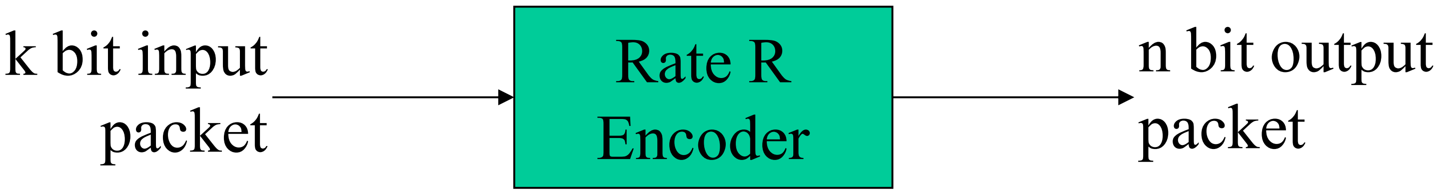
EEENGM4221:
Broadband Wireless Communications

Lecture 8: ‘Simple’ HARQ

Dr Simon Armour

Error Control

- Fundamentally, Error Control Strategies are based on the introduction of controlled redundancy into the data.
- The code rate, R, defines the ratio of input bits, k, to output bits, n, for a given encoder, hence R=k/n
  - Since redundancy is added R<1
  - Output data rate < Input data rate
- Thus, throughput is sacrificed in return for the ability to detect or correct errors



Ref: Lin & Costello

ARQ (1)

- Automatic Repeat reQuest (ARQ) exploits the redundancy to detect errors
  - Commonly, a Cyclic Redundancy Check Code is used
    - A CRC-N adds N bits to the data; R=k/(k+N)
- A protocol is employed which enables the destination node to inform the source node of the result of the error check either by:
  - Sending an acknowledgement (ACK)
  - Sending a negative acknowledgement (NACK)
  - The NACK may be implicit, i.e. if no ACK received, assume NACK
  - Given the relatively high likelyhood of errors in wireless communications, implicit ACKs are not a wise option!
- The source will then retransmit any packets believed to be in error

Ref: Lin & Costello

ARQ (2)

- Throughput is reduced:

$$f_{d,throughput} = f_{d,\text{nominal}}\left(1-\sum_{r=1}^{\infty}\left(P_p\right)^r\right)$$

- However, if P_p is small, the first few terms of the summation dominate and:

$$f_{d,throughput} \rightarrow f_{d,\text{nominal}}\left(1-P_p\right) \qquad \text{as} \qquad P_p \rightarrow 0$$

ARQ (3)

- In principle, ARQ guarantees near zero packet errors
  - If at first you don't succeed, try,try,try…try again!
  - However, it is possible that errors go undetected
    - This becomes less likely as the code rate reduces
- The reduction in packet errors is achieved at the expense of increased packet delay
- The 'round trip delay' of successful packet transmission increases with the number of repeat transmissions required
- Ultimately, it may prove better to abandon a packet after a certain number of retries if it is delayed too long to be of use
  - This is again a function of the QoS Requirements of the application

Ref: Lin & Costello

ARQ (4)

- The mean total round trip delay of a packet is related to the mean round trip delay of a single transmission attempt and the number of retransmissions required:

$$D_{total,mean} \geq (r+1)D_{1,mean}$$

- If the round trip delay of a single packet transmission is constant, the statistics of the likely total round trip delay are trivial

Ref: Lin & Costello

ARQ (5)

- However, if the round trip delay per transmission attempt is not constant, the statistics of the total delay are non-trivial
- Variable round trip times combine with ARQ error control to cause delay jitter
- So ARQ degrades throughput and delay in order to improve error rates
- Given the harsh nature of wireless channels, ARQ is essential for all but the most error tolerant of applications

Ref: Lin & Costello

FEC (1)

