Error Coding

- Transmission process may introduce errors into a message.
  - Single bit errors versus burst errors
- Detection: e.g. CRC
  - Requires a check that some messages are invalid
  - Hence requires extra bits
  - "redundant check bits"
- Correction
  - Forward error correction: many related code words map to the same data word
  - Detect errors and retry transmission

Parity

- Even parity
  - Append parity bit to 7 bits of data to make an even number of 1s
  - Odd parity accordingly defined.
  - 1 in 8 bits of overhead?
    - When is this a problem?
- Can detect a single error
  - 100010
- But nothing beyond that
2-D Parity

- Make each byte even parity
- Finally, a parity byte for all bytes of the packet
- Example: five 7-bit character packet, even parity

```
0110100
1011010
0010110
1101001
1001011
10001010
```

Effectiveness of 2-D Parity

- 1-bit errors can be detected, corrected
- Example with even parity per byte:

```
0110100
1011010
000
1110101
1001011
10001100
```

- Odd number of 1's

Effectiveness of 2-D Parity

- 2-bit errors can also be detected
- Example:

```
0110100
1011010
00011
1110101
1001011
10001100
```

- Odd number of 1's

- What about 3-bit errors? >3-bit errors?
Cyclic Redundancy Codes (CRC)

- Commonly used codes that have good error detection properties
  - Can catch many error combinations with a small number of redundant bits
- Based on division of polynomials
  - Errors can be viewed as adding terms to the polynomial
  - Should be unlikely that the division will still work
- Can be implemented very efficiently in hardware
- Examples:
  - CRC-32: Ethernet
  - CRC-8, CRC-10, CRC-32: ATM

Link Flow Control and Error Control

- Dealing with receiver overflow: flow control.
- Dealing with packet loss and corruption: error control.
- Actually these issues are relevant at many layers.
  - Link layer: sender and receiver attached to the same "wire"
  - End-to-end: transmission control protocol (TCP) - sender and receiver are the end points of a connection
- How can we implement flow control?
  - "You may send" (windows, stop-and-wait, etc.)
  - "Please shut up" (source quench, 802.3x pause frames, etc.)

Flow Control: A Naïve Protocol

- Sender simply sends to the receiver whenever it has packets.
- Potential problem: sender can outrun the receiver.
  - Receiver too slow, small buffer overflow, ...
- Not always a problem: receiver might be fast enough.
Adding Flow Control

- Stop and wait flow control: sender waits to send the next packet until the previous packet has been acknowledged by the receiver.
  - Receiver can pace the sender
- Drawbacks: adds overheads, slowdown for long links.

Window Flow Control

- Stop and wait flow control results in poor throughput for long-delay paths: packet size/roundtrip-time.
- Solution: receiver provides sender with a window that it can fill with packets.
  - The window is backed up by buffer space on receiver
  - Receiver acknowledges the a packet every time a packet is consumed and a buffer is freed

Window Limitations

Throughput = \( \frac{\text{Window Size}}{\text{Roundtrip Time}} \)
Error Control: Stop and Wait Case

- Packets can get lost, corrupted, or duplicated.
- Duplicate packet: use sequence numbers.
- Lost packet: timeouts and acknowledgements.
  - Positive versus negative acknowledgements
  - Sender side versus receiver side timeouts
- Window based flow control: more aggressive use of sequence numbers (see transport lectures).

What is Used in Practice?

