Energy-Efficient Adaptive Wireless Network Design

Paul J.M. Havinga, Gerard J.M. Smit, Martinus Bos
University of Twente, department of Computer Science, Enschede, the Netherlands
Phone: +31 534894619, Fax:+31 534894590
havinga@cs.utwente.nl

Abstract

Energy efficiency is an important issue for mobile computers since they must rely on their batteries. We present an energy-efficient highly adaptive architecture of a network interface and novel data link layer protocol for wireless networks that provides Quality of Service (QoS) support for diverse traffic types. Due to the dynamic nature of wireless networks, adaptations are necessary to achieve energy efficiency and an acceptable quality of service. The paper provides a review of ideas and techniques relevant to the design of an energy efficient adaptive wireless network.

1. Introduction

Wireless communications have experienced a rapid growth over the last decade. Portable computers like PDAs and laptops that use wireless communication to interact with the environment rely on their limited battery energy for their operation. Energy consumption is a limiting factor in the amount of functionality that can be placed in these devices. More extensive and continuous use of network services by multimedia applications will only aggravate this problem. Multimedia applications are characterised by their various media streams. Each stream can have different quality of service requirements. Depending on the service class and QoS of a connection a different policy can be applied to the communication protocol by the application such that it minimises energy consumption. For example, by avoiding error-control overhead for connections that do not need it and by never transmitting stale data, efficiency is improved.

While low-power components and subsystems are essential building blocks for portable systems, we have shown in our research [7][8] that a system-wide approach that incorporates the low-power vision into all layers of the system is beneficial because there is a vital relationship between hardware architecture, operating system and applications, where each benefits from the others.

In this paper we address the issue of *energy efficiency* in protocols for wireless networks and the host network interface, and in doing so incorporate all layers of the mobile system. The purpose of this paper is to provide a summary of some improvements that can be made at several layers of the system. Some of these improvements have been previously mentioned by others, but usually they are specific to only certain layers. We argue that communication metrics should be fed back to the applications in order to make mot of the available resources and to attain a near optimal balance between QoS and energy conservation.

2. Energy-efficient wireless communication

In this section we will discuss some techniques that can be used to reduce the energy consumption that is needed for the wireless communication. This can be accomplished efficiently and effectively only by a careful design of all layers in a wireless system. *Adaptability* of the protocols is a key issue. To illustrate our system level approach we discuss in the following examples how energy efficiency touches all layers of the system.

Physical layer – At the lowest level we need to apply an energy-efficient radio that can be in various operating modes (like variable RF power and different sleep modes) such that it allows a dynamic power management. Energy can also be saved if it can adapt its modulation techniques and basic error-correction schemes. The bandwidth a radio offers influences its energy consumption.

Medium access layer – In an energy-efficient MAC protocol [3] the basic objective is to minimise the time the radio needs to be powered. Since the overhead introduced due to state transitions is also significant, minimising the number of transitions also reduces energy consumption. By scheduling data transfers in bulk, an inactive terminal is allowed to doze and power off the receiver. Consequence is that the network interface must be reactivated at a scheduled time. Unsuccessful actions of the transceiver due to collisions and errors should be avoided, and the protocol should adapt to the dynamic environment.

Logical link control layer – Due to the dynamic nature of wireless networks, *adaptive error control* can give significant gains in effective bandwidth and energy efficiency. This avoids applying error-control overhead to connections that do not need it, and it allows to selectively match the required QoS and the conditions of the radio link. Above these error-control adaptations, a slot scheduler in the base-station can also adapt its traffic scheduling to the error conditions of wireless connections of a mobile. The scheduler can try to avoid periods of bad error conditions by not scheduling non-time critical traffic during these periods. *Flow control mechanisms* are needed to prevent buffer overflow, but also to discard stale packets. Depending on the service class and QoS of a connection, a different flow control can be applied such that it minimises the required bandwidth and energy consumption. For instance, in a video application it is useless to transmit images that are already outdated.

Network/transport layer – Errors on the wireless link can be propagated in the protocol stack. In the presence of a high packet error rate and periods of intermittent connectivity, characteristic for wireless links, some network protocols (such as TCP) may overreact to packet losses, mistaking them for congestion. TCP responds to all losses by invoking congestion control and avoidance algorithms. These measures result in unnecessary increases in energy consumption and deterioration of QoS. The limitations of TCP can be overcome by a more adequate congestion control during packet errors. These schemes choose from a variety of mechanisms to improve end-to-end throughput, such as local retransmissions, split connections and forward error correction [1].

