



Concatenated hybrid ARQ - a flexible scheme for wireless real-time communication

Elisabeth Uhlemann, Tor Aulin, Lars K. Rasmussen and Per-Arne Wiberg

Halmstad University Post-Print

N.B.: When citing this work, cite the original article.

©2002 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Uhlemann E, Aulin T, Rasmussen L, Wiberg P. Concatenated hybrid ARQ - a flexible scheme for wireless real-time communication. In: Eighth IEEE Real-Time and Embedded Technology and Applications Symposium: proceedings : September 25-27, 2002, San Jose, California. Los Alamitos, Calif.: IEEE Computer Society Press; 2002. p. 35-44.

DOI: <http://dx.doi.org/10.1109/RTTAS.2002.1137379>

Copyright: IEEE

Post-Print available at: Halmstad University DiVA
<http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-404>

Concatenated Hybrid ARQ - A Flexible Scheme for Wireless Real-Time Communication*

Elisabeth Uhlemann^{1,2}, Tor Aulin², Lars K. Rasmussen^{2,3} and Per-Arne Wiberg¹

¹Computing and Commun. Lab. Halmstad University Halmstad, Sweden
{bettan, pelle}@ide.hh.se

²Dept. of Computer Engineering Chalmers University of Tech. Göteborg, Sweden
tor@ce.chalmers.se

³Inst. for Telecommun. Research University of South Australia Mawson Lakes, Australia
larsr@ce.chalmers.se

Abstract

The concept of deadline dependent coding (DDC) has previously been suggested by the authors for maximizing the probability of delivering the required information before a given deadline in a wireless communication system. In this paper, these principles are further developed using concatenated codes with iterative decoding, providing a new level of flexibility and robustness for DDC protocols. The strategy of DDC is to combine different coding and decoding methods with automatic repeat request (ARQ) techniques in order to fulfill the application requirements within a wireless real-time communication system. These requirements are formulated as two Quality of Service (QoS) parameters: deadline (t_{DL}) and probability of correct delivery before the deadline (P_d), leading to a probabilistic view of real-time communication. An application can negotiate these QoS parameters with the DDC protocol, thus creating a flexible and fault-tolerant scheme.

1. Introduction

The tremendous development in wireless communication has provided opportunities in many related fields, such as remote sensing, monitoring, and control. New technologies and products are introduced into the market at an ever-increasing rate. This wireless evolution offers improvements for industrial applications, where traditional wireline solutions have prohibitive problems in terms of cost and feasibility. Wireline implementations are for example not cost efficient for large, temporary production lines, and may not be feasible at all for systems including rotating or high mobility machinery such as measurement and control of moving objects. Another sample application, requiring wireless access, is

communication to and from different kinds of vehicles in factory automation situations.

A time-critical system or a real-time system is characterized by the fact that it has deadlines to meet. In this work, we are interested in real-time communication systems where the timeliness of the delivered data is equally important as the correctness of the delivered data. This provides a challenge for wireless real-time communication due to the harsh communication environment encountered when transmitting over a wireless channel. The inherent consequence is a relatively high error rate, making the wireless channel unreliable in comparison to copper wire local loops or optical wireline channels. This has limited the extensive use of wireless access in real-time systems. Implementations of wireless communication systems for industrial use, guaranteeing real-time delivery, have been attempted, however, no general framework has been suggested. As a consequence, it is not straightforward to evaluate the general dependability of such systems with respect to real-time delivery and fault-tolerance.

In this paper, we exploit a probabilistic view of real-time communication in order to provide a systematic approach for the development of efficient wireless real-time communication protocols. The main contribution of the paper is to propose a deadline dependent coding scheme based on a retransmission strategy involving concatenated codes and iterative soft decision decoding [1]. This scheme results in improved fault-tolerance in presence of communication errors and permits a flexible admission control scheme capable of providing a trade-off between the worst-case delivery time and the quality of the delivered data. Each component used in the coding scheme is made deadline dependent. The benefits of the protocol in terms of real-time communication are demonstrated by computer simulations using simple channel models, providing benchmark performance limits.

*This work is funded by the national Swedish Real-Time Systems research initiative ARTES (www.artes.uu.se), supported by the Swedish Foundation for Strategic Research.

1.1. Quality of Service Using a Probabilistic View of Real-Time Communication

The literature in the field of real-time systems often discusses two different classes of systems, *hard* and *soft* real-time systems. In a hard real-time system, late delivery cannot be tolerated. In contrast, in a soft real-time system a specified low probability of late delivery is tolerated, while permitting performance degradation by relaxing the real-time constraints. In this work we use a probabilistic view of the real-time constraints. This means that it is not meaningful to talk about hard or soft real-time communication systems. Instead we talk about a deadline for delivery and the probability of success in delivering correct information before this deadline. Thus, we use two parameters: deadline, t_{DL} , and the probability of correct delivery before the deadline, P_d . They can be viewed as Quality of Service (QoS) parameters of a real-time communication system. A protocol layer can negotiate values of these parameters with an upper or a lower layer. The protocol can then either reject the request or accept it, guaranteeing the delivery based on the given values of the QoS parameters. If the request cannot be accepted the application can possibly re-negotiate. One of the objectives of the real-time communication protocol is to maximize the probability that the communication system will be able to accept the transmission request. Note that a hard real-time system defined using these QoS parameters will have $P_d=1$, a situation that cannot be achieved in any physical system due to noisy channel conditions, [2].

1.2. Deadline Dependent Coding

Within the context of a probabilistic view of real-time constraints, the digital communication problem also has an elegant probabilistic formulation due to Shannon [3]. A fundamental result was formulated in [3] known as Shannon's *channel capacity*. The channel capacity incorporates into one composite parameter the effects of channel parameters such as thermal noise, constrained bandwidth, and limited signal power. The channel capacity is a fundamental upper limit for the achievable data rate over channels described by these parameters. Considering a wireless radio channel in this perspective, bandwidth is limited since the radio spectrum is a limited natural resource. The radio spectrum is assigned according to strictly enforced rules and thus, a fully utilized frequency band cannot easily be complemented by additional resources. Furthermore, wireless devices are often battery-powered and therefore the transmitted signal power should be limited to prolong battery life. In addition, limited transmit power also limits the inherent interference generated by transmitters in a wireless multiple access system.