- Forward Error Correction Coding (FEC) exploits redundancy to correct errors
- Given the received packet (which may contain errors) a FEC decoder has the task of choosing the most likely packet to have been input to the encoder
- Typically, lower rate codes can correct more errors
  - They can be compared by 'coding gain'
    - How many fewer dBs of SNR required to achieve the same BER or PER with the code than without
  - Some codes are better than others and achieve more coding gain at a given rate
    - Often, this is at the expense of increased complexity

FEC (2)

- Thus a FEC code reduces the PER at the expense of also reducing throughput
- The effects on delay are minimal
- FEC cannot guarantee that the decoded packet is correct!
- There are lots of different FEC codes in common use: Hamming, Reed-Solomon, BCH, convolutional, Turbo, LDPC, etc

Ref: Lin & Costello

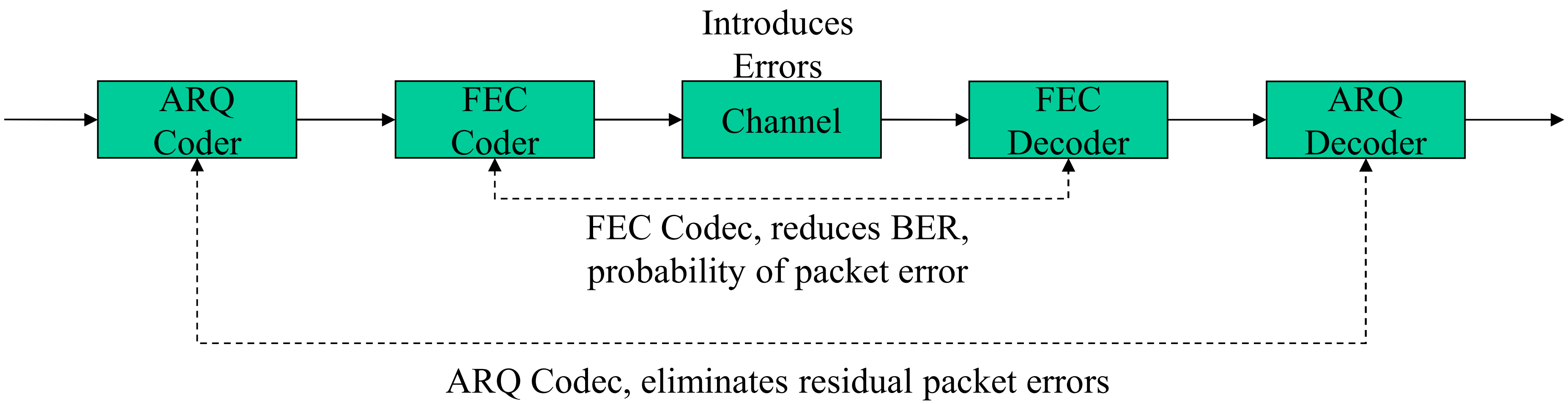
HARQ (1)

- ARQ and FEC both control errors
  - ARQ can guarantee near zero error at the expense of throughput and delay
  - FEC can reduce errors at the expense of throughput but with minimal implications for delay
- Often, the best result is achieved by combining an ARQ code with a FEC code
- This is sometimes referred to as Hybrid ARQ (HARQ)
- There are various clever forms of HARQ but the simplest is just a concatenation of two codecs

HARQ (2)

- Concatenation of codes results in net rate:

$$R_{net} = R_{ARQ} R_{FEC} = \frac{k_{ARQ} k_{FEC}}{n_{ARQ} n_{FEC}}$$



Ref: Lin & Costello

Review of Lecture 8

- We have reviewed how ARQ and FEC can be combined for ‘simple’ HARQ
  - ARQ codec ensures near zero errors
  - FEC codec reduces the number of retransmissions required
- This has implications for QoS
- This is the simplest form of HARQ. More advanced strategies covered in the next lecture will be:
  - Chase Combining
  - Incremental Redundancy

Ref: