to base station) and downlink (base station to mobile host) 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 by which a mobile needs to turn its receiver on. If the wake-up call of the communication is implemented with a low-power low-performance radio, instead of the high-performance high-energy consuming radio, then the required energy can be reduced even more.

- *Migrate as much as possible work to the base-station.* *(P4)*

  In a centralised wireless system architecture, the base-station that is connected to the fixed network and a mains outlet, can perform many tasks in lieu of the mobile. The calculation of a traffic control that adheres the QoS of all connections is an example of such task. At higher levels, the base-station can also perform tasks to process control information, or to manipulate user information that is being exchanged between the mobile device and a network-based server (see Chapter 2).

Note that these principles can reduce the energy consumption of the wireless interface. The energy consumption of the mobile system is much more complex and comprises many issues. The total achieved energy reduction is thus based on many trade-offs. For example, grouping traffic in multimedia video streams to minimise the number of transitions requires the data to be buffered in the client's memory. The required amount of energy needed for buffering reduces the effect of the energy savings principle in some sense.

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. Due to the dynamic character of wireless multimedia systems and time-varying radio channel conditions, flexibility and adaptation play a crucial role in achieving an energy efficient design.

We have chosen to adopt *Asynchronous Transfer Mode* (ATM) mechanisms for the wireless network. We have no intention to build the full-blown B-ISDN ATM protocol stack, but merely adopt the small, fixed size packet and the QoS mechanisms. In the next section we give a short introduction to ATM and motivate why it is suitable for building an energy-efficient wireless network.

5.4 ATM

The challenge of designing a network that can cope with all different service types led to the development of the *Asynchronous Transfer Mode* (ATM). ATM is able to support different kind of connections with different QoS parameters. ATM technology provides deterministic or statistical guarantees with connection-oriented reservations. The original intent of ATM was to form a backbone network for high speed data transmission regardless of traffic type. Later, ATM has been found to be capable of more. Today, ATM scales well from backbone to the customer premises networks and is independent of the bit rate of the physical medium. By preserving the essential characteristics of ATM transmission, wireless ATM offers the promise of improved performance and QoS, not attainable by other wireless communication systems like cellular systems, cordless, or wireless LANs. In addition, wireless ATM access provides location independence that removes a major limiting factor in the use of computers and powerful telecom equipment over wired networks [58].

ATM transports data in small, fixed size (in B-ISDN ATM 53-byte) packets called *cells*. Having a fixed cell size allows for a simple implementation of ATM devices, and results in a more deterministic behaviour. Small cells have the benefit of a small scheduling granularity, and hence provide a good control over queuing delays. This also allows rapid switching that supports any mix of delay-sensitive traffic and bursty data traffic at varying bit rates. ATM carries cells across the network on connections known as *Virtual Circuits*. With a Virtual Circuit the flow of data is controlled at each stage in its path from source to destination. In ATM, the QoS requirements of Virtual Circuits are a key element as it relates to how cells for a Virtual Circuit are processed. The connection-oriented nature allows the user to specify certain QoS parameters for each connection. Network resources are reserved upon the acceptance of a Virtual Circuit, but they are consumed only when traffic is actually generated.

5.4.1 ATM service classes

The ATM service architecture uses procedures and parameters for traffic control and congestion control whose primary role is to protect the network and end-system to achieve network performance objectives. The design of these functions is also aimed at reducing network and end-system complexity while maximising network utilisation. The *ATM service categories* represent service building blocks and introduce the possibility for the user to select specific combinations of traffic and performance parameters. Most of the requirements that are specific to a given application may be resolved by choosing an appropriate ATM Adaptation Layer (AAL). However, given the presence of a heterogeneous traffic mix, and the need to adequately control the allocation of network resources for each traffic component, a much greater degree of flexibility, fairness and utilisation of the network can be achieved by providing a selectable set of capabilities within the ATM-layer itself.

The ATM forum has specified the following ATM Service Categories (ASC). ATM Service Category relates quality requirements for a given set of applications and traffic characteristics to network behaviour.

- *Constant Bit Rate* (CBR). A category based on constant (maximum) bandwidth allocation. This category is used for connections that require constant amount of bandwidth continuously available during the connection lifetime. CBR is oriented to serve applications with stringent time delay and jitter requirements (like telephony), but is also suitable for any data transfer application which contains smooth enough traffic.
- *Variable Bit Rate* (VBR) for statistical (average) bandwidth allocation. This is further divided into real-time (rt-VBR) and non-real-time (nrt-VBR), depending on the QoS requirements. Rt-VBR is intended to model real-time applications with sources that transmit at a rate which varies in time (e.g. compressed images) and have strict delay constraints. Video-conference is a suitable application, in which the real-time constraint should guarantee a synchronisation of voice and image, and the network resources are efficiently utilised because of the varying bandwidth requirements due to compression. Nrt-VBR is for connections that carry variable bit rate traffic with no strict delay constraints, but with a required mean transfer delay and cell loss. Nrt-VBR can be used for data-transfer like response-time critical transaction processing (e.g. airline reservation, banking). The undetermined time constraints give the possibility to use large buffers.
- *Available Bit Rate* (ABR) where the amount of reserved resources varies in time, depending on network availability. The variation managed by the traffic control mechanisms is reported to the source via feedback traffic. Compliance to the variations from the feedback signal should guarantee a low cell loss ratio for the application. Generally, it is necessary to use large buffers to offer ABR service on the network due to the burst nature of the service. It has no guaranteed cell transfer delay, but just a minimum guaranteed bandwidth. This category provides an economical support to those applications that show vague requirements for throughput and delay and requires a low cell loss ratio. Applications are typically run over protocol stacks like TCP/IP, which can easily vary their emission as required by the ABR rate control policy.
- *Unspecified Bit Rate* (UBR) has no explicit resource allocation and does not specify bandwidth or QoS requirements. Losses and error recovery or congestion control mechanisms could be performed at higher layers, and not at lower network layers. UBR can provide a suitable solution for less demanding applications like data applications (e.g. background ftp) that are very tolerant to delay and cell loss. These services can take advantage of any spare bandwidth and will profit from the resultant reduced tariffs.

5.4.2 Admission control and policing

Setting up a virtual connection involves taking information on the required service class and QoS. Using this information the system negotiates along the path from source to destination in order to reserve the necessary resources. A traffic contract specifies the negotiated characteristics of a virtual connection at an ATM User Network Interface (UNI). Each QoS parameter consists of a value pair, one representing the low end, and the other the high end. This is called the tolerable range.

Once admitted, the system continually checks that the virtual connection sends data according to its allowance, known as *policing*. When the value of the delivered QoS parameter falls outside the tolerable range, the contract is be violated.

Functions related to the implementation of QoS in ATM networks are usage parameter control (UPC) and connection admission control (CAC). In essence, the UPC function (implemented at the network edge) ensures that the traffic generated over a connection conforms to the declared traffic parameters. Excess traffic may be dropped or carried on a best-effort basis. The CAC function is implemented by each switch in an ATM network to determine whether the QoS requirements of a connection can be satisfied with the available resources.

5.4.3 Wireless ATM

At the moment there are already wireless LANs and wireless systems offering data services and mobile data. These mobile systems offer low bit rate wireless data transmission with mobility and roaming possibility. Wireless LANs offer mobility only in restricted, smaller areas of coverage without wide area roaming capabilities. The achieved bit rates are generally greater than with current mobile systems. The third generation mobile telecommunication systems, such as UMTS (Universal Mobile Telecommunication System) aim to achieve data services of up to 2 Mbit/s, which is a significant improvement over the second-generation mobile systems. However, the importance of speech service may overrun the 2 Mbit/s data service goals [58]. The third generation wireless networks will enable mobiles to carry integrated multimedia. Wireless ATM networks can be useful for these new generation wireless networks because of its ability to handle traffic of different classes and integrate them into one stream. A wireless ATM network consists generally of a cluster of base stations interconnected by a wired ATM network (see Figure 1).

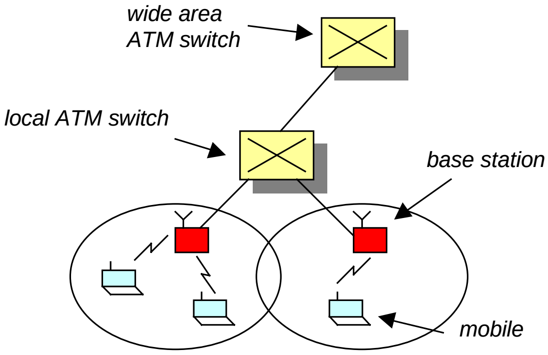

Figure 1: Wireless ATM architecture.

Originally, ATM was characterised by bandwidth on demand at megabits per second rates; it operates at very low bit error rate environments, supports packet switched transport, virtual circuit connections, and statistical sharing of the network resources among different connections.

Wireless networking is inherently unreliable and the bandwidth supported is usually lower than that of fixed networks. Various forms of interference on the wireless link result in high error rates, and thus introduces delay, jitter and an even lower effective bandwidth. Mobility of the user makes these problems even more dynamic and introduces the need for handover mechanisms when the user comes in reach of a different base-station.

The characterisation of ATM – that was designed for wired networks – seems rather contradictory with the operating conditions of wireless networks. Even with high redundancy introduced at several layers (i.e. physical, medium access control, transport and applications) the quality of service may not be guaranteed. Therefore, when adopting ATM in a wireless environment we need to adopt a more dynamic approach to resource usage. *Applications* must adapt their QoS requirements on the current operating environment. Explicit renegotiation of the QoS of a connection about the available resources between the application and the wireless system is essential in wireless ATM systems.

Since a connection typically involves both a fixed and a wireless part, the wireless link should support similar mechanisms as the fixed ATM network. Therefore it has to support all traffic types taking into account the characteristics of the wireless medium. The medium access protocol should be able to bridge the fixed and wireless world and provide ATM services transparently over a wireless link.

Wireless ATM is a topic on which many research activities are going on, e.g. Magic WAND [57], MEDIAN [18], NTT AWA [43]. Most projects aim to extend ATM to the mobile terminal. The main difference can be observed in air interface. No project explicitly addresses reduction in power consumption as a major issue.

5.5 Energy-efficient error control

Since high error rates are inevitable to the wireless environment, *energy-efficient error-control* is an important issue for mobile computing systems. This includes energy spent in the physical radio transmission process, as well as energy spent in computation, such as signal processing and error control at the transmitter and the receiver.

Error-control mechanisms traditionally trades off complexity and buffering requirements for throughput and delay [46][48][15]. In our approach we apply energy consumption constraints to the error-control mechanisms in order to *enhance energy efficiency under various wireless channel conditions*. In a wireless environment these conditions not only vary dynamically because the physical conditions of a communication system can vary rapidly, but they can also vary because the user moves from an indoor office environment to a crowded city town. Not only the characteristics could have changed, it is even possible that a complete different infrastructure will be used [71]. The communication interface of the mobile must not only be able to adapt to these situations and provide the basic functionally, it must also do it energy efficient in all these situations. At the same time, the Quality of Service guarantees of the various connections should still be supported. In some cases it may be impossible to maintain the QoS guarantees originally promised to the application as the channel degrades, for example when the user moves into a radio shadow where the radio loses physical layer connectivity.

5.5.1 The error model

In any communication system, there have always been errors and the need to deal with them. Wireless networks have a much higher error rate than the normal wired networks. The errors that occur on the physical channel are caused by phenomena such as signal fading, transmission interference, and user mobility.

In characterising the wireless channel, there are two variables of importance. First, there is the Bit Error Rate (BER) – a function of Signal to Noise Ratio (SNR) at the receiver -, and second the burstiness of the errors on the channel. Figure 2 presents a graphical view of packets moving through this channel.

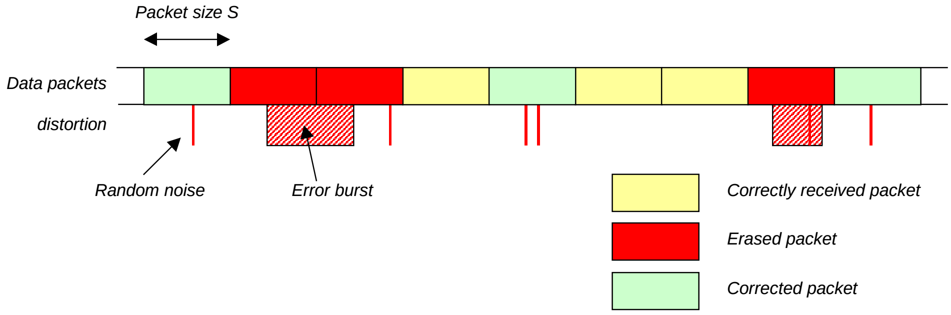

Figure 2: Error characteristics and packet erasures.

This leads to two basic classes of errors: packet erasures and bit corruption errors [21][83]. Error control is applied to handle these errors.

Note that when the bit errors are independent, the packet error rate (*PER*) is related to the size of the packet (*s*) and the bit error rate (*BER*) as

$$PER = 1 - (1 - BER)^s \qquad (1)$$

While this does not take into account the bursty nature of a wireless link, it gives an idea of the influence of the packet length on the error rate of a packet. Even one uncorrected bit error inside a packet will result in the loss of that packet. Each lost packet directly results in wasted energy consumption, wasted bandwidth, and in time spent. This loss might also result in the additional signalling overhead of an ARQ protocol [45]. Because of this, it is important to simultaneously adapt the error control mechanism when the packet size is maximised to minimise the number of transitions. In Section 5.7.2 we will analyse the effects of packet length and energy efficiency in more detail.

5.5.2 Error-control alternatives

There are a large variety of error-control strategies, each with its own advantages and disadvantages in terms of latency, throughput, and energy efficiency. Basically there are two methods of dealing with errors: retransmission (Automatic Repeat reQuest (ARQ) and Forward Error Correction (FEC). Hybrids of these two also exist. Within each category, there are numerous options. Computer communication generally implements a reliable data transfer using either methods or a combination of them at different levels in the communication protocol stack. Turning a poor reliability channel into one with moderate reliability is best done within the physical layer utilising signal space or binary coding techniques with soft decoding. FEC is mainly used at the data link layer to reduce the impact of errors in the wireless connection. In most cases, these codes provide less than perfect protection and some amount of residual errors pass through. The upper level protocol layers employ various block error detection and retransmission schemes (see e.g. [67][39]).

- With *FEC* redundancy bits are attached to a packet that allow the receiver to correct errors which may occur. In principle, FEC incurs a fixed overhead for every packet, irrespective of the channel conditions. This implies a reduction of the achievable data rate and causes additional delay. When the channel is good, we still pay this overhead. Areas of applications that can benefit in particular from error-correction mechanisms are *multicast applications* [74][65][61]. Even if the QoS requirement is not that demanding, insuring the QoS for all receiving applications is difficult with retransmission techniques since multiple receivers can experience losses on different packets. Individual repairs are not only extremely expensive, they also do not scale well to the number of receivers. Reducing the amount of feedback by the use of forward error correction, leads to a simple, scalable and energy-efficient protocol.

  Several studies have shown that adaptive packet sizing and FEC can significantly increase the throughput of a wireless LAN, using relative simple adaptation policies (e.g. [21][24][60]). Note that, due to the burst errors, FEC block codes might require interleaving to spread the errors over the whole packet. However, burst error events on the indoor wireless channel caused by slow-moving interference may last for hundreds of milliseconds, rendering interleaving infeasible for time-critical (delay and jitter) applications [29].

