Towards A Model for Making A Trade-off Between QoS And Costs

Lodewijk T. Smit, Gerard J.M. Smit, and Paul J.M. Havinga
Department of Computer Science, University of Twente, Enschede, the Netherlands
email:smitl@cs.utwente.nl

Jos A. Huisken, Kees G.W. Goossens, and John T.M.H. Dielissen
Philips Research Laboratory, Prof. Holstlaan 4, 5656 AA Eindhoven, the Netherlands

Abstract

Mobile handheld devices need to have low energy consumption and should be able to adapt to their environment. To achieve this, a model is needed that makes a trade-off between Quality of Service (QoS) and costs at run-time. A case study of a rake receiver in combination with a turbo decoder has been done to find out the requirements of such a model. Our conclusion is that it is not feasible to express the whole behavior of the system in an analytical way due to a complex external unpredictable time-variant environment. Instead a control system is proposed.

1 Introduction

Due to limited energy resources, mobiles need to have low energy consumption. To obtain a low-power consumption the system has to be energy efficient and effective. Energy efficient means the use of low-power hardware and efficient algorithms. Effective means doing the work that is needed to satisfy the quality constraints only. Most applications are designed for a worst-case scenario and waste energy in *not* worst-case situations by doing more work than is strictly needed. To save energy, applications have to adapt to the environment of the mobile to do just enough work to satisfy the quality constraints. The environment of the mobile has a dynamic nature, and the changes are unpredictable beforehand. Therefore the applications have to adapt to their environment at run-time, instead of making choices at design time by a designer.

Our goal is to develop a model that allows us to make a trade-off between quality of service and energy consumption. The model can be used to minimize the energy consumption of the whole system while satisfying certain quality constraints. Further, the model has to be 'light weight' in terms of processing power to prevent that computing a decision consumes more energy than the energy consumption saved by the model. The development of this model is part of the Chameleon project. A more in-depth discussion about the different methods to save energy for mobile multimedia handheld devices and the Chameleon project can be found in [8]. The model mentioned above will be referred to as the Chameleon model.

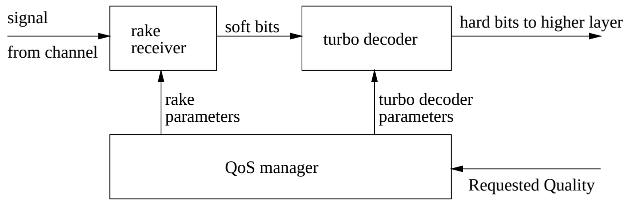

Figure 1. The System

To investigate what this will mean in practice, we decided to do a case study with a rake receiver [5] combined with a turbo decoder [2, 3], which can be used in a third generation telephony [4] terminal. The general idea is that there is a trade-off between the rake receiver and the turbo decoder. A very good rake receiver (consuming a lot of energy) gives a good output to the turbo decoder, so the turbo decoder has an easy job correcting the few errors. On the other hand a bad rake receiver (consuming only a little energy) needs a good turbo decoder (consuming a lot of energy) to obtain the same bit error rate (BER) as in the previous case.

2 System Description

Figure 1 shows the system configuration. The rake receiver receives signals from different paths with different delays (there are multiple paths due to reflections) and combines them to construct one output signal. Each recognized path element is correlated with a code in a finger to retrieve the original signal. This basic principle of a rake receiver is shown in Figure 2. For a more detailed description, see [4, 9].

