## Error Control Systems with Feedback

So far, only *forward error correction* (FEC) has been considered. With information flow in both directions of a channel, other options become available. The most important error control methods for channels with feedback are as follows:

**1.** Automatic-repeat-request (ARQ) protocols.
**2.** Type-I hybrid-ARQ protocols.
**3.** Type-II hybrid-ARQ protocols.





## Pure ARQ Protocols

The objective of a pure ARQ protocol is a system that will detect an error burst, discard the affected packet (the message is broken up into *packets* of length $k$), and request a retransmission.

The most frequently used error-detecting codes in ARQ protocols are the CRC codes.





## Performance Measures for ARQ Protocols

The following two measures are often used to evaluate the performance of an ARQ protocol.

**accepted packet error rate** The percentage of packets accepted by the receiver that contain one or more bit/symbol errors.

**throughput** The average number of encoded data packets accepted by the receiver in the time it takes the transmitter to send a single $k$-bit data packet.





## Average number of transmissions

Let $P_r$ be the probability that a retransmission request is generated for a received packet. If the random variable $T$ is defined as the number of times a packet must be transmitted before it is accepted, then according to Eq. 15-3 in [Wic] its expectation $T_r$ is given by

$$T_r = \mathrm{E}[T] = \frac{1}{1 - P_r}.$$

This expression is useful in calculating the throughputs of the basic ARQ protocols.

## Retransmission Protocols

In selecting a retransmission protocol, the designer must strike a balance between the complexity of the design and the throughput of the resulting system. The three basic retransmission protocols are the following:

1. Stop-and-wait (SW-ARQ).
2. Go-back-$N$ (GBN-ARQ).
3. Selective repeat (SR-ARQ).





## Stop-and-Wait (1)

In the stop-and-wait (SW-ARQ) protocol, the transmitter sends out a packet and waits for an acknowledgment. The receiver responds by sending an acknowledgment (ACK) if the packet was deemed error-free, or it sends a retransmission request (RQ) if the packet contained a detectable error pattern; see [Wic, Fig. 15-1]. The transmitter is idle while waiting for the acknowledgment.

The primary benefit of the SW-ARQ protocol is that there is no need for packet buffering at the transmitter or receiver. The primary disadvantage of the protocol is that it provides poor throughput, in particular when the propagation delays are long (satellite communications, etc.).





## Stop-and-Wait (2)

If $\Gamma$ denotes the number of bits that could be transmitted during the idle time of the transmitter, then the throughput of the SW-ARQ protocol is

$$\eta_{SW} = \frac{k}{T_r(n+\Gamma)} = R\left(\frac{1-P_r}{1+\Gamma/n}\right),$$

where $R$ is the rate of the error-detecting code used in the protocol.





## Go-Back-$N$ (1)

If we are willing to allow for some buffering in the transmitter, the go-back-$N$ (GBN-ARQ) protocol can be implemented; see [Wic, Fig. 15-2]. In this protocol the transmitter sends packets in a continuous stream.

When the receiver detects an error, it sends an RQ for that packet and waits for its second copy (and ignores all subsequent packets until the second packet is received, so receiver buffering is avoided). The transmitter responds by resending the requested packet and *all subsequent packets* (so buffering is necessary in the sender).

## Go-Back-$N$ (2)

If $\Gamma$ again denotes the number of bits that can be transmitted during the "round-trip delay", then the number of packets that can be transmitted (at least partially) during this time is (note that we disagree with Eq. 15-6 in [Wic])

$$N - 1 = \left\lceil \frac{\Gamma}{n} \right\rceil,$$

and the throughput of the GBN-ARQ protocol is

$$\eta_{GBN} = \left(\frac{k}{n}\right)\left(\frac{1}{1+(T_r-1)N}\right) = R\left(\frac{1-P_r}{1+P_r(N-1)}\right).$$





## Selective Repeat

If we allow for buffering in both the transmitter and the receiver, we can implement a selective-repeat (SR-ARQ) protocol; see [Wic, Fig. 15-3]. The transmitter sends a continuous stream of packets and responds to retransmission requests by sending the requested packet and thereafter resuming the transmission where it was stopped.