The significance of channel capacity is that as long as the communication rate is kept below the channel

capacity, an arbitrarily low error rate can in principle be obtained *if* infinitely long signals are used. From coding theory, we know that most finite-length codes are good, provided they are sufficiently long. Decoding complexity, however, may prohibit the use of codes beyond a certain length. When a real-time communication system is used, we are not only concerned with decoding complexity but also transmission time. The question is how well we can perform when complexity requirements in terms of time to decode *and* time to transmit have to be considered.

A new type of communication protocols may offer desirable benefits for wireless real-time communication as demonstrated in [4] where deadline dependent coding (DDC) was introduced. The main idea behind the concept of DDC is to make all aspects of the communication protocol deadline dependent. The protocol should also attempt to minimize the required bandwidth, the transmitted energy and the time required to successfully deliver the information. The QoS parameters t_{DL} and P_d are mapped onto a retransmission protocol, which plays the role of maximizing the probability of correct delivery before a deadline, using a minimum of resources. The DDC protocol performs a series of transmissions triggered by a retransmission protocol, providing increasingly more information for decoding the closer we get to the deadline.

The principles of DDC are illustrated in Figure 1. In the beginning of the time window, a message is transmitted. The receiver decodes the message and checks the quality of the decoder output. If the decoder output does not meet a pre-determined quality or reliability level, a retransmission is requested. This procedure will be repeated until the requested level of P_d has been achieved. The strength of the coding scheme, the decoder strategy, the guiding principles of the retransmission scheme and the maximum allowed number of retransmissions are all components and features to be designed according to the QoS parameters. The maximum number of retransmissions allowed is determined directly by the deadline while other components are designed to provide the required P_d before the deadline. It should be noted that once a request has been accepted by the protocol, the probability of delivering a packet, with the required QoS parameters P_d and t_{DL} , is guaranteed to hold.

A retransmission scheme is continuously adapting to instantaneous channel conditions. A series of retrans-

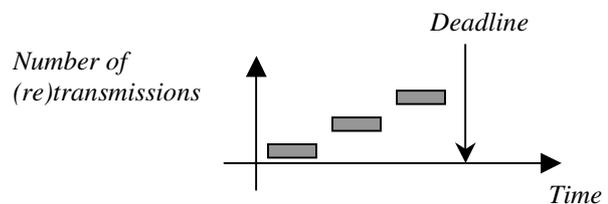


Figure 1. The Deadline Dependent Coding transmission suite.

missions is initiated when the channel is bad, thus contributing to the robustness of the protocol, while only a negligible number of retransmissions are required when the channel is well behaved. We may therefore be able to use a relatively short code intended for good channel conditions, thereby saving bandwidth resources. Using a regular scheme where no retransmissions are allowed, we have to use a significantly longer code to get the same performance during bad channel conditions as using a retransmission scheme with a short code. As the channel conditions improve, we must continue to use the same long code, requiring more resources than the immediate channel conditions call for.

So far, we have discussed static and dynamic channel properties in terms of error rate and amount of resources used. In a multiple access radio system, multiple access interference (MAI) is encountered. Hence, it is important to limit the interference created by the acting nodes. Consequently, another advantage of a carefully designed retransmission scheme is that we only use the required amount of redundancy suitable for the current channel condition and thus the amount of MAI is reduced. As the deadline approaches, it becomes increasingly more important to get the information delivered regardless of possible undesirable MAI. As a consequence, increasingly more channel resources are allocated in order to meet the probabilistic requirements for delivery before the deadline.

The DDC scheme differs from existing communication protocols in a number of ways. Most importantly, DDC explicitly uses the deadline to control the transmission suite. There are a number of real-time communication protocols, e.g., [5] and [6] that are best effort protocols and consequently do not give any guarantees or explicit predictions on the probability of delivery. Other protocols have been developed, which guarantee hard deadlines [7], [8]. They depend, however, on a reliable channel, while DDC in contrast strives to maximize the probability of correct delivery over an unreliable channel.

1.3. DDC Using Concatenated Codes with Iterative Decoding

From the results in the literature (e.g., [4]-[9]), the principles of DDC tend to provide a new promising design approach for an efficient and fault-tolerant real-time communication protocol for critical deadline dependent communication over unreliable wireless channels. This work is therefore based on DDC, but incorporating serially concatenated codes with iterative decoding. Concatenated codes using iterative decoding is a way of providing long codes with manageable decoding complexity, [10],[1]. The iterative decoder is sub-optimal and hence less complex, but for certain channels, it approaches the performance of the optimal decoder in an iterative fashion. The resulting DDC protocol presented here is based on concatenated hybrid automatic repeat request (CHARQ) techniques,

creating a flexible and dependable scheme to meet real-time constraints. Each packet has a certain t_{DL} and P_d required by the user or the application. These two parameters will be translated into a maximum number of retransmissions allowed and a maximum number of iterations allowed for the sub-optimal iterative decoding algorithm per transmission.

If simple, cheap transmitters and receivers are required, e.g., a mobile sensor with limited battery supply, the mapping of the QoS parameters onto the CHARQ-DDC protocol may be done using a look up table. If the transmitter and receiver can be more costly, the mapping can be done adaptively based on the current estimated channel conditions.