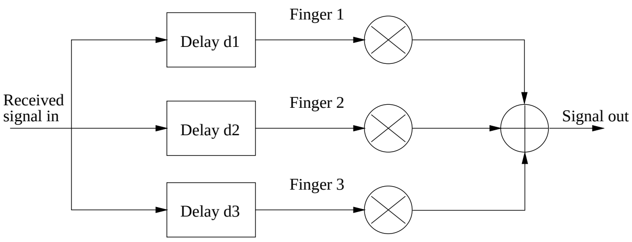

Figure 2. Basic Principle Of A Rake Receiver

The soft output signal of the rake receiver is a signed x bits value. A high output value means a high probability that a 1 was transmitted; a low value means a high probability that a 0 was transmitted. The turbo decoder is an error correcting decoder, which will get the values of the rake receiver as input and produces hard bits (0 or 1) on the output. A block diagram is of the turbo decoder is shown in Figure 3. The turbo decoder is constructed out of two decoders, an interleaver, and a deinterleaver. The output of each decoder is passed to each other, to improve the output of the decoders by use of a-priori information. This iterative principle gives the turbo decoder a better performance than a conventional decoder. Between the decoders, the data is (de)interleaved. More details about turbo decoding can be found in e.g. [10].

The Quality of Service (QoS) manager (see Fig. 1) determines the (current) values for the parameters of the rake and the turbo decoder making a tradeoff between the costs and the performance using the Chameleon model. The minimal requested quality is an input constraint for the QoS manager.

There are a lot of parameters that affect the quality (and consequently the costs). However, many parameters are determined by the external environment (e.g. number of paths and interference). These "external" parameters have a considerable influence on the system. Therefore the QoS manager has to take into account the effects of changes of these parameters, but the QoS manager cannot change them. Further, the values of some parameters are determined at or already before design time (e.g. chiprate) and cannot be changed (easily) at run-time.

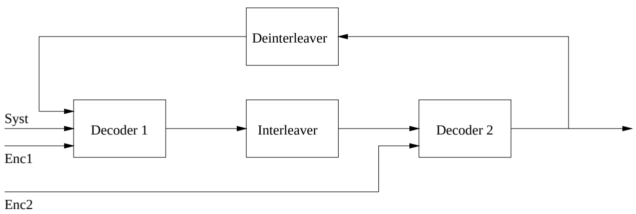

Figure 3. Block Diagram Of Turbo Decoder

The two most important trade-off parameters for a rake receiver and a turbo decoder that can be changed by the Chameleon model are the number of fingers and the number of iterations. The number of fingers is equal to the number of multipaths that are combined, the number of iterations states how many times the turbo decoding algorithm runs before making a hard decision about the value of the output bit. At first start, we will restrict our QoS manager to these two parameters. Further, we define that quality is equivalent to the bit error rate performance and that costs are equivalent to the arithmetic complexity.

3 Model

Our goal is to construct a model that minimizes the costs given a set of constraints. This looks like a typical (non-linear) optimization problem known from operations research [12]. The cost function to be minimized is the sum of the cost function for the rake receiver and the cost function for the turbo decoder. The set of constraints exists of (1) a quality function that must satisfy a minimum threshold for the required minimum quality of service and (2) limits on the range of parameters (e.g. the number of finger must be between 1 and 10). Note that the quality function is not the sum of two quality functions for the rake receiver and the turbo decoder, because the quality of the turbo decoder is dependent on the quality output of the rake receiver. The cost and quality functions will be discussed below in more detail.

3.1 Cost Function

The essence of a rake receiver is correlation of the incoming chips (transmitted pulses) with a code. If the number of operations for this correlation are c per chip, then the total number of operations per bit needed for the arithmetic of the rake receiver are $c \cdot sf \cdot co \cdot ch$, where sf is the spreading factor (sf is equal to number of chips per bit), co is the number of correlators and ch is the number of channels. Note that correlators are needed also for channel estimation and searching, so the number of correlators is greater than the number of fingers. Further, most rake receiver designs are more complex (having tracking algorithms, advanced filters, etc.) that make the performance better and the arithmetic complexity higher. In most cases, after the correlation the signal is multiplied with a gain factor before the combining operation, also raising the computation costs. So, the exact costs depends on the design and implementation, but in general the costs per bit are proportional of $sf \cdot co \cdot ch$.

