**The Data Link Layer:**

Responsible for delivering frames of information over a single link. Handles transmission errors and regulates the flow of data

**Design issues**

**Frames:** Link layer accepts packets from the network layer, and encapsulates them into frames that it sends using the physical layer; reception is the opposite process.

**Possible services**:

Unacknowledged connectionless service

* Frame is sent with no connection / error recovery. Example - Ethernet

Acknowledged connectionless service

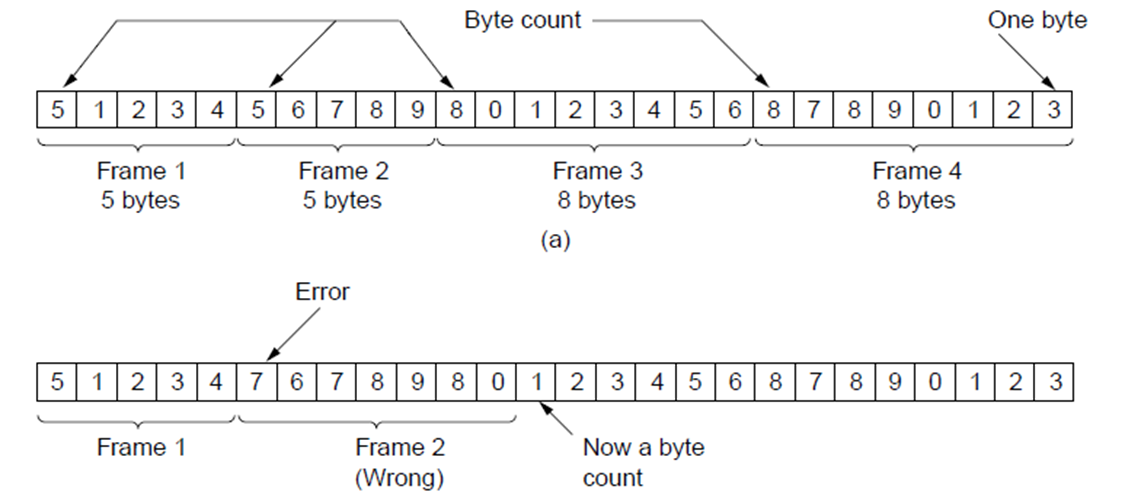
* Frame is sent with retransmissions if needed. Example - WiFi

Acknowledged connection-oriented service

* Connection is set up; rare. This was common in past

**Framing methods**

**Byte count:** Frame begins with a count of the number of bytes in it. This is simple, but it is difficult to resynchronize after an error.

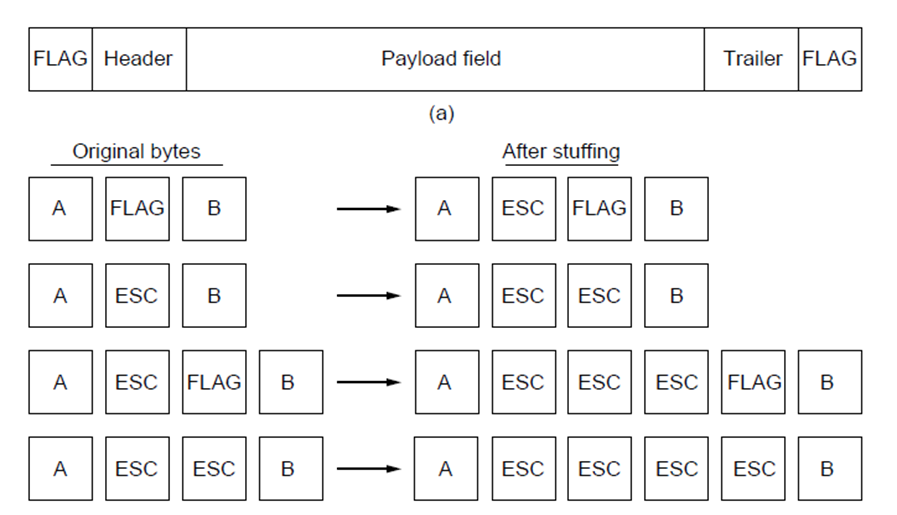


Error

Case

Expected Case

**Flag bytes with byte stuffing:** Special flag bytes delimit frames; occurrences of flags in the data must be stuffed (escaped.) This is longer, but easy to resynchronize after error. This is used in PPP (Point-to-Point Protocol)



Need to escape extra ESCAPE bytes too!

