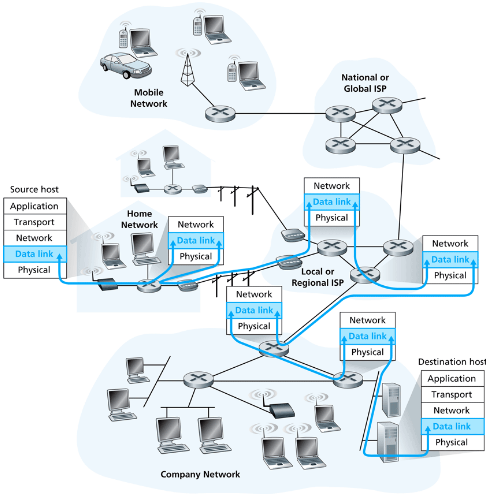
Error Detection and Multiple Access

Daniel Zappala

CS 460 Computer Networking
Brigham Young University

Link Layer

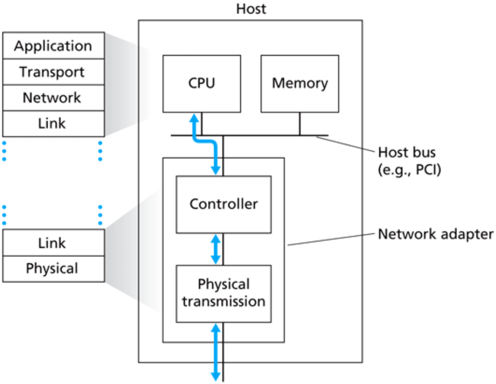


- link layer is responsible for transferring a frame of data between two nodes over a single link
- many different types of links
    - wired and wireless
    - point-to-point vs shared medium
- each link can potentially run a different protocol (Ethernet, 802.11), may provide different services (reliability, multicast)

Link Layer Services

- framing, data transmission
  - encapsulate data into a frame, adding header and/or trailer
  - may use link-layer addressing, likely different from IP addresses
  - negotiate channel access on shared medium
- reliable delivery
  - seldom used on links with low error rates (fiber)
  - wireless links have high error rates, may include retransmission
  - why implement this at both link layer and transport layer?
- flow control
- error detection and correction
  - errors caused by noise, collisions
  - identify and correct some bit errors others cause packet loss or retransmission
- half or full duplex transmission
- broadcast, multicast on shared medium

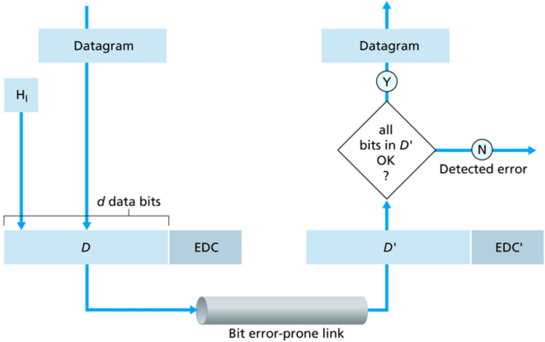
Hardware



- adaptor or (NIC) combines physical and link layers
- sending
  - encapsulate IP packet in a frame
  - add header and trailer: error checking, flow control
- receiving
  - check/correct errors
  - handle flow control
  - extract IP packet and deliver to IP

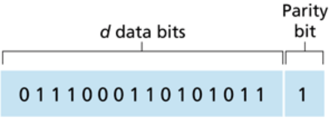
Error Detection and Correction

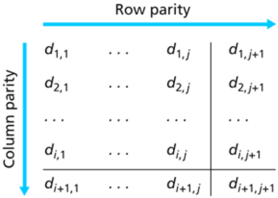
Error Detection and Correction

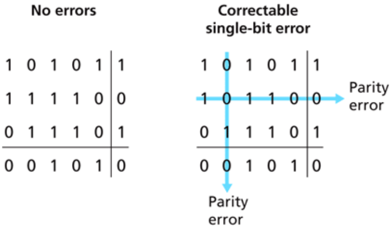


- EDC = Error Detection and Correction bits
- not 100% reliable - very small chance that some errors are missed
- more EDC bits ⇒ better detection and correction

