

Ad hoc and Sensor Networks

Chapter 6: Link layer protocols

Holger Karl



Computer Networks Group
Universität Paderborn

Goals of this chapter – Link layer tasks in general

- **Framing** – group bit sequence into packets/frames
 - Important: format, size
 - **Error control** – make sure that the sent bits arrive and no other
 - Forward and backward error control
 - Flow control – ensure that a fast sender does not overrun its 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
- ! Understand the issues involved in turning the radio communication between two neighboring nodes into a somewhat reliable *link*



SS 05

Ad hoc & sensor networks - Ch 6: Link layer protocols

2

Overview

- **Error control**
- Framing
- Link management



SS 05

Ad hoc & sensor networks - Ch 6: Link layer protocols

3

Error control

- Error control has to ensure that data transport is
 - Error-free – deliver exactly the sent bits/packets
 - In-sequence – deliver them in the original order
 - Duplicate-free – and at most once
 - Loss-free – and at least once
- Causes: fading, interference, loss of bit synchronization, ...
 - Results in bit errors, bursty, sometimes heavy-tailed runs (see physical layer chapter)
 - In wireless, sometimes quite high average bit error rates – 10^{-2} ... 10^{-4} possible!
- Approaches
 - Backward error control – ARQ
 - Forward error control – FEC



SS 05

Ad hoc & sensor networks - Ch 6: Link layer protocols

4

Backward error control – ARQ

- Basic procedure (a quick 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
 - Optimal timer setting?
 - 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

- Alternating bit – at most one packet outstanding, single bit sequence number
- Go-back N – send up to N packets, if a packet has not been acknowledged when timer goes off, retransmit all unacknowledged packets
- Selective Repeat – when timer goes off, only send that particular packet



How to use acknowledgements

- Be careful about ACKs from different layers
 - A MAC ACK (e.g., S-MAC) does not necessarily imply buffer space in the link layer
 - On the other hand, having both MAC and link layer ACKs is a waste
- Do not (necessarily) acknowledge every packet – use cumulative ACKs
 - Tradeoff against buffer space
 - Tradeoff against number of negative ACKs to send



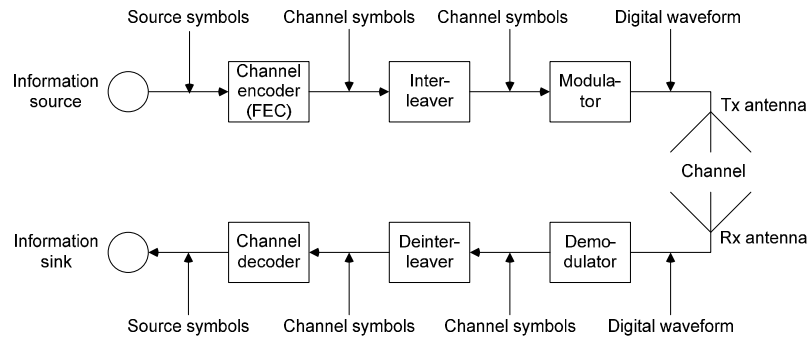
When to retransmit

- Assuming sender has decided to retransmit a packet – when to do so?
 - In a BSC channel, any time is as good as any
 - In fading channels, try to avoid bad channel states – postpone transmissions
 - Instead (e.g.): send a packet to another node if in queue (exploit multi-user diversity)
- How long to wait?
 - Example solution: Probing protocol
 - 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



Forward error control

- Idea: 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



Block-coded FEC

- Level of redundancy: **blocks of symbols**
 - Block: k p-ary source symbols (not necessarily just bits)
 - Encoded into n q-ary channel symbols
- Injective mapping (**code**) of p^k source symbols ! q^n channel symbols
- Code rate**: $(k \log p) / (n \log q)$
 - When $p=q=2$: k/n is code rate
- For $p=q=2$: Hamming bound – code can correct up to t bit errors only if

$$2^{n-k} \geq \sum_{i=0}^t \binom{n}{i}$$

- Codes for (n,k,t) do not always exist



Popular block codes

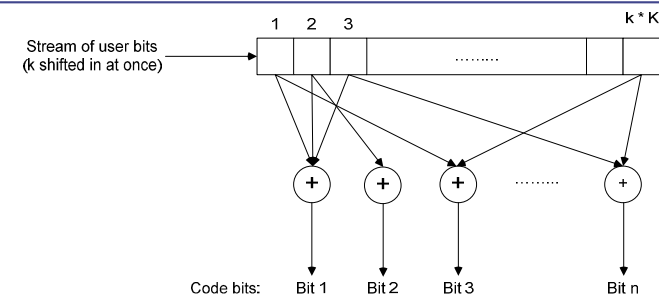
- Popular examples
 - Reed-Solomon codes (RS)
 - Bose-Chaudhuri-Hocquenghem codes (BCH)
- Energy consumption
 - E.g., BCH encoding: negligible overhead (linear-feedback shift register)
 - BCH decoding: depends on block length and Hamming distance (n, t as on last slide)

$$E_{dec} = (2nt + 2t^2) \cdot (E_{add} + E_{mult})$$

- Similar for RS codes



Convolutional codes

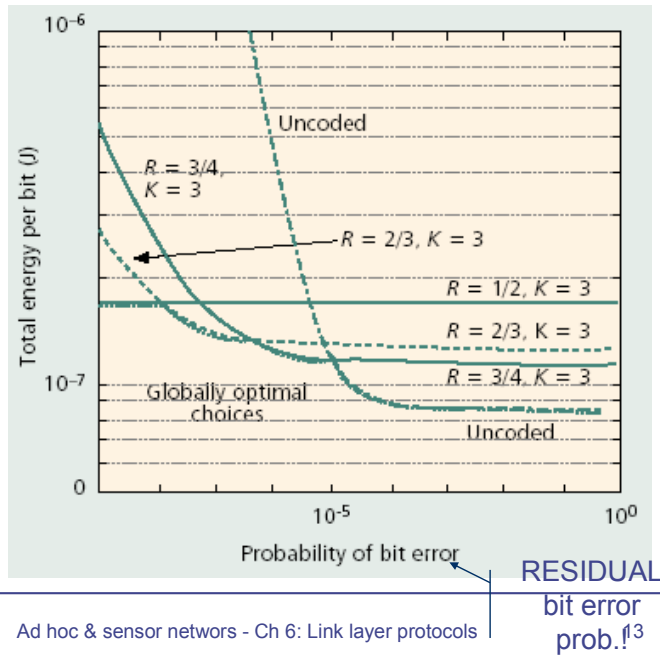


- Code rate: ratio of k user bits mapped onto n coded bits
- Constraint length K determines **coding gain**
- Energy
 - Encoding: cheap
 - Decoding: Viterbi algorithm, energy & memory depends exponentially (!) on constraint length



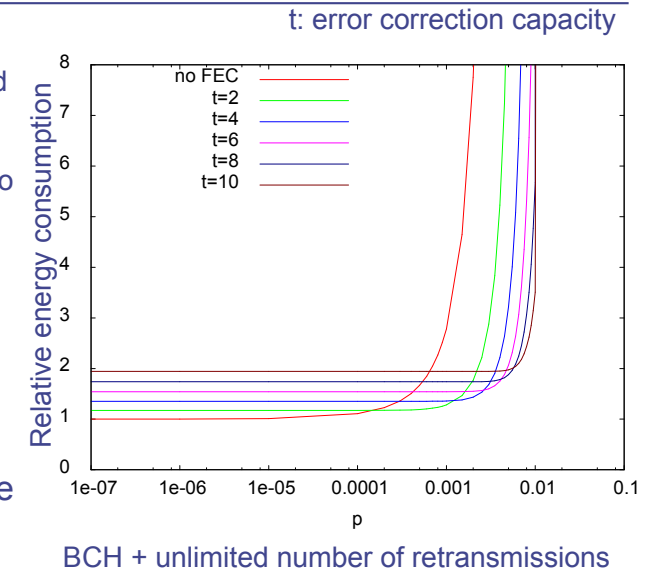
Energy consumption of convolutional codes

- Tradeoff between coding energy and reduced transmission power (coding gain)
- Overall: block codes tend to be more energy-efficient



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**



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!



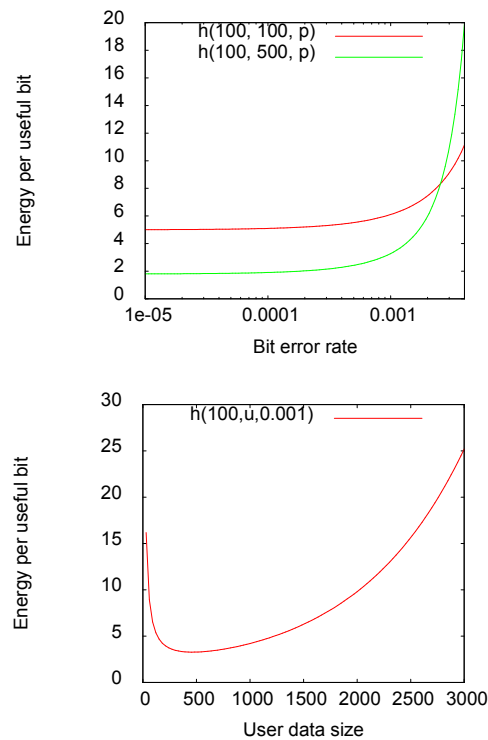
Overview

- Error control
- **Framing**
- Link management



Frame, 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
- Notation: $h(\text{overhead}, \text{payload size}, \text{BER})$



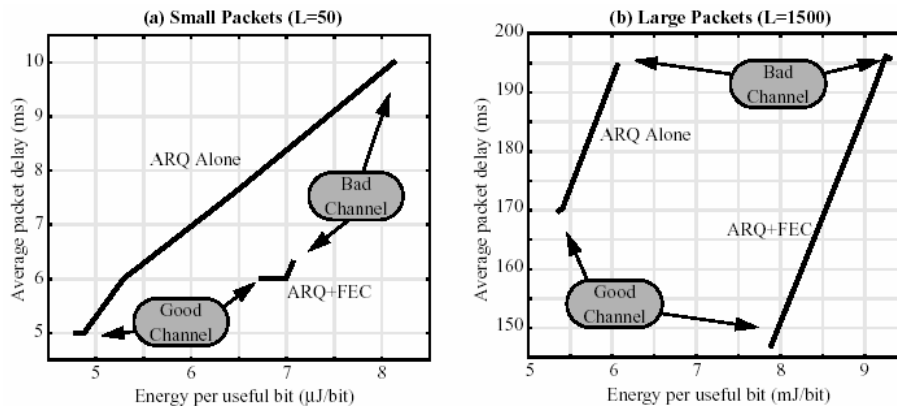
Dynamically adapt frame length

- 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, ...)
 - Example: Use number of attempts T required to transmit the last M packets as an estimator of the packet error rate (assuming a BSC)
 - Details: homework assignment
- Second problem: how long are observations valid/how should they be aged?
 - Only recent past is – if anything at all – somewhat credible



Putting it together: ARQ, FEC, frame length optimization

- Applying ARQ, FEC (both block and convolutional codes), frame length optimization to a Rayleigh fading channel
 - Channel modeled as Gilbert-Elliot



Overview

- Error control
- Framing
- **Link management**



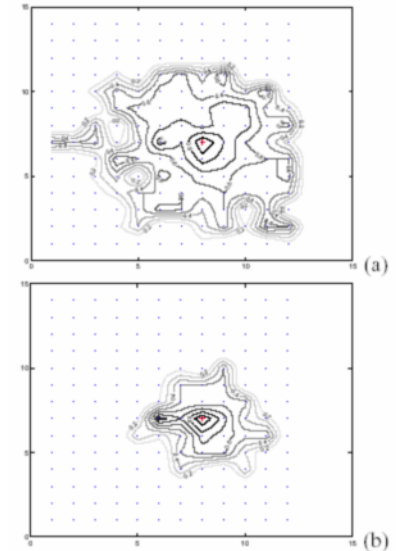
Link management

- Goal: decide to which neighbors that are *more or less* reachable a link should be established
 - Problem: communication quality fluctuates, far away 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



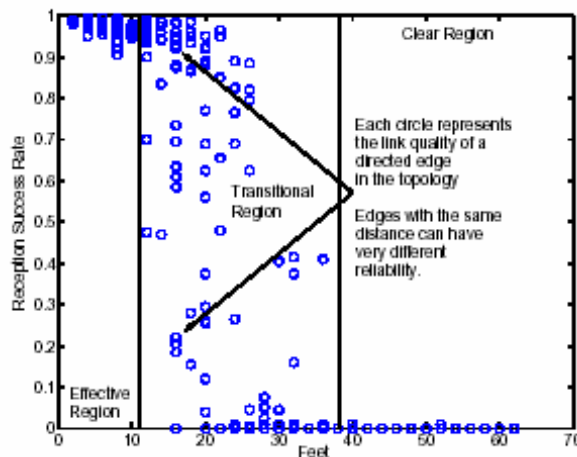
Link quality characteristics

- Expected: simple, circular shape of “region of communication” – not realistic
- Instead:
 - Correlation between distance and loss rate is weak; iso-loss-lines are not circular but irregular
 - Asymmetric links are relatively frequent (up to 15%)
 - Significant short-term PER variations even for stationary nodes



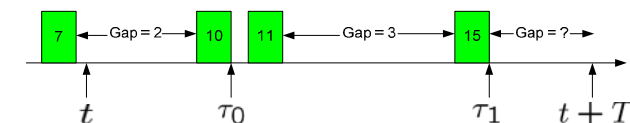
Three regions of communication

- **Effective region:** PER consistently < 10%
- **Transitional region:** anything in between, with large variation for nodes at same distance
- **Poor region:** PER well beyond 90%



Link quality estimation

- How to estimate, on-line, in the field, the actual link quality?
- Requirements
 - Precision – estimator should give the statistically correct result
 - Agility – estimator should react quickly to changes
 - Stability – estimator should not be influenced by short aberrations
 - Efficiency – Active or passive estimator



- Example: WMEWMA only estimates at fixed intervals

$$P_n = \alpha P_{n-1} + (1 - \alpha) \frac{r_n}{r_n + f_n}$$

r_n : received packets in interval
 f_n : packets identified as lost



Conclusion

- Link layer combines traditional mechanisms
 - Framing, packet synchronization, flow control
- with relatively specific issues
 - Careful choice of error control mechanisms – tradeoffs between FEC & ARQ & transmission power & packet size ...
 - Link estimation and characterization

