Energy efficiency of error correction on wireless systems

Paul J.M. Havinga
University of Twente, department of Computer Science, the Netherlands
Havinga@cs.utwente.nl

Abstract -- **Since high error rates are inevitable to the wireless environment, energy-efficient error-control is an important issue for mobile computing systems. We have studied the energy efficiency of two different error correction mechanisms and have measured the efficiency of an implementation in software. We will show that it is not sufficient to concentrate on the energy efficiency of error control mechanisms only, but the required extra energy consumed by the wireless interface should be incorporated as well. A model is presented that can be used to determine an energy-efficient error-correction scheme of a minimal system consisting of a general-purpose processor and a wireless interface. As an example we have determined these error-correction parameters on two systems with a WaveLAN interface.**

I INTRODUCTION

Low energy consumption is a key issue for portable wireless network devices like computers like PDAs and laptops where battery energy is a limited resource. Power consumption is becoming the limiting factor in the amount of functionality that can be placed in these devices [8]. More extensive and continuous use of network services will only aggravate this problem [21]. 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, including error coding.

This paper describes how energy efficiency in the wireless link can be enhanced via adaptive error control and we investigate the energy efficiency of error control mechanisms. The focus is on the higher protocol layers (link and transport) layers that offer a high potential gain of energy efficiency.

Error-control mechanisms traditionally trades off complexity and buffering requirements for throughput and delay [13][14][2]. 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 [20]. 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.

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 to choose from. Computer communication generally implements a reliable data transfer using either methods or a combination of them at different levels in the 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. [19][10]).

Adaptive error control allows the error control strategy to vary as the channel conditions vary. 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 likely to be optimal in terms of energy efficiency all the time, and adaptive error control seems likely a source of efficiency gain.

Scope

In our study we focus on the energy efficiency of local error-control schemes for communication protocols on the wireless channel. We take into account the computational costs associated with error correcting codes in addition to the communication costs associated with radio transmission. In particular we concentrate on error control schemes 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. In multimedia traffic important parameters are jitter, delay, reliability, and throughput [6]. In the presence of stringent QoS and energy consumption constraints it is very hard to use retransmission techniques as basic error control mechanism because they affect the QoS when the error conditions become bad. Although good designed retransmission schemes can be optimal with respect to energy efficiency, they can introduce intolerable low performance to fulfil the required QoS of the application. Although error correction schemes in most cases consume more energy due to increased computation and communication, they can provide the constant quality and stringent delay provisions required for multimedia traffic. Moreover, since continuous-media services like audio, video and graphics can tolerate some residual corruption, ARQ can be avoided altogether for these applications [7]. Nevertheless, ARQ may be valuable on time-varying wireless channels, where QoS guarantees will be difficult or expensive.

0-7803-5668-3/99/$10.00 © 1999 IEEE.

Related work

Error control is an area in which much research has been performed. Books on error control, such as [12], cover the basic FEC and ARQ schemes well. More recently, much work has focussed on error control in wireless channels. Adaptive error control is mainly used to improve the throughput on a wireless link [4][5]. Most relevant work that also relates the error coding strategy to energy consumption is by Zorzi [23][24] and Lettieri [11].

Overview

A goal of this research is to determine which error-correcting mechanisms should be used in a wireless environment with stringent QoS, and which parameters they should apply to obtain the most energy efficient solution for a variety of error conditions in a wireless environment. In particular we will study the well-known Reed-Solomon code and the less known EVENODD code [1] that was designed for a system of redundant disks (RAID). We will concentrate on an implementation in software. Dedicated hardware may be used, though in a system, which requires the flexibility to alter error control schemes on a stream by stream basis, a software solution may be preferable. The total energy consumption per useful bit will depend both on the energy of transmission and the energy of redundancy computation. We will show that the computational cost associated with FEC cannot be ignored. Furthermore, the trend has been toward smaller communication cells, e.g. with the size of an office room, thus requiring lower transmit power. The ratio of computational to transmit power under these circumstances is therefore only likely to increase. An additional goal is therefore to determine which parameters should be used when the cost for communication is incorporated as well.