- Using *ARQ*, feedback is propagated in the reverse direction to inform the sender of the status of packets sent. The use of ARQ results in an even more significant increase of delay and delay variations than FEC [66]. The retransmission requires additional buffering at the transmitter and receiver. A large penalty is paid in waiting for and carrying out the retransmission of the packet. This can be unacceptable for systems where Quality of Service (QoS) provisioning is a major concern, e.g. in wireless ATM systems. These communications will include video, audio, images, and bulk data transfer, each with their own specific parameter settings regarding for example jitter, delay, reliability, and throughput [19]. Solutions to provide a predictable delay at the medium access control layer by reserving bandwidth for retransmission are possible [27], but spoil bandwidth.

ARQ schemes will perform well when the channel is good, since retransmissions will be rare, but perform poorly when channel conditions degrade since much effort is spent in retransmitting packets. Another often ignored side effect in ARQ schemes is that the round-trip-delay of a request-acknowledge can also cause the receiver to be waiting for the acknowledge with the receiver turned 'on', and thus wasting energy.

- *Hybrids* do not have to transmit with maximum FEC redundancy to deal with the worst possible channel. Under nominal channel conditions, the FEC will be sufficient, while under poor channel conditions ARQ will be used. Although more efficient than the pure categories, a hybrid system is still a rigid one since certain channel conditions are assumed.
- *Adaptive error control* allows the error-control strategy to vary as the channel conditions vary. The error control can be FEC, ARQ, or a hybrid. The wireless channel quality is a function of the distance of user from base station, local and average fading conditions, interference variations, and other factors. Furthermore, in packet data systems the bursty nature of data traffic also causes rapid changes in interference characteristics. In a wireless channel, link adaptations should occur frequently because of the rapid changes in signal and interference environment. In such a dynamic environment it is likely that any of the previous schemes is not optimal in terms of energy efficiency all the time. Adaptive error control seems likely a source of efficiency gain.

Adaptive error control can be added fairly easily to a MAC protocol and link layer protocols. First of all, the adaptive error-control techniques have to be present in the sender and receiver.

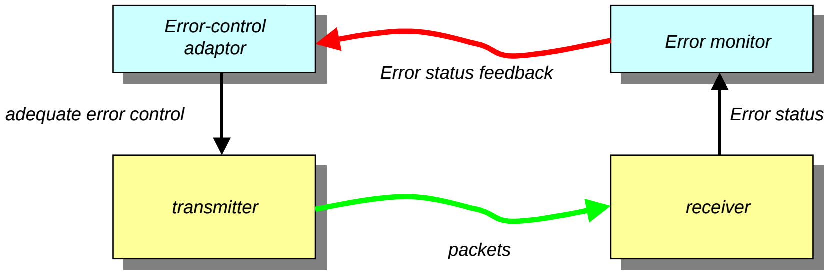

Figure 3: Feedback loop for adaptive error control.

Secondly, a *feedback loop* is required to allow the transmitter to adapt the error coding according to the error rate observed at the receiver. Normally, such information consists of parameters such as mean carrier-to-interference ratio (C/I) or signal-to-noise ratio (SNR), standard deviation of SNR channel impulse response characterisation, bit error statistics (mean and standard deviation), and packet error rate. The required feedback loop limits the responsiveness to the wireless link conditions. Additional information can be gathered with a technique that performs link adaptation in an implicit manner by purely relying on acknowledgement (ACK/NACK) information from the radio link layer.

Depending on the application, the adaptation might not need to be done frequently. If, for example, the application is an error-resilient compression algorithm that when channel distortion occurs, its effects will be a gradual degradation of video quality, then the best possible quality will be maintained at all BERs ([3][56][76][77]).

A more detailed comparison of the performance of ARQ and FEC techniques has been made by many researchers (e.g. [44],[66] and [85]), and is not part of our research.

The choice of energy-efficient error-control strategy is a strong function of QoS parameters, channel quality, and packet size [44]. Since different connections do not have the same requirements concerning e.g. cell loss rate and cell transfer delay, different error-control schemes must be applied for different connection types. The design goal of an error-control system is to find optimum output parameters for a given set of input parameters. Input parameters are e.g. channel BER or maximum delay. Examples of output parameters are FEC code rate and retransmission limit. The optimum might be defined as maximum throughput, minimum delay, or minimum energy consumption, depending on the service class (or QoS) of a connection. Real-time traffic will prefer minimum delay, while most traditional data services will prefer a maximum throughput solution. All solutions in a mobile environment should strive for minimal energy consumption.

5.5.3 Local versus end-to-end error-control

The networking community has explored a wide spectrum of solutions to deal with the wireless error environment. They range from local solutions that decrease the error rate observed by upper layer protocols or applications, to transport protocol modifications and proxies inside the network that modify the behaviour of the higher level protocols [23].

Addressing link errors near the site of their occurrence seems intuitively attractive for several reasons.

- It is most efficient that the error-correcting techniques to be tightly coupled to the transmission environment because they understand their particular characteristics [31].
- Entities on the link are likely to be able to respond more quickly to changes in the error environment, so that parameters such as FEC redundancy and packet length are varied with short time.
- Performing FEC on an end-to-end basis implies codes that deal with a variety of different loss and corruption mechanisms, even on one connection. In practice this

implies that different codes have to be concatenated to deal with every possible circumstance, and the resulting multiple layers of redundancy would be carried by every link with a resultant traffic and energy consumption penalty [30]. End-to-end error control requires sufficient redundancy for the worst case link, resulting in a rate penalty on links with less impairment. Local error control requires only extra bandwidth where it is truly needed.

- Practically, deploying a new wireless link protocol on only those links that need it is easier than modifying code on all machines. Application-level proxies address this problem to some extend, but they are currently constrained to running end systems, whereas local error control can operate on exactly the links that require it [23].

Despite these attractions, trying to solve too much locally can lead to other problems. In the case of local error control for wireless links, there are at least three dangers [23].

- Local error control alters the characteristics of the network, which can confuse higher layer protocols. For example, local retransmission could result in packet reordering or in large fluctuations of the round-trip time, either of which could trigger TCP timeouts and retransmissions.
- Both local and end-to-end error control may respond to the same events, possibly resulting in undesirable interactions, causing inefficiencies and potentially even instability.
- End-to-end control has potentially better knowledge of the quality requirements of the connection. For example, a given data packet may bear information with a limited useful lifetime (e.g. multimedia video traffic), so error control that will cause the delay to exceed a certain value is wasted effort. It might be better to drop a corrupt video packet, than to retransmit it, since retransmission may make the next packet late.

Given the significant advantages of local error control, we will pursue a local approach for the lower layers of the communication protocol stack. However, while we propose that the primary responsibility for error control fall to the local network, there is no reason to dogmatically preclude the involvement of higher level protocols. In particular, the application should be able to indicate to the local network the type of its traffic and the QoS expectations.

The lowest level solution to local error-control is by using hardware error-control techniques such as adaptive codecs and multi-rate modems. While these are attractive in terms of simplicity, they may leave a noticeable residual error rate. In addition, while they reduce the average error rate, they cannot typically differentiate between traffic of different connections. A MAC and link-layer approach that is able to apply error control on a per-traffic basis is an attractive alternative. These protocols, such as IEEE 802.11 [41], MASCARA [5], and E^{2}MaC [33], are – or can be made – traffic-aware (rather than protocol-aware) by tailoring the level of error control to the nature of the traffic (e.g. bounding retransmission for packets with a limited lifetime).

5.5.4 Related work

Error control is an area in which much research has been performed. Books on error control, such as [46], cover the basic FEC and ARQ schemes well. More recently, much work has focussed on error control in wireless channels. Some error-control scheme alternatives and their implications have been discussed in Section 5.5.1.

Adaptive error-control is mainly used to improve the throughput on a wireless link [21][24]. Schuler presents in [66] some considerations on the optimisation and adaptation of FEC and ARQ algorithms with focus on wireless ATM developments. The optimisation, with respect to the target bit error rate and the mapping of the wireless connection quality to the ATM QoS concept, is discussed in detail. Eckhardt and Steenkiste [23] argue and demonstrate that protocol-independent link-level local error control can achieve high communication efficiency even in a highly variable error environment, that adaptation is important to achieve this efficiency, and that inter-layer coexistence is achievable.

In [82] it has been shown that classic ARQ strategies could lead to a considerable waste of energy (due to several reasons: more communication overhead, more transitions, longer communication time, etc.). They propose an adaptive scheme, which slows down the transmission rate when the channel is impaired. This scheme saves energy without a significant loss in throughput. Several other solutions have been proposed [3][55], but the focus with these solutions is mainly on increasing the throughput, and not on preserving QoS and energy efficiency.

Classic ARQ protocols overcome errors by re-transmitting the erroneously received packet, regardless of the state of the channel. Although in this way these retransmission schemes *maximise the performance* – as soon as the channel is good again, packets are received with minimal delay – the consequence is that they expend energy [82]. When the tolerable delay is large enough, ARQ outperforms error-correction mechanisms, since the residual error probability tends to zero in ARQ with a much better energy efficiency than error correction methods [85].

Most relevant work that relates the error coding strategy to energy consumption is by Zorzi and Lettieri. Zorzi describes in [82] and [85] an adaptive probing ARQ strategy that slows down the transmission rate when the channel is impaired without a significant loss in throughput. A modified scheme is also analysed, which yields slightly better performance, but requires some additional complexity. Lettieri ([45]) describes how energy efficiency in the wireless data link can be enhanced via adaptive frame length control in concert with adaptive error control based on hybrid FEC and ARQ. The length and error coding of the frame going over the air and the retransmission protocol are selected for each application stream based on QoS requirements and continually adapted as a function of varying radio channel conditions.

All error-control techniques introduce latency, a problem that is more prominent with limited bandwidth. This poses the problem that low latency (for interactivity) and high reliability (for subjective quality) are fundamentally incompatible under high traffic conditions [29]. Some multimedia applications might, however, be able to use the possibly corrupt packet. With *multiple-delivery transport service* multiple possibly corrupt but increasingly reliable versions of a packet are delivered to the receiving application [29]. The application has the option of taking advantage of the earlier arriving corrupt packet to lower the perceived latency, but eventually replaces them with the asymptotically reliable version.

In the concept of *incremental redundancy* (IR) [60], redundant data, for the purpose of error correction, is transmitted only when previously transmitted packets of information are received and acknowledged to be in error. The redundant packet is combined with the previously received (errored) information packets in order to facilitate error correction decoding. If there is a decoding failure, more redundancy is transmitted. The penalty paid for increased robustness and higher throughput is additional receiver memory and higher delay.

5.6 Energy-efficient wireless network design

This section describes the basic principles and mechanisms of the network interface architecture implemented in our research, and our energy efficient medium access control protocol for wireless links, called E^{2}MaC. The protocol and the architecture are targeted to a system in which quality of service (including the incurring energy consumption) plays a crucial role. The ability to integrate diverse functions of a system on the same chip provides the challenge and opportunity to do system architecture design and optimisations across diverse system layers and functions [73].

As mentioned before, two key requirements in mobile multimedia systems are:

- *Requirement 1*: the need to maintain quality of service in a mobile environment and,
- *Requirement 2*: the need to use limited battery resources available efficiently.

We have tackled these problem by making the system highly adaptive and by using energy saving techniques through all layers of the system. Adaptations to the dynamic nature of wireless networks are necessary to achieve an acceptable quality of service. It is not sufficient to adapt just one function, but it requires adaptation in several functions of the system, including radio, medium access protocols, error control, network protocols, codecs, and applications. Adaptation is also a key to enhancing battery life. Current research on several aspects of wireless networks (like error control, frame-length, access scheduling) indicate that continually adapting to the current condition of the wireless link have a big impact on the energy-efficiency of the system [13][16][36][45][41]. In our work these existing ideas and several new ideas have been combined into the design of adaptive energy efficient medium access protocols, communication protocol decomposition, and network interface architecture [37] using the previously mentioned principles P1, P2, P3, and P4.

5.6.1 System overview

The goals of low energy consumption and the required support for multiple traffic types lead to a system that is based on reservation and scheduling strategies. The wireless ATM network is composed of several base-stations that each handle a single radio cell[2] possibly covering several 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 few mobiles. The backbone of the base-stations is a wired ATM network. In order to avoid a serious mismatch between the wired and wireless networks, the wireless network part should offer similar services as the wired network.

The general theme that influences many aspects of the design of the data link protocol is adaptability and flexibility. This implies that for each connection a different set of parameters concerning scheduling, flow control and error control should be applied.

We do not intend to handle all aspects of a full-blown wireless ATM network that provides all possible services. We adapt some features of ATM because they can be used quite well for our purpose. To implement the full ATM stack would require a large investment in code and hardware. The QoS provisions of ATM fit quite well with the requirements of multimedia traffic. This provides much more possibilities for differentiating various media streams than an often used approach in QoS providing network systems with just two priority levels (real-time versus non-real-time) [17], or even multiple priority levels [33].

However, when adopting ATM in a wireless environment, we need a much more dynamic approach to resource usage. The small size packet structure and small header (in B-ISDN ATM 48 bytes data and 5 bytes header) allows for a simple implementation. Small cells have the benefit of a small scheduling granularity, and hence provide a good control over the quality of a connection. The fixed size also allows a simple implementation of a flexible buffering mechanism that can be adapted to the QoS of a connection. Also a flexible error control mechanism has advantage when these cells are adopted. When the base station is connected via a wired ATM network, then the required processing and adaptation can be minimal since they use the same cell structure and the same quality characteristics.

The system contains several QoS managers. Applications might need resources under control of several QoS managers. The QoS managers then need to communicate with each other via a wired network and wirelessly with applications on mobiles. The key to providing service quality will be the scheduling algorithm executed by the QoS manager that is typically located at the base-station. This QoS manager tries to find a (near) optimal 'schedule' that satisfies the wishes of all applications.

[2] Note that the term cell here is different from the term cell used to denote the basic transmission unit in ATM.

Each mobile can have multiple unidirectional connections with different Quality of Service requirements. Five service categories have been defined under ATM (see Section 5.4.1): constant bit rate (CBR), real time VBR, non-real time VBR, unspecified bit rate (UBR), and available bit rate (ABR). The scheduler gives priority to these categories in the same order as listed here possibly using different scheduling algorithms for each category.

The base-station receives transmission requests from the mobiles. The base-station controls access on the wireless channel based on these requests by dividing bandwidth into *transmission slots*. The key to providing QoS for these connections will be the scheduling algorithm that assigns the bandwidth. The premise is that the base-station has virtually no processing and energy limitations, and will perform actions in courtesy of the mobile. The main principles are (using the principles *P*1 to *P4* of Section 5.3): avoid unsuccessful actions by avoiding collisions and by providing provisions for adaptive error control, minimise the number of transitions by scheduling traffic in larger packets, synchronise the mobile and the base-station which allows the mobile to power-on precisely when needed, and migrate as much as possible work to the base-station.

The layers of the communication protocol are summarised in Figure 4. The column in the middle represents the layers used by the base-station; the columns on the left and right represents the layers used by the mobile.

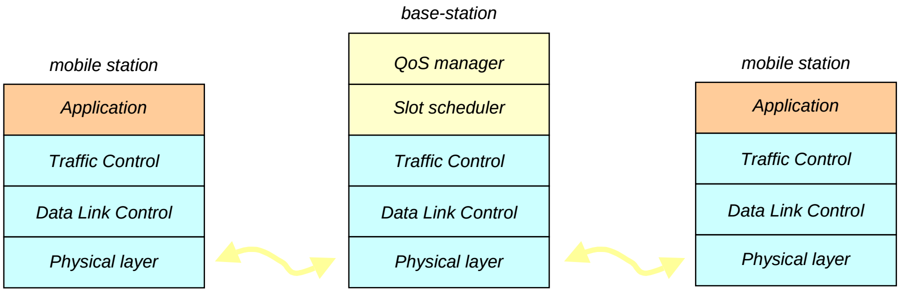

Figure 4: Protocol stack

The lower layers exist in both the mobile and the base station. The *Data link control* manages the data-transfer with the physical layer (using the E^{2}MaC protocol), and *Traffic control* performs error control and flow control. The base-station contains two additional layers: the *Slot Scheduler* that assigns slots within frames to connections, and the *QoS manager* that establishes, maintains and releases virtual connections.