1.4. Overview of the Paper

This paper discusses different real-time aspects of the CHARQ-DDC scheme and the corresponding mapping of the QoS parameters. Each component of the CHARQ-DDC scheme is designed with the real-time application in mind. Consequently, the paper initially describes the different components in the scheme and their real-time design criteria. Section 2 contains a description of concatenated coding, iterative decoding, and a stopping criterion for the iterative decoding process in order to keep decoding time and complexity at a minimum. Section 3 describes the retransmission scheme, which is truncated due to the real-time constraints. In Section 4 we combine the contents of the two previous sections and describe the CHARQ-DDC protocol. The benchmark performance results of CHARQ-DDC, obtained by computer simulations, are presented in Section 5, and in Section 6 the results are interpreted from a real-time point of view. Concluding remarks are found in Section 7.

2. Iterative Decoding for Concatenated codes

Concatenated codes were introduced in [10] as a way of providing long codes with manageable decoding complexity. Serially concatenated codes consist of a cascade of an inner code, an interleaver and an outer code. The interleaver is used as an integral component, [1], permuting the bits from the outer decoder and thus creating longer, more powerful codes. The concatenated structure allows for a partitioned decoding process, where each component code is decoded separately. A modular iterative decoding strategy is then applied to the corresponding concatenated codes, exchanging information between the two component decoders [1], Figure 2. The exchange of *a posteriori probability* (APP), in terms of so-called extrinsic information [1], has proven most effective, and thus the inner decoder produces APPs of the inner information symbols, which is then passed through the deinterleaver and used by the outer decoder. The outer decoder, in turn, produces APP information,

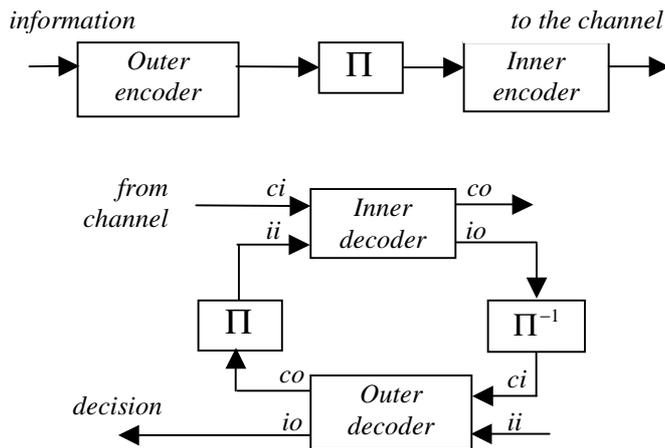


Figure 2. Encoder and decoder structure for a serially concatenated code. Π denotes the interleaver.

which is then passed via the interleaver back to the inner decoder. This exchange is recursively continued until a stopping criterion is met. The resulting decoder is sub-optimal with respect to optimal maximum likelihood (ML) decoding, but in most cases the result will converge to the ML performance in an iterative fashion.

Optimal ML decoding for the complete code is NP-hard and infeasible for large block lengths. However, the concatenated structure allows for optimal ML decoding of each component code separately. Information from these separate decoding processes is then exchanged between the component decoders in an iterative fashion. Since the component codes have very short block lengths as compared to the total code, a considerable complexity reduction is achieved even for a reasonably large number of iterations. The complexity is increasing linearly with the number of iterations as opposed to exponentially with the interleaver size [11]. In addition, only two relatively simple decoder components are needed rather than the very complex decoder for the total code. So, the concatenated structure enables iterative decoding, an elegant iterative solution to a complicated problem, resulting in substantial complexity reduction [1].

From a real-time communication point-of-view the total block size is also of concern. It is however possible to cater for a variety of block sizes by selecting appropriate component codes and interleaver sizes. A time and safety critical application benefits from the long powerful code yielding reliable communication, while the processing time of the iterative decoder is kept low. Within a DDC protocol, the iterative decoding algorithm also gives the opportunity to always deliver something to the receiver just before the deadline. We can offer a fast tentative response and progressively provide iterative refinements. This last-minute delivery can be complemented by a

measure of reliability or quality of the delivered data based on current APP information.

In an iterative APP decoder, reliability information in terms of APP information or extrinsic information is updated recursively [1]. Initially, there is much to be gained from iterating, but in most cases the performance reaches a point of diminishing returns. The number of iterations needed for convergence varies between packets and is generally not known. A common approach for stopping the iterative decoding process is to allow for a fixed number of iterations. This may lead to unnecessary iterations, or to performance degradation if the process is terminated prematurely. Applying a performance based stopping criterion these problems can be addressed. The convergence behavior of iterative APP decoding algorithms has been paid considerable attention recently. In [12] the average behavior of the output extrinsics as a function of the number of iterations is studied for parallel concatenated codes. Here, the extrinsic information is translated into equivalent log-likelihood ratios (LLRs), which provides a one-to-one mapping of extrinsics onto a scalar LLR [1]. A stopping rule based on thresholding the average LLR over a packet is found to be efficient. In this work, we consider the approach for serially concatenated codes. It should be noted that other methods, such as using an additional error detection code or using the error detection capabilities of the constituent codes might also be used. We have chosen the average LLR threshold because it is simple and provides information about the current error rate.

The stopping criterion is intended to stop the iterative process once the mean LLR is above the threshold, however, if the decoding algorithm does not converge for a specific packet, this may never happen. As we are dealing with real-time communication we must have an upper limit on the time to decode a packet. Since the decoding complexity, and hence the time required to deliver a packet with sufficient quality, is directly related to the number of iterations, we consequently need to have an upper limit on the number of iterations. Examining the convergence behavior in a concatenated system, it is generally noticed that for a majority of packets, convergence is observed after a fixed number of iterations. Consequently, in this work the maximum allowed number of iterations is set accordingly. A non-negligible number of packets may, however, converge after less iterations.

3. Automatic Repeat Request

In a packet based system, an Automatic Repeat reQuest (ARQ) scheme [13] can be used. Whenever a packet arrives, the receiver may choose to reject the packet, and instead request a retransmission through a feedback channel. To determine whether or not a retransmission should be requested the receiver checks the quality or the reliability of the received packet. One of the main

components in a DDC protocol is the retransmission scheme. Since we consider a time-limited channel, we must limit the number of retransmissions, hence the ARQ system becomes truncated [14]. The maximum number of retransmissions allowed is chosen according to the deadline.