With regard to the turbo decoder, the number of operations per bit is about linear to the number of iterations of the turbo-decoding algorithm. There exist different turbo decoding algorithms with different costs and performance. Three frequently used algorithms are Max-Log-MAP, Log-MAP and SOVA. The costs per bit are shown in the table below [6]:

Operation	Max-Log-Map	Log-MAP	SOVA
max ops	5 x 2^M-2	5 x 2^M-2	$3(M+1)+2^M$)
additions	10 x 2^M+11	15 x 2^M+9	2 x 2^M+8
mul by ± 1	8	8	8
bit comps			$6(M+1)$
look-ups		5 x 2^M-2	
total ops	15 x 2^M-17	25 x 2^M+13	3x2^M+9xM+25
for M=3	103	213	76

A turbo decoders contains also two (de)interleavers. The number of operations per bit for the interleavers dependents on their implementation. We assume these costs about i. Each iteration the decoding algorithm is executed twice. The number of operations needed for the turbo decoder per bit is about $n \cdot (2 \cdot 103 + 2 \cdot i)$, using the Max-Log-MAP algorithm and an encoder with 3 memories, where n is the number of iterations. The (third generation) Universal Mobile Telecom System (UMTS) always uses three memories in the encoder. So, the costs are linear with the number of executed iterations. Note that this is only the arithmetic cost, e.g. control costs are not included.

The cost function of the system is simply the sum of the cost function of the rake receiver and the cost function of the turbo decoder.

3.2 Quality Function

Problems arose when we tried to define the quality function which gives the bit-error-rate for a certain situation. Different approaches we used:

- Analytical derivation of quality function
- Approximate a quality function
- Eliminate the need for a quality function

3.2.1 Analytical derivation of quality function

For both the rake receiver and the turbo decoder, there are a lot of parameters that influence the quality of service. Most of them are unpredictable (quickly changing) environment parameters. There are almost no analytical expressions known from literature for expressing the quality of service, and those that are, are restricted to special cases. Most results in both rake receiver and turbo decoder research areas are obtained by simulations, not by analytical work. A complicating factor is that the quality of the turbo decoder is dependent on the quality of the output of the rake receiver, so the quality function cannot be split up easily into two separate parts. Most research focuses on the rake receiver or on the turbo decoder, but usually not on their combination. Furthermore, most research for the turbo decoder assumes an additive white Gaussian noise (AWGN) channel model, whereas research with regard to the rake receiver usually assumes a Rayleigh fading or Racian channel model. These differences make it difficult to combine these two subjects and to derive a quality function for the whole system. Our conclusion is that when experts with thorough system knowledge use simulations for small parts of the whole system due to the complexity, it is not feasible to derive an analytical function for the whole system.

3.2.2 Approximate a quality function

If it is not feasible to define a quality function that gives an exact indication of the quality, we can try to derive a quality function that gives an approximation of the quality. Because it's not possible to do it in an analytical way, we need data. The idea is to derive an analytical function by means of regression and interpolation. To get the necessary data we could use results from literature or do a lot of simulations. We started with the first possibility.

Researchers only cover one or a few aspects of the rake receiver or turbo decoding due to the complexity. All the parameters except one or a few are fixed. Results are presented for this specific aspect. However, different researchers use different simulation environments by making different assumptions (such as different channel models, neglecting sometimes certain things (e.g. Doppler effect), simplifying the real world (e.g. assuming a time invariant environment)) and fixing different parameters. These differences make it hard to combine the different results and to draw valid conclusions for the overall system. Beside this fact, the amount of data is very limited. Most times, only a few plots are presented. So, there are only a few data points. However, due to the large number of parameters, there is a very large N-dimensional space. This means that there is not enough data to derive an analytical function for the quality of service. Note that doing the simulations by ourselves will not solve the last problem.