The definition of the protocol in terms of multiple phases in a frame is similar to other protocols proposed earlier. The E^{2}MaC protocol goes beyond these protocols by having minimised the energy consumption of the mobile within the QoS requirements of a connection. The features of the protocol are support for multiple traffic types, per-connection flow control and error-control, provision of service quality to individual connections, and energy efficiency consideration.

5.6.2 E^{2}MaC protocol

In the E^{2}MaC protocol the scheduler of the base station is responsible for providing the required QoS for the connections on the wireless link and tries to minimise the amount of energy spend by the mobile. It uses the four main principles *P1* to *P4*. The protocol is able to provide near-optimal energy efficiency (i.e. energy is spent for the actual transfer only) for a mobile within the constraints of the QoS of all connections.

The protocol uses fixed-length frames of multiple slots. Each slot has a fixed size. A slot determines the time-frame in which data can be received or transmitted. The base-station and mobile are completely synchronised (the time unit is a slot), which allows the mobile to power-on precisely when needed. The base-station controls the traffic for all mobiles in range of the cell and broadcasts the schedule to the mobiles.

E^{2}MaC frame structure

The frame is divided in time-slots that can have three basic types: *traffic control*, *registration request*, and *data*. Only the traffic control type has a fixed position at the start of the frame[3]. 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 in the registration request slots, the traffic control slots and data slots are collision-less.

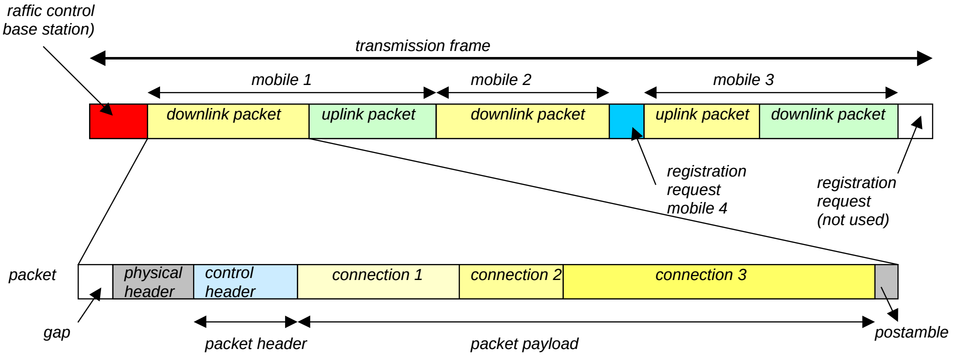

Figure 5: Example of a transmission frame.

The *traffic control slot* (TCS) 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 slot 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)

[3] Implementation restrictions (that cause the time to interpret the traffic control to be significant) might cause the traffic control to be located somewhere in the previous transmission frame.

the connection queue status for all connections, 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 a traffic control slot with the size of one ATM cell when we assume that a frame has maximal 256 slots and a mobile-ID is 16 bits. These numbers seem sufficient in a micro-cellular network in which the cells have the size of one or several rooms. To allow mobiles to power down completely for a while, it might be useful to have a timestamp in the traffic control slot as well. A mobile is not required to receive the TCS of each frame. Depending on the QoS of its connections, it might receive the TCS at a lower frequency.

The *registration request slots* are used by a mobile for two purposes: 1) to announce that it wants to connect to the base station, and 2) for connection management. This traffic is contention based. Because no data traffic is carried during this period, back-off values can be kept short [53]. All slots of a frame that are not used for a connection are registration request slots that can be utilised by a mobile to request a new connection or to update the status of the connection queues of the mobile (the buffer status). To be effective, the base station slot scheduler must know the state of each connection to avoid to assign in vain uplink slots to idle connections. The buffer status is generally forwarded to the base station in each uplink packet, but when no uplink connection is available, it can use a registration request slot to transmit a control connection message with the buffer status. Registration requests allow mobiles that have entered the cell to register at the base-station.

Both the mobile and the base-station use the *data slots* to send the actual data.

The overhead introduced in the physical layer can be significant, e.g. for WaveLAN it can be up to *virtual* 58.25 bytes (for guard space (gap in which the silence level is measured), interfacing delay (required to synchronise to the internal slotsync moments), preamble and postamble, see Figure 6). Moreover, with this interface that has a throughput of 2 Mbit/s, a transition time from sleep to idle of 250 μs already takes *virtually* 62.5 bytes (500 bits). The overhead required to power-up is thus already more than the transmission of one ATM cell.

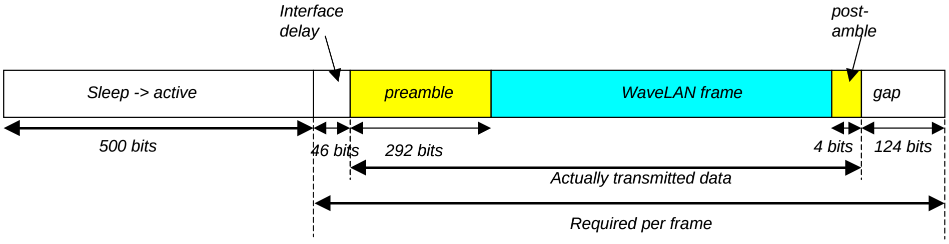

Figure 6: WaveLAN physical layer block format.

This shows that efficient data transmission (in terms of bandwidth utilisation and energy consumption) can only be achieved if the number of ATM cells transmitted is not too small. (In Section 5.7.2 the consequences of the transitions are analysed in more detail.) So, the data cells from one mobile are grouped together as much as possible within the QoS restraints. These cells form a packet that is a sequence of ATM cells possibly for multiple connections. Each packet is constituted of a header, followed by the payload consisting of ATM cells generated by the same mobile. Control messages that do not need the payload can use the header only. Because in general the transition-overhead between transmit and receive modes is much less than the transition overhead between power down modes, transmission packets and reception packets for one mobile are placed right after each other in the frame. A mechanism in which a frame is divided into an uplink part and a downlink part uses the available bandwidth more efficiently, but requires more transitions for the mobile. More details about the slot scheduling can be found in Section 5.6.4.

The header of a packet contains 1) information about the actual length of the data for each connection, 2) the parameters of the error coding applied for each connection, and 3) flow control information about all transmission and reception queues of the network interface.

5.6.3 QoS manager

The *QoS manager* establishes, maintains and releases wireless connections between the base-station and the mobile and also provides support for handover and mobility services. Applications contact the QoS manager when setting up a connection. The QoS manager will inform the applications when they should adapt their data streams when the QoS of a connection has changed significantly. Figure 7 gives a schematic overview of the service model.

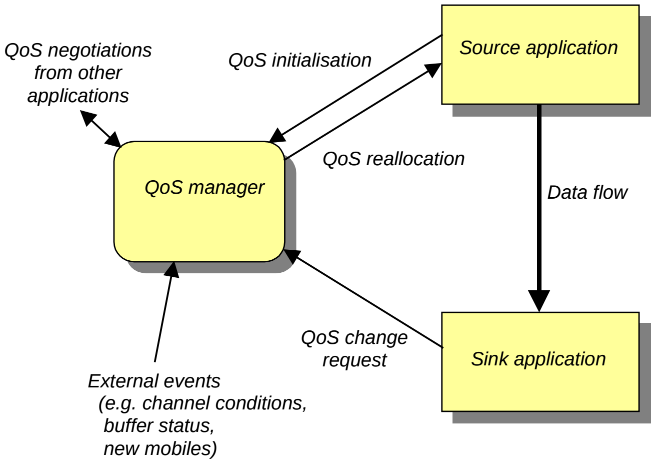
QoS negotiations from other applications
QoS initialisation
Source application
QoS reallocation
QoS manager
Data flow
QoS change request
Sink application
External events (e.g. channel conditions, buffer status, new mobiles)

Figure 7: The service model for adaptive applications.

QoS support in wireless networks involves several considerations beyond those addressed in earlier work on conventional wireline networks. Wireless broadband access is subject to sudden variations in bandwidth availability due to the dynamics of the wireless channel and the service demand (e.g. mobiles moving in and out the base station's coverage area, interactive multimedia connections). In traditional networks based on fixed terminals and high-quality/high capacity links it is feasible to provide 'hard' QoS guarantees to users. However, in the mobile environment, mobility and the need for efficient resource utilisation require the use of a 'soft' QoS model [64].

Multimedia networking requires at least a certain minimum QoS and bandwidth allocation for satisfactory application performance. This minimum QoS requirement has a wide dynamic range depending on the user's quality expectations, application usage modes, and application' tolerance to degradation. In addition, some applications can gracefully adapt to sporadic network congestion while still providing acceptable performance. The soft QoS model is suitable for adaptive multimedia applications capable of gracefully adjusting their performance to variable network conditions. The QoS manager matches the requirements of the application with the capabilities of the network. Figure 7 conceptually illustrates the role of adaptive applications in the QoS model.

The application requests a new connection for a certain *Service Class* that defines the media type (e.g. video, audio, data), interactivity model (e.g. multimedia browsing, videoconference), and various QoS traffic parameters (e.g. required bandwidth, allowable cell loss ratio). The service classes allow multimedia sessions to transparently adapt the quality of the connection when the available resources change marginally without the need to further specify details and without explicit renegotiations.

Network resource allocation is done in two phases. First, the QoS manager checks the availability of resources on the base-stations coverage area at connection setup. The necessary resources are estimated based on the required service. The new connection is accepted if sufficient resources are estimated to be available for the connection to operate within the service contract without affecting the service of other ongoing connections. Otherwise, the connection is refused. Second, while the connection is in progress, dynamic bandwidth allocation is performed to match the requirements of interactive traffic and the available resources. When the available bandwidth changes (because congestion occurs, or the error conditions change drastically), the QoS manager reallocates bandwidth among connections to maintain the service of all ongoing connections within their service contracts. The resulting allocation improves the satisfaction of under-satisfied connections, while maintaining the overall satisfaction of other connections as high as possible. In [64] a bandwidth reallocation algorithm is described that fits well to the QoS model used by the QoS manager.

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 QoS manager on the base-station. The required QoS is determined by classical parameters like *throughput*, *reliability*, *jitter* and *delay*. The quantitative QoS parameters used by the protocol are:

- The required bandwidth, expressed in the number of data slots required in a frame and the frequency that a connection will use slots in a frame.
- jitter, the allowable variation in delay in a frame.

The Traffic Control uses two additional parameters:

- allowable delay, expressed in number of ATM cells
- reliability, the percentage of cells that may be erased due to buffer overflow or errors

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 [34]. 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 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.

Connection management

Control messages are exchanged between the functional entities. These messages are used 1) to perform data link layer flow control between the connection queues of the mobile and the base-station, and 2) to manage connections, i.e. to setup new connections, to update the QoS of current connections, or to release connections.

Just like any packet, a control connection packet contains the buffer status of the connection queues. This status is used by the slot scheduler to make a proper schedule.

Each mobile has at least one *control connection*. When a mobile enters a cell it uses a registration request slot to register itself to the base-station. In this slot contentions can occur with requests from other mobiles. 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 slot it indicates (using the ID of the mobile that it has acknowledged) the connection and assigns a connection-ID of the a 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.

The control connection is scheduled at a rate corresponding to the most stringent requirement of all established connections of the mobile. This requirement can stem from maximal delay, jitter, etc. It will in general be mapped to a deadline time at which a control message or normal data connection of any connection needs to be established by the base station. We will name this interval the *maximum Cell Transfer Delay* (maxCTD).

The mobile can use control messages to change the parameters of existing connections, and request new connections. Possible commands are:

- **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. The value of *s* is determined by the requirements of all connections. The connection with the most stringent requirement will in general dictate the value of *s*. 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.
- **update QoS**. This message can be used to change the current QoS of a connection.
- **new connection**. New connections can be made using the current control connection. 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.

5.6.4 Slot scheduler

The notion of QoS over a wireless link has been the focus of much recent research, and several scheduling algorithms have been proposed [19][59][68]. This section describes a framework by which various scheduling mechanisms can be build that incorporates the QoS requirements and uses the four principles of energy efficient design (see Section 5.3) [4].

The *slot scheduler* on the base-station (*Principle P4*) assigns bandwidth and determines the required error coding for each individual connection. The QoS manager provides the service contracts used.

For a proper slot assignment, the slot scheduler needs to know the current state of each connection. For the downlink direction, the scheduler acquires this information directly by monitoring the corresponding queues in the base station. For the uplink direction, this information can be obtained through the implementation of a dedicated protocol, which can be a *polling* scheme or a *contention* scheme. The polling based mechanism, often proposed for its implementation simplicity, requires a polling interval based on *maxCTD*. The polling scheme introduces a maximal delay equal to the polling interval. The contention based scheme has a delay of one frame (when no contention occurs). Polling and contention are quite different also in the utilisation of the channel bandwidth. The polling mechanism uses a number of slots that linearly increases with the number of mobiles with a slope that grows as the required polling interval decreases. The number of contention slots is practically independent from *maxCTD*. The advantage of the polling scheme is that it can give better guarantees since no contention can occur.

[4] Up to now we have just implemented a simple scheduling algorithm [42].

In E^{2}MaC we use a combination of both polling and contention. In E^{2}MaC all packets include the buffer status of the connection queues. Thus, when there are enough uplink connections (either normal data packets or control connection packets), then the slot scheduler will receive the connection queue status frequently. If this is not sufficient (for example because a connection queue receives more data than anticipated), then a contention slot can be used to transfer a recent buffer status update to the slot scheduler using a control packet.

A schedule is broadcast to all mobiles so that they know when they should transmit or receive data (*Principle P1* and *P3*). In composing this traffic control, the slot scheduler takes into account: the state of the downlink and uplink queues, and the radio link conditions per connection. The slot scheduler is designed to preserve the admitted connections as much as possible within the negotiated connection QoS parameters. It schedules all traffic according to the QoS requirements and tries to minimise the number of transitions the mobile has to make (*Principle P2*). It schedules the traffic of a mobile such that all downlink and uplink connections are grouped into packets taking into account the limitations imposed by the QoS of the connections. The grouping of traffic in larger packets is also used by other protocols to increase the efficiency (both in terms of bandwidth and energy consumption) of the protocol. In general there are three phases: uplink phase, downlink phase, and reservation phase. In the downlink phase the base station transmits data to the mobiles, and in the uplink phase the mobiles transmit data to the base station. In the reservation phase mobiles can request new connections. We will refer to this mechanism as *phase grouping*.

In our protocol we have in principle similar phases, but these are not grouped together in a frame according to the phase, but are grouped together according to the mobile involved. In our protocol we thus group the uplink and downlink phase *of one mobile*. We will refer to this mechanism as *mobile grouping*.

Figure 8 shows the two grouping strategies. In mobile grouping the uplink and downlink packets for a mobile are grouped sequentially (if possible) so that the mobile can power down longer and make minimal transitions between power modes. The power consumption of the WaveLAN modem when transmitting is typical 1675 mW, 1425 mW when receiving, and 80 mW when in sleep mode [80]. Increasing the sleep time period of the radio thus significantly improves the energy efficiency of the wireless network. Moreover, due to the large power-transition times, this mechanism might give the mobile enough time to enter a power-down mode at all. This is shown in the figure where Mobile 2 with phase grouping cannot enter sleep mode after reception of the downlink packet, but is forced to idling[5]. Because the operating modes of phase grouping for a mobile are spread in the frame, the power-mode transition times T_{sleep} to enter sleep mode, and $T_{wake\text{-}up}$ to wake from sleep mode limits the time a mobile can stay in sleep mode.