A Hybrid ARQ (HARQ) scheme, first suggested in [15], uses an error control code in conjunction with the retransmission scheme. Consequently, it tries to decode the received code word first and only requests a retransmission if the uncertainty of the decoding decision is considered too high, i.e., if the decoder output is below a certain reliability threshold. There are different methods of determining whether a decoding decision is sufficiently reliable and hence different criteria for requesting a retransmission. The choice of method significantly affects the character of the retransmission scheme. When an error detection code, e.g. a cyclic redundancy check code (CRC) [16], is used for requesting a retransmission we have a two-code approach since two different codes are concatenated [10] for HARQ purposes. Alternatively, in a one-code approach, e.g., [17], reliability information within the decoding process is identified and used to determine whether we need a retransmission or not.

In a pure ARQ scheme, rejected packets are discarded. Previously received packets may, however, be used for so-called packet combining, in order to improve performance. There are two major types of packet combining, diversity combining, [18], and code combining, [19]. Diversity combining is in general done before the decoder whereas code combining is generally done within the decoder.

4. Concatenated Hybrid ARQ

By *Concatenated Hybrid ARQ* (CHARQ) we mean a HARQ scheme using concatenated codes as the error control code, but also using concatenation between retransmissions. Now, soft extrinsic information may also be passed between retransmissions to be used in the iterative decoding process. The concept of concatenated HARQ is a relatively new principle. The first time a turbo code, i.e. a parallel concatenated code with iterative decoding, was used in a HARQ scheme was in [20]. The turbo code is applied in a two-code approach, where an outer error detection CRC code is used both for terminating the iterations and for generating retransmission requests.

The CHARQ protocol presented here is a HARQ scheme based on serially concatenated block codes. The convergence behavior of the iterative decoder yielding the average LLR threshold stopping criterion is used to define the applied retransmission criterion. Consequently, we have a one-code approach where a retransmission is requested when the iteration-stopping criterion is not fulfilled after the maximum allowed number of iterations. Since the retransmission scheme is intended for wireless

real-time applications, the number of retransmissions is limited and the ARQ scheme becomes truncated. The final decoder results are always delivered prior to the deadline, regardless of quality.

We apply the diversity combining technique *equal gain combining* [21]. This implies that the receiver just averages over all received copies of a packet to produce a combined packet for decoding. This requires a slightly more complex decoder, since the erroneous copy has to be stored to enable the combining procedure.

Since the error control code used in this HARQ scheme is a concatenated code with iterative decoding, another kind of packet combining technique is also possible. The extrinsic information of the code bits, co , from the outer decoder, Figure 3, which when interleaved is the same as the *a priori* information on the information bits, ii , to the inner decoder, can be saved and used as *a priori* information when decoding the *new copy*, r_1 , of the packet received from a retransmission, [20]. The extrinsic information to the inner decoder from the previous transmission is denoted ii_0 in Figure 3. It should be noted that the information saved from the previous retransmission is only used once and we will have a new updated value in the next iteration.

The strategy of saving the extrinsic information can be seen as a code combining technique. The process of saving the extrinsic information requires no extra memory since this information is already saved to be passed and updated between the decoders. It is simply not reset when a new copy of a packet arrives. The strategy of saving the extrinsic information can also be seen as a doping procedure to speed up convergence of the iterative decoder, [22]. The concept of doping is based on providing side information to the decoding process. The extrinsic information from the decoding process of the previous transmission constitutes useful side information. The use of prior knowledge leads to a constructive bias, speeding up convergence. Considering the strategy of saving the extrinsic information as a doping operation, equal gain combining may be used in conjunction with the saved extrinsic information, leading to joint diversity and code combining.

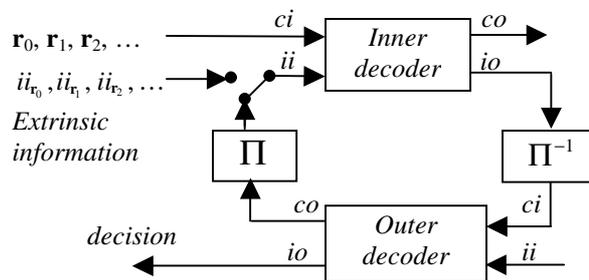


Figure 3. Code combining or doping with the extrinsic information from the previous transmission.

5. Performance Results

We have used binary phase shift keying (BPSK) modulation [2] throughout this work, as it is a commonly used method in many existing hardware platforms. A simple and frequently used mathematical model for the communication channel is the additive white Gaussian noise (AWGN) channel [2] which models thermal noise present in all electronic equipment. Further, we consider serially concatenated block codes based on two binary Reed Solomon RS(7,3) codes with a pseudo-random interleaver of size 945 used in conjunction with the ARQ protocol [23]. An error free feedback channel is assumed throughout this work.

In a wireless communication system amplitude fluctuations due to variations in the radio propagation environment are commonly experienced [21]. However, since this paper is mainly focusing on the *development* of a powerful iterative DDC scheme intended for real-time communication rather than its strict real-time performance analysis, we consider a static AWGN channel for benchmarking the protocol design. For this purpose it is also convenient to assume an error free feedback channel. The reported performance experiments therefore represent best-case limits rather than examples of performance over specific wireless channels.

The maximum number of iterations was chosen, based on simulation results, to be seven and the maximum number of retransmissions allowed was chosen to be four. Consequently, the maximum number of iterations that can ever occur is $7 \cdot 5 = 35$, yielding an upper bound on the execution time or communication time of each task. The LLR threshold was set to ± 10 .

The following procedure is applied:

1. Use iterative decoding on the received packet, checking the LLR threshold after each iteration. If the stopping criterion is fulfilled go to 3. If seven iterations are reached, go to 2.
2. If four retransmissions are made, go to 3, else request a retransmission. Do packet combining, if any. Go to 1.
3. Output hard quantized values.