- No flow or error control.
  - E.g. regular Ethernet, just uses CRC for error detection
- Flow control only.
  - E.g. Gigabit Ethernet
- Flow and error control.
  - E.g. X.25 (older connection-based service at 64 Kbs that guarantees reliable in order delivery of data)

Switching and Media Access Control

- How do we transfer packets between two hosts connected to the a switched network?
- Switches connected by point-to-point links -- store-and-forward.
  - Multiplexing and forwarding
  - Used in WAN, LAN, and for home connections
  - Conceptually similar to ‘routing’
  - But at the datalink layer instead of the network layer
- Multiple access networks -- contention based.
  - Multiple hosts are sharing the same transmission medium
  - Used in LANs and wireless
  - Need to control access to the medium
A Switch-based Network

- Switches are connected by "point-to-point" links.
- Packets are forwarded hop-by-hop by the switches towards the destination.
  - Many forms of forwarding
- Many datalink technologies use switching.
  - Virtual circuits: Frame-relay, ATM, X.25, ...
  - Packets: Ethernet, MPLS, ...

Three techniques for switching

- **Global addresses - connection-less**
  - Routers keep next hop for destination
  - Packets carry destination address
- **Virtual circuits - connection oriented**
  - Connection routed through network to set up state
  - Packets forwarded using connection state
- **Source routing**
  - Packet carries path

Global Address Example
Global Addresses

• Advantages
  - Stateless - simple error recovery
• Disadvantages
  - Every switch knows about every destination
    - Potentially large tables
  - All packets to destination take same route
  - Need special approach to fill table

Simplified Virtual Circuits Example

Virtual Circuits

• Advantages
  - Efficient lookup (simple table lookup)
  - Can reserve bandwidth at connection setup
  - Easier for hardware implementations
• Disadvantages
  - Still need to route connection setup request
  - More complex failure recovery - must recreate connection state
• Typical use → fast router implementations
  - ATM - combined with fix sized cells
  - MPLS - tag switching for IP networks
Source Routing

- Advantages
  - Switches can be very simple and fast
- Disadvantages
  - Variable (unbounded) header size
  - Sources must know or discover topology (e.g., failures)
- Typical uses
  - Ad-hoc networks (DSR)
  - Machine room networks (Myrinet)

Comparison

<table>
<thead>
<tr>
<th>Source Routing</th>
<th>Global Addresses</th>
<th>Virtual Circuits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Size</td>
<td>Word</td>
<td>OK – Large address</td>
</tr>
<tr>
<td>Router Table Size</td>
<td>None</td>
<td>Number of hosts</td>
</tr>
<tr>
<td>Forward Overhead</td>
<td>Best</td>
<td>Table lookup</td>
</tr>
<tr>
<td>Setup Overhead</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Error Recovery</td>
<td>Tell all hosts</td>
<td>Tell all switches</td>
</tr>
</tbody>
</table>
Most Popular:
Address Lookup-based Approach

- Address from header:
  - Absolute address (e.g. Ethernet)
  - (IP address for routers)
  - (VC identifier, e.g. ATM)
- Next hop: output port for packet.
- Info: priority
- We will see how this table is filled (learning bridges)

### Multiple Access Protocols

- Prevent two or more nodes from transmitting at the same time over a broadcast channel.
  - If they do, we have a collision, and receivers will not be able to interpret the signal
- Several classes of multiple access protocols
  - Partitioning the channel, e.g. frequency-division or time division multiplexing
    - With fixed partitioning of bandwidth - not flexible
  - Taking turns, e.g. token-based, reservation-based protocols, polling based
  - Contention based protocols, e.g. Aloha, Ethernet
- Next lecture

### Fiber Distributed Data Interface (FDDI)

- One token holder may send, with a time limit.
  - Known upper bound on delay.
- Optical version of 802.5 token ring, but multiple packets may travel in train: token released at end of frame.
- 100 Mbps, 100km.
- Optional dual ring for fault tolerance.
- CDI: FDDI over unshielded twisted pair, shorter range
Other “Taking Turn” Protocols

- Central entity polls stations, inviting them to transmit.
  - Simple design – no conflicts
  - Not very efficient – overhead of polling operation
- Stations reserve a slot for transmission.
  - For example, break up the transmission time in contention-based and reservation based slots
    - Contention based slots can be used for short messages or to reserve time.
    - Communication in reservation based slots only allowed after a reservation is made.
  - Issues: fairness, efficiency