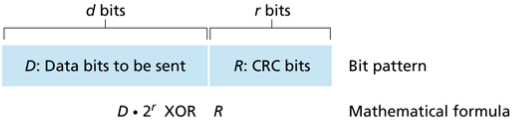
Parity Checking





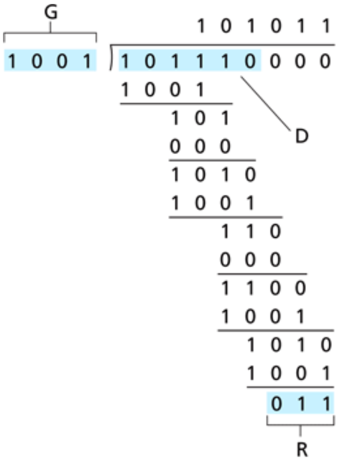


CRC



- treat D data bits as a binary number
- choose an $r+1$ pattern, generator G, known by sender and receiver
- sender
  - choose r CRC bits such that $D * 2^r \oplus R$ exactly divisible by G (modulo 2)
- receiver
  - divide data received by G
  - if non-zero remainder: error detected
  - can detect all burst errors less than $r+1$ bits
- CRC-32: Ethernet, FDDI

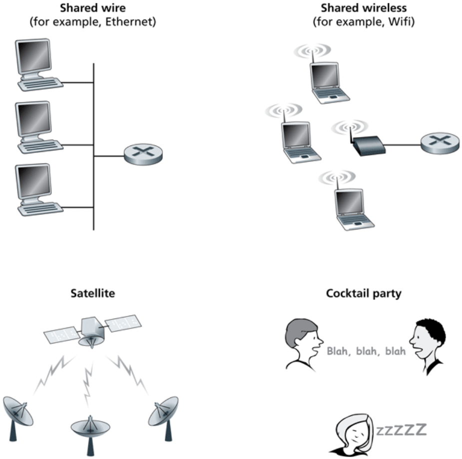
CRC Example



- transmit $D * 2^r \oplus R = 101110011$

Multiple Access Links

- links shared among many nodes



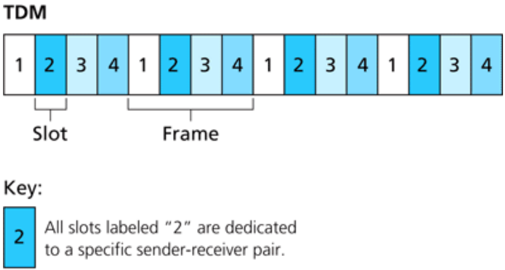
Multiple Access Protocol

- protocol for sharing a single broadcast channel
- protocol determines when each node can transmit
  - must avoid collisions
  - collision: two or more simultaneous transmissions lead to interference so that a signal can't be received
- challenge: communication about channel sharing must use the channel itself!
- ideal protocol for a channel of rate $Rbps$
  - when one node wants to send, transmit at rate R
  - when m nodes want to transmit, each sends at R/m
  - fully decentralized: no master node, no clock synchronization
  - simple to implement

Types of MAC Protocols

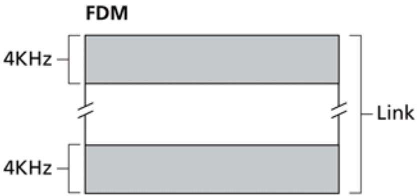
- channel partitioning
  - divide channel into pieces (time slots, frequency, code)
  - allocate each piece to a single node
- random access
  - listen to see if anyone else is sending
  - any node can send if the channel is clear
  - detect and recover from collisions
- taking turns
  - nodes coordinate with a master or a token
  - on its turn, node can send as much as it needs to (up to some maximum size)

TDMA: Time Division Multiple Access



- divide channel into slots based on time
- one slot per node per round
- unused slots are idle
- used in GSM cell phones

FDMA: Frequency Division Multiple Access



- divide channel into frequency bands
- one frequency band per node
- unused time in frequency band is idle
- used in conjunction with TDMA