In Figure 4, schemes allowing from 0 to 4 retransmissions are considered and compared in terms of bit error rate (BER) plotted against E_b^0/N_0 , (in dB) where E_b^0 is the bit energy for the regular scheme with no retransmissions and N_0 the one-sided spectral density of the channel noise. For high signal to noise ratio, $\text{SNR} = E_b^0/N_0$, all the curves tend towards the regular scheme without ARQ, since the probability of a retransmission tends to zero. For very low SNR, the curves of the different schemes are also on top of each other, since the probability of a retransmission tends to one. Virtually all transmissions are rejected and thus there is nothing to gain from a retransmission since the final packet must be accepted. The region between these two extreme values,

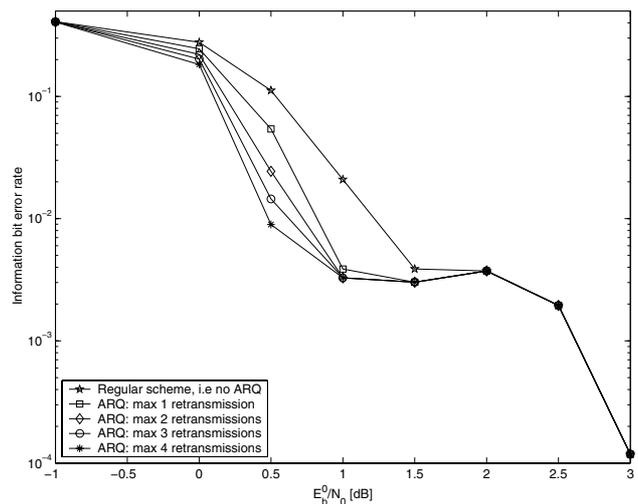


Figure 4. Information BER as a function of E_b^0/N_0 for different ARQ schemes. Erroneous packets are discarded.

however, is the working area of the ARQ schemes and a noticeable gain can be observed. The reason for showing the different truncation lengths in one figure is to see the improvements when more time is allocated to allow for additional retransmissions.

At high SNR, a loss in performance due to the stopping criterion can be observed. This is a result of the fact that the threshold is fixed for all values of SNR. The performance in terms of BER for a particular stopping criterion should, however, be compared to the number of iterations actually needed. It can be concluded that, although we have a considerable performance loss for high SNR, the average number of iterations needed to fulfill the stopping criterion is low [23]. It follows that we gain real-time benefits at the expense of BER performance. A truncated ARQ system offers most improvement in the region where the initial probability of retransmission is relatively high at the same time as the probability for accepting a packet within the allowed number of retransmissions is high. This is the case for the region 0-1.5 dB [23]. Consequently, in this region, the stopping criterion is good since we obtain faster convergence at virtually no performance loss. It follows that even though the chosen stopping criterion has an error plateau at 1.5-2 dB, it is still relevant in the SNR area where we expect to operate.

In Figure 5 the BER for the same schemes are shown, but now the extrinsic information is saved and used when decoding after a retransmission. It can be seen that the BER for low SNR is improved as compared to the case in Figure 4 without any packet combining technique. Note that the curves representing a regular scheme with no ARQ in both Figure 4 and Figure 5 are the same since no packet combining is made for these curves and the same stopping criterion is used.

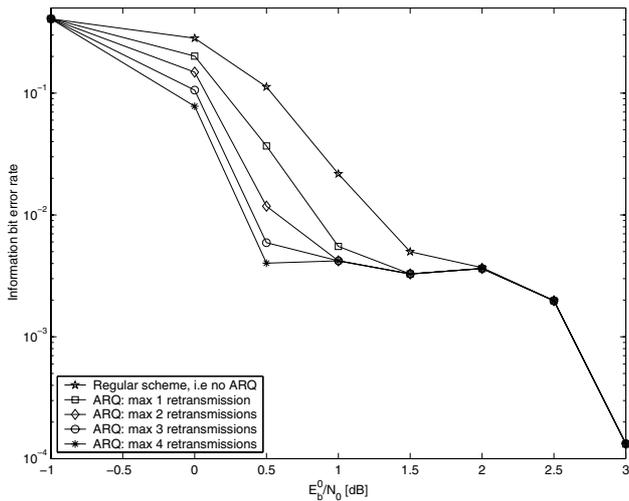


Figure 5. BER as a function of E_b^0/N_0 for different ARQ schemes. The extrinsic information is used for doping.

When equal gain combining is applied, additional memory is required to store erroneous copies used in the combining process, but the gain in terms of BER is considerable as can be seen in Figure 6. In the region from 0-1 dB, it is observed that the BER increases when the SNR is increasing. However, the SNR is defined here with reference to a regular scheme without retransmissions. An ARQ scheme has an effective SNR, which is related to the average number of retransmissions. Equal gain combining makes use of old copies pertaining to the same packet, and hence constructively exploits the increased effective SNR. Going from 0 dB to 1 dB, more and more transmissions are considered reliable after the first transmission as opposed to the second transmission [23], and thus, the effective SNR decreases, even though the reference SNR increases. Consequently, a scheme using an increased effective SNR to improve performance will also suffer when the effective SNR is reduced. This together with the imperfect stopping criterion explains the increase in BER observed between 0 dB and 1 dB.

In the region from 0 dB to 1.5 dB the schemes using between one and four retransmissions have identical performance and their curves are on top of each other. From 2 dB to 3 dB all five schemes are the same since the probability of a retransmission is zero. At -1 dB, however, we only have data from the regular scheme and the scheme allowing one retransmission. This is due to the very low probability of errors occurring in a scheme using equal gain combining of two retransmissions or more. We simply have not encountered any errors during this simulation.

