S-72.630 Capacity enhancement methods for radio interface
ARQ Schemes

Markku Liinaharja

March 25, 2004

Outline

- Classification of error control strategies
- Basic ARQ schemes and their performance
- Improving ARQ performance
- Hybrid ARQ schemes (types I and II)
- An ARQ scheme with diversity combining
- An adaptive ARQ algorithm
- Some adaptive ARQ schemes
- Concluding remarks

Classification of error control strategies

- Two basic categories of error control strategies for data communications
- Forward error correction (FEC) schemes: one-way communication without a feedback channel. The decoder at the receiver attempts to retrieve the original information block using an error-correcting code.
- Automatic repeat request (ARQ) schemes: two-way communication where the receiver checks the correctness of each received block by using a high-rate error-detecting code and sends a positive (ACK) or negative (NAK) acknowledgement to the transmitter accordingly. All incorrectly received packets are retransmitted. Cyclic redundancy check (CRC) codes are typically used for error detection.
- Combinations of ARQ and FEC are known as hybrid ARQ (HARQ) schemes. These are further divided into type-I (HARQ-I) and type-II (HARQ-II) HARQ schemes.

Basic ARQ schemes

- Based on retransmission strategies, there are three basic types of ARQ schemes: stop-and-wait (SW) ARQ, go-back-N (GBN) ARQ and selective-repeat (SR) ARQ.
- The SW ARQ scheme is simple, but extremely inefficient because after sending a codeword, the transmitter waits idly for an acknowledgement before sending the next one.
- In the GBN ARQ scheme, after receiving a NAK for a packet, the transmitter resends the packet in question and the N-1 packets that follow (a sliding window of length N is used). Unlike SW ARQ, GBN ARQ is a continuous scheme, but still inefficient, especially if the feedback delay is long.
- The SR ARQ scheme is the most efficient basic ARQ scheme because only the packets with detected errors are retransmitted.

Measuring the performance of ARQ schemes

- *Throughput efficiency* η is defined as the average number of information bits successfully accepted by the receiver per one transmitted bit.
- A variation of η is *packet throughput* T, which is the average number of packets successfully accepted by the receiver per one transmitted packet.
- By definition, η and T are both numbers between 0 and 1, and they describe the efficiency of transferring data.
- For continuous schemes, where the transmitter is never idle, $T = 1/\mathrm{E}(R)$, and $\eta = (k/n) \cdot T$, where n is the packet size in bits, k is the number of information bits per packet, and R is the (random) number of transmission attempts needed until a packet is received successfully.
- Other performance measures include delay metrics, for example the average delay of a message consisting of a random number of packets.

Modelling the channel environment

- In the simplest case, we have a memoryless binary symmetric channel (BSC) with bit error rate (BER) equal to ϵ. In this channel, the packet error rate (PER),that is, the probability P_e that there is at least one bit error in a received n-bit packet, is simply $1 - (1 - \epsilon)^n$. A memoryless BSC is equivalent to an AWGN channel.
- Time-varying channel environments are often modelled by finite-state Markov chains, where the states are characterized by bit or packet error probabilities. The simplest such model is the Gilbert-Elliott model, which has two states as shown in Figure 1.

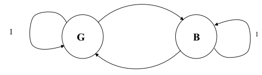

Figure 1: The Gilbert-Elliott channel model.

Performance comparison of basic ARQ schemes

- The throughput of the SR ARQ scheme in a BSC with BER ϵ can be shown to be

$$\eta_{\mathrm{SR}} = \frac{k}{n}(1 - P_e) = \frac{k}{n}(1 - \epsilon)^n,$$

  where again n is the packet size and k is the number of information bits per packet.
- Figure 2 shows the throughputs of SW, GBN and SR ARQ schemes when $n = 200$, $k = 184$, and the round-trip delay (feedback delay) is equal to the transmission time of 9 packets.