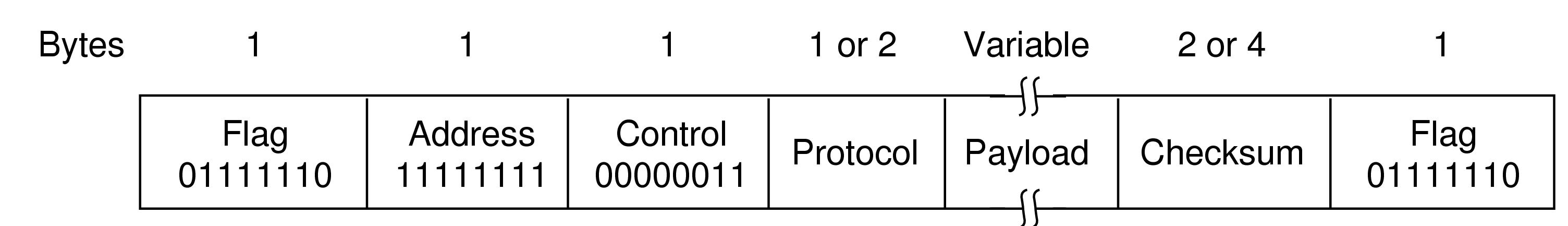
Frame

format

Stuffing

examples

**PPP (Point-to-Point Protocol)** is a general method for delivering packets across links. Framing uses a flag (0x7E) and byte stuffing. There is an“Unnumbered mode” (connectionless unacknowledged service) which is used to carry IP packets. Errors are detected with a checksum.



IP packet

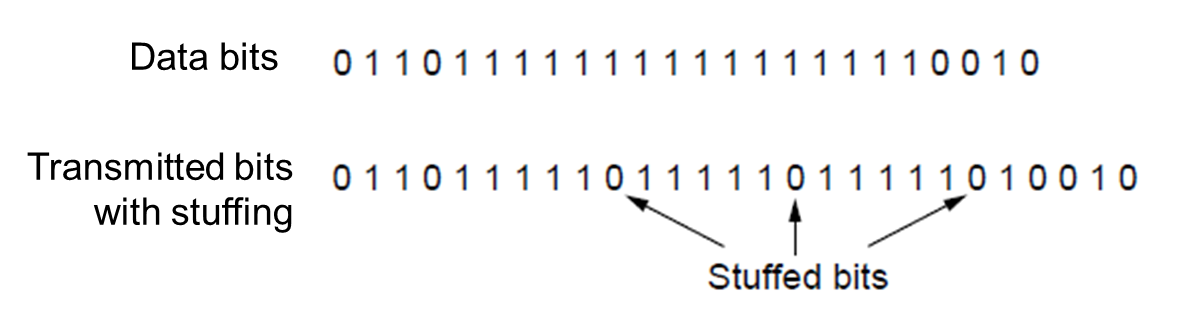
0x21 for IPv4

b’cast addr

header

**Flag bits with bit stuffing:** Stuffing done at the bit level.

* Frame flag has six consecutive 1s (not shown)
* On transmit, after five 1s in the data, a 0 is added
* On receive, a 0 after five 1s is deleted
* Used in HDLC (formerly commonly used)



**Error control:** Error control repairs frames that are received in error. This requires errors to be detected at the receiver. Typically, retransmit the unacknowledged frames. A timer protects against lost acknowledgements.

**Flow control:** Prevents a fast sender from out-pacing a slow receiver. Flow control is mostly the responsibility of the Transport layer in practice (e.g., TCP)

**Error detection and correction:** Error codes add structured redundancy to data so errors can be either detected, or corrected.

|  |  |
| --- | --- |
| Error detection codes: | Error correction codes: |
| * Parity | * Hamming codes |
| * Checksums | * Binary convolutional codes |
| * Cyclic redundancy codes | * Reed-Solomon and Low-Density Parity Check codes   (Mathematically complex, widely used in real systems) |

**Forward Error Control (FEC):** a technique used for controlling errors in data transmission over unreliable or noisy communication channels. The central idea is the sender encodes his message in a redundant way by using an error-correcting code (ECC).