We now have the question of whether saved extrinsic information may improve the equal gain combining technique or speed up the convergence of the iterative decoder. Observing the BER of a scheme using equal gain

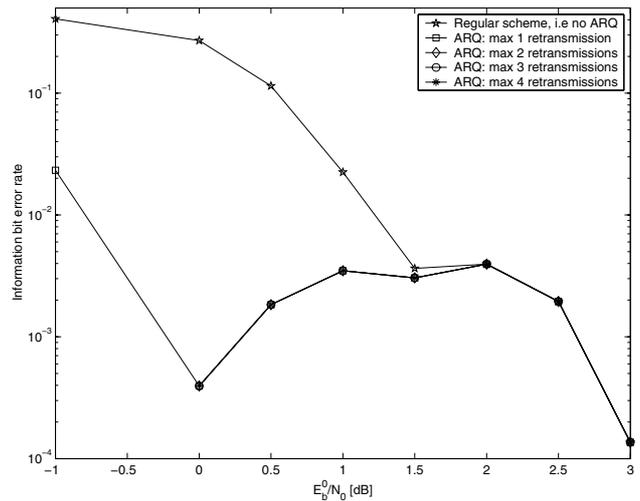


Figure 6. BER as a function of E_b^0/N_0 for different ARQ schemes. Equal gain combining is used.

combining, while simultaneously saving the extrinsic information, we see no immediate improvement, but can, however, conclude that the inclusion of the extrinsic information does not reduce performance in terms of BER.

The performance of an ARQ scheme is usually measured both in terms of BER and in terms of throughput. In our case we are not only concerned with the number of retransmissions, but also the number of iterations made for each retransmission. We want to know if the packet combining technique considered improves both the throughput and the convergence speed. First we compare the scheme saving the extrinsic information to that where no packet combining technique is used. The relative frequency of occurrence of each iteration is plotted in Figure 7. Note that after seven iterations a retransmission occurs. Consequently, iteration number eight in Figure 7 corresponds to the first iteration of a newly received retransmission. The black columns in Figure 7 correspond to the scheme discarding erroneous copies of a packet. When a new packet arrives, the iterative process is restarted. This can be seen as a staircase characteristic with a step every seventh iteration, starting at one since each new packet requires a few iterations to converge. The lighter columns correspond to the scheme saving the extrinsic information. The number of iterations decreases smoothly due to the fact that the decoder does not have to restart every time a retransmission arrives. Note that the results for the first seven iterations are the same for the two different schemes, as no retransmissions have yet occurred.

In Figure 8 the equal gain combining schemes are compared in a similar way. Here, one scheme is using the saved extrinsic information and the other scheme is not. The most noticeable difference between Figure 7 and Figure 8 is that equal gain combining significantly reduces

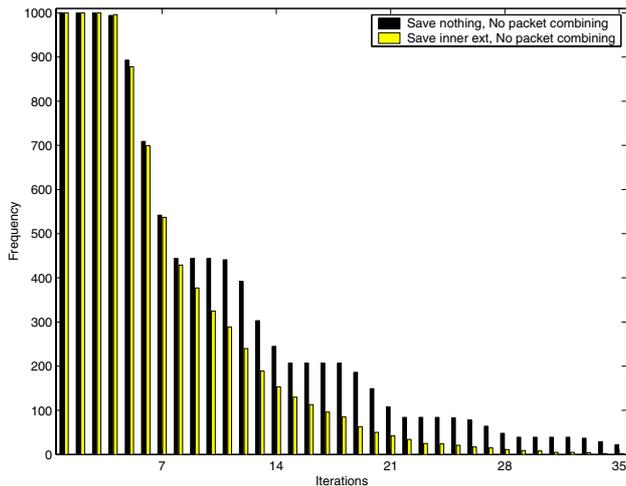


Figure 7. Relative frequency of occurrence for each iteration. A scheme without packet combining is compared to a scheme saving the extrinsic information, plotted at $E_b^0/N_0=0.5$ dB.

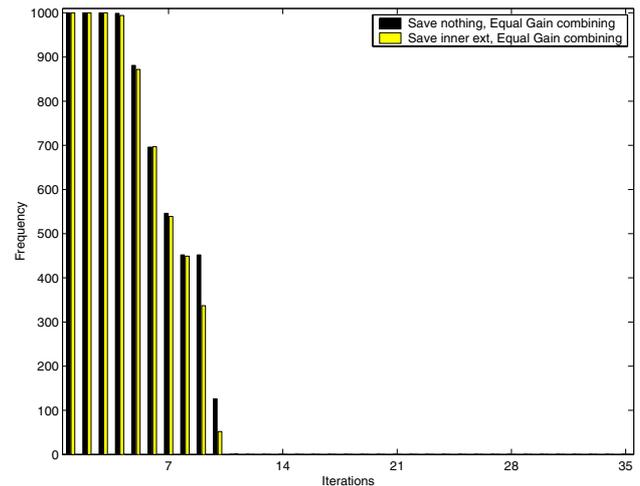


Figure 8. Relative frequency of occurrence for each iteration, plotted at $E_b^0/N_0 = 0.5$ dB. Both schemes employ equal gain combining - one is saving the extrinsic information, the other is not.

the number of iterations. For the first seven iterations in Figure 8 the two schemes are the same, since no retransmissions have been made and hence no packet combining is performed. However, we can still see an improvement in the convergence speed when the extrinsic information is saved and used, since the columns for iteration nine and ten have a lower frequency for this scheme. Additional results, together with a more detailed analysis can be found in [23].

Clearly, the best performance results are obtained when equal gain combining is applied. Therefore, this technique should be used whenever the increased complexity required to store erroneous copies of a packet is affordable. Saving the extrinsic information does not require any increase of the decoder complexity but still improves performance regardless of whether equal gain combining is used or not.

In Figure 7 and Figure 8 the relative frequency of occurrence for each iteration is given as measure of decoder complexity. The x-axis may also be seen as a measure of logic time. However, when a real-time scheme is considered, the transmission propagation time has to be taken into account. Consequently, an iteration cannot be compared directly to a retransmission. In the bar charts on relative frequency of occurrence for each iteration, one may use one time unit to make one iteration, while a retransmission with a following iteration takes, for example, five time units.