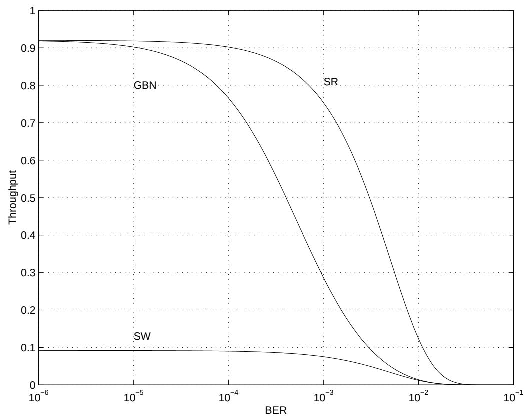

Figure 2: Comparison of the three basic ARQ schemes.

Improving ARQ performance

- The throughput of basic ARQ schemes falls down rapidly as the channel PER increases.
- One way to address this problem is to add error correction capability to the scheme (hybrid ARQ schemes).
- Another possibility is to use adaptive ARQ schemes, where the transmission mode is changed according to the estimated channel conditions. Channel state estimation is done by observing the acknowledgements received by the transmitter.
- The adaptivity can be based, for example, on varying the packet length or the transmitted number of copies of a packet.

HARQ-I schemes

- The first transmission and all retransmissions of a packet are identical codewords containing redundant bits for both detecting and correcting errors.
- It is possible to use just one block code for simultaneous error detection and error correction. Another way is to use two separate codes: an inner code (such as CRC) for error detection and an outer code for error correction.
- If an uncorrectable error pattern is detected in the received packet, the receiver asks for retransmission.
- These schemes are best suited for channels where the level of noise and interference is fairly constant.
- Adaptive HARQ-I schemes with adjustable code rates have been suggested for time-varying channel environments. For example, a family of BCH codes with constant length could be used.

HARQ-II schemes

- ‘ARQ with memory:’ an incorrectly received packet is stored by the receiver to be used for decoding together with the retransmission block of parity bits.
- The method of ‘incremental redundancy:’ the parity bits for error correction are sent only when they are needed.
- Two codes are used: a high-rate block code (like CRC) for error detection, and a 1/2-rate code for error correction.
- Unlike HARQ-I schemes, HARQ-II schemes have ‘built-in adaptivity:’ in good channel conditions, they operate like basic ARQ schemes with redundancy just for error detection; in poor channel conditions, the error correcting part becomes more dominant.

An ARQ scheme with diversity combining

- The HARQ-II schemes are a special case of *code combining* methods where ‘packets are concatenated to form noise-corrupted code words from increasingly longer and lower-rate codes.’ (Wicker)
- Another form of packet combining is *diversity combining*, where ‘the individual symbols from multiple, identical copies of a packet are combined to create a single packet with more reliable constituent symbols.’ (Wicker)
- A method proposed by Sindhu (*IEEE Trans. Commun.*, May 1977): calculate the bitwise modulo-2 sum (or logical XOR) of two erroneous copies of a packet to locate the errors. The error correction fails only when there are overlapping bit errors in the two copies of the packet.
- The performance of an ARQ scheme (called EARQ) based on this idea was studied by Chakraborty et al. (*IEEE Commun. Lett.*, July 1998), see Figure 3.

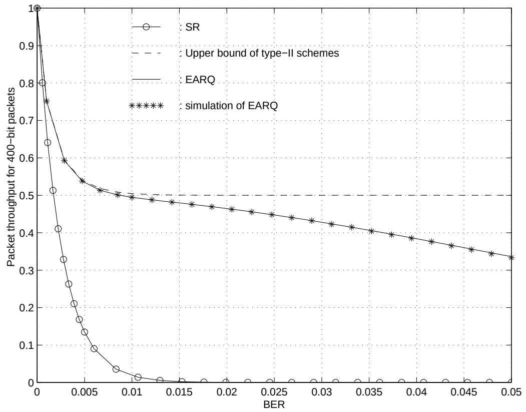

Figure 3: Comparison of SR ARQ, ideal HARQ-II and EARQ schemes.

