TSG-RAN Working Group1 meeting #2 **TSGR1#2(99)039**
Yokohama 22-25, February 1999

Agenda Item:

Source: Lucent Technologies

Title: Application of serially concatenated codes with interleavers to low data rate services

Document for: Discussion

Introduction

In this document, we propose an alternative to the channel code for low-quality services, i.e., those services that require a bit error probability of the order of 10^{-3}. So far, the code foreseen for these applications is a rate 1/3 (and also a rate ½) convolutional codes with 256 states decoded using the Viterbi algorithm. The data rate is 8 kbit/s, and frames of 10 and 20 msec are considered.

We propose, as alternatives to the rate 1/3 solution based on the convolutional code, to use a serially concatenated convolutional code (SCCC) [1] formed by two 4-state constituent codes with information block sizes of 80 and 160 bits.

The two solutions will be compared based on bit and frame error probabilities estimated by simulations over the additive Gaussian channel.

The present solution

So far, for low data rate services, rate ½ and 1/3 256-state convolutional codes with Viterbi decoding have been considered. We consider information frame size of 80 bits; the information block is encoded using the convolutional codes in a trellis-terminated fashion.

The generating matrices of the codes are

$$G(Z) = \left(1 + Z^2 + Z^3 + Z^5 + Z^6 + Z^7 + Z^8, 1 + Z + Z^3 + Z^4 + Z^7 + Z^8, 1 + Z + Z^2 + Z^5 + Z^8\right) \quad \text{Rate } 1/3$$
$$G(Z) = \left(1 + Z^2 + Z^2 + Z^4 + Z^8, 1 + Z + Z^2 + Z^3 + Z^5 + Z^7 + Z^8\right) \quad \text{Rate } 1/2$$

In the following, we will describe two alternatives to the rate 1/3 code. Similar proposal can be made for the rate ½ case.

The proposed alternative

The second alternative is based on a rate 1/3, serially concatenated convolutional code (SCCC) based on the concatenation of a 4-state, punctured rate 2/3 outer code, and a 4-state, rate ½ inner code, obtained by the same rate ½ recursive convolutional encoder with generating matrix. The interleaver joining the two codes acts on the outer encoded bits, and has a size of 123 (80x3/2 +3 bits for outer trellis termination), and 243

(160x3/2 + 3 bits for outer trellis termination).

A block diagram of the encoder is shown in Figure 1.

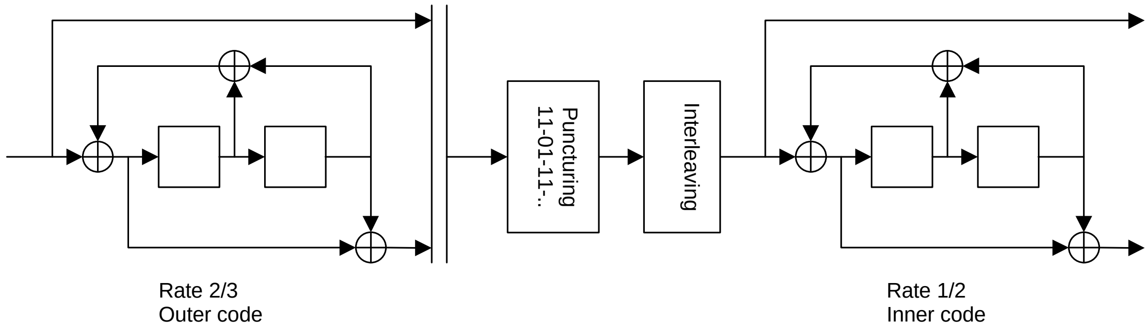

Figure 1: Structure of the rate 1/3 SCCC encoder.

The decoder is an iterative decoder based on two SISO modules, one for the outer and the second for the inner codes, and is shown in Figure 2