6. Mapping the QoS parameters to the DDC scheme

The principles of DDC are illustrated in Figure 1. The number of retransmissions, the retransmission criterion

and the packet combining technique are features that will all affect the DDC scheme in different ways. As stated before, in the beginning of the time window, a message is transmitted. The iterative decoder tries to decode the message while checking the stopping criterion. If it reaches the maximum allowed number of iterations, a retransmission is made. This procedure will be repeated until the requested level of P_d has been achieved.

From the bar charts in Section 5 we have the probability of each iteration occurring and from the BER curves we have the probability of bit error for different truncation lengths. Combining these two diagrams, we can plot the probability of correct delivery before deadline, P_d , as a function of time [9]. We assume again an error free feedback channel and we do not take into account the possible variations in transmission time of a packet, as these factors will affect the compared schemes equally. For example, the time it takes for each transmission or retransmission may vary due to traffic intensity, buffer sizes, propagation environment, etc. These statistical variations become part of the constraints imposed on the protocol design, affecting the QoS parameters through P_d . Selecting a specific time duration for each transmission translates into a probability for the transmission to be completed within the duration. The corresponding analysis is therefore strongly dependent on specific implementation issues and thus, here we adopt the concept of logical time. The true transmission time in a statistical representation will affect all the considered schemes equally and hence merely correspond to a scaling of the x-axis in some manner. In addition we do not put an absolute value on the time to perform one iteration, since it will be hardware dependent.

In Figure 9 four different schemes are compared, scheme A, where erroneous copies of a message are

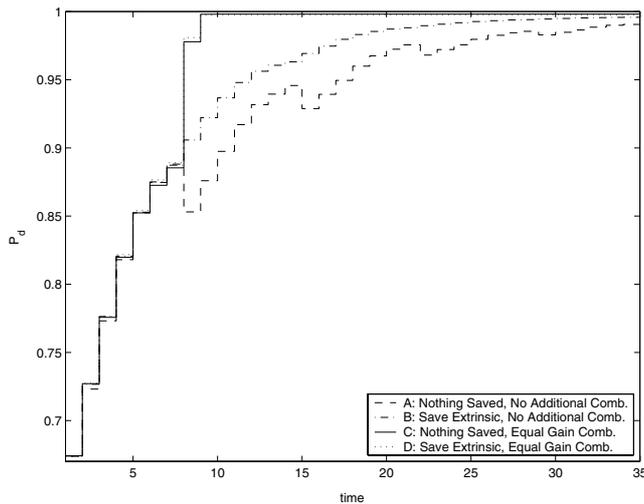


Figure 9. P_d plotted as a function of time for schemes using different packet combining techniques. The signal to noise ratio is 0.5 dB.

discarded, scheme B, saving the extrinsic information, scheme C, using equal gain combining and finally scheme D, using both equal gain combining and saving the extrinsic information. All the schemes have curves with a staircase characteristic. The length of a step in the staircase depends on the time to do one iteration and, in case of a retransmission every seventh iteration, the transmission time together with the time to do one iteration. The height of each step yields the performance improvement. For each iteration some additional bits can be successfully decoded and results in a new step. For the first seven iterations or steps the performance of the schemes are the same since no packet combining can be done until a retransmission has occurred. As iteration number eight is the first iteration of a newly received packet, this is where we can see a difference between the schemes. Scheme A, which discards erroneous copies of a message, will actually lose in performance in the beginning when a retransmission is requested. This is due to the fact that the whole packet is deemed erroneous and hence even correctly decoded bits will be discarded. Scheme B on the other hand does not have to restart but uses the saved extrinsic information; hence a better performance is achieved faster. Scheme C and D both use equal gain combining and the gain at the eighth iteration is considerable. Scheme D also uses extrinsic information to speed up convergence and we can conclude that the procedure does actually achieve slightly better performance faster. However, it is only for the first few iterations, after which performance is virtually identical. Recall that the extrinsic information is only used once, i.e. in iteration number eight, fifteen, and so on.

It should be noted that at every step there is an increased probability that the packet is correct and retransmissions will cease. Hence, using all allowed

retransmissions and iterations is in fact the worst case. Most of the time we will stop earlier because the stopping criterion is fulfilled. The BER should be considered as a mean of all trials in the simulation regardless of when the stopping criterion for each trial is fulfilled.

The P_d is obviously SNR dependent. The retransmission scheme can handle temporary amplitude fluctuations due to channel noise, but if the conditions deteriorate over a longer period of time, the protocol should adapt accordingly. Some knowledge of the current channel conditions is available in most wireless communication system, and consequently can be used to guide the selection of the DDC protocol components in terms of coding and decoding strategies, ARQ scheme, and retransmission strategy such that the required QoS is maintained.

The QoS parameters will thus translate into the maximal number of retransmissions allowed and the number iterations permitted for each retransmission, given the components of the DDC protocol. If the requested P_d can be achieved after the number of iterations and retransmission given by the t_{DL} – the request is accepted. If the request is rejected the application can renegotiate by stating a lower value on one or both of the QoS parameters. Depending on the application in question one may choose to prolong the deadline or reduce the requested P_d . If, for example, we have a sensor value that is over-sampled but the correctness is critical one may choose to prolong the deadline. A multimedia application may, on the other hand choose to reduce the requested P_d in order to meet the required deadline.

The mapping of the QoS parameters can be done using a lookup table or, if more advanced structures are allowed the mapping can be done in runtime. The latter will of course enable additional flexibility by continuously adapting to channel variations. Which packet combining method to use is determined at design time by the complexity requirements and the memory resources allowed for the decoder.