An adaptive ARQ algorithm

- Proposed by Yao (*IEEE Trans. Commun.*, Jan. 1995)
- An adaptive ARQ scheme with two transmission modes: mode L ('low error') meant for 'good' channel conditions, and mode H ('high error') meant for 'bad' channel conditions.
- The algorithm for detecting changes in channel state and switching between transmission modes: in mode L, if the transmitter receives α contiguous NAKs, switch to mode H; in mode H, if β contiguous ACKs are received, switch to mode L.
- The transmission modes L and H can be characterized, for example, by packet size or the number of copies of the packet that are transmitted at once.

Adaptive SR schemes with variable packet size

- It makes a big difference, what packet size the SR ARQ scheme uses. If there are two transmission modes using packet sizes n_1 and n_2, one interesting idea is to set $n_1 = 2n_2$.

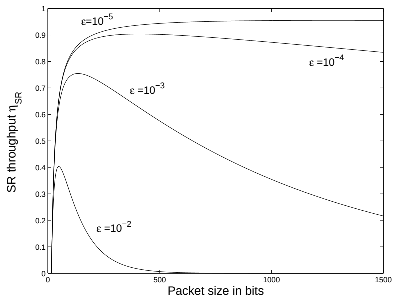

Figure 4: The SR throughput as a function of packet size with four different BER values when $n - k = 16$ (16-bit CRC is used).

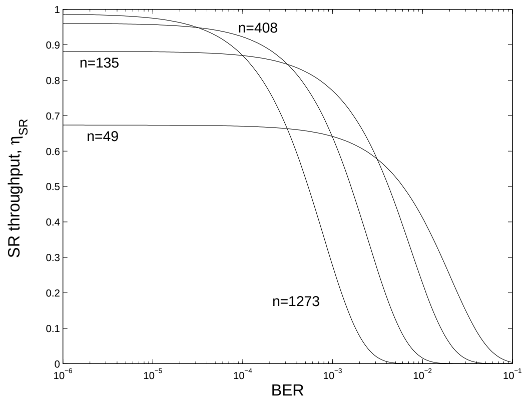

Figure 5: The SR throughput as a function of the BER with four different packet sizes when $n - k = 16$ (16-bit CRC is used). The four packet sizes are chosen to be the optimal ones corresponding to BER values 10^{-5}, 10^{-4}, 10^{-3} and 10^{-2}.

Adaptive GBN schemes

- At high PER values, the performance of the GBN ARQ scheme can be improved by sending multiple copies of the packet at once, as Figure 6 demonstrates.

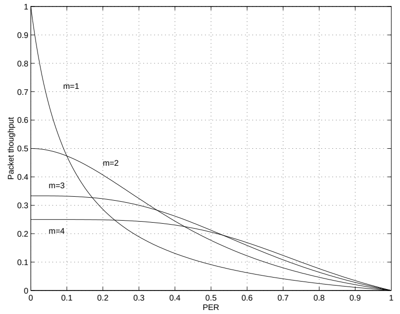

Figure 6: Comparison of m-copy GBN schemes with different values of m when $N = 10$.

Performance analysis of adaptive ARQ schemes

- Time-varying channel environments are represented by hidden Markov models (HMMs), where states are characterized by bit or packet error probabilities.
- Based on this assumption, the entire system, consisting of the channel environment and the adaptive scheme, is modelled by a Markov chain.
- The performance of an adaptive scheme is measured by its average throughput efficiency over the states of the system, which is a function of the design parameters of the adaptive scheme.
- Optimization of the design parameters, which define the channel sensing algorithm of the scheme, has also been considered.

Concluding remarks

- ARQ schemes are not suitable for delay-sensitive applications where there is no time for retransmissions.
- ARQ is needed in applications where it is necessary to receive packets without errors, for example file transfer. With relatively few redundant bits for error detection, the probability of undetected error can be made extremely small.
- It is possible to improve the performance of the basic ARQ schemes significantly without using error correction. Adaptive SR schemes with variable packet size are especially interesting.
- However, if the channel PER stays high for long periods of time, use of FEC becomes necessary. HARQ-II schemes using advanced coding methods, such as turbo codes or zigzag codes for error correction, are an appealing choice.
- A lot of research remains to be done on ARQ schemes in multiantenna systems. Both code combining and diversity combining need to be considered.