[5] A power-optimised network interface could stop receiving the downlink packet after it has received data for mobile 2, and thus also enter sleep mode.

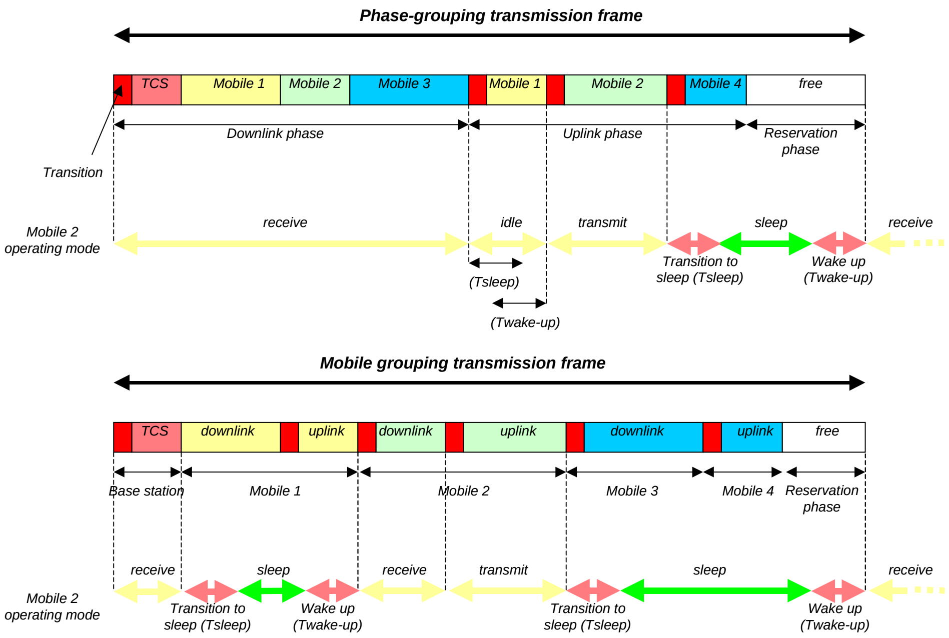

Figure 8: Grouping strategies in a transmission frame.

Notice that in the mobile grouping strategy there is more transition overhead (i.e. one transition per mobile) since the base station does not transmit its data to the mobiles in one packet during the downlink phase of phase grouping. The transition overhead consists of guard space (gap), interfacing delay, preamble, and postamble (see for an example Figure 6). The transition overhead involved with each transmission packet is the reason that the available bandwidth of mobile grouping is less than the available bandwidth in phase grouping. However, since the traffic of a mobile is grouped, the mobile can enter a low-power mode (sleep) for a longer time. In fact, with phase grouping, the mobile is in general forced to receive the complete downlink packet, and will ignore the data not destined for the mobile. The consequences of using mobile grouping on the channel efficiency and the energy consumption is analysed in detail in Section 5.7.2.

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. In this case, 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.

The slot scheduler maintains two tables: a *request table* and a *slot schedule table*. The request table maintains several aspects of the current connections handled by the base station (like the connection type, the connection queue size and status, the error state of the channel with mobile, the assigned bandwidth, the requested reliability). The slot schedule table reflects the assigned number of slots to connections, and the error coding to be applied. This table is essentially broadcast as Traffic Control Slot (TCS) to the mobiles.

These two tables are used by the QoS manager and slot scheduler to assign bandwidth to connections. Since these entities are implemented as software modules on the base-station, their implementation can be adapted easily to other scheduling policies if needed.

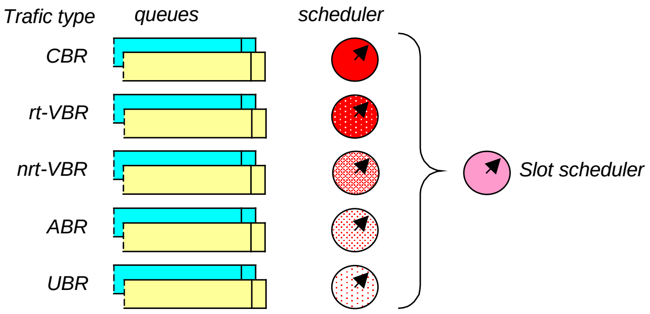

Figure 9: Scheduling per traffic type.

Each ATM service class is assigned a priority, from high to low: CBR, rt-VBR, nrt-VBR, ABR, and UBR. The scheduler gives high priority to CBR and VBR traffic. These traffic sources can reserve bandwidth that the scheduler will try to satisfy. CBR traffic is assigned a maximum bandwidth. If this is not used, the bandwidth will be used by other connections. VBR traffic (both real-time and non-real-time) bandwidth is assigned according to the current traffic flow, up to a specified (average) maximum. The bandwidth adjustment is depending on the current traffic load and the traffic generated by the VBR source. The reservation is updated dynamically in each frame. ABR and UBR traffic, on the other hand, is treated with lower priority and without reservation. Within the same traffic type, the different connections are treated using a scheduling scheme that incorporates the specific requirements of the traffic type (see Figure 9). Real-time traffic requires a fair queuing algorithm. Non-real time traffic can use a more simple scheduling like a round robin mechanism [59].

Dealing with errors

The slot-scheduler dynamically adapts error coding and scheduling to the current conditions in the cell. The error coding required for a specific connection is determined by the error rate observed at the receiver and the required quality of the connection. The slot scheduler retrieves the monitored channel status via a backward connection. It indicates to the network interface which error coding scheme to use. The slot scheduler has to dynamically adapt its schedule when 1) connections are added or removed, 2) connections change their QoS requirements, and 3) the channel between mobile and base station has significant change in error condition.

The scheduler further tries to avoid periods of bad error conditions by not scheduling non-time critical traffic during these periods. Hard-real time traffic (CBR and rt-VBR) remains scheduled, although it has a higher chance of being corrupted. Note that the error conditions perceived by each mobile in a cell may differ. Since the base-station keeps track of the error conditions per connection, it can give mobiles in better conditions more bandwidth. This can lead to a higher average rate on the channel, due to the introduced dependency between connections and channel quality [16]. In Section 5.6.7 we will give more details on this adaptive error control.

5.6.5 Buffer status coding and flow control

Each connection has its own connection queues with customised flow control. Flow control is needed to prevent buffer overflow. ATM cells of a connection on which the maximum allowed delay is exceeded, for example due to bad error conditions, will be discarded by Traffic Control.

The connection queues of the connections can have a different size and replacement policy. The slot scheduler takes this into account in determining the schedule. The scheduler will be able to assign slots most effectively if it has an accurate notion of the transmission buffer status of each mobile.

A coding associates the number of cells N in the queue to a number of bits that represent the status. The coding of N in a number of bits determines the accuracy. There is a trade-off between the information accuracy and the cost of the information in terms of number of bits to be transmitted. The slot scheduler uses the status information to assign up to I slots to that connection. There are several alternatives for queue size coding.

A *linear coding* associates I with the number of cells in the transmission queues. The maximum number of cells (M) that can be indicated with linear coding is determined by the number of bits C used for the coding ($M=2^C-1$). At the base station, the scheduler can assign up to N cell-sized slots to the requesting connection using the relation $I=\min(M,N)$.

The simplest implementation of linear coding requires just one bit to code the connection queue status consists of a stop/run mechanism. The flow control information is also used by the slot-scheduler to assign slots for connections. If a transmission queue indicates **stop**, then this means that the queue is empty and does not need slots. If it indicates **run**, then the queue contains data and it needs slots. If a reception queue indicates **stop**, then it means that it cannot accept more data. A **run** on a reception queue means that it will accept more data. However, with this simple mechanism the scheduler does not know the buffer occupancy. Therefore, it either needs a threshold of multiple cells (and consequently introduces a delay), or the scheduler might assign too much bandwidth for a connection.

Adding more bits to the coding relieves this problem. For example, when four bits are used to code the buffer occupation, then the scheduler can assign slots using I=min(15,N). Since it is accurate, it minimises the number of assigned and unused slots. However, the effect of this algorithm is that – because of the upper-bound M=15 – this coding tends to reduce the differences among the connections. Therefore, it penalises connections with congested buffers.

A logarithmic coding introduces some inaccuracy, but allows a better representation of the buffer occupancy. The coding uses the following relation:

$$I = \begin{cases} 0 & N=0 \\ 1 + [\log_2(N)] & 1 \leq N \leq 2^{M-1} \\ M & N > 2^{M-1} \end{cases} \qquad (2)$$

Although this coding also has an upper-bound, it has a much larger range. Therefore, the scheduler can reveal connections with congested buffers.

5.6.6 The architecture of an energy efficient and adaptive network interface

One of the functional modules of a *Mobile Digital Companion* (MDC) is the network module. This module provides the interface between the external world and the different modules of the MDC. The processor on the MDC is responsible for the establishment of the connections between the modules, but also negotiates with the external infrastructure about the QoS of the connections between network module and the modules that are at the end-point of connections. Once a connection between modules is established, they autonomously communicate with each other in the Companion.

On the Network Module the *Data link control* manages the data-transfer with the physical layer, and *Traffic control* performs error control and flow control. Figure 10 depicts the basic blocks of the architecture of the Network Module. The number of connection queues is dynamic and the figure is just an example.

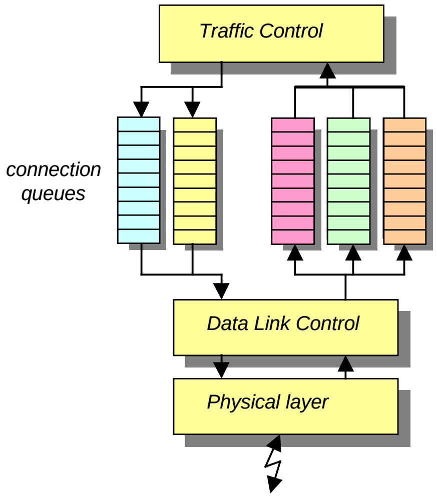

Figure 10: The network interface architecture.

The *Data Link Control (DLC)* performs the traffic allocation of data in the transmission queues. The actual admission decision of connections is made by the QoS management, which informs the Data Link Control using a traffic control packet (either transmitted over the air for the mobile or internally for the base-station). Data Link Control regulates the flow of ATM cells between the physical layer and a local *buffer*. The buffer is organised in such a way that it has a small queue for each connection. This buffer is only meant to store ATM cells for a short time, just enough to implement an effective error control mechanism. When the Data Link Control has to transmit data for a certain connection, it forwards the ATM cells from the transmission queue to the physical layer. On reception it will receive the ATM cells and store them in the queue assigned for that connection. The Data Link Control performs error detection on each ATM cell. The overhead required for error control will be fixed, so that the slot size will not vary.

The *Traffic Control (TC)* controls the flow of data from the connection queues to the corresponding end-points and applies an adaptive error control scheme that operates on individual virtual connections. The choice of an energy efficient error control strategy is a function of QoS parameters, radio channel quality, and packet length. Therefore the architecture of the network interface uses a dynamic error control adapted to these parameters. Each individual connection may use error control schemes that are both adaptive and customised. The selection of the error control scheme and the required size of the queue depend on the QoS constraints imposed on each connection, such as delay constraints or loss-less transfer constraints. This avoids applying error control overhead to connections that do not need it, and allows the possibility to apply it selectively to match the required QoS and the conditions of the radio link. The error control will be based on adaptive error *correcting* techniques. Although well designed retransmission schemes can be energy efficient, they are much more complex to implement (they require a protocol with control messages, sequence numbers, retry counters, etc.) and can introduce intolerable low performance in delay, jitter and bandwidth to fulfil the required QoS of the connection [36]. The redundant data needed to implement the error correction, will be multiples of ATM cells, so that they fit well in a transmission frame. Status information about the channel conditions and the rate of not-correctable errors are fed-back to the Slot Scheduler at the base station. The Slot Scheduler will try to match the radio conditions to the required fault tolerance, and adapt the required error code and required bandwidth accordingly.

5.6.7 Adaptive error control

A wireless channel quality is dynamic because of the rapid changes in signal and interference environment. The wireless channel quality is a function of the distance of user from base station, local and average fading conditions, interference variations, and other factors. Furthermore, in packet data systems the bursty nature of data traffic also causes rapid changes in interference characteristics.

Due to the dynamic nature of wireless networks, adaptive error control can give significant gains in bandwidth and energy efficiency (see Section 5.5.2). The input parameters for an adaptive error-control system can be classified into two main groups: requirements by the upper protocol layers and momentary transmission quality. Adaptation of the error control can be influenced by three considerations [66]:

1. The FEC redundancy can be adapted to the channel bit error rate and induced energy consumption [36]. The error control system has to find a balance between the added redundancy and the bit error rate and energy consumption.
2. The error control algorithm can be adapted to the required quality. For a wireless connection that tolerates a specified cell loss rate, the error control parameters can be tailored to just meet the requirements.
3. The performance of various error-correcting methods depends on the actual error statistics of the transmission channel. While the FEC technique is generally more suitable for uniformly distributed bit errors, the ARQ technique is optimal for large error bursts, which can hardly be corrected by FEC.

Both the packet length and the BER determine the packet error rate (PER) according to Equation (1). Thus, adaptation is also required when the slot scheduler adapts its packet size in order to minimise the number of transitions. In fact, the Slot Scheduler and the Traffic Control need to work in concert to optimise the overall frame structure.

In our system the channel status information will be gathered by the receivers and forwarded to the Slot Scheduler at the base station. The scheduler determines then, incorporating the QoS requirements of the individual connections, and the observed error rate the changes that have to be applied.

Error control can be applied at multiple layers in the communication protocol stack (see Section 5.5). In our system we apply different error-control techniques at the various layers of the protocol stack. Figure 11 shows the error protocol stack of our system.

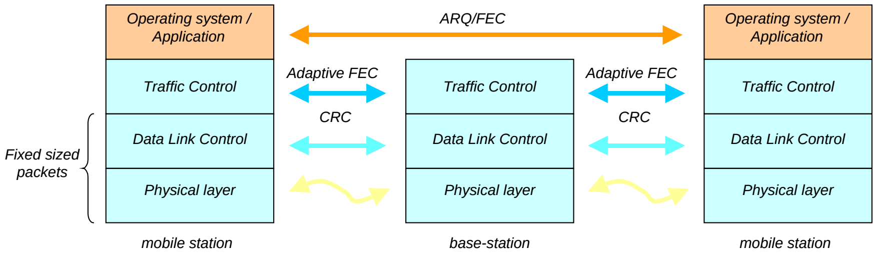

Figure 11: Error control protocol stack.

We do not design the physical layer, but concentrate on the higher layers. At the physical layer we assume that there will be some error correction. This, however, provides less than perfect protection and some amount of residual errors pass through.

At the Data Link Layer we use the E^{2}MaC protocol. An essential property of this MAC protocol is that it uses fixed-length frames of multiple fixed-length slots. This property allows the network interface to power-on their radio precisely when needed. It also simplifies the design of the data link control. The consequence, however, is that we cannot apply efficiently adaptive error control at this layer, since adaptive error control will change the size of a cell. Depending on the quality of the radio device that is being used, the Data Link Control Layer can use a fixed Forward Error Correction to reduce the number of corrupted cells that were caused by random noise. In our current implementation we only added a one byte CRC to each cell. This CRC is used by the Traffic Control to detect corrupted cells.