First we will provide in section II the general strategy that we will use for error correction. Section III shows the results of an implementation of the mechanisms on a general-purpose processor. The energy efficiency factor of our implementation and the pre-conditions for it will be determined for both mechanisms. The results are used in section IV to determine the energy efficiency of a minimal system consisting of a general-purpose processor and a wireless interface. As an example we determine the energy efficiency of error control with a WaveLAN card. We will finish with some conclusions in section V.

II ERROR CONTROL

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.

The error model

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. Studies have shown that this leads to three classes of errors: packet loss, packet truncation, and bit corruption errors. A *hierarchical error control* mechanism – on which basic error control is applied on a relatively small amount of data to overcome the basic BER, and additional layers on which FEC in combination with ARQ is applied on larger amounts of data to overcome burst errors – can tackle these types of errors. However, given the high variability of the error environment, these techniques would need to be applied *adaptively* to avoid unnecessary overhead when conditions are good. 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. [3][4][5][17]).

In this study we do not deal with the physical layer or on the mechanisms involved like coding and power control. Our starting point is on the interface between the physical transceiver and the higher-level communication system. Most wireless transceiver hardware of wireless systems has some error correction mechanisms build-in to overcome or reduce the impact of the errors on the physical channel. However, it is usually inefficient to provide a very high degree of error correction, and some residual errors pass through. The network communication layer therefore does not see the raw physical channel, but a channel modified by the physical layer that has a residual error characteristic [22]. The most relevant errors that occur are block errors covering a period of up to a few hundred milliseconds. If the length over which the error correction is applied is larger than the burst size, then it is possible that a sufficiently strong error correcting code will be enough to overcome even the burst errors.

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 present in the sender and receiver.

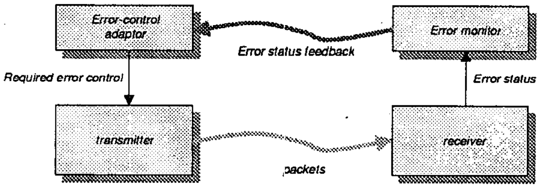

Figure 1: Feedback loop for adaptive error control.

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 characterization, 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.

The encoding packet model

The basis for most currently designed wireless systems will likely be packet switching communication schemes that manage data transfer in blocks that contain multiple symbols (of size s). Errors are assumed to be detected by some detection technique, and the whole packet will be discarded. The residual channel characteristic after the physical and link layer processing is then based on *erasures*, i.e. missing packets in a stream [18]. Figure 2 shows a graphical representation of the error correction mechanism. The sender collects a number of *source data packets* in a buffer. When the buffer is full, the data is encoded, and the encoded data is transmitted. The receiver is able to reconstruct the original data from a subset of the encoded data, and so can allow the erasure of some packets.

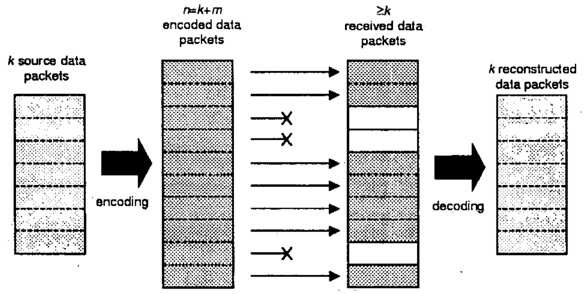

Figure 2: Graphical representation of error correction.

We will denote the number of source packets as k, the packet size as S, the number of redundant packets m, and the number of encoded packets as n. Such a code is called an (n,k) code and allows the receiver to recover from m $(=n-k)$ losses in a group of n encoded packets. This structure can be seen as an (S) x $(k + m)$ array in which the columns represent a packet of length S, the first k columns represent the source data packets, and the last m columns represent the redundant packets. All packets together build up one *frame*. Figure 3 gives a graphical representation of this scheme.