Random Access Protocols

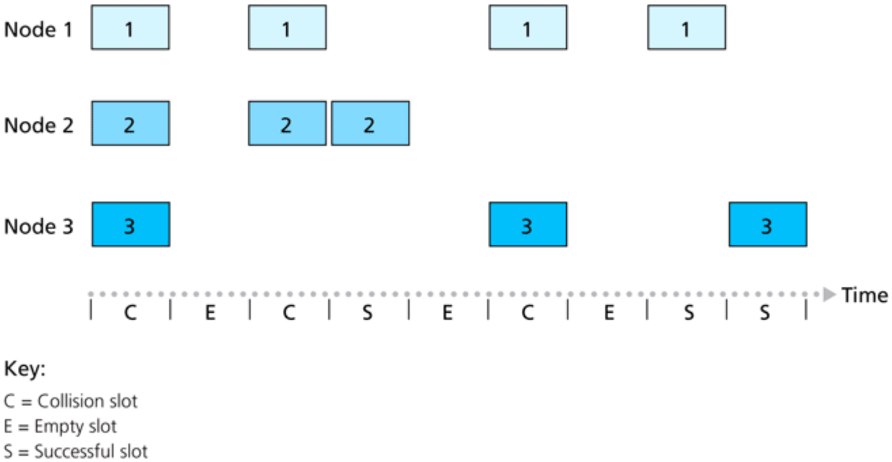
Random Access

- to send a packet
    - transmit at full channel rate
    - some protocols require listening first to see if channel free
- must detect and recover from collisions
- examples
    - ALOHA: slotted and unslotted
    - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

- assumptions
  - all frames same size
  - time is divided into equal size slots = time to transmit 1 frame
  - nodes transmit frames only at beginning of slots
  - nodes are synchronized
  - if 2 or more nodes transmit in slot, all nodes detect collision
- operation
  - when node wants to send, transmit in next slot
  - if no collision, node can send new frame in next slot
  - if collision, node retransmits frame in each subsequent slot with probability p until it succeeds

Slotted ALOHA Example



Slotted ALOHA Pros and Cons

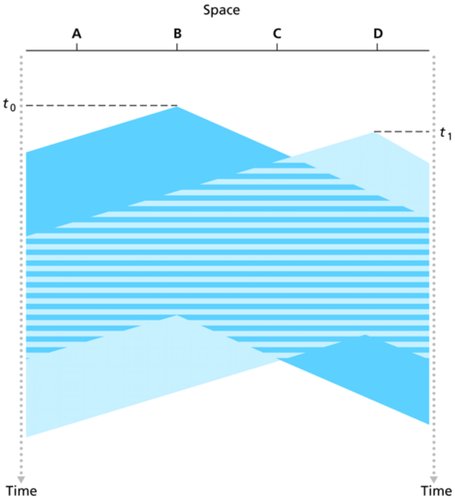
- pros
  - single active node can continuously transmit at full rate of channel
  - highly decentralized
  - simple
- cons
  - not very efficient
    - channel utilization is 37% (see book)
    - excess collisions cause wasted slots
    - too many idle slots
  - clock synchronization required
- nodes should be able to detect collision in less than time to transmit packet
  - rest of the slot is wasted
  - but unslotted ALOHA efficiency is even worse (18%)

CSMA: Carrier Sense Multiple Access

- solution to problems with ALOHA
- listen before you transmit (carrier sensing)
  - if the channel is busy, wait until later
  - otherwise, send the data

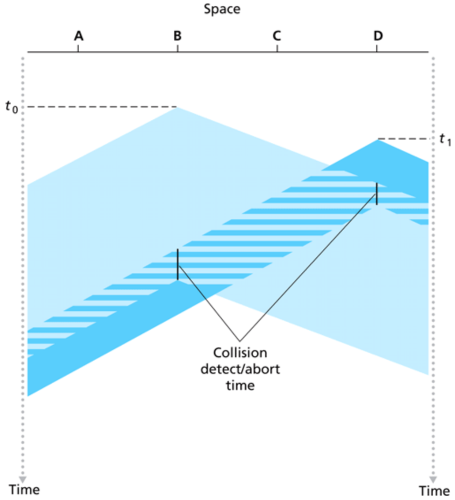
CSMA Collisions

- collisions occur even with carrier sensing: propagation delay means two nodes can start before they hear each other
- collision wastes all time when packets are transmitted



CSMA/CD: Collision Detection

- add ability to stop transmission once a collision is detected
- collision detection
  - easy for wired links: measure signal strengths and compare transmitted and received signals
  - hard for wireless links: turn off receiver when transmitting
- very successful - used in Ethernet, 802.11 networks



Taking Turns

Taking Turns

- channel partitioning
  - share channel efficiently and fairly at high load
  - inefficient at low load: high delay, only $1/N$ of bandwidth
- random access
  - efficient at low load: low delay, full bandwidth
  - inefficient at high load: collisions
- taking turns
  - try to have the best of both types
  - send at full rate when it is your turn
  - share fairly when everyone wants a turn

Types of Protocols

- polling
  - master node checks with each other node to see if it wants to send
  - problems
    - polling overhead
    - latency
    - single point of failure
- token passing
  - pass a control token among all nodes in a ring
  - problems
    - token passing overhead
    - latency before you get the token again
    - token can be lost
  - used in FDDI: fiber-based token rings