Traffic Control is able to apply adaptive error control since it operates with multiples of cells. The error correction mechanism then operates on relative large blocks. Any block error correction mechanism could be used. Generally, block codes such as Bose, Chaudhuri and Hockuenghem (BCH) and Reed-Solomon codes require a decoder capable of performing arithmetic operations in finite fields [51]. A comparison between application-specific integrated circuit (ASIC), FPGA, and digital signal processing (DSP) implementations of the decoder shows that the performance of FPGA-based designs lean more toward that of ASICs, but retain flexibility more like DSPs [11][28]. Unfortunately, good VLSI designs for codes using BCH or Reed-Solomon codes do not map well to FPGAs [4]. A code that does not require finite-field arithmetic, but basically only exclusive-OR operations, is the EVENODD code [8]. The EVENODD code was originally designed for a system of redundant disks (RAID). We have studied the EVENODD error correcting mechanism, and compared it with Reed-Solomon in Appendix A.

In the Traffic Control of the network interface we monitor the condition of the wireless channel of the receiver on three ways. The first method is to monitor the number of corrupted cells using the CRC of each cell. Second, we monitor the rate of corrupted cells that the Traffic Control was not able to correct. The last method is to use the information that is provided directly from the radio hardware. The measured channel condition is returned to the transmitter such that the adaptive mechanisms there can make a determination of how to format outgoing packets. The status information that is gathered from these methods is forwarded to the Slot Scheduler at the base station (either using a special field in each uplink packet if the status originates from a mobile, or via an internal connection if the status originates from the base station). The Slot Scheduler can then decide to adapt the error control and simultaneously adapt the assigned bandwidth of a connection to the required fault-tolerance. The modification of the error-control parameters needs to be done synchronously at the base station and the mobile. The slot scheduler therefore indicates the error coding that should be applied for a connection in the traffic control slot that is transmitted in each frame.

Depending on the application, the adaptation might not need to be done frequently. If for example the application is an error-resilient compression algorithm that when channel distortion occurs, its effects will be a gradual degradation of video quality, then the best possible quality will be maintained at all BERs [56].

Note that with adaptive error control the energy efficiency is increased, but it cannot guarantee a reliable connection. Higher level protocols in the operating system or in the application are needed to ascertain this, if required. End-to-end error control has potentially better knowledge of the quality requirements of the application (see also Section 5.5).

To ensure a reliable operation, a confirmed service for the control protocol might be needed as well. This already indicates that adaptive error control introduces a significant increase in complexity. More research needs to be done to find a feasible implementation with low complexity and high efficiency. Simplifications in which only a minimal set of error-control mechanisms is used might quite well turn out to be the most optimal solution.

Avoiding bad periods – Above these error control adaptations, the slot scheduler can also adapt its scheduling policy to the error conditions of wireless connections to a mobile. The scheduler tries 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 more bandwidth when these have better conditions. 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 [16].

To ensure long-term fairness a special mechanism can be used that gives *credits* to connections that are not scheduled due to their error conditions. If a mobile is in error-state, the slot scheduler then adds credits for the appropriate connections. This credit mechanism is not applied to real-time traffic, since stale packets will be dropped. When the error state conditions become better, the slot scheduler schedules the aggregate credits to slots for these connections.

5.6.8 Application interface

Multimedia applications typically communicate multiple streams of data with different types and QoS requirements. If multimedia applications want to achieve optimal performance in an efficient way, they must be aware of the characteristics of the wireless link. Simply relying on the underlying operating system software and communication protocols to transparently hide all the peculiarities of a wireless channel compromises energy consumption and achievable QoS.

By providing the application feedback on the communication, the application can take advantage of the peculiarities and the different data streams over the wireless link. The quality of service over the wireless link and the required energy consumption can be optimised by selecting appropriate parameters for the network interface and network protocols, and by adapting the data-streams.

Recent developments on the internet show streaming audio/video players (like RealPlayer [63]) that dynamically change the frame rate when available network bandwidth changes. The application notes these changes implicitly, i.e. the application senses that available bandwidth is too low because it gets data too late. Actually, only scaling down frame rates is automatic. Once an appropriate lower frame rate is chosen it will not be changed back when more bandwidth becomes available, as this cannot be noted implicitly.

When the link status is available to the applications, scaling in both ways becomes possible. As bandwidth can be used only once, it is better to have one authority that divides it instead of having for example two applications that note an increase in bandwidth and both of them start negotiating higher frame rates with the other end of their transmission. In the end this should average out, but a lot of energy will be wasted before changes settle. The operating system seems to be the right place to put the authority.

Although current audio and video codecs may not benefit from the information, the network interface can make notifications of interesting events. Examples are: 1) the bandwidth dropping below a certain level and 2) the latency in transmission of the last *x* frames being below a certain limit. When 1) is noted, the codec might drop sample rate accordingly or in case of video maybe even switch from color to black and white. In the case that 2) occurs the application could decide to do less buffering, which is more energy efficient.

Once new codecs that allow fast switching of resolution, frame rate, color/black and white become available, mobiles can take advantage of these notifications from the network interface. When mobile power reduction is taken seriously, these news codecs will emerge as chips with a billion transistors allow implementing them.

The system needs some mechanism in the operating system to tie hardware and user applications together. The MOBY DICK Project uses *Inferno* from Lucent Technologies

Bell Labs [20]. In Inferno communicating programs are multithreaded by nature. The mechanism to notify applications of hardware triggers is implemented as an entry point in the namespace of each application through which messages can simply be transferred. Threads block while reading from a channel until the other end writes a message.

To clarify the idea, here is a small code example of a video transmitter that can generate both color frames and black/white frames:

```
....
x:=sys->open(".../connctl",OREAD);    # open x as a control channel
spawn netwatcher(chan x,ref usecolor);# start a netwatcher thread
while not eof(v_in)                   # while not end of video stream
       generateandsendframe(v_in,usecolor); # send video frames
....

void
netwatcher(chan control,ref int usecolor) {
   while (1) {
         msg := <- control;           # read control msg from channel
         "parse message";
         usecolor = 0 or 1 depending on contents of message;
   }
}
```

5.6.9 *Implementation*

We have implemented a test-bed of the network interface that we can use to experiment with the various techniques and mechanisms for e.g. error control and MAC protocol. It is build with off-the-shelf components to allow a short design cycle.

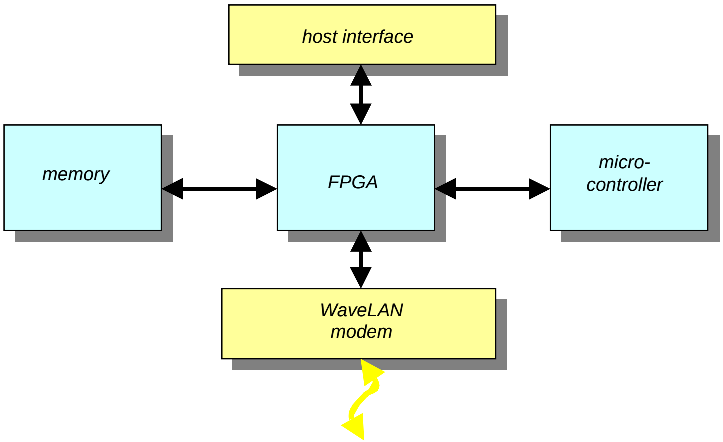

Figure 12: Network interface test-bed architecture.

Figure 12 shows the architecture of the network interface test-bed. The three basic components are:

- *memory* (512 kBytes SRAM) that will be used to implement the connection queues. The amount of buffering that is actually needed depends on the applied error control. Since retransmission is to be implemented by the applications (modules) we just need to have enough buffering to implement an error correction mechanism that is able to correct a small number of cells per connection[6].
- An *FPGA* (Xilinx XC4010) controls the dataflow between the radio and the host and provides basic error detection and error correction functions.
- A *microcontroller* (PIC 16C66) implements the Traffic Control and the Data Link Control (see Section 5.6.6). It controls the functions to be performed by the FPGA, controls the radio modem, and does the power-management. The queues that are stored in the memory are controlled by the microcontroller. It performs the control operations to setup, maintain and release the queues. It collects the status information of the queues and the radio, and transfers this to the QoS manager and slot scheduler in the base-station. Besides these basic functions it further provides miscellaneous operations like initialising the radio modem and gathering status information about the quality of the radio channel.

Figure 13 shows a photograph of the network interface implementation.

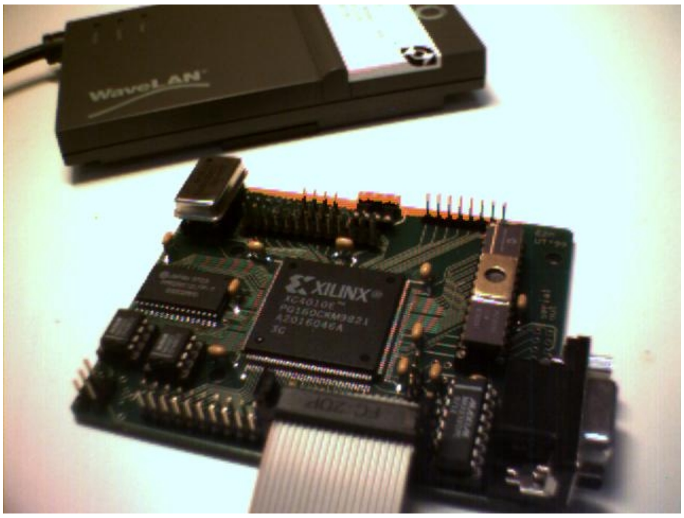

Figure 13: Network interface implementation.

We use a WaveLAN modem as the physical layer. The WaveMODEM is a RF module that converts a serial transmit data stream from the host into Radio Frequency (RF) modulated signals. When the RF signal is received, the RF signal will be demodulated into a serial data stream to the host. The raw data rate is 2 Mb/s. The WaveMODEM

[6] Note that this assumes that the connections have a guaranteed throughput (to the modules, and over the wireless channel).

operates half duplex, i.e. the modem is either transmitting or receiving. The modem provides the basic functionality to send and receive frames of data. It does not include a Medium Access Control Protocol, but provides signalling information like carrier sense.

The FPGA controls the data-flow between the radio and the host. It uses the memory to implement the queues. The FPGA does not perform any control-type operations, it just follows the instructions given by the microcontroller. The microcontroller controls the traffic-flow from the radio and from the host. It therefore performs the queue setup and administration. This administration is used to setup VCI mapping tables and queue address maps in the FPGA. It thereby uses the connection type to determine which flow-control and error control to use. It receives control messages (from either the base-station via the Traffic Control Slot, or from the host) when new connections have to be initialised, changed or removed, and when data has to be received or transmitted.

Cells arriving from the host can be protected against errors that might occur on the wireless channel by adding redundant cells. The FPGA provides the computation-intensive functions of the error control, that the microcontroller can use to build the packet that is protected by the required error correcting code. In this case, the FPGA works alongside a microcontroller that implements the remainder of the error control algorithm.

When the network interface has received traffic from the wireless link, it forwards this to the host using a previously established connection. On reception of an ATM cell, the FPGA simply looks up the VCI mapping table and the queue memory map to determine where to store the cell.

While transferring a cell to memory that originates from the wireless link, it performs *error detection* using a CRC check. Errors are reported to the microcontroller that can determine to initiate error correction on the received packet. Just like the error encoding mechanism, the error correction is being performed in a close collaboration between the FPGA and the microcontroller. The basic compute intensive operations are being performed by the FPGA, and the irregular control functions are being performed by the microcontroller.

The WaveLAN modem has a raw bandwidth of approximately 4830 ATM-cells per second. When we have a frame-rate of 100 Hz, then each frame is about 48 ATM cells large. The memory is capable to store 8000 64-byte cells, which is equivalent to about 1.6 seconds of continuous traffic.

5.6.10 Wireless communication with multiple radio's

The mobiles are expected to spend most of their time in sleep modes. This, however, also implies that they can neither transmit nor receive radio transmissions most of the time. As discussed before, a main source of unessential energy consumption is due to the costs of just being connected to the network. In E^{2}MaC we have tried to minimise this overhead, but still the receiver has to be switched on from time to time, just to discover whether the base-station has some messages waiting.

Another means of discovery is to use a low power RF detection circuit to wake the mobile out of sleep mode. Such a circuit can be quite small, but cannot be used to transfer bits. We could use a very low power receiver for the signalling only. This receiver can be used to wake-up a mobile and transfer connection setup requests or connection queue status information from the base station. It uses the same synchronisation mechanism between mobile and base-station, but 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.

Note that the energy per bit transmitted or received tends to be lower at higher bit rates. For example, the WaveLAN radio operates at 2Mb/s and consumes 1.8 W, or 0.9 μJ/bit [79]. A commercially available FM transceiver (Radiometrix BIM-433) operates at 40 kb/s and consumes 60 mW, or 1.5 μJ/bit [61]. This makes the low bit rate radio less efficient in energy consumption for the same amount of data. However, there is a trade-off when a mobile has to listen for a longer period for a broadcast or wake-up from the base station, then the high bit rate radio will consume about 30 times more energy than the low bit rate radio. Therefore, the low bit rate radio must only be used for the basic signalling, and as little as possible for data transfer.

Another method to increase energy efficiency might be achieved by providing adequate support for broadcast or multicast. Energy can be saved when mobiles do not need to request a certain datum separately, but when the base station transmits it as a broadcast.

5.7 Evaluation of the E^{2}MaC protocol

The E^{2}MaC protocol is designed to provide QoS to various service classes with a low energy consumption of the mobile. The base-station which has 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^{2}MaC protocol then offers the tools to the system to reduce the energy consumption that is needed for the wireless interface.

In the design of the protocol all main principles of energy efficient MAC design are used (see Section 5.3). 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. In this section we will show how these principles are used in the E^{2}MaC protocol and evaluate the attainable gain in energy reduction.

We define the energy efficiency *e* as the energy dissipation needed to transfer the a certain amount of data (e.g. a packet) divided by the total energy dissipation used for that.

$$e = \frac{\textit{Energy dissipation to transfer a certain number of bits}}{\textit{Total energy dissipation}} \qquad (3)$$

5.7.1 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 Slot (TCS) 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 on time. Note that the mobile does not need to receive the TCS of each frame.

After a connection has been set up, and the mobile has no data to send, it can simply tell the base-station that it will sleep for some time. The time is determined by the QoS of all connections of the mobile. 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 station will let the connection sleep again for the same period. This mechanism allows the mobile to sleep for a long period, and still be certain that it can acquire a slot within a bounded time. In this way the mobile reserves periodically a slot in a frame, and the bandwidth spent depends on the tolerable delay.

A mobile that just has to listen if there is *downlink traffic* waiting at the base-station wasts much energy. The E^{2}MaC protocol therefore tries to minimise the amount of energy needed by broadcasting such information in the Traffic Control slot of a frame. It is assumed that the mobile and base station can keep in sync for a reasonable time and thus can turn on their receiver just in time to receive the Traffic Control Slot. The moment at which the receiver has to be turned on depends on the accuracy of the clock. This allows a mobile to sleep when for some time a connection is not used. When the mobile wakes up, and the synchronisation with the base station has become unreliable, then it needs to scan for the TCS. The cost 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^{2}MaC protocol there is always an inevitable overhead due to the *traffic control*. In the E^{2}MaC 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, is selfish behaviour since it reduces the freedom of the slot-scheduler. Only when the load on the wireless channel is moderate, is the scheduler 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.