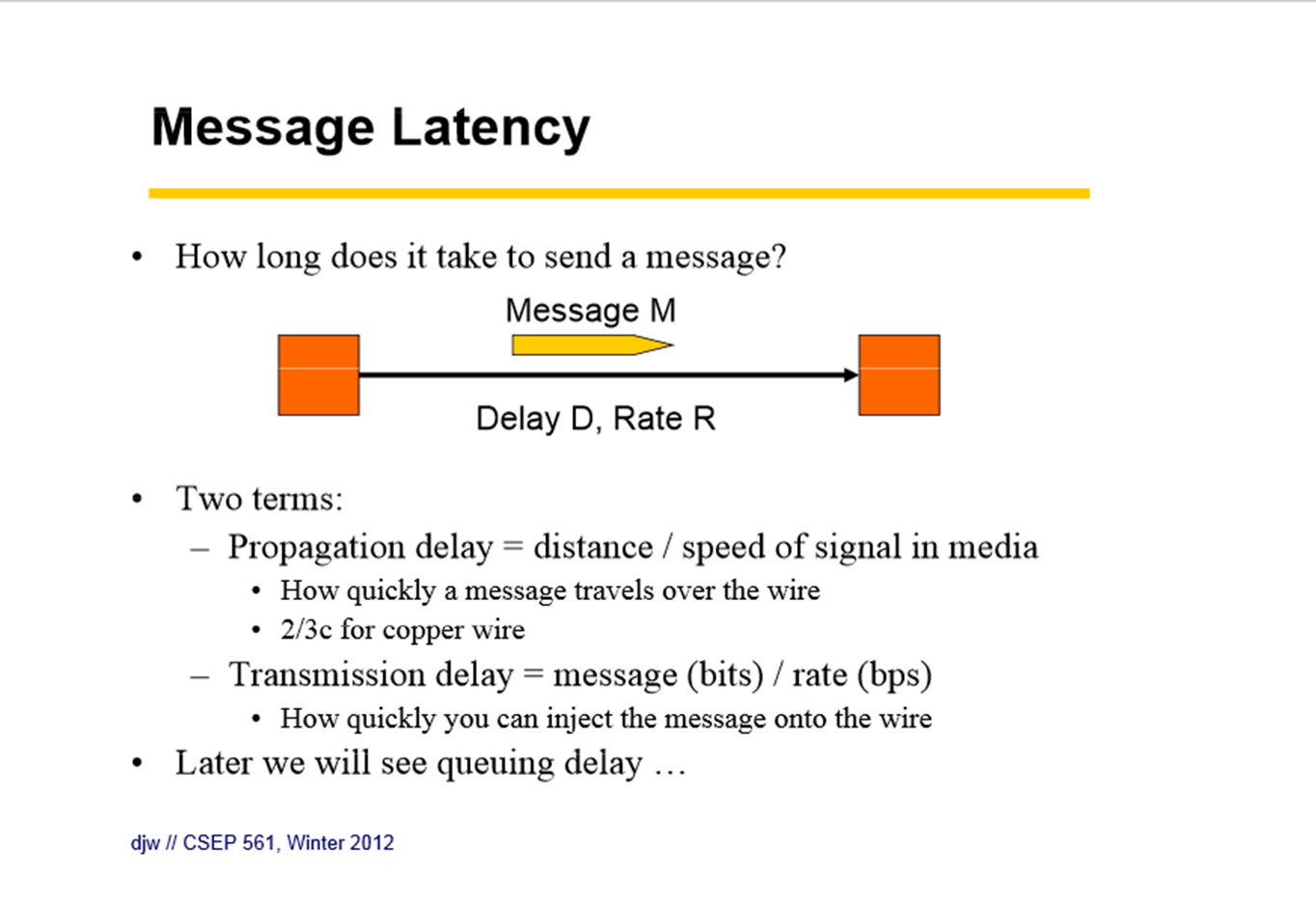
**Parity:** a [bit](http://en.wikipedia.org/wiki/Bit) added to the end of a string of [binary code](http://en.wikipedia.org/wiki/Binary_code) that indicates whether the number of bits in the string with the value [one](http://en.wikipedia.org/wiki/1_%28number%29) is [even](http://en.wikipedia.org/wiki/Even_number) or [odd](http://en.wikipedia.org/wiki/Odd_number). Parity bits are used as the simplest form of [error detecting code](http://en.wikipedia.org/wiki/Error_detection_and_correction). Parity bit is added as the modulo 2 sum of data bits

