



**Institute for the Wireless  
Internet of Things**

at Northeastern University

# EECE 5155

## Wireless Sensor Networks (and The Internet of Things)

**Prof. Francesco Restuccia**

Email: [f.restuccia@northeastern.edu](mailto:f.restuccia@northeastern.edu)



# Link Layer Objectives

- **Framing**
  - Group bit sequence into packets/frames
  - Format and size are important
- **Error control**
  - Forward and backward error control
- **Flow control**
  - Ensure that a fast sender does not overrun a slow(er) receiver
- **Link management**
  - Discovery and manage links to neighbors
  - Do not use a neighbor at any cost, only if link is good enough  
**(reliable link)**



# Error Control

## ➤ Objectives of error control

- Error-free
- In-sequence
- Duplicate-free
- Loss-free

## ➤ Causes of packet loss

- Fading, Interference, loss of bit synchronization
- Results in bit errors, often bursty
- Bit error rates can be high ( $10^{-2}$ - $10^{-4}$  possible)



# Error Control

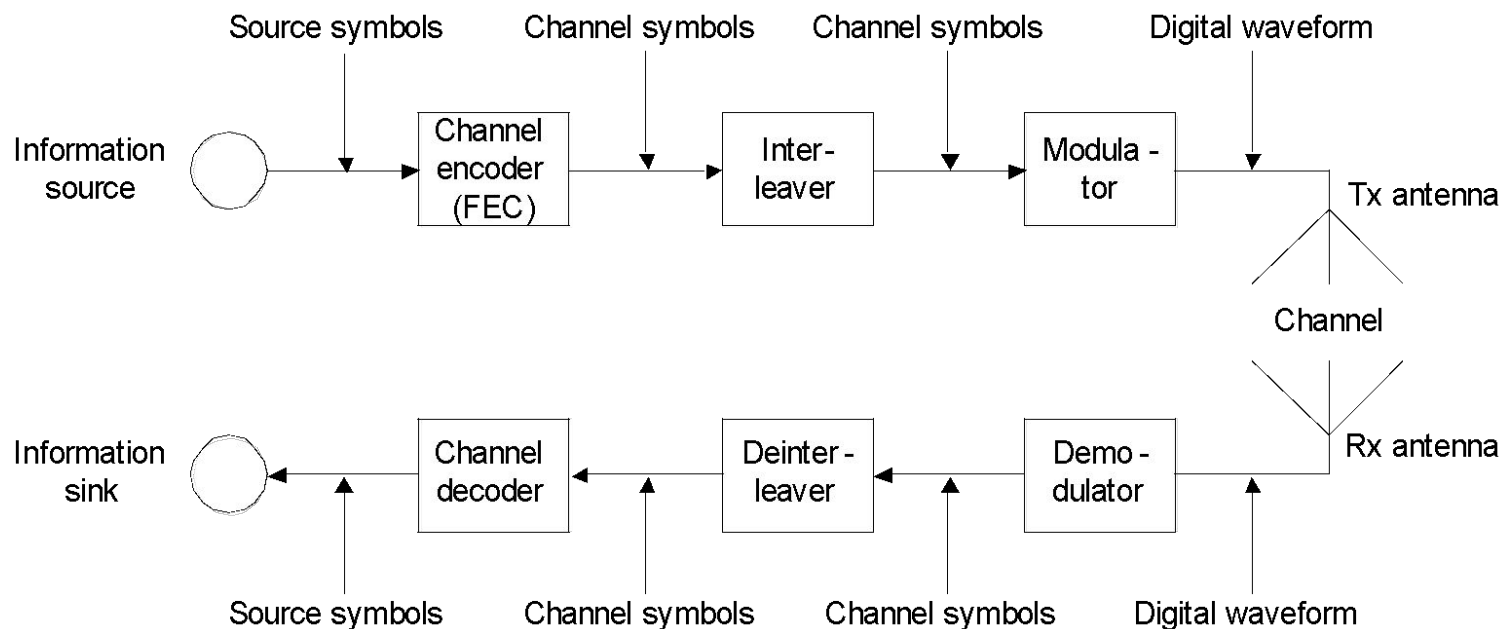
## Possible Approaches

1. Forward Error Control (FEC)
2. Automatic Repeat Request (ARQ)
  - Backward error control
3. Hybrid Schemes



# Forward Error Control (FEC)

- Endow symbols in a packet with **additional redundancy** to withstand a limited amount of random permutations
  - Additionally: interleaving – change order of symbols to withstand burst errors