3.2.3 Eliminate the need for a quality function

After this observation, the question raised whether it is really necessary that the quality of service function has to be known. Of course, this is the most convenient situation, but in a lot of other real life circumstances, things can be regulated without knowing all the parameters and all the details. E.g. a thermostat can perfectly regulate the temperature without knowing anything about the environment (like incoming sunlight, status of fireplace, open doors) just by measuring temperature and regulating the amount of appended heat. The only "knowledge" of the system is that adding heat will increase the temperature.

Something similar can be done in our case. We know that changing a parameter will make the quality better or worse. However, the exact amount of the difference is not exactly known. To get a better indication about how much a change will contribute to a quality increase or decrease, a first derivative could be calculated over the last differences. There are a few potential problems with this approach. First, the number of parameters on the receiver side that could be changed should be low. Due to the fact that it is unknown what the quality increase or decrease will be, different combinations have to be tried out, if the quality has to be changed. If there are too many parameters, the number of possibilities will be too large. Secondly, as already stated, this approach could not give a prediction of the quality on beforehand. Consequently, a kind of learning curve is needed, before a 'stable' situation is reached. A third problem is a parameter with a quality function with more than one optimum. The discovered optimum could be a local optimum instead of a global optimum.

The configuration will look like Figure 4.

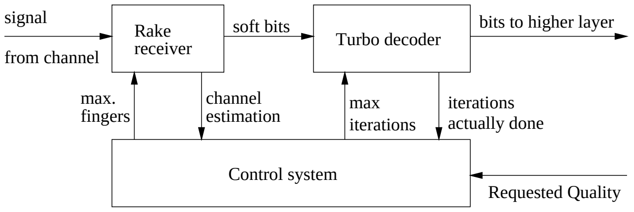

Figure 4. The Control System

Quality Measurement

Both the rake receiver as well as the turbo decoder can be implemented with a facility to deliver information that can be used to determine the quality of service. The rake receiver can deliver the channel estimation information (which has to be collected anyway). The channel estimation provides information about the number of paths and their phase and amplitude, so this can be used as an indication of the quality of the input channel. With regard to the turbo decoder early stop algorithms [7, 13] are known, which prevent unnecessary iterations if the data no longer contains errors (that the turbo decoder is able to correct). The average number of iterations that is needed to correct the data is a measure for the BER performance of the output of the turbo decoder.

The quality of the system is determined by the final output, so the output of the turbo decoder has to be compared with the given quality constraint. The quality information given by the rake receiver is however also useful. This information can be used as extra information used to fine-tune the decisions made by the model (e.g. ensure that the number of finger is always equal or smaller than the number of paths).

Quality Function Characteristic

In case of the rake receiver, adding fingers will give a better performance, until the number of fingers will become larger than the number of recognized multipaths [1]. If the number of fingers becomes too large, then the performance will degrade, because of adding noise for the unrecognized paths to the signal with the maximal ratio combining. The plot of the quality as function of the number of fingers will look like the right plot of Figure 5.

In case of the turbo decoder, the quality will be the same or better after each iteration, but never be lower. The first iterations deliver more gain than the last iterations [11]. Therefore, the plot of the quality (y-axis) as function of the number of iterations (x-axis) of the turbo decoder will look like the left plot of Figure 5. Note that the exact numbers are unknown and depend on a lot of parameters.

Control

With the known cost function and the data collected through measurements of the quality, decision rules must make the trade-offs to minimize the costs within the given constraints. Due to limited space, not all the decisions rules are discussed here. The principle is that a prediction for the quality and cost increase or de-

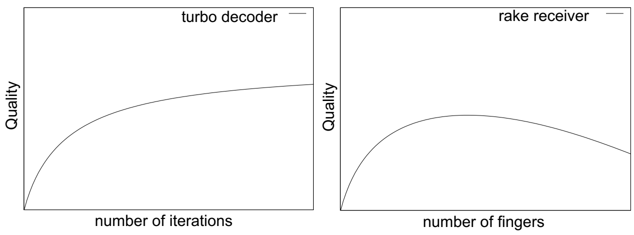

Figure 5. Quality Functions For The Turbo Decoder (left) And The Rake Receiver (right)

crease can be made using the first derivative. The first derivative of the cost function is always known. The first derivative of the quality function must be based on the information gathered from changes in the history. In case there is no history, small changes can be made to parameters to collect information to obtain the first derivative. These changes are not allowed to cause a violation of the quality constraint, so they should always (temporarily) improve the quality.