5.7.2 Minimise the number of transitions

The number of *transitions* between transmitting, receiving, idle, sleep, and off is minimised by the system. 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. There are basically three effects that contribute to the required energy for a transition from sleep to transmission:

1. the required time and energy to change the power mode from sleep to idle. For example, the WaveLAN MODEM interface will become stable and operative within 250 μs after it was signalled to wake-up from sleep mode [79].
2. the required time and energy the interface has to be in idle and transmission mode, but not transmitting actual data. This is the overhead required to initiate and terminate the actual transmission. This time includes the required gap (guard time), interfacing delay, preamble, and afterwards the postamble. Also, as an example, for the WaveLAN interfaces this can take 466 bits per frame.
3. the required time and energy to enter the sleep mode after transmission (WaveLAN documentation does not specify this).

These effects greatly influence the required energy for the transmission of a packet. 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 required to power-up is thus already more than the transmission of one ATM cell.

We will first evaluate in paragraph A the consequences on channel efficiency and energy consumption of the mobile grouping mechanism used in the E^{2}MaC protocol. We will compare this with phase grouping as commonly used in other MAC protocols. Then we will evaluate in paragraph B the consequences of the packet size being used.

A. Overhead

The maximal throughput of the network is determined by 1) the required guard space and physical overhead between slots, 2) the overhead in transmitting control information, and 3) by error control. The transition-overhead (see previous paragraph) to wake-up after a sleep can be done in parallel with a different communication stream and does not influence the throughput of the network. Higher-level protocol issues that might reduce the throughput (like reservation of bandwidth for e.g. mobility or error control) are not considered here.

We assume the wireless physical header and trailer to be a fact that cannot be changed or improved with a MAC protocol, although the protocol can try to minimise the number of times that these are required. Grouping of uplink and downlink traffic of one mobile

(*mobile grouping*) implicates that there is some space between sending and receiving to allow the transceiver to switch its operating mode from sending to receiving (i.e. guard space, preamble, postamble). This has a negative effect on the capacity of the wireless channel. The advantage is that it allows the mobile (i.e. the radio device) to turn its power off for a longer period, and that it makes less power-state transitions. If we would, in contrast to the mobile grouping of uplink and downlink traffic, group the downlink traffic from the base-station to all mobiles (*phase grouping*), then the space between sending and receiving that is required for mobile grouping, is not present for the downlink traffic (see Figure 8). Most MAC protocols group the traffic from the base-station, mainly because of its efficient use of the available bandwidth. However, there are consequences of phase grouping related to the energy consumption:

1) The receiver of the mobile must be on for a longer period (i.e. during the whole downlink period because it needs to synchronise using the preamble of the radio packet). If the radio would be capable of synchronising during the transmission of a downlink packet, then the mobile might be able to power down during the downlink phase. However, this will still cause an additional energy consuming power-state transition (from power-down/sleep/idle mode to an active transmission mode.

2) The period between two operations is too small to enter sleep mode. This period is determined by the time needed to enter sleep mode (T_{sleep}) and the time needed to wake-up ($T_{wake\text{-}up}$).

These two effects lead to higher energy consumption for the mobile. This shows that there is a trade-off between performance (channel efficiency) and energy consumption. The energy gained with mobile grouping depends on 1) the amount of data in the downlink phase that is not destined to the mobile, but must be received by the radio device because it is stored in the downlink transmission frame[7]. And 2) the amount of time between receiving the downlink packet and the uplink packet. Only when this time exceeds the time required to enter sleep mode (T_{sleep}) plus the time needed to wake-up ($T_{wake\text{-}up}$), then energy can be saved. Otherwise, the mobile must remain idle, waiting for its time to transmit its uplink packet.

We will now evaluate the effects of mobile grouping on the available bandwidth and on the energy consumption. We will compare this with the phase grouping mechanism. The properties of interest are:

TCS The size of the traffic control slot. In our implementation we use one ATM cell-sized slot.

O The overhead to transmit a packet. The overhead *O* consists of the overhead when the interface must be idle O_{idle} (required for guard space and interfacing delay), plus the overhead O_p required for preamble and postamble. The interfacing delay is caused by two factors. First, the delay caused by the

[7] If the network interface is able to skip packets in the downlink phase that are transmitted after the mobile has received its packet, then the downlink schedule order determines the amount of data to be received

wireless interface to synchronise to its internal syncslots. Second, we have an additional delay because we must also synchronise to the time slots that the MAC protocol uses. The MAC protocol uses fixed time slots, but since each packet is not a multiple of this slot size (because of the overhead in the wireless interface) we need to incorporate a delay with an average length of the size of a time slot divided by two.

T_{sw} This is the time needed by the wireless interface to enter sleep mode T_{sleep} plus the time $T_{wake\text{-}up}$ needed to wakeup from sleep mode to an active mode (idle, receive or transmit).

C The number of bytes used for the collision phase (reservation phase).

O_{total} The total overhead in a frame that is introduced to transmit the actual data over the wireless link.

F The size of a transmission frame.

TD The total size available for a mobile to transmit data packets. This can be expressed with:

$$TD = F - O_{total}$$

D The size of a packet (uplink and downlink) used by a mobile. We assume that the whole frame is used, and that all mobiles have an equal share. The size of an uplink and a downlink packet is thus dependent on the number of mobiles using the frame. It can be expressed with:

$$D = TD / 2M$$

M The number of mobiles, each with uplink and downlink packets.

All properties can be expressed in bytes. When a property is related to time, then we use the virtual overhead that expresses the number of bytes the wireless channel can transmit in that time.

In our analysis we will assume that each mobile has both uplink connections and downlink connections that both have similar bandwidth requirements. We further assume a packet length that allows a mobile to enter sleep mode. Thus, the packet length is greater than T_{sw}.

Evaluation phase grouping

Figure 14 shows a typical phase grouping transmission frame with three mobiles, each using downlink and uplink packets.

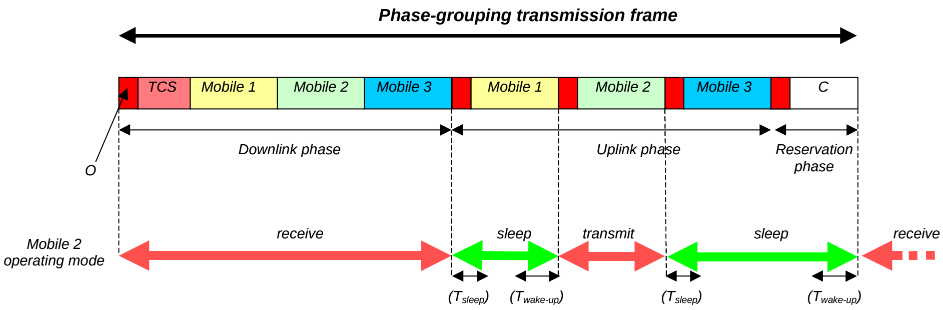

Figure 14: Phase grouping transmission frame

In general we have M mobiles, each with uplink and downlink packets. The total *overhead* O_{total} can be expressed with:

$$O_{total} = O + TCS + M.O + O + C$$

or,

$$O_{total} = TCS + (M{+}2).O + C \qquad (4)$$

We will now determine the total time a mobile can enter sleep mode. This time can be used to evaluate the energy consumption a mobile needs for its wireless interface.

We assume that a mobile is required to receive the whole downlink packet from the base station. Whether a mobile is able to transmit its uplink packet depends on the schedule made by the base station. In our analysis we evaluate the sleep period of a mobile that is scheduled as second in the uplink phase (e.g. Mobile 2 in Figure 14). We can then divide the uplink period in three phases: pre-uplink, uplink (in which the mobile transmits its data), and post-uplink.

When there is just one mobile, then we do not have a pre-uplink phase. The mobile can only sleep in the contention phase. The total sleep time T_{sleep} of the mobile in this situation is:

$$T_{sleep}\,(M{=}1) = O + C - T_{sw}$$

When there are more mobiles, then we have all phases. The pre-uplink sleep period is:

$$T_{sleep\text{-}pre} = (D + O) - T_{sw}$$

The post-uplink sleep period is during the remaining (M-2) data packets from the other mobiles:

$$T_{sleep\text{-}post} = (D + O)\,.\,(M{-}2) - T_{sw}$$

Together with the collision phase this gives a total sleep time for $M > 1$:

$$T_{sleep}\,(M{>}1) = (D + O) - T_{sw} + (D + O)\,.\,(M{-}2) + O + C - T_{sw}$$

Thus:

$$T_{sleep} = \begin{cases} O + C - T_{sw} & M = 1 \\ (D + O) \,.\, (M\text{-}1) + O + C - 2\,T_{sw} & M > 1 \end{cases} \qquad (5)$$

Evaluation mobile grouping

We will now evaluate mobile grouping using the same assumptions as applied for phase grouping. Figure 15 gives an example of a mobile grouping transmission frame.

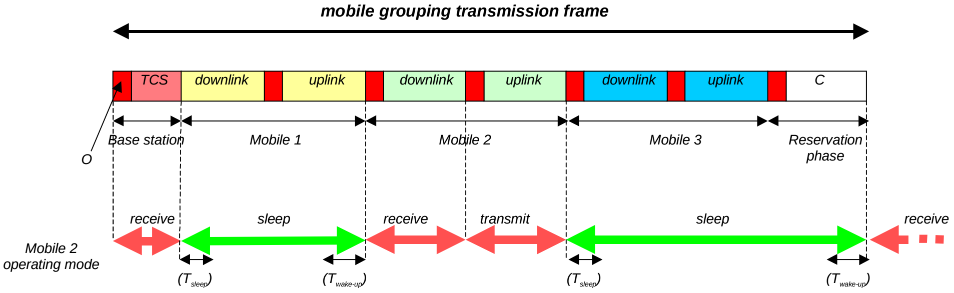

Figure 15: Mobile grouping transmission frame

In general we have M mobiles, each with uplink and downlink packets. The total overhead O_{total} can be expressed with:

$$O_{total} = O + TCS + O + 2M.O + O + C$$

or,

$$O_{total} = TCS + (2M{+}3).O + C \qquad (6)$$

We can divide the uplink period in three phases: pre-uplink, uplink (in which the mobile transmits its data), and post-uplink.

When there is just one mobile, then we do not have a pre-uplink phase. The mobile can only sleep in the contention phase. The total sleep time T_{sleep} of the mobile in this situation is:

$$T_{sleep}\,(M{=}1) = O + C - T_{sw}$$

When there are more mobiles, then we have all phases. The pre-uplink sleep period is:

$$T_{sleep\text{-}pre} = (2\,D + O) - T_{sw}$$

The post-uplink sleep period is during the remaining (M-2) data packets from the other mobiles:

$$T_{sleep\text{-}post} = 2\,(D + O)\,.\,(M\text{-}2) - T_{sw}$$

Together with the collision phase this gives a total sleep time for $M > 1$:

$$T_{sleep}\,(M{>}1) = (2\,D + O) - T_{sw} + 2\,(D + O)\,.\,(M\text{-}2) + O + C - T_{sw}$$

Thus:

$$T_{sleep} = \begin{cases} O + C - T_{sw} & M = 1 \\ (2M-2)\,D - 2\,T_{sw} + (2M\text{-}2)O + C & M > 1 \end{cases} \qquad (7)$$

We will now apply the characteristics of the WaveLAN modem to these equations.

F 2544 bytes. (The transmission rate is 2 Mb/s. When we use a frame rate of 100 Hz, then the frame size is approximately 2544 bytes.)

TCS 53 bytes. (one ATM cell)

O 71 bytes. (O_p = 37 bytes, O_{idle} = 22 bytes. The internal time slots are 24 bytes, which results in an average synchronisation delay of 12 bytes)

C 53 bytes. (one ATM cell)

T_{sw} 73 bytes (T_{sleep}=10 bytes (unspecified by specs), $T_{wake\text{-}up}$=63 bytes)

The results are shown in Figure 16 and Figure 17.

Figure 16 shows the total overhead O_{total} caused by the two mechanisms. As expected, phase grouping induces less overhead than mobile grouping. When there are many mobiles using the frame (both in the uplink and in the downlink direction), then the overhead constitutes a significant part of the total available bandwidth. We can increase the frame size by lowering the frame rate frequency of 100 Hz. This would reduce the overhead that is required to transmit a certain amount of data, but will increase the latency.

Also shown in the figure is the packet size D (under the assumption that the whole frame is used). This clearly shows that the packet size D becomes rather small when the number of mobiles using the frame increases. The packet size of mobile grouping is a little bit smaller than of phase grouping because of the larger overhead.

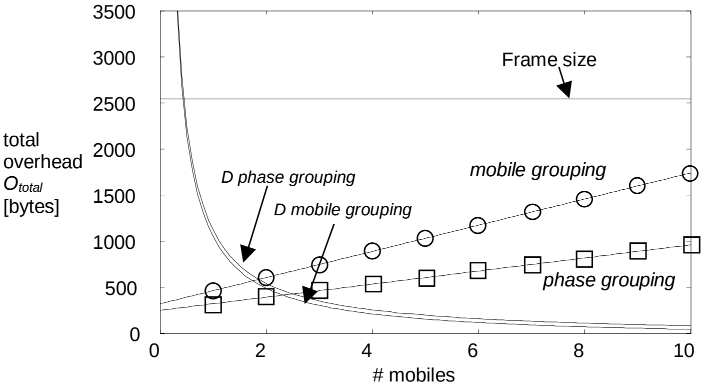

Figure 16: Total overhead versus number of mobiles.

When we look at the consequences for energy consumption, then mobile grouping is more advantageous. This is shown in Figure 17. The sleep period per mobile is larger when using mobile grouping compared to phase grouping.

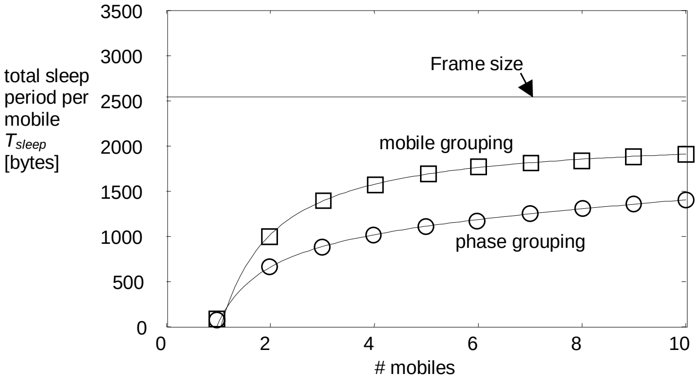

Figure 17: Total sleep period per mobile versus number of mobiles.

The figure show that the increase in the total sleep period is already high with a small number of mobiles. As the overhead increases with the number of mobiles using the wireless channel, mobile grouping seems particularly attractive for systems with a small cell size (e.g. pico-cellular with the size of an office-room). In these systems the number of mobiles in one cell will in general be small, and the available bandwidth high. Mobile grouping strategy will then have a small overhead while allowing a large sleep period.

Note that the assumptions we have made are conservative for two reasons. First, we assumed that the whole frame is used. If this is not the case, then the sleep period can be larger. However, when using phase grouping, a mobile is in general forced to receive the whole downlink packet, and cannot enter sleep mode in that phase; whereas mobile grouping only needs to receive the TCS. Second, we assumed that the amount of uplink traffic is equal to the amount of downlink traffic. This might be true for voice applications (mobile phone), but is in general not true for applications running on a mobile computer. For these applications the downlink traffic in general will use more bandwidth. The disadvantage of having one large downlink packet (phase grouping) then becomes even more apparent.