# Backward Error Control – ARQ

- **Basic procedure** (recap)
  - Put header information around the payload
  - Compute a checksum and add it to the packet
    - Typically: **Cyclic redundancy check (CRC)**
      - Quick, low overhead, low residual error rate
  - Provide feedback from receiver to sender
    - Send **positive** or **negative acknowledgement**
  - Sender uses timer to detect that acknowledgements have not arrived
    - Assumes packet has not arrived
  - If sender infers that a packet has not been received correctly, sender can retransmit it
    - What is maximum number of retransmission attempts? If bounded, at best a semi-reliable protocols results



# Standard ARQ Protocols

## ➤ Stop-and-Wait

- Buffer only one packet, retransmit if no ACK before timer expires

## ➤ Go-back-N

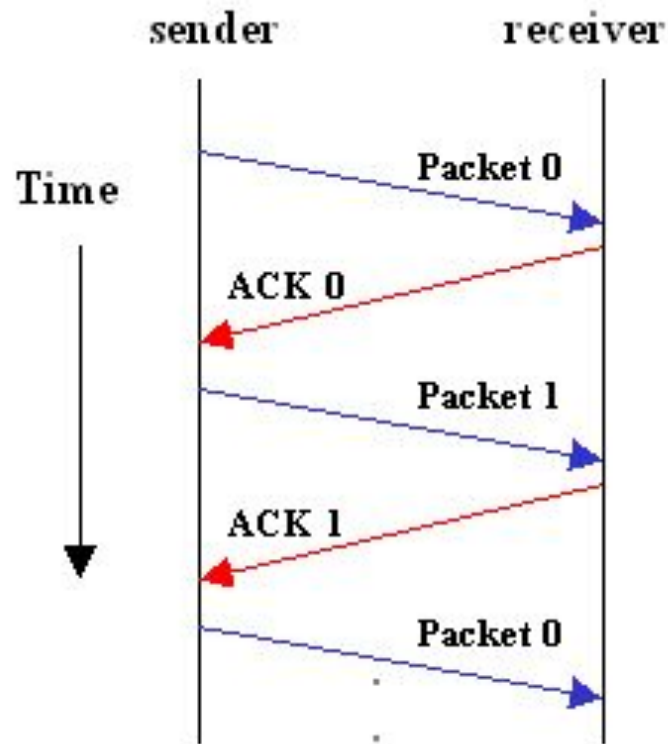
- Buffer N packets, send up to N packets, if a packet has not been acknowledged when timer goes off, retransmit all unacknowledged packets