Operating system level – Another way to avert the high cost (either performance, energy consumption or money) of wireless network communication is to avoid use of the network when it is expensive by predicting future access and fetching necessary data when the network is cheap. In the higher level protocols of a communication system caching and scheduling can be used to control the transmission of messages. This works in particular well when the computer system has the ability to use various networking infrastructures, with varying and multiple network connectivity and with different characteristics and costs [5].

Modern high-performance network protocols require that all network access goes through the operating system, which adds significant overhead to both the transmission path (typically a system call and data copy) and the receive path (typically an interrupt, a system call, and a data copy). Aside from causing performance problems, this also incurs significant energy consumption. To address the performance problem, several user-level communication architectures have been developed that remove the operating system from the critical communication path [2]. The same principles may also be used to improve energy efficiency.

3. Energy-efficient wireless networking design

In the Moby Dick project we develop and define the architecture of a new generation of handheld computers, the so-called *Mobile Digital Companion*. This section describes the basic principles and mechanisms of the network interface architecture being implemented in the Moby Dick project, our energy-efficient medium access control for wireless links, called E^2MaC, and how adaptability is included into all layers of the system. The implementation is described briefly.

3.1. Moby Dick

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 believe *adaptation* is of fundamental importance to the success of low-power handheld systems. 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. Current research on several aspects of wireless networks (like error control, frame-length, access scheduling) indicate that adapting to the current condition of the wireless link continuously has a large impact on the energy-efficiency of the system (e.g. [4][6][9][11][12][14][16]). In the Moby Dick project [13] 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. We have a reconfigurable systems-architecture which, in combination with a QoS driven operating system and network protocols, can cope with the inherent dynamics of a mobile environment. The protocol and the architecture are targeted to a system in which quality of service and energy consumption plays a crucial role.

3.2. System overview

The wireless network is composed of several base-stations where each handles a single radio cell covering several mobile stations. We consider an office environment in which the cells are small and have the size of one or several rooms. Small cells save energy as the transmitters can be low powered and they also provide a high aggregate bandwidth since small cells need to be shared with only few mobiles. The backbone of the base-stations is a wired network.

The base-station controls access by dividing bandwidth into transmission slots. The premise is that the base-station has virtually no processing and energy limitations, and will perform actions in courtesy of the mobile. The main principles are: 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, and synchronise the mobile and the base-station which allows the mobile to power-on precisely when needed.

The QoS provisions of ATM with its small fixed sized cells (48 bytes data, 5 bytes control) 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), or even multiple priority levels. The small cells of ATM 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 and error control mechanism that can be adapted to the QoS of a connection. 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 layers of the communication protocol are summarised in Figure 1. 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.

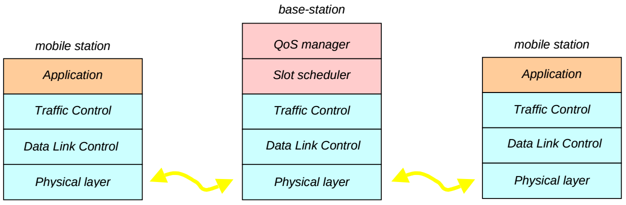

Figure 1: Protocol stack.

The lower three layers exist in both the mobile and the base station. The *Data Link Control* manages the data-transfer with the physical layer, 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.

3.3. 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.

Multimedia networking requires at least a certain minimum QoS 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 applied 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.

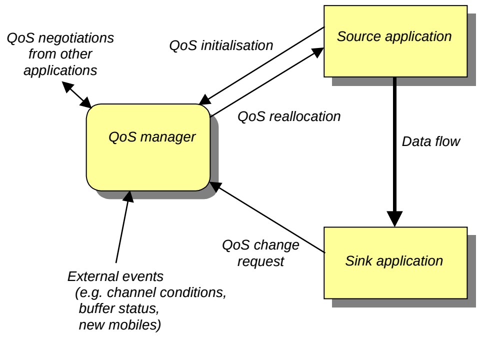

Figure 2: The service model for adaptive applications.

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 allows 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 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. In [10] a bandwidth reallocation algorithm is described that fits well to the QoS model used by the QoS manager.

3.4. Slot scheduler

The *slot scheduler* controls the traffic in one cell. It assigns bandwidth to applications and determines the required error coding for each individual connection. 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 (see Section 3.6).

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 to the mobiles.

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 according to the error rate observed at the receiver. The slot scheduler retrieves this status information via a backward connection and indicates to the network interface which error coding scheme to use. 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 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 [4].