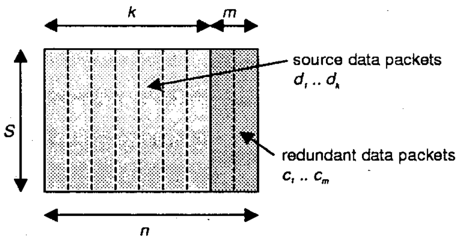

Figure 3: Representation of an encoded frame.

In communication systems error correcting codes are used to protect packets of data that are transmitted over some network. The most general technique for tolerating m simultaneous failures with m redundant packets is a technique based on *Reed-Solomon* (RS) coding [16]. This technique requires computation over finite fields and results in a complex implementation. An attractive alternative might be a scheme like *EVENODD* that only requires simple exclusive-OR operations and that it is able to tolerate two failures [1].

III IMPLEMENTATION AND RESULTS

In the next sections we show the results of a software implementation of both error-correction mechanisms. We are aware of the fact that a software implementation is not the most energy-efficient solution, and might not provide enough performance. However, there exist many applications and systems that do not need high performance and cannot use the capabilities and advantages of dedicated hardware. For example, a notebook computer can also benefit from an energy-efficient solution, although not the most optimal possible when the required hardware would be available. A software implementation has further a number of specific advantages compared to a hardware solution:

- The use of a microprocessor allows *very rapid adaptations* to varying error conditions (burst size, frequency) and required QoS from applications. The adaptation to perform can be applying another error control scheme, or adapting some parameters of the error control scheme.
- A software implementation allows us to experiment with a large set of error control schemes, and experience in 'real life' how applications behave. When we have a good feeling of the behaviour of the schemes, then we could compose a subset of error control schemes that is suitable to be implemented in hardware, either in an FPGA, a DSP, or a custom chip.
- The error control can easily and efficiently be embedded in various layers of the communication protocol where the data is buffered anyway. Little extra overhead is expected with a good engineered and well-integrated error correction mechanism.
- A standard processor also allows the use of relatively large memories, and thus allows for much larger block lengths than standard custom chips (that typically allow a block length of up to 255 bytes [15]). In a wireless office environment burst errors of 1 to 100 ms can be expected. To handle these large erasures at relatively high speed (say 2 Mb/s), a large block size is needed.

We will assume the energy consumed by the algorithm to be linearly related to the amount of time needed for the processor to do its calculations. Although this possibly introduces some inaccuracy, we believe that this still gives a good approximation of reality. Both implementations are written in C and are portable across many platforms. The code is written straightforward, and only uses the most obvious optimisations (like the use of a multiply lookup table for RS coding). Handcrafted code that makes good use of the specific features of the processor (like registers, caching and the use of special instructions) might achieve significant speedups.

In the following we will compare the energy-efficiency factors using a rating that indicates the amount of energy consumed to process one byte, using:

$$E = 1 / (\textit{time to process 1 byte } [\mu s] \; . \; \textit{required power } [mW]) \qquad (1)$$

We define the *energy efficiency e* as the amount of data processed divided by the energy that is consumed to process that data.

Error correction

The data model that we have used in our measurements is an (S) x $(k + (n-k))$ array of symbols with size s. The energy efficiency e_{ec} of RS coding and of EVENODD coding (approximation when n-k>2) for k large is approximately:

$$e_{ec} = E_{ec} / (n-k) \; . \; k \qquad (2)$$

The factor E_{ec} is determined by the implementation of the coding. We first measured E_{ec} on a PentiumPro 133 MHz for EVENODD and RS using n-k=2. The effect of the inherent overhead is getting less when the packet size and the number of source packets k is enlarged, because it will be amortized over more data. The column *constraints* indicates when the values have become stable. Note that the EVENODD algorithm operates on symbols of size s, whereas the RS algorithm becomes stable in terms of a packet size S (consisting of multiple symbols s). To summarize we have the following results:

Table 1: Energy efficiency factors of error correcting codes for n-k=2.

mechanism	*constraints*	$E_{ec\ encoding}$	$E_{ec\ decoding}$
EVENODD	$k \geq 5; s \geq 32$ bytes	100	90
Reed-Solomon	$k \geq 5; S \geq 850$ bytes	28	40

The implementations show that a constant performance and constant energy efficiency is reached for small values of k and for small data sizes. The RS code has a larger initial overhead than EVENODD. The RS code is attractive because it is the most general technique capable of tolerating n-k simultaneous failures. The complexity and the requirement of computations in the finite field make the encoding however about four times less energy efficient than EVENODD. The EVENODD mechanism on the other hand, can only sustain two packet erasures, but is more flexible and gives a further reduction in energy consumption for decoding when the burst-error size is small [9].