The overhead caused by the transmission of the traffic control (TCS) 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 20 shows the effect of the frame size on the energy efficiency versus the load per mobile. The overhead for error control (i.e. to transmit a CRC and redundant data) also reduces the throughput. The required guard space between slots influences the throughput, but has no effect on the energy consumption. The size of the guard space depends on the hardware of the transceiver.

B. Packet size

Figure 18 shows the effect on the energy efficiency with respect to the packet size *s* and the overhead introduced with one transition. The overhead *O* that is required to transmit a packet consists of the virtual overhead O_v to wake-up from sleep mode, plus the overhead when the interface must be idle O_{idle} (required for guard space and interfacing delay), plus the physical overhead O_p that is required before the interface can transmit the actual data. The energy consumption when transmitting is E_{tx}, when waking-up $E_{wake-up}$, and when idling E_{idle}. The energy efficiency e_{packet} of transmitting one packet of size *s* is then:

$$e_{packet} = \frac{E_{tx}.s}{E_{tx}.s + E_{wake\text{-}up}.O_v + E_{idle}.O_{idle} + E_{tx}.O_p} \qquad (8)$$

We use a simplified energy model in which the energy consumption is equal in all states (waking up, idling and transmitting). For WaveLAN we have O_v = 62.5 bytes, O_{idle} = 21.25 bytes, and O_p = 37 bytes, which gives to a total overhead of 120 bytes (2.27 ATM cells).

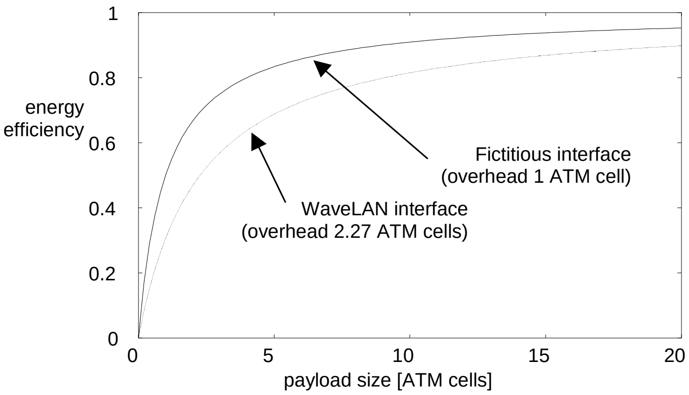

Figure 18: Energy efficiency vs. payload size as a function of overhead.

Figure 18 shows that the packet size has a big influence on the energy efficiency. It also shows the energy efficiency if we would have used a fictitious interface that has an overhead of one ATM cell. Small packet sizes are not efficient because the total overhead is large. So, when the protocol bundles communication in bursts from one mobile, much energy can be saved when compared with a scheme that for example requires two transitions per ATM cell (i.e. change the mode from idle to transmission, and back to idle).

The discussion above shows that the large transition times of a wireless interface make a large packet size profitable. However, this is valid for ideal situations in which no errors occur only. According to Equation (1) the packet error rate (PER) depends on the bit error rate (BER) and the packet size. The overhead imposed is caused by two main factors: overhead in time (power up, power down, inter-frame gap (which is the required guard space between two transmissions), transmission mode transitions) and overhead in bits transmitted over the air (preamble, MAC control header, postamble).

When we consider the *goodput,* which is the throughput a user will see, as a function of BER and packet size, then we only need to incorporate the overhead where errors influence the transmission. Since power up/down and transmission mode transitions of different mobiles can occur in parallel in time, they do not influence the goodput as well.

When we study the WaveLAN modem characteristics that are also depicted in Figure 6, then we can specify various quantities of interest. Let:

I = inter-frame guard space, 15.5 bytes

P = length of preamble (36.5) plus postamble (0.5), 37 bytes

M = length of MAC control header, in E^{2}MaC, 48 bytes

D = number of data bytes

The goodput g normalised to the raw bit rate of the radio can be specified as:

$$g = \frac{D}{I + P + M + D} \quad (1 - BER)^{D + M} \qquad (9)$$

We have plotted this equation with the goodput *g* versus packet size *s* for various bit error rates in Figure 19. This figure clearly shows that when the channel conditions are bad, large packet sizes lead to a low goodput. If the QoS of the connection requires a better goodput, then the error control mechanism has to be adapted, or the packet size has to become smaller. The issues of packet size and error control coding are intertwined, since the amount and kind of coding needed will depend on similar factors as with packet sizing.

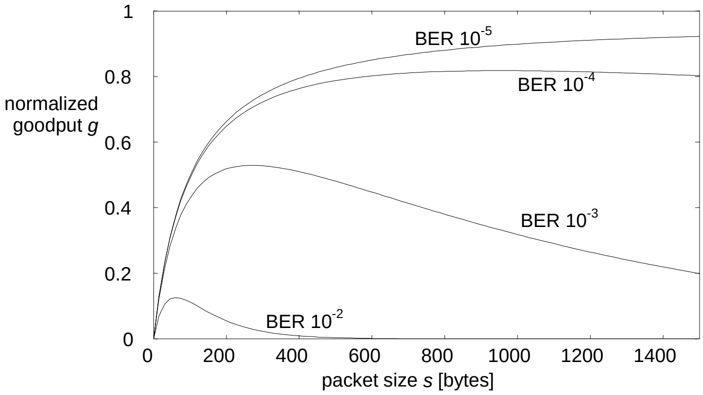

Figure 19: Goodput vs. packet size on WaveLAN for various BER.

There seems to be a trade-off concerning the packet size between energy efficiency (minimal transitions thus large packet size) and goodput (adequate packet size, not too large). However, the Traffic Control module can also adapt the error control mechanism, such that it divides the packet into smaller segments that each has their own error control. Both possible adaptations (either more redundancy on large packets or several smaller packets with less redundancy) require extra energy that is needed for the error control.

5.7.3 Avoid unsuccessful actions

In our approach unsuccessful transfers are minimised because the chance of a collision is small and the base station tries to avoid periods of bad error conditions.

- *Errors*

The error control is applied on individual connections and is tailored to the traffic type and required QoS of the connection. This structure, combined with an adaptive error control allows for an error control scheme that does not perform error control when it is not needed, but on the other hand does not give a too low reliability for a connection.

The scheduler at the base-station plays an active role in the error control. It not only determines the required error coding for each connection, it also tries to avoid periods of bad-error conditions by not scheduling non-time critical traffic during these periods. How profitable this latter approach is, depends on factors like the typical size of an error burst and on 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 amount of energy that otherwise would have been wasted 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.

- *Collisions*

The chance of a *collision* in the E^{2}MaC 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 this section we will compare the energy efficiency of a mobile with *uplink traffic* using the E^{2}MaC protocol and Slotted Aloha, a collision based protocol that is often used as a reference and is also used in many systems as the basis of the access mechanism. 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 the access mechanism in Slotted Aloha

In Slotted Aloha, time is divided into slots [2]. 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. Therefore, a backoff procedure is of no concern in the analysis of energy consumption [48]. The energy dissipated is determined by the time that the transmitter and receiver must be 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 π that it is successfully received by the base station is:

$$\pi = (1 - p)^{n-1} \qquad (10)$$

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. The average number of transmissions υ needed to send a message successfully is given by:

$$\upsilon = \frac{1}{(1 - p)^{n - 1}} = (1 - p)^{1 - n} \qquad (11)$$

Every time the mobile attempts to send a message, the receiver is switched on to receive possible positive acknowledgements. Using υ, the average time T_{tx} that the transmitter is active for a successful packet transmission is determined to be:

$$T_{tx} = \upsilon \,.\, T_{data} \qquad (12)$$

Similarly, the time T_{rx} per packet that the receiver is switched on in order to receive an acknowledgement is given by:

$$T_{rx} = \upsilon \,.\, T_{ack} \qquad (13)$$

The total energy dissipation is given by the time that the transmitter is on, plus time that the receiver is turned on to receive the possible acknowledgements, multiplied by the power dissipations of each of these functions.

The energy efficiency e_{sa} is thus determined by:

$$e_{sa} = \frac{T_{data} \,.\, P_{tx}}{T_{tx} \,.\, P_{tx} + T_{rx} \,.\, P_{rx}} \qquad (14)$$

in which T_{data} is the time to transmit one packet, T_{tx} is the time to successfully transmit a packet, T_{rx} the time to receive the acknowledge, P_{tx} the power dissipation for transmission and P_{rx} the energy dissipation for reception.

Energy efficiency of the access mechanism in E^{2}MaC

The access mechanism used in E^{2}MaC is based on a TDMA structure where the base station assigns time (slots) to mobiles in which they are allowed to transmit. Since in the E^{2}MaC 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, the overhead is determined by the reception of the traffic control.

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. When not each frame is used by the connections of a mobile, then the mobile does not need to receive the TCS either. We will only analyse the worst case in which a mobile needs to receive the traffic control once per frame.

The time that the receiver has to be on per frame to receive the Traffic Control Slot is determined by the number N of slots in a frame. Since no collisions can occur, acknowledgements are not needed on this level. The energy efficiency of the access mechanism e_{e2mac} is thus:

$$e_{e2mac} = \frac{p \,.\, T_{tx} \,.\, P_{tx}}{p \,.\, T_{tx} \,.\, P_{tx} + (T_{rx} \,.\, P_{rx}) / N} \qquad (15)$$

where N is the number of slots in a frame, p the probability that a mobile sends a packet in a frame, T_{rx} is the time needed to receive the traffic control slot, T_{tx} the time to transmit the packet, P_{tx} the power dissipation for transmission and P_{rx} the energy dissipation for reception.

Comparison

In our analysis we will assume that the energy consumption for transmission is equal to reception, thus $P_{tx} = P_{rx}$. This approximates the power consumption characteristics of the WaveLAN 2.4 GHz modem [79]. 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 will use $T_{rx} = 1/8\ T_{tx}$.

Figure 20 shows the energy efficiency characteristics of Slotted Aloha for a various number of mobiles, and for E^{2}MaC for two frame-sizes.

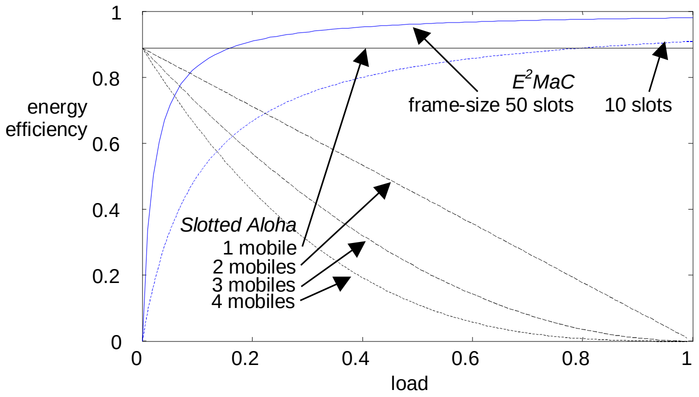

Figure 20: Energy efficiency vs. load for uplink traffic on Slotted Aloha and E^{2}MaC.

It is difficult to make a fair comparison for several reasons. Slotted Aloha requires an explicit acknowledgement as part of the MAC protocol. In E^{2}MaC no acknowledgements are required at that level, but there will be acknowledgements at higher layers of the protocol stack. In our comparison we will only incorporate MAC level issues.

Further, the energy efficiency of the E^{2}MaC protocol is independent of the activity of the users, in contrast to Slotted Aloha where the efficiency strongly depends on the activity of other users. Therefore, the indicated load for Slotted Aloha is the total load in a cell, and for E^{2}MaC it is the load per mobile.

The energy efficiency of the E^{2}MaC 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 communicating with a small packet size at a low rate), then the energy consumed by the E^{2}MaC 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, with Slotted Aloha, when 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^{2}MaC 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 $P_{tx} > P_{rx}$. This, however, has little influence on the characteristics, and the conclusions remain the same.

5.8 Related work

In 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 focused on supply voltage and circuit design [54]. 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.

Lettieri [45] shows 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 inspired by the high error rate on wireless links, where a high error rate on a large frame might not be efficient. A similar effect is reached when the error control would adapt to the current error condition of the radio channel. This is the approach that we have taken in E^{2}MaC. A similar point of view to reduce energy consumption (i.e. incorporating the error condition of the radio channel) is used in [16]. They try to avoid transmission during bad channel periods in order to reduce the number of unsuccessful transmissions. Both protocols, however, lack QoS provisions. In E^{2}MaC 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 [41] 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. The mobiles have to be synchronised to wake up at the same time the base station announces buffered frames for the receiver. Afterwards the mobiles request the frame from the base station. This mechanism 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 since any packet is being transmitted using a CSMA-like technique, and collisions can incur an indefinite delay.

HIPERLAN [25] is the wireless LAN specified by the ETSI. Its energy saving is based on two mechanisms: a dual data rate radio, and buffering. Because HIPERLAN is based on a broadcast channel, each station needs to listen to all packets in its range. To decide whether the station is the destination of a packet, each packet is divided into a low-power low bit-rate (1.4706 Mb/s) part to transmit acknowledgement packets and the packet header, and a high power high bit-rate (23.5294 Mb/s) part to transmit the data packet itself. HYPERLAN does not need a dedicated base station, but any station can become a so-called forwarder. Forwarders use a forwarding mechanism to build the infrastructure. The physical size of a HIPERLAN is thus a function of the current position of all stations. Power saving is based on a contract between at least two stations. The station that wants to save power is called the power-saver, and the station that supports this is the power-supporter. Power-supporters have to queue all packets destined for one of its power-savers. Forwarders and power-supporters are not expected to be mobiles since they have to receive, buffer, and forward packets sent to one of its clients. The p-saver is active only during pre-arranged intervals. Since this interval is minimal 500 ms, it cannot be used for most time-bounded traffic [81].

The R-TDMA protocol is shown to be energy efficient [48], but this is mainly due to the reservation scheme that is used to provide QoS for real-time connections. This clearly shows the advantage of having a time slotted reservation scheme. Other energy saving techniques are not applied.

The protocol design of the energy-conserving medium access control (EC-MAC) protocol [70][69] is related to the E^{2}MaC 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 E^{2}MaC has with its **sleep** command (so that the mobile does not need to receive all Traffic Control Slots). It further does not apply some of the energy saving mechanisms such as dynamic flow control and dynamic adaptations to varying error conditions. The protocol uses phase grouping of traffic.

The principle of synchronisation between mobile and base station has been used for some time in paging systems [54]. 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 [53] 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, and has no explicit mobile grouping of traffic.

5.9 Conclusions

In this chapter we have first pointed out that separating the design of the protocol from the context in which it exists, leads to penalties in performance and energy consumption that are unacceptable for wireless, multimedia applications. Then, we have presented an architecture of a highly adaptive network interface and 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^{2}MaC protocol are: avoid unsuccessful actions, minimise the number of transitions, and synchronise the mobile and the base-station. We have shown in this chapter that considerable amounts of energy can be saved using these principles. The protocol is able to provide near-optimal energy efficiency (i.e. energy is spent for the actual transfer only) for a mobile within the constraints of the QoS of all connections, and only requires a small overhead. Most of the resulting energy waste comes from the relatively long transition times between the various operating modes of current wireless radio's. Minimising these transition times in future radio designs will be beneficial and will further reduce the energy consumption significantly.

A particular novel mechanism of the E^{2}MaC protocol is the mobile grouping of traffic. This mobile grouping strategy reduces the number of operating mode transitions between transmitting, receiving, active, and sleep, and maximises the possible sleep period of the transceiver. We have made the involved trade-off between performance and energy efficiency in favour of the latter because energy efficiency is one of our main concerns, and because the overhead in our system will be small (because we group all traffic as much as possible and we use small cell size with only a few mobiles per cell). Future work can be found in the development and analysis of wireless scheduling algorithms to provide QoS bounds to the various traffic types that incorporates the energy efficiency principles as determined in this chapter.