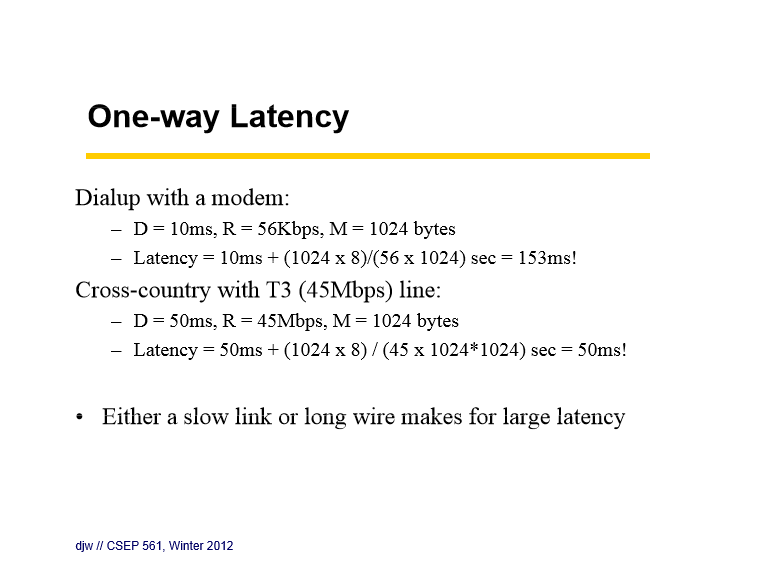
* + Ex: 1110000 🡪 11100001
  + Ex: 1 error, 11100101; detected, sum is wrong
  + Ex: 3 errors, 11011001; detected sum is wrong
  + Ex: 2 errors, 11101101; *not* *detected*, sum is right!

**Checksum:** Checksum treats data as N-bit words and adds N check bits that are the modulo 2N sum of the words Improved error detection over parity bits. This detects bursts up to N errors and detects random errors with probability 1-2N. However it is vulnerable to systematic errors, e.g., added zeros.

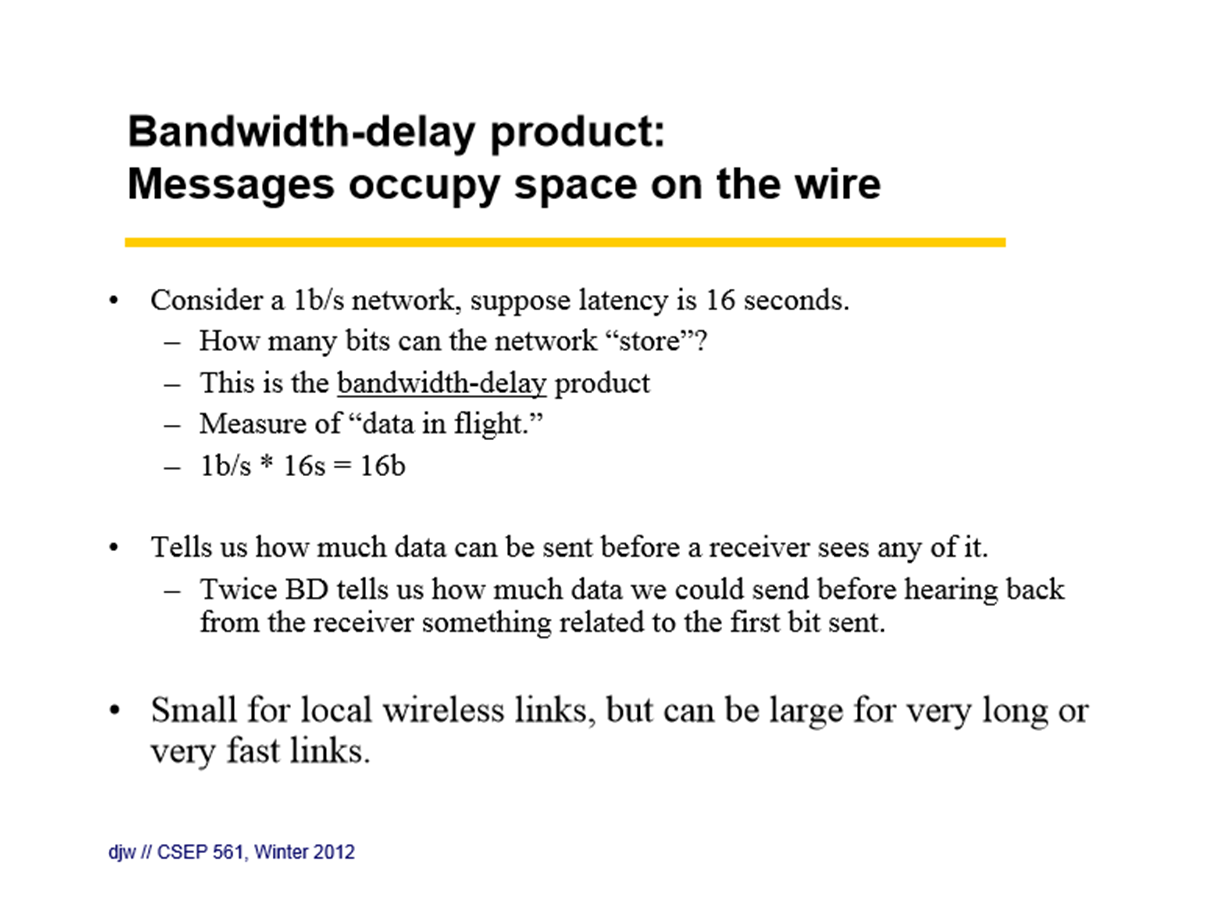
**Cyclical Redundancy checks (CRC):** Adds bits so that transmitted frame viewed as a polynomial is evenly divisible by a generator polynomial. Stronger detection than checksums: E.g., can detect all double bit errors. It is not vulnerable to systematic errors. Also has property that all bursts of 32 bits or less are detected as well as all bursts affecting an odd number of bits.