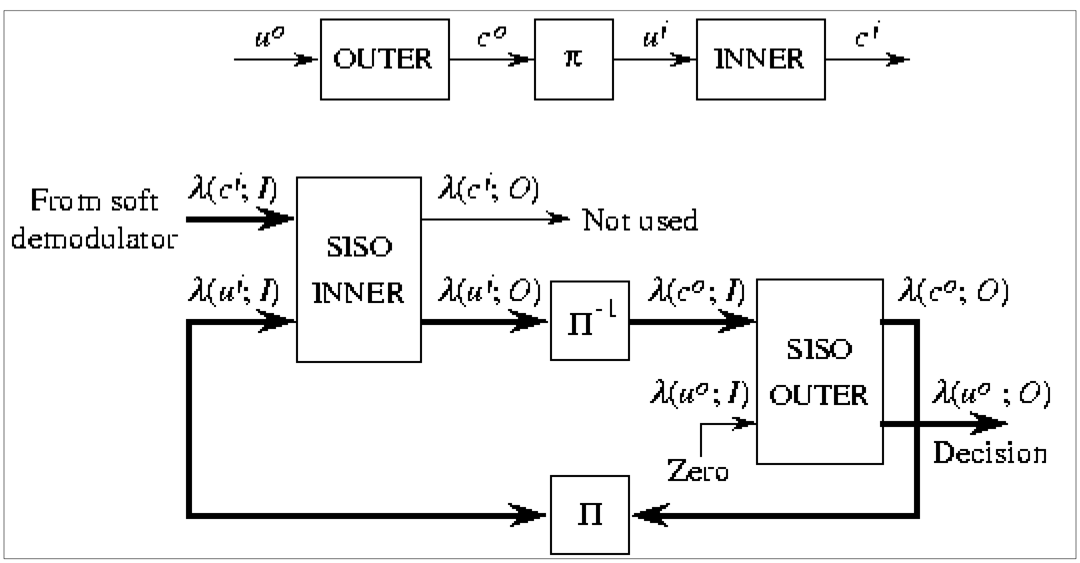

Figure 2: Decoder structure for SCCC

The choice of this code is based on the fact that we have proposed the same solution for medium-high quality, medium-high bit rate data services (see the companion document [2]), and offers then the advantage of homogeneity.

Performance comparison

In Figure 3, we present the frame error probability performance obtained by simulating the two aforedescribed solutions on the AWGN channel. The SCCC scheme uses optimised Nortel interleavers. The same performance hierarchy can be expected on Phase 2 channel.

To be fair, also the convolutional code has been decoded using a MAP decoding algorithm, instead of the Viterbi algorithm. Its performance, however, are almost the same, being the MAP decoder slightly better in terms of bit error probability. The SCCC decoder use 10 and 100 iterations.

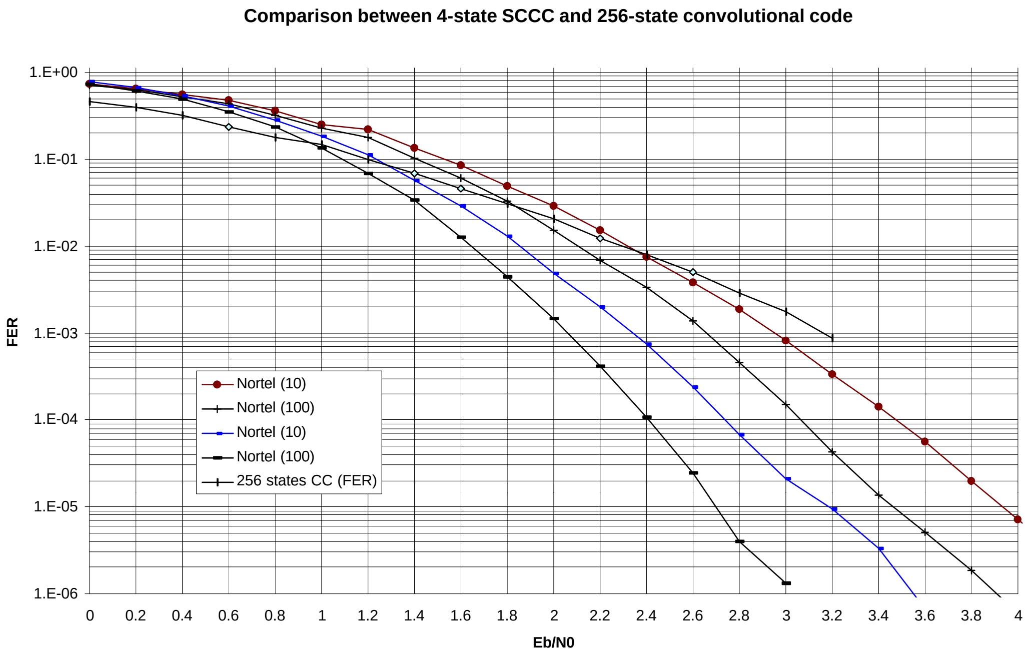

Figure 3: Comparison among the two codes proposed for the low data rate services on the AWGN channel in terms of frame error probability. Information block sizes of 80 and 160 are considered. 10 and 100 iterations.