7. Conclusions

We have put forward a framework for transmitting real-time data over a radio channel based on the Quality of Service parameters deadline (t_{DL}) and probability for correct delivery before deadline (P_d), using a scheme called CHARQ-DDC. The required values for the deadline and the probability of correct delivery can be negotiated by the application, thus forming a transmission mechanism allowing flexible admission control. Serially concatenated codes with iterative decoding were incorporated into a DDC scheme. The stopping criterion for the iterative decoding process was used as a retransmission criterion constructing a concatenated hybrid ARQ scheme with iterative decoding. The goal was to reject packets with

slow or no convergence early and request a retransmission to hopefully receive a packet with fast convergence.

Two different packet combining techniques for the CHARQ scheme were presented and evaluated. Initially the diversity combining technique equal gain combining was applied, resulting in substantial performance improvements. However, since the CHARQ system is based on serially concatenated codes using iterative decoding, a different category of packet combining techniques can be used. The iterative decoder passes extrinsic information between its constituent component decoders, which can be saved from one transmission and then used in the decoding process of a retransmission. This technique can be viewed as a code combining technique, and using it leads to performance improvement in terms of BER and convergence speed at very low additional cost in decoder complexity. Saving the extrinsic information to use in the decoding process of a retransmission can also be viewed as a doping operation of the iterative decoder and with this observation in mind, it can be used in conjunction with equal gain combining. It is concluded that the extrinsic information does speed up the convergence process of the iterative decoder.

The QoS parameters were then mapped onto the CHARQ-DDC protocol to create a fault-tolerant scheme in presence of communication errors. The mapping of the QoS parameters to the CHARQ-DDC scheme allows a flexible admission control.

8. References

- [1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo codes," *Proceedings of the International Conference on Communications*, Switzerland, pp. 1064-1070, May 1993.
- [2] J. G. Proakis, *Digital Communications*, McGraw-Hill Book Co., New York, 1995, 3rd edition.
- [3] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379-423 and pp. 623-656, October 1948.
- [4] H. Bengtsson, E. Uhlemann and P.-A. Wiberg, "Protocol for wireless real-time systems", *Proc. of the 11th Euromicro Conference on Real Time Systems*, York, England, UK, June 9-11, 1999.
- [5] W. Zhao, J. A. Stankovic and K. Ramamritham, "A window protocol for transmission of time-constrained messages," *IEEE Transactions on Computers*, vol. 39, no. 9, pp. 1186-1203, September 1990.
- [6] S.-K. Kweon, K. G. Shin and Q. Zheng, "Statistical real-time communication over Ethernet for manufacturing automation systems," *Proceedings of the IEEE Real-Time Technology and Applications Symposium*, Phoenix, Arizona, pp. 192-202, May 1999.
- [7] T. F. Abdelzaher, E. M. Atkins and K. G. Shin, "QoS negotiation in real-time systems and its application to automated flight control," *IEEE Transactions on Computers*, pp. 1170-1183, Nov. 2000.
- [8] J. Kim and K. G. Shin, "Performance evaluation of dependable real-time communication with elastic QoS," *Proceedings of the International Conference on Dependable Systems and Networks*, pp. 295-303, 2001.
- [9] E. Uhlemann, T. M. Aulin, L. K. Rasmussen and P.-A. Wiberg, "Deadline dependent coding - a framework for wireless real-time communication," *Proceedings of the 7:th International Conference on Real Time Computing Systems and Applications*, RTCSA 2000, Cheju Island, South Korea, pp. 135-142, Dec. 12-14, 2000.
- [10] G. D. Forney, Jr., *Concatenated Codes*, M.I.T. Press, Cambridge, MA, 1966.
- [11] P. Robertson, E. Villebrun and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the Log domain," *Proceedings of the IEEE International Conference on Communications (ICC)*, Seattle, Washington, USA, pp. 1009-1013, June 1995.
- [12] A. C. Reid, T. A. Gulliver and D. P. Taylor, "Convergence and errors in turbo-decoding," *IEEE Transactions on Communication*, pp. 2045-2051, December 2001.
- [13] S. Lin, D. J. Costello Jr. and M. J. Miller, "Automatic-repeat-request error control schemes," *IEEE Communications Magazine*, vol. 22, no. 12, pp. 5-16, December 1984.
- [14] E. Malkamäki, *Performance of Error Control over Block Fading Channels with ARQ Applications*, Ph.D. dissertation, Helsinki University of Technology, October 1998.
- [15] J. M. Wozencraft and M. Horstein, "Coding for two-way channels", *Research Laboratory of Electronics*, MIT, MA, Technical Report 383, 1961.
- [16] S. Lin and D. J. Costello Jr., *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983.
- [17] H. Yamamoto and K. Itoh, "Viterbi decoding algorithm for convolutional codes with repeat request," *IEEE Transactions on Information Theory*, vol. 26, no. 5, pp. 540-547, September 1980.
- [18] P. Sindhu, "Retransmission error control with memory," *IEEE Transactions of Communications*, vol. 25, no. 5, pp. 423-429, May 1977.
- [19] D. Chase, "Code combining - a maximum-likelihood decoding approach for combining an arbitrary number of noisy packets," *IEEE Transactions of Communications*, vol. 33, no. 5, pp. 385-393, May 1985.
- [20] K. R. Narayanan and G. L. Stüber, "A novel ARQ technique using the turbo coding principle," *IEEE Communications Letters*, vol. 1, no. 2, pp. 49-51, March 1997.
- [21] M. Schwartz and W. R. Bennett, *Communication Systems and Techniques*, McGraw-Hill, Inc. 1966.
- [22] S. ten Brink, "Code doping for triggering iterative decoding convergence," *Proceedings of the IEEE International Symposium on Information Theory, ISIT 2001*, Washington D.C., USA, p. 235, June 2001.
- [23] E. Uhlemann, *Hybrid ARQ using serially concatenated block codes for real-time communication - an iterative decoding approach*, Licentiate Thesis, Chalmers University of Technology, Göteborg, Sweden, October, 2001. Available at: www.ce.chalmers.se/staff/bettan/EU_lic.pdf.