Due to the dynamic environment, the quality and the value of the first derivative could change every moment. The first derivative of the quality functions gives only a rough guess about what will happen when changing the parameter, not an exact estimate.

The same approach can be used with an extended set of parameters (not only the number of turbo decoding iterations and rake fingers). However, as mentioned, the set has to be limited and has to satisfy the rules that there is maximum one optimum.

4 Conclusion And Future Work

The quality of service of the current situation or a prediction for the future can not be determined analytically, due to unpredictable influences of the environment which have a considerable impact on the quality of service of the system. Our proposal is to measure the quality and to control the system dependent on these measurements using predefined decision rules. A next step in our research is to simulate the whole system to define the decision rules in detail and to evaluate this approach.

Acknowledgements

This research is conducted within the Chameleon project (TES.5004) supported by the PROGram for Research on Embedded Systems & Software (PROGRESS) of the Dutch organization for Scientific Research NWO, the Dutch Ministry of Economic Affairs and the technology foundation STW. We thank Philips for a very pleasant stay at the Natlab research laboratory in Eindhoven during the research period.

References

[1] F. Adachi. Effects of orthogonal spreading and rake combining on ds-cdma forward link in mobile radio. *IEICE Trans. Commun.*, E80-B(11):1703–1712, Nov. 1997.

[2] C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding: Turbo-codes. *IEEE Transactions on Communications*, 44(10):1261–1271, Oct. 1996.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: Turbo codes. In *Proc. International Conference on Communications (ICC)*, pages 1064–1070, May 1993.

[4] T. Ojanperä. *Wideband CDMA for Third Generation Mobile Communications*. The Artech House Universal Personal Communications Series. Artech House, 1998. ISBN: 0-89006-735-X.

[5] R. Price and P. Green. A communication technique for multipath channels. In *Proceedings of the IRE*, volume 46, pages 555–570, Mar. 1958.

[6] P. Robertson, E. Villebrun, and P. Hoeher. A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain. In *Proc. International Conference on Communications (ICC)*, pages 1009–1013, June 1995.

[7] R. Y. Shao, S. Lin, and M. P. Fossorier. Two simple stopping criteria for turbo decoding. *IEEE Transactions on Communications*, 47(8):1117–1120, Aug. 1999.

[8] G. J. Smit, P. J. Havinga, M. Bos, L. T. Smit, and P. M. Heysters. Reconfiguration in mobile multimedia systems. In *Proc. of PROGRESS workshop 2000*, pages 95–105, Oct. 2000.

[9] G. L. Turin. Introduction to spread-spectrum antimultipath techniques and their application to urban digital radio. *Proc. of the IEEE*, 68(3):328–353, Mar. 1980.

[10] M. C. Valenti. *Iterative Detection and Decoding for Wireless Communications*. PhD thesis, Virginia Polytechnic Institute and State University, July 1999.

[11] C. C. Wang. On the performance of turbo codes. In *Proc. IEEE MILCOM 1998*, pages 987–992, Oct. 1998.

[12] W. L. Winston. *Operation Research : Applications and algorithms*. Duxbury press, 3 edition, 1994.

[13] A. Worm, H. Michel, F. Gilbert, G. Kreiselmaier, M. Thul, and N. Wehn. Advanced implementation issues of turbo-decoders. In *Proc. 2nd International Symposium on Turbo-Codes and Related Topics*, Sept. 2000.