3.5. Network interface

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

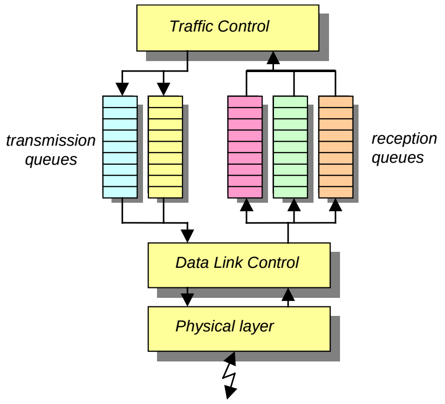

Figure 3: The network interface architecture.

Figure 3 depicts the basic blocks of the architecture of the Network Interface. The number of connection queues is dynamic and the figure is just an example.

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 manager, which informs the Data Link Control (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*. This buffer is only meant to store ATM cells for a short time, just enough to implement an effective error-control mechanism. The buffer is organised in such a way that it has a small queue for each connection. 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 and basic error correction on each ATM cell.

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 which energy-efficient error-control strategy to use, is a function of QoS parameters, radio channel quality, and packet length. The selection of the error-control scheme and the required size of the queue depends on the QoS constraints imposed on each connection, such as delay constraints or loss-less transfer constraints.

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 [11]. 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 bandwidth accordingly.

Each connection has its own connection queues with *customised flow control*. Flow control is needed to prevent buffer overflow. Depending on the type of data carried by a connection, a replacement policy and size can be set for each connection. Flow control information is transmitted in the header of each data packet.

3.6. Energy-efficient MAC protocol (E^{2}MaC)

The E^{2}MaC protocol uses fixed-length frames. The frame is divided in time-slots that can have three basic types: *traffic control*, *registration request*, and *data*. The base-station controls the traffic for all mobiles in range of the cell and broadcasts the schedule in the *traffic control slot* (TCS), the first packet of a frame.

The overhead introduced in the physical layer can be significant, e.g. for WaveLAN [15] 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 4).

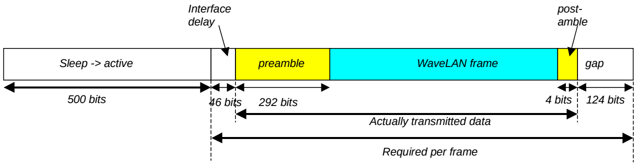

Figure 4: WaveLAN physical layer block format.

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). 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.

Efficient data transmission (in terms of bandwidth utilisation and energy consumption) can only be achieved if the number of ATM cells transmitted sequentially is not too small. 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. 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.

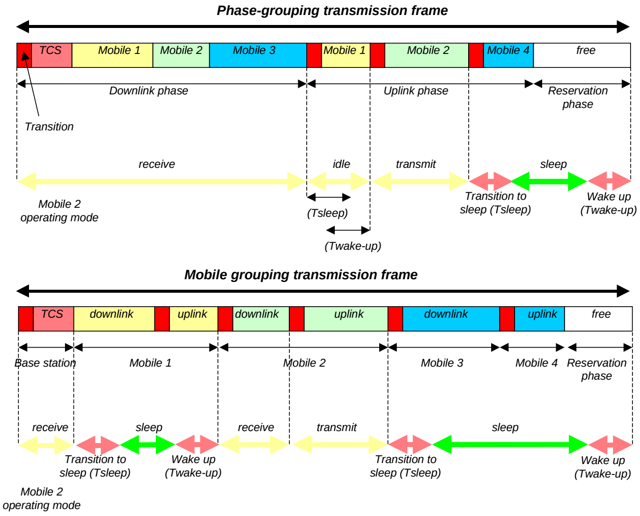

Figure 5: Grouping strategies in a transmission frame.

Grouping of uplink and downlink traffic of one mobile (which we will refer to as *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. 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 5). 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. Most MAC protocols group the traffic from the base-station, mainly because of its efficient use of the available bandwidth.

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. 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 ($T_{wake\text{-}up}$), energy can be saved. Otherwise, the mobile must remain idle, waiting for its time to transmit its uplink packet.

Evaluation

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. We assume a network interface with the characteristics of the WaveLAN modem [15]. The power consumption of the WaveLAN modem when transmitting is typical 1675 mW, 1425 mW when receiving, and 80 mW when in sleep mode. Increasing the sleep time period of the radio thus significantly improves the energy efficiency of the wireless network. We use a frame rate of 100 Hz. 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.

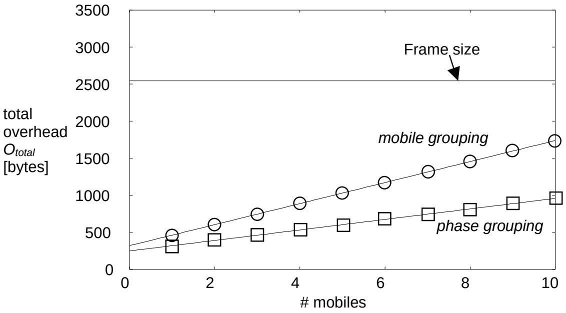

Figure 6: Total overhead versus number of mobiles.