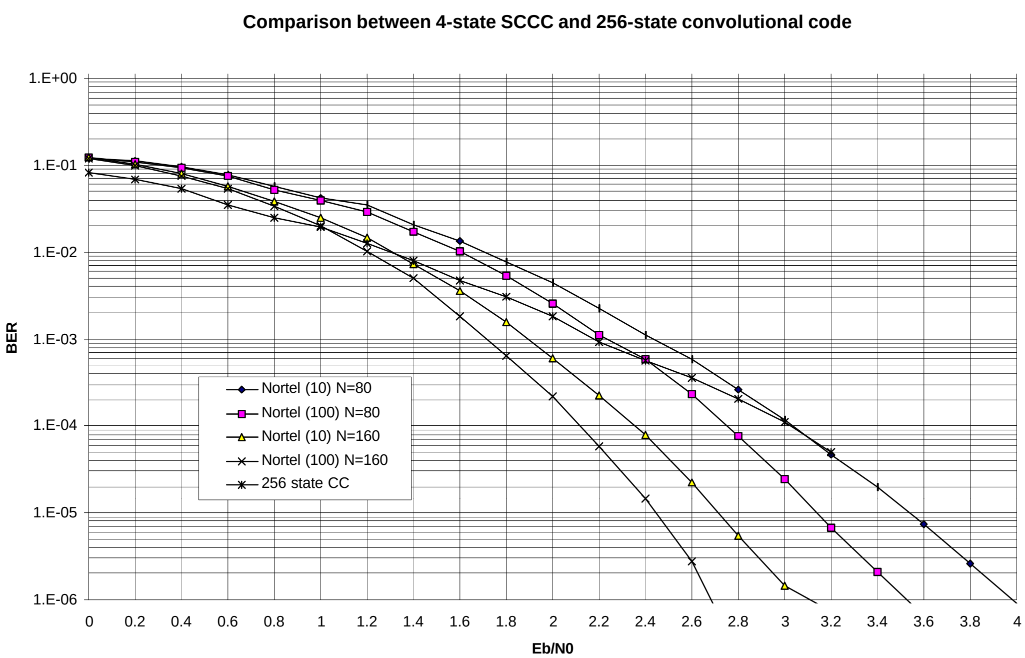

Figure 4: Comparison among the two codes proposed for the low data rate services on the AWGN channel in terms of bit error probability. Information block sizes of 80 and 160 are considered. 10 and 100 iterations.

The analogous results in terms of bit error probability are reported in Figure 4. The simulation results suggest the following considerations:

- In terms of frame error probability the SCCC outperforms the convolutional code below 10^{-2} owing to its larger free distance (22 for N=80 and 24 for N=160 for SCCC, with respect to 18 for the 256 state convolutional code). At FER=10^{-3} the gain of SCCC is 0.3 dB (10 iterations N=80), 0.6 dB (100 iterations N=80), 0.9 dB (10 iterations N=160), 1.2 dB (100 iterations N=160).
- In terms of bit error probability, the SCCC outperforms the convolutional code below 10^{-4} for N=80, and below 10^{-2} for N=160. In this last case, at $\mathbf{10^{-3}}$ the convolutional code looses 0.3 dB for 10 iterations and 0.5 for 100 iterations, and at $\mathbf{10^{-6}}$ it looses 1 dB at 10 iterations and 1.4 at 100 iterations.

Complexity considerations

We have not yet performed a detailed comparative study on the implementation complexity of the two schemes, but we believe that the 4-state SCCC decoder is significantly less complex than the 256-state convolutional Viterbi decoder.

Conclusions

As a conclusion, we strongly suggest to reconsider the choice of the 256-state convolutional code as the channel code for low-quality, low data rate applications. The alternative proposed here, which is based on the use of the same code proposed for the high-quality services, offers significant gains over the convolutional code for both bit and frame error probabilities, and we believe it requires a lower implementation complexity.

References

[1] S. Benedetto, D. Divsalar, G. Montorsi, F. Pollara, "Soft-input soft-output modules for the construction and distributed iterative decoding of code networks", *European Transactions on Telecommunications*, vol. 9, n. 2, March-April 1998.

[2] Lucent TSGR1#2(99)036