IV ERROR-CORRECTION ENERGY-EFFICIENCY OF A MINIMAL COMMUNICATION SYSTEM

The *communication overhead* involved mainly depends on the number of additional bits that are transmitted. The communication energy efficiency of transmitting an (n,k) redundant code (with $m = n\text{-}k > 0$) equals:

$$e_{com} = E_{com} \, . \, k\,/\,(n - k) \qquad (3)$$

in which E_{com} is determined by the energy consumption of the wireless interface.

In Figure 4 both functions are plotted (with (n-k) constant), and the trade-off is shown clearly. The energy efficiency of error correction increases when the number of source packets k decreases, and the energy efficiency of *communication* e_{com} shows a greater efficiency when the ratio between the number of source packets k and the number of redundant packets (n-k) is large. The value of k where the communication overhead equals the error correction overhead depends on the implementation of the coding and on the energy efficiency of the communication.

Note that the maximum loss rate that can be sustained equals $(n - k)\,/\,n$. So, when determining the most energy efficient value of k for a constant number of redundant packets (n-k), k must always be below the required value. In [9] is shown that by using (2) and (3) the energy efficiency of a system composed of a transceiver for the communication and a codec for the encoding and decoding of the packets equals to:

$$e_{tot} = \{\, (n - k)\,/\,E_{com} \, . \, k \; + \; (n - k) \, . \, k\,/\,E_{ec} \,\}^{-1} \qquad (4)$$

So, once we know the values of E_{com} and E_{ec}, this equation can be used to adapt the parameters of the error-correcting code, such that an energy efficient implementation is achieved.

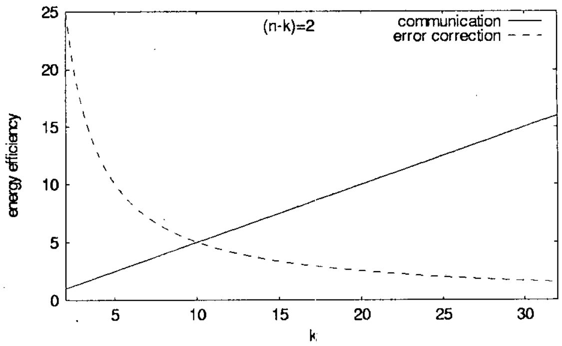

Figure 4: System energy efficiency e vs. number of source packets k.

As an example we will determine the energy efficiency of a small system consisting of a WaveLAN PCMCIA card with a Pentium Pro and a StrongARM SA-1100. The energy efficiency of the StrongARM is about 70 times better than the PentiumPro. Figure 5 shows the result.

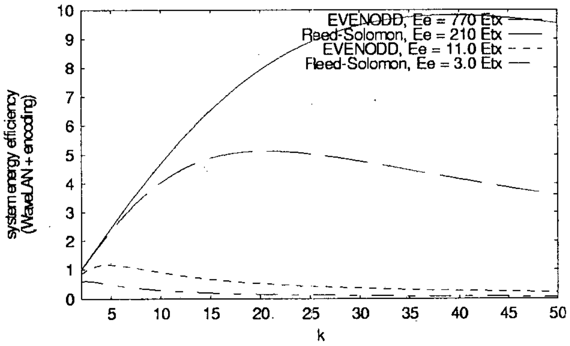

Figure 5: Energy efficiency of error correction on systems with WaveLAN.

