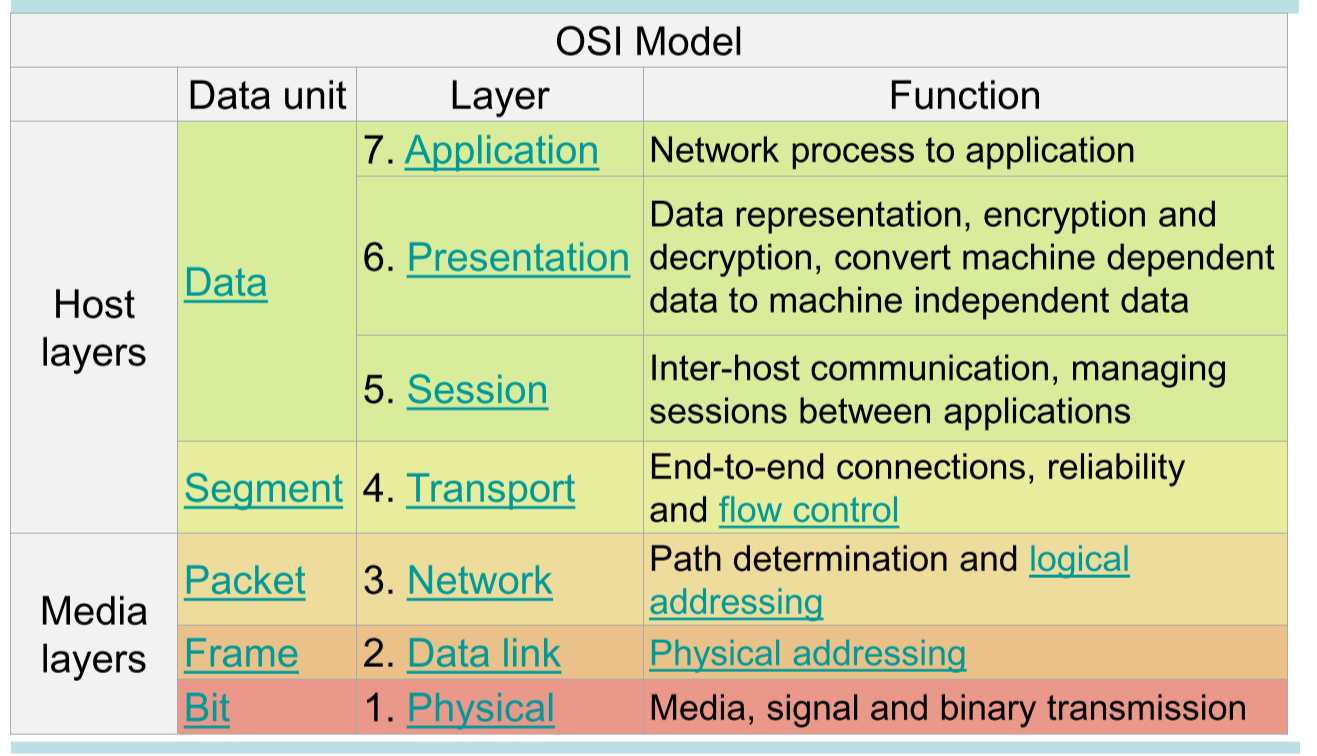
CE3005 Lecture Notes

# Network Layers and Physical Resilience

## Layered Network Architecture

* Network is organized as a stack of layers
  + Each layer offers services to the layer above, and processes information from the layer below, using an interface
  + Allows ease of design – other layers aren’t affected by changes to one layer
* **Protocol**: set of rules that govern communication between two peering parties (computers)
  + Format of messages
  + Order of messages
  + Actions that should be taken upon transmitting/receiving of messages
* **Network architecture**: set of layers and protocols that enable developers to build systems

## OSI (Open Systems Interconnection) 7-Layer Model



* Message starts off at application layer, moves down to physical layer
* Each layer adds a **header** to the message it receives – layer-specific information
* At the receiving end, headers are stripped as the message travels up the OSI layers