Figure 6 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. This would reduce the overhead that is required to transmit a certain amount of data, but will increase the latency.

When we look at the consequences for energy consumption, then mobile grouping is more advantageous. This is shown in Figure 7. The sleep period per mobile is larger when using mobile grouping compared to phase grouping.

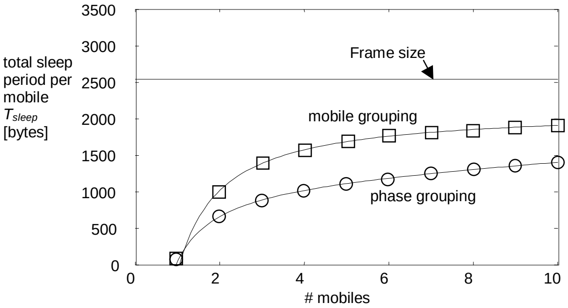

Figure 7: Sleep period per mobile versus number of mobiles.

The envisioned cell size for the Mobile Digital Companion is small (pico-cellular with the size of an office-room). Therefore, the number of mobiles in one cell will in general be small and the protocol will encounter a small overhead. When the transition times T_{sleep} and $T_{wake\text{-}up}$ become shorter, mobile grouping becomes even more advantageous.

3.7. Implementation

We have made a test-bed for the network interface that we used to experiment with the various techniques and mechanisms for e.g. error control and the MAC protocol. The base station will handle the TCP/IP protocol in lieu of the mobiles, and uses a lightweight protocol with the mobile to transfer the packets. Note that while we propose to eliminate the dependency on software-based protocol stacks from the mobile, there is no reason to dogmatically preclude the involvement of the mobile from ‘standard’ protocols.

The testbed is build with off-the-shelf components to allow a short design cycle. Currently we use a WaveLAN modem as the physical layer. The raw data rate is 2 Mb/s. The modem provides the basic functionality to send and receive frames of data. It does not include a Medium Access Protocol, but provides signalling information like carrier sense and collision detect.

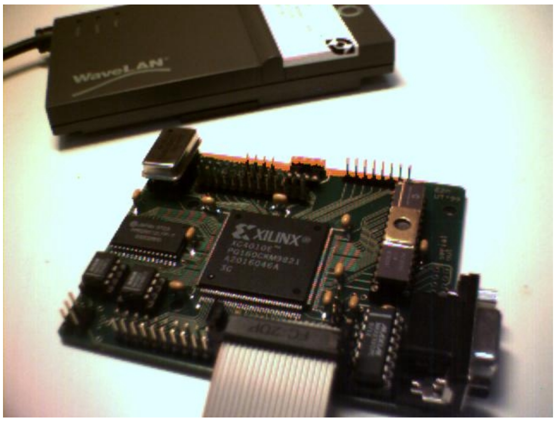

Figure 8: Network interface implementation.

Figure 8 shows a photograph of the network interface implementation. The FPGA controls the data-flow between the radio and the host. 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 the host. It therefore performs the queue setup and administration. This administration is used to setup VCI mapping tables and queue address maps stored in the FPGA. It thereby administrates 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. 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 when to initiate error correction on the received packet. The error correction is being performed in a close interaction 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 corrected packet will be forwarded to its destination. Cells arriving from the host can be protected against errors that might occur on the wireless channel by adding redundant cells. Just like the error correction mechanism, the FPGA provides the compute intensive functions that the microcontroller can use to build the packet that is protected by the required error correcting code.

4. Conclusions

We have presented an architecture for 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.

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, 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.

References

[1] Balakrishnan H., Padmanabhan V.N., Seshan S., Katz R.H.: A comparison of mechanisms for improving TCP performance over wireless links", *ACM SIGCOMM'96*, Stanford, August 1996.

[2] Bhoedjang, R.A.F., Rühl T., Bal H.E.: “User-level network interface protocols”, *Computer*, November 1998, pp. 53-60.

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

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

[5] Ebling, M.R., Mummert, L.B., Steere D.C.: “Overcoming the Network Bottleneck in Mobile Computing”, *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, Dec. 1994, Santa Cruz, CA.

[6] 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.

[7] Havinga P.J.M., Smit G.J.M., Bos M.: “Energy efficient wireless ATM design”, *wmATM'99*, June 2-4 1999.

[8] Havinga P.J.M., Smit G.J.M.: “Octopus: embracing the energy efficiency of handheld multimedia computers” , *proceedings ACM/IEEE Mobicom* 1999.

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

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

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

[16] 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.