The throughput of the SR-ARQ protocol is simply

$$\eta_{SR} = \left(\frac{k}{n}\right)\left(\frac{1}{T_r}\right) = R(1-P_r).$$





## Noisy Feedback Channels

The following additional protocols can be implemented for a noisy feedback channel.

- ▷ Each time the transmitter sends out a packet, a timer for that packet is started. If a response is not obtained for that packet after a given period of time, it is assumed that the response is an RQ.
- ▷ When the receiver sends an RQ, the receiver starts a timer for that RQ. If a new copy of the packet is not received after a given period of time, the RQ is sent again.
- ▷ If the receiver receives a packet that has already been accepted, an ACK is sent and the packet is discarded.





## Hybrid-ARQ Protocols

In hybrid-ARQ protocols, each packet is encoded for both error detection and error correction. The FEC portion corrects the most common error patterns. Hybrid protocols provide throughput similar to that of FEC systems, while offering reliability performance typical of ARQ protocols.

Hybrid-ARQ protocols are divided into the simpler type-I protocols and the more advanced type-II protocols.

## Type-I Hybrid-ARQ Protocols (1)

Type-I hybrid-ARQ protocols can be implemented using either one-code or two-code systems. In a two-code system, the data is first encoded using a high-rate error-detecting code (CRC codes are frequently used). The encoded data is then encoded once again using a FEC code.

An example of a single-code type-I hybrid-ARQ protocol is given in [Wic, Example 15-4]. In [Wic, Fig. 15-11] and [Wic, Fig. 15-12] the performance of a pure ARQ protocol and a type-I hybrid-ARQ protocol are compared with respect to reliability and throughput.





## Type-I Hybrid-ARQ Protocols (2)

The single-code type-I hybrid-ARQ protocols are based on the following idea.

If the minimum distance of a linear block code satisfies the condition

$$d_{\min} \geq \lambda + l + 1,$$

then the code is capable of correcting all error patterns with $\lambda$ or fewer errors and simultaneously detecting all error patterns with $l$ ($l > \lambda$) or fewer errors.





## Packet Combining Systems

These systems offer adaptive code rates for different channel conditions. Two distinct types:

**code combining** The individual transmissions (of one data block) are encoded at some rate $R$. If the receiver has $N$ packets that have caused retransmission requests, these packets are concatenated to form a single packet encoded at rate $R/N$.

**diversity combining** Multiple identical copies of a packet are used to create a single packet whose symbols are more reliable than those of any of the individual copies. Symbol voting is used in hard-decision systems and symbol averaging in soft-decision systems.





## Type-II Hybrid-ARQ Protocols (1)

Type-II hybrid-ARQ protocols are code combining systems where $N$ (the number of combined packets) is limited to 2. They adapt to changing channel conditions through the use of *incremental redundancy*. The transmitter in these systems responds to retransmission requests by sending additional parity bits to the receiver. These additional bits allow for increased error-correction capability.

## Type-II Hybrid-ARQ Protocols (2)

The “original” type-II hybrid-ARQ protocol proposed by Wang and Lin:

1. A $k$-bit message is first encoded using a high-rate $(n, k)$ error detecting code $\mathbf{C}_1$ to form an $n$-bit packet $P_1$.

2. $P_1$ is encoded using a $(2n, n)$ systematic *invertible* code $\mathbf{C}_2$.

3. The $n$ parity bits (denoted by $P_2$) from the $\mathbf{C}_2$ codeword are saved in a buffer, while the $\mathbf{C}_1$ codeword $P_1$ is transmitted.

4. If the initial transmission of $P_1$ is found to contain errors, it is stored in a buffer, a retransmission request is sent, and the transmitter responds by sending $P_2$.





## Type-II Hybrid-ARQ Protocols (3)

5. The received version of $P_2$ is first inverted and checked for errors. If there are detected errors, the received noisy version of $P_2$ is appended to the earlier received (also noisy) version of $P_1$ to form a noise-corrupted $\mathbf{C}_2$ codeword. After decoding, the resulting $n$-bit word is again checked for errors using $\mathbf{C}_1$. If errors are detected, the process continues, with the transmitter alternating transmission of $P_1$ and $P_2$ until the decoding/ error detection process is successful.

For example, shortened half-rate cyclic codes have been used as $\mathbf{C}_2$-codes.