**Latency, delay and rate:**



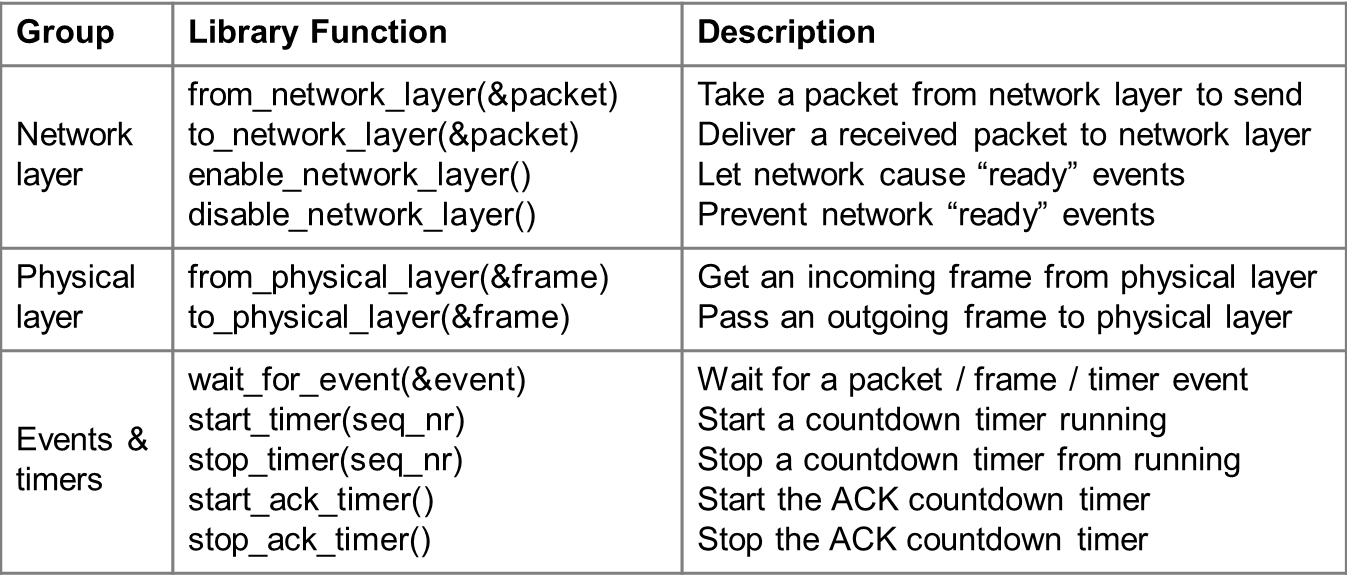


**Bandwidth-delay product:**



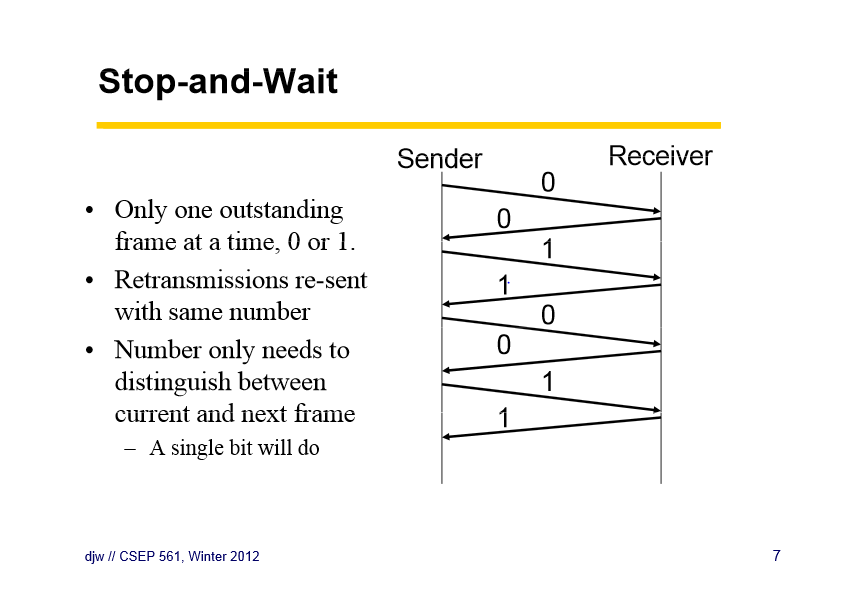
**Pipelining:** multiple frames outstanding in the network at any instant.

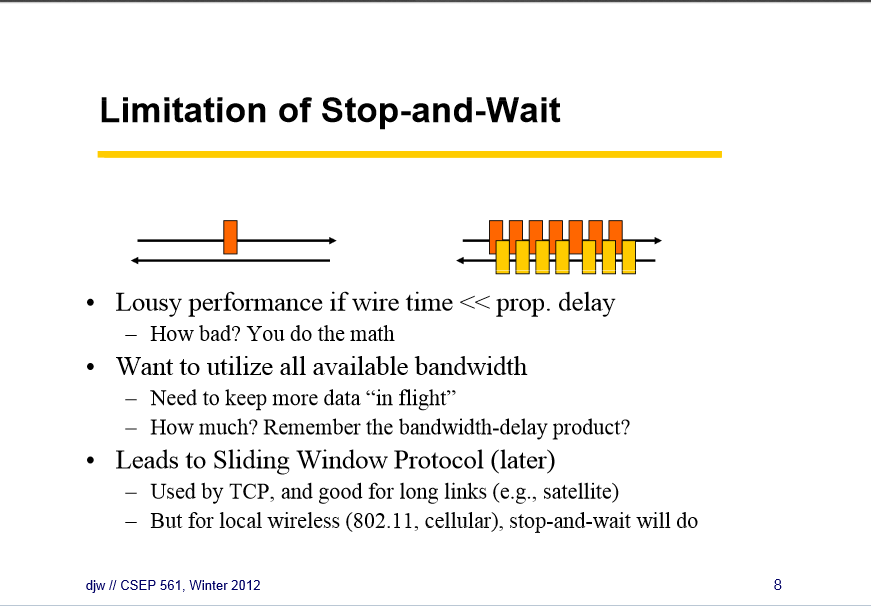
**Network interface (NIC) card:** a computer hardware component that connects a computer to a computer network. The network controller implements the electronic circuitry required to communicate using a specific physical layer and data link layer standard such as Ethernet, Fibre Channel, Wi-Fi or Token Ring. This provides a base for a full network protocol stack, allowing communication among small groups of computers on the same local area network (LAN) and large-scale network communications through routable protocols, such as Internet Protocol (IP).



**Stop-and-wait protocol:**

Protocol that ensures sender can’t outpace receiver. The receiver returns a dummy frame (ack) when ready. Only one frame is sent out at a time – this is called stop-and-wait. This gives us flow control. This is an intermediate protocol in which we’ve dealt with one issue (flow control) but not another (error control). It is instructive, but does not represent any real protocol (as it will deadlock if there are errors).

****

****

**ARQ**: (Automatic Repeat request) adds error control to Stop-and-Wait. The receiver acks frames that are correctly delivered. The sender sets timer and resends frame (if no ack.) ARQ is also called PAR (Positive Acknowledgement with Retransmission)