## ➤ Selective Repeat

- Buffer N packets, when timer goes off, only send unacknowledged packets

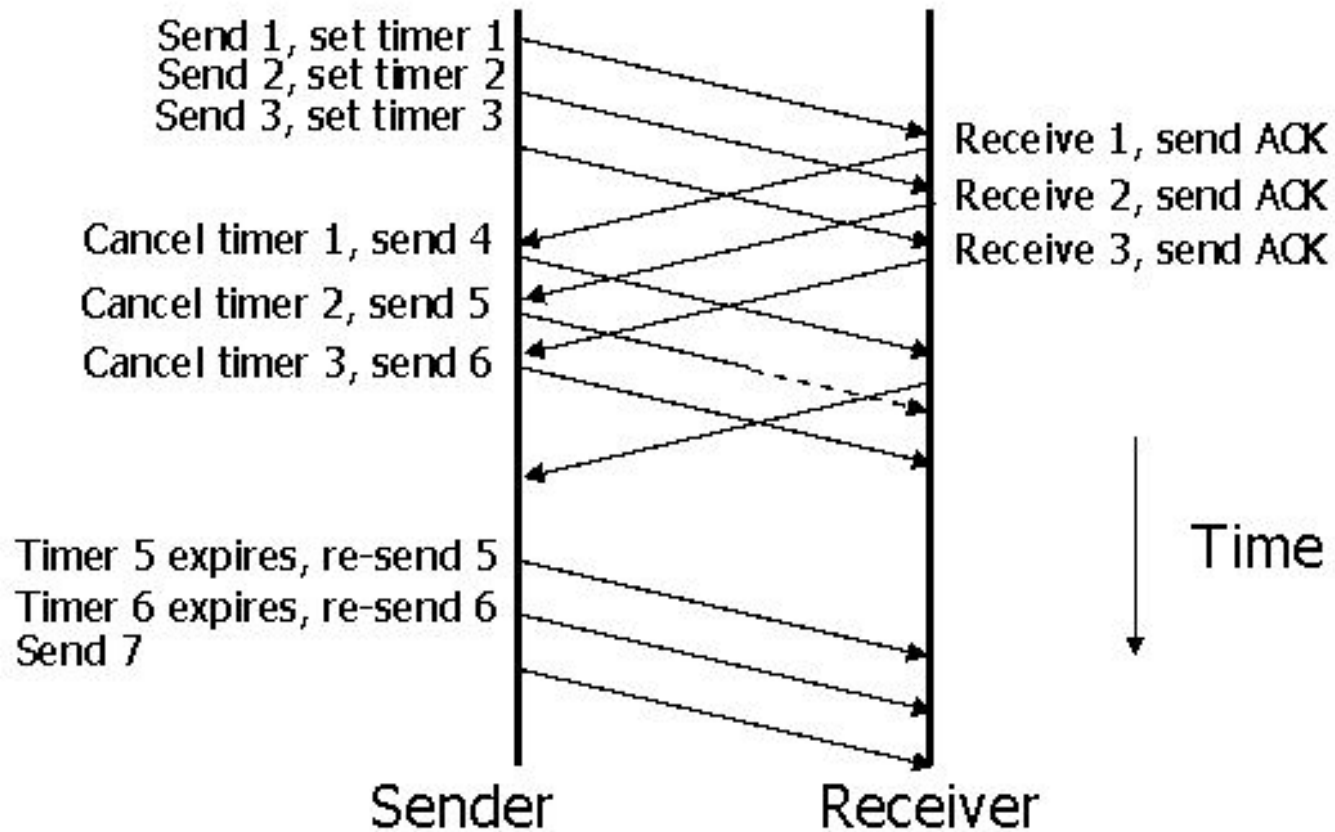


# Stop-and-Wait

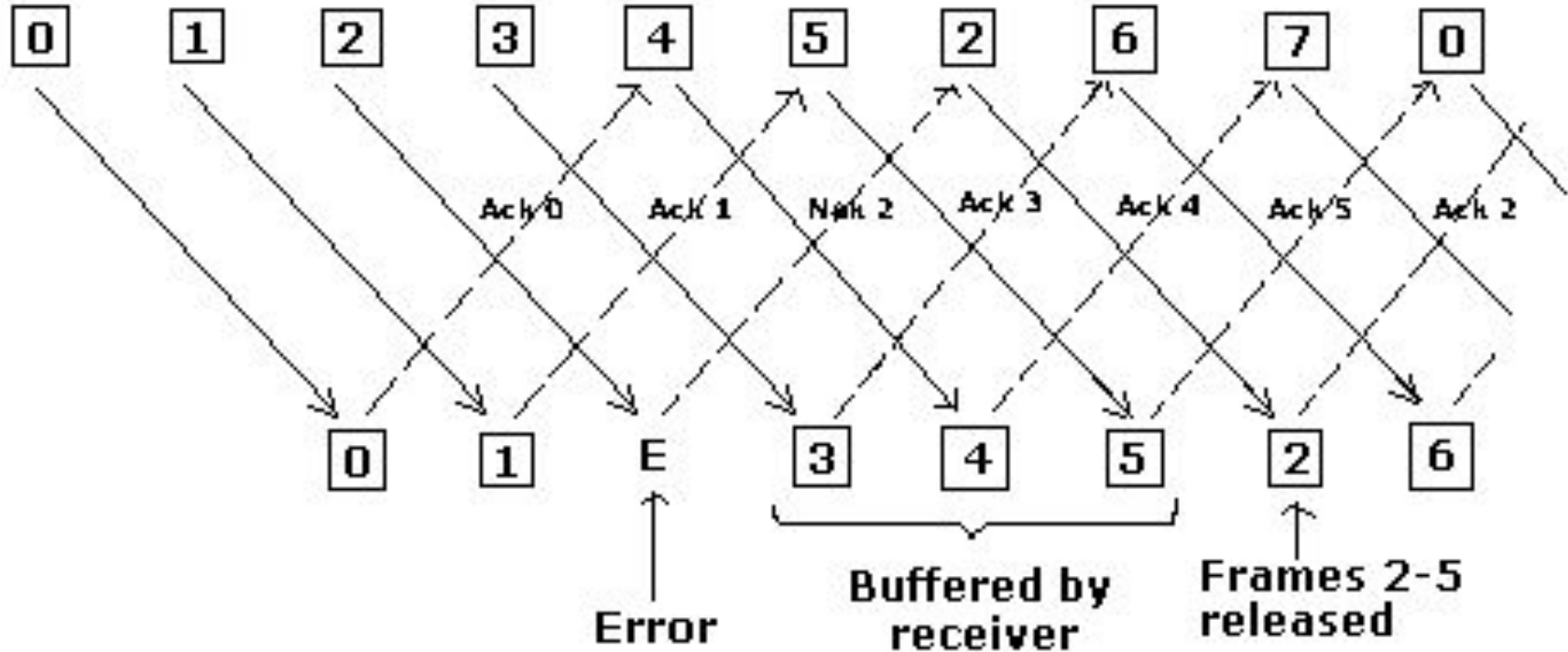




# Go-back-N



# Selective Repeat



# Design Considerations



# When To Retransmit?

- Assuming sender has decided to retransmit a packet: when do we do so?
  - In a Binary Symmetric Channel (BSC), any time is as good as any
  - In fading channels, **try to avoid bad channel states** – postpone transmissions
- How long to wait?
  - Example solution: **Probing protocol** (Zorzi and Rao)
  - Idea: reflect channel state by two protocol modes, “normal” and “probing”
  - When error occurs, go from normal to probing mode
  - In probing mode, periodically send short packets (acknowledged by receiver) – when successful, go to normal mode

M. Zorzi and R. R. Rao, "Error control and energy consumption in communications for nomadic computing," in *IEEE Transactions on Computers*, vol. 46, no. 3, pp. 279-289, March 1997, doi: 10.1109/12.580424.



# Comparison: FEC vs. ARQ

- FEC
  - Constant overhead for each packet
  - Not (easily) possible to adapt to changing channel characteristics
- ARQ
  - Overhead only when errors occurred (expect for ACK, always needed)
- Both schemes have their uses ! **hybrid schemes**



# Optimize: Power Control On a Link Level

- Further controllable parameter: transmission power
  - Higher power, lower error rates – less FEC/ARQ necessary
  - Lower power, higher error rates – higher FEC necessary
- Tradeoff!



# Optimize: Packet Size

- Small packets: low packet error rate, high packetization overhead
- Large packets: high packet error rate, low overhead
- Depends on bit error rate, energy consumption per transmitted bit
- For known bit error rate (BER), optimal frame length is easy to determine
- Problem: how to estimate BER?
  - Collect channel state information at the receiver (RSSI, FEC decoder information, ...)
  - For example, use number of attempts  $T$  required to transmit the last  $M$  packets as an estimator of the packet error rate (assuming a BSC)
- Second problem: how long are observations valid/how should they be aged?
  - Only recent past is credible



# Optimize: Link Management

- Decide to which neighbors (which can be *more or less* reachable) a link should be established
  - Communication quality fluctuates
  - Far neighbors can be costly to talk to, error-prone
  - Quality can only be estimated
- Establish a ***neighborhood table*** for each node
  - Partially automatically constructed by MAC protocols