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. Therefore, a hybrid MAC protocol that can operate in two modes and that is optimised for both network types will probably be the most efficient.

We have shown that energy-awareness must be applied in almost all layers of the network protocol stack. Instead of trying to save energy at every separate layer, like trying to implement TCP efficiently for wireless links [5], we have shown that applying energy saving techniques that impact all layers of the protocol stack can save more energy. To achieve maximal performance and energy efficiency, *adaptability* is important, as wireless networks are dynamic in nature. Adaptability cannot be effectively implemented in one separate layer. Furthermore, if the application layer is provided with feedback on the communication, advantage can be taken of the differences in data streams over the wireless link. To allow this, feedback is needed from almost all layers: the physical layer provides information on link quality, the medium access layer on effectiveness of its error correction, and the Data Link Control layer on buffer usage and error control. Also, if the transport layer is provided with proper feedback, it can make better differentiation between the needs for congestion control and retransmission.

Migration of some functionality from the mobile, for example to the base-station, allows reduction of the complexity of mobiles. Only a few simple components are now needed for the implementation of the network interface. Added complexity in the base-station or other parts of the fixed network is justified because they can be better equipped and are not battery powered.

The programming paradigm of *Inferno* is well suited for transparent distribution and migration of functionality. Inferno also allows easy implementation of feedback through layers of the network protocol stack up to the level of the applications.

References

[1] Abnous A, Rabaey J.: "Ultra-Low-Power Domain-Specific Multimedia Processors," *Proceedings of the IEEE VLSI Signal Processing Workshop*, San Francisco, October 1996.

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

[3] Agrawal P., Chen J-C, Kishore S., Ramanathan P., Sivalingam K.: "Battery power sensitive video processing in wireless networks", *Proceedings IEEE PIMRC'98*, Boston, September 1998.

[4] Ahlquist G.C., Rice M., Nelson B.: "Error control coding in software radios: an FPGA approach", *IEEE Personal Communications*, August 1999, pp. 35-39, 1999.

[5] Akyildiz I.F., McNair J., Martorell L.C., Puigjaner R., Yesha Y.: "Medium Access Control protocols for multimedia traffic in wireless networks", *IEEE Network*, pp.39-47, July/August 1999.

[6] Balakrishnan H., et al.: "A comparison of mechanisms for improving TCP performance over wireless links", *Proceedings ACM SIGCOMM'96*, Stanford, CA, USA, August 1996.

[7] Bauchot F., Decrauzat S. Marmigere G., Merakos L., Passa N.: "MASCARA, a MAC protocol for wireless ATM", *proceedings ACTS Mobile Summit*, pp. 556-562, Granada, Spain, Nov. 1996.

[8] Blaum M., et al.: "EVENODD: an efficient scheme for tolerating double disk failures in RAID architectures", *IEEE Transactions on computers*, Vol. 44, No 2, pp. 192-201, February 1995.

[9] Birk Y. and Keren Y.: "Judicious Use of Redundant Transmissions in Multi-Channel ALOHA Networks with Deadlines", *proceedings IEEE Infocom'98*, pp. 332-338, March 1998.

[10] 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.

[11] Bowers H., Zhang H.: "Comparison of Reed-Solomon codec implementations", *Technical rep. UC Berkele*y, http://infopad.eecs.berkeley.edu/~hui/cs252/rs.html.

[12] Chen T.-W., Krzyzanowski P., Lyu M.R., Sreenan C., Trotter: "A VC-based API for renegotiable QoS in wireless ATM networks", *Proceedings IEEE ICUPC'97*, 1997.

[13] Chen T.-W., Krzyzanowski P., Lyu M.R., Sreenan C., Trotter: "Renegotiable Quality of Service – a new scheme for fault tolerance in wireless networks", *Proceedings FTCS'97*, 1997.

[14] 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.

[15] Cho, Y.J., Un, C.K.: "Performance analysis of ARQ error controls under Markovian block error pattern", *IEEE Transactions on Communications.*, Vol. COM-42, pp. 2051-2061, Feb-Apr. 1994.

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

[17] Choi S., Shin K.G.: “A cellular wireless local area network with QoS guarantees for heterogeneous traffic”, *Mobile networks and applications* 3, pp. 89-100, 1998.

[18] Ciotti C., Borowski J.: "The AC006 Median Project – Overview and State-of-the-Art", *ACTS Mobile Summit*, Granada, Nov. 96, http://www.imst.de/mobile/median/median.html.

[19] Colombo G., Lenzini L., Mingozzi E., Cornaglia B., Santaniello R.: Performance evaluation of PRADOS: a scheduling algorithm for traffic integration in a wireless ATM network”, *Proceedings of the fifth annual ACM/IEEE international conference on mobile computing and networking (MobiCom’99*), pp. 143-150, August 1999.

[20] Dorward S., Pike R., Presotto D., Ritchie D., Trickey H., Winterbottom P.: “Inferno”, *Proceedings COMPCON Spring’97*, 42nd IEEE International Computer Conference, 1997, URL: http://www.lucent.com/inferno.

[21] Eckhardt D., Steenkiste P.: “Measurement and analysis of the error characteristics of an in building wireless network”, *Proceedings of the SIGCOMM ‘96 Symposium on Communications Architectures and Protocols*, pp. 243-254, Stanford, August 1996, ACM.

[22] Eckhardt D., Steenkiste P.: “A trace-based evaluation of adaptive error correction for a wireless local area network”, *Journal on Special Topics in Mobile Networking and Applications (MONET)*, special issue on Adaptive Mobile Networking and Computing, 1998.

[23] Eckhardt D.A., Steenkiste P.: “Improving wireless LAN performance via adaptive local error control”, *Sixth IEEE International conference on network protocols (ICNP’98)*, Austin, October 1998.

[24] Elaoud, M, Ramanathan, P.: “Adaptive Use of Error-Correcting Codes for Real-time Communication in Wireless Networks”, *proceedings IEEE Infocom’98*, pp. 548-555, March 1998.

[25] ETSI: "High Performance Radio Local Area Network (HIPERLAN)", *draft standard ETS 300 652*, March 1996.

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

[27] Figueira, N.R., Pasquale, J.: “Remote-Queueing Multiple Access (RQMA): Providing Quality of Service for Wireless Communications”, *proceedings IEEE Infocom’98*, pp. 307-314, March 1998.

[28] Goslin G.R.: “Implement DSP functions in FPGAs to reduce cost and boost performance”, EDN magazine, 1996, http://www.ednmag.com/reg/1996/101096/21df_05.htm.

[29] Han R.Y., Messerschmitt: “Asymptotically reliable transport of multimedia/graphics over wireless channels”, *Proc. Multimedia Computing and Networking*, San Jose, Jan. 29-31, 1996.

[30] Haskell P., Messerschmitt D.G.: “In favor of an enhanced network interface for multimedia services”, *IEEE Multimedia Magazine*, 1996.

[31] Haskell P., Messerschmitt D.G.: “Some research issues in a heterogeneous terminal and transport environment for multimedia services*”, Proc. COST #229 workshop on adaptive systems, Intelligent Approaches, Massively Parallel Computing and Emerging Techniques in Signal Processing and Communications*, Bayona, Spain, Oct. 1994.

[32] Havinga P.J.M., Smit G.J.M., Bos M.: “Energy efficient wireless ATM design”, *proceedings second IEEE international workshop on wireless mobile ATM implementations (wmATM’99)*, pp. 11-22, June 1999.

[33] Havinga P.J.M., Smit G.J.M., Bos M.: “Energy efficient wireless ATM design”, to appear in *ACM/Baltzer Journal on Mobile Networks and Applications (MONET), Special issue on Wireless Mobile ATM technologies, Vol. 5, No 2., 2000.*

[34] Havinga P.J.M., Smit G.J.M.: “Low power system design techniques for mobile computers”, *CTIT technical report 97-32*, the Netherlands, 1997.

[35] Havinga P.J.M., Smit G.J.M.: “The Pocket Companion's Architecture”, *Proceedings Euromicro Summer School on Mobile Computing '98*, pp. 25-34, Oulu, Finland, August 1998.

[36] Havinga, P.J.M., “Energy efficiency of error correcting mechanisms for wireless communication”, *CTIT technical report 98-19*, 1998, the Netherlands.

[37] 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.

[38] Hettich A., Evans D., Du Y., Lott M., Fifield R.: “Fast uplink signalling for an ATM radio interface using energy burst with random access”, *proceedings wmATM’99*, pp.167-176, June, 1999.

[39] Huitema, C.: “The case for packet level FEC”, *Proceedings 5th workshop on protocols for high speed networks*, pp. 109-120, Sophia Antipolis, France, Oct. 1996.

[40] Hyden E. A., “Operating System support for Quality of Service, *Ph.D. thesis, University of Cambridge*, 1994.

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

[42] Klein Gebbink J.P.A., Nienhuis M.L.: “An energy efficient wireless Communication system with Quality of Service”, *Ms. Thesis University of Twente*, July 1999.

[43] Kohiyama, K., Hashimoto A.: “Advanced Wireless Access System”, *Telecom’95*, Geneva, October 1995.

[44] Lettieri P., Schurgers C., Srivastava M.B.: “Adaptive link layer strategies for energy efficient wireless networking”, ACM WINET.

[45] 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.

[46] Lin S., Costello D.J. Jr.: “Error control coding: fundamentals and applications”, *Prentice-Hall*, 1983.

[47] Lin, S., Costello, D.J., Miller, M.: “Automatic-repeat-request error-control schemes”, *IEEE Comm. Magazine*, v.22, n.12, pp. 5-17, Dec 1984.

[48] 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.cs.utwente.nl/~havinga/papers/macenergy.ps, 1997.

[49] Liu, H., El Zarki, M.: “Delay bounded type-II hybrid ARQ for video transmission over wireless networks”, *proceedings Conference on Information Sciences and Systems*, Princeton, March 1996.

[50] Lorch, J., Smith, A. J.: ”Software strategies for portable computer energy management”, *IEEE Personal Communications Magazine*, 5(3):60-73, June 1998.

[51] MacWilliams, F.J., Sloane, N.J.A.: “The theory of error-correcting codes”, *North-Holland Publicing Company*, Amsterdam, 1977.

[52] Makrakis D.M., Mander R.S., Orozco-Barbosa L., Papantoni-Kazakos P.: “A spread-slotted random-access protocol with multi-priority for personal and mobile communication networks carrying integrated traffic”, *Mobile Networks and Applications* 2, pp.325-331, 1997.

[53] 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.

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

[55] Mathis, M., et al., “RFC2018: TCP selective acknowledgement option”, Oct. 1996.

[56] Meng T.H., Hung A.C., Tsern E.K., Gordon B.M.: "Low-power signal processing system design for wireless applications", *IEEE Personal communications*, Vol. 5, No. 3, June 1998.

[57] Mikkonen J., Kruys J.: “The Magic WAND: a wireless ATM access system”, *proceedings ACTS Mobile Summit*, pp. 535-542, Granada, Spain, Nov. 1996.

[58] Mikkonen J.: “Wireless ATM overview”, *Mobile Communications International*, Issue 28, pp. 59-62, Feb. 1996.

[59] Moorman, J.R., Lockwood J.W.: “Multiclass priority fair queuing for hybrid wired/wireless quality of service support”, *Proceedings of the second ACM international workshop on Wireless Mobile Multimedia (WoWMoM’99)*, pp. 43-50, August 1999.

[60] Nobelen R. van, Seshadri N., Whitehead J., Timiri S.: “An adaptive radio link protocol with enhanced data rates for GSM evolution”, *IEEE Personal Communications*, pp. 54-64, February 1999.

[61] Nonnenmacher, J., Biersack, E.W.: “Reliable multicast: where to use Forward Error Correction”, *Proceedings 5th workshop on protocols for high speed networks*, pp. 134-148, Sophia Antipolis, France, Oct. 1996.

[62] Radiometrix, "Low Power UHF Data Transceiver Module", http://www.radiometrix.co.uk/products/bimsheet.htm

[63] RealPlayer, http://www.realplayer.com

[64] Reiniger D., Izmailov R., Rajagopalan B., Ott M., Raychaudhuri D.: “Soft QoS control in the WATMnet broadband wireless system”, *IEEE Personal Communications*, pp. 34-43, February 1999.

[65] Rizzo, L.: “Effective Erasure Codes for Reliable Computer Communication Protocols”, *ACM Computer Communication Review*, Vol. 27- 2, pp. 24-36, April 97.

[66] Schuler C.: "Optimization and adaptation of error control algorithms for wireless ATM”, *International Journal of Wireless Information Networks*, Vol. 5, No. 2, April 1998.

[67] Shacham, N., McKenney, P.: “Packet recovery in high-speed networks using coding and buffer management”, *Proceedings IEEE Infocom’90*, San Fransisco, pp. 124-131, May 1990.

[68] Shakkottai S., Srikant R.: “Scheduling real-time traffic with deadlines over a wireless channel”, *Proceedings of the second ACM international workshop on Wireless Mobile Multimedia (WoWMoM’99)*, pp. 35-42, August 1999.

[69] Sivalingam, K.M., Chen J.C., Agrawal, P., Srivastava, M.B.: "Design and analysis of low-power access protocols for wireless and mobile ATM networks", *Journal on special topics in mobile networking and applications (MONET)*, June 1998.

[70] 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.

[71] Smit G.J.M., et al.: "Overview of the Moby Dick project", *Proceedings Euromicro Summer School on Mobile Computing '98*, pp. 159-168, Oulu, Finland, August 1998.

[72] Smit, G.J.M., Havinga, P.J.M., van Opzeeland, M., Poortinga, R.: "Implementation of a wireless ATM transceiver using reconfigurable logic", *proceedings IEEE wmATM'99*, pp. 241-250, June 2-4 1999.

[73] Srivasta M.: "Design and optimization of networked wireless information systems", *IEEE VLSI workshop*, April 1998.

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

[75] Su W., Gerla M.: "Bandwidth allocation strategies for wireless ATM networks using predictive reservation", *IEEE Globecom '97*, 1997.

[76] Swann R., Kingsbury N.: "Error resilient transmission of MPEG-II over noisy wireless ATM networks", *IEEE proceedings of the International Conference on Image Processing*, Santa Barbara, October 1997.

[77] Swann R.: "Bandwidth efficient transmission of MPEG-II Video over noisy mobile links", *Signal Processing*, Vol. 12, No. 2, pp. 105-115, April 1998.

[78] Truman T.E.: "A methodology for the design and implementation of communication protocols for embedded wireless systems", *Ph.D. thesis, University of California, Berkeley*, spring 1998.

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

[80] WaveMODEM 2.4 GHz Data Manual, Release 2, AT&T 1995.

[81] Woesner H., Ebert J., Schläger M., Wolisz A.: "Power-saving mechanisms in emerging standards for wireless LANs: The MAC level perspective", *IEEE Personal Communications*, Vol. 5, No. 3, June 1998.

[82] Zorzi, M., Rao, R. R.: "Error control and energy consumption in communications for nomadic computing", *IEEE transactions on computers*, Vol. 46, pp. 279-289, March 1997.

[83] Zorzi, M., Rao, R. R.: "On the impact of burst errors on wireless ATM", *IEEE Personal Communications,* August 1999, pp.65-76.

[84] Zorzi, M., Rao, R. R.: "On the statistics of block errors in bursty channels", *IEEE transactions on communications, 1998.*

[85] Zorzi, M: "Performance of FEC and ARQ Error control in bursty channels under delay constraints", *VTC'98*, Ottawa, Canada, May 1998.