The figure shows only the transmitting side, thus encoding and sending the data. The graphs with E_e=210 E_{tx} and E_e =770 E_{tx} represent the StrongARM efficiency. The general tendency is that the energy efficiency increases with increasing k (because less communication is needed), up to a certain k_{max} where the encoding cost is becoming the limiting factor. When the efficiency of encoding is high (like the StrongARM implementation of EVENODD), then k_{max} is high also. If the efficiency of encoding is low (like the RS encoding on a PentiumPro), then the communication cost is negligible.

This shows that is of no use to have a larger k than k_{max}, not only because it is less efficient, but also because it has lower error-correcting capabilities. On the other side, k must be chosen small enough to sustain the bad state probability of the system in a certain environment. This k is called k_{min}. For n-k=2 the tolerable error rate ε is determined by $\varepsilon<2/(k+2)$, thus $k<2/\varepsilon-2$. Both k_{max} (for energy efficiency) and k_{min} (for error correcting capabilities) thus determine the appropriate choice of k in a system. The general consensus that it is profitable to minimise the number of bits over the air-interface [5] is thus not correct and effective when considering the energy efficiency of a system.

General strategy

The value of k_{max} is the maximum of the graph from relation (4). When implementing an energy efficient error correction mechanism there are several parameters that must be taken into account.

- *Ratio R*. The ratio R between the energy efficiency factor of error correction and the energy efficiency factor of communication is not dependent on the environment and the current state of the wireless channel. It is static for a certain system configuration and can be determined when designing or configuring the system.
- *Bad state probability e*. This probability is highly dependent on the current situation of the wireless channel. The parameter determines the required coding mechanism, the number of source packets k, and the number of redundant packets n-k.
- *Maximum burst error-size*. This parameter is also dependent on the current situation and mainly influences the required buffer space, but also influences the error correcting capabilities of the mechanism. If the required QoS can sustain the delays that are introduced with the buffer, then the size can be chosen large.

When all these parameters are known, then the error correction mechanism can be chosen such that it adheres to the required QoS (incorporating the error correcting capabilities and latency) at the most energy efficient way using the relation (4) described in this section.

V CONCLUSION

We have studied two different error correction mechanisms with different characteristics and capabilities, i.e. EVENODD and RS. The implementations of these mechanisms on a general-purpose processor in *C* show that they already reach constant performance and constant energy efficiency for small values of k and for small data sizes. The RS code is attractive because it is the most general technique capable of tolerating n-k simultaneous failures. The complexity and the requirement of computations in the finite field make the encoding however about four times less energy efficient than EVENODD. The EVENODD mechanism on the other hand can only sustain two packet erasures, but it allows the reconstruction of $2(k$-$1)$ erased *symbols* (and not packet erasures as with RS) which increases its flexibility and gives a further reduction in energy consumption for decoding when the burst-error size is small.

We have shown that it is not sufficient to concentrate on the efficiency of error control mechanisms only, but the required extra energy consumed by the wireless interface should be incorporated as well. From energy efficiency perspective it is not always profitable to minimise the number of bits transmitted over the air. We have provided a strategy to determine the most energy efficient error correction scheme of a minimal *system* consisting of a general-purpose processor and a wireless communication interface.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

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

[8] 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 ICPCW'97*, Dec. 1997.

[9] Havinga, P.J.M.: "Energy efficiency of error correcting mechanisms for wireless communication", *CTIT Technical Reports 1998*, TR-CTIT 98-19, September 1998, the Netherlands.

[10] 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.

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

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

[13] 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.

[14] 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.

[15] "L64711/12/13/14 Reed-Solomon Encoders/decoders", *LSI logic*, http://www.lsilogic.com/products/unit5_7d.html.

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

[17] 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.

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

[19] 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.

[20] Smit, G.J.M., et al. "Overview of the Moby Dick project", *Euromicro summer school on mobile computing*, Oulu, Aug. 1998.

[21] Stemm, M, et al.: "Reducing power consumption of network interfaces in hand-held devices", *Proceedings mobile multimedia computing MoMuc-3*, Princeton, Sept 1996.

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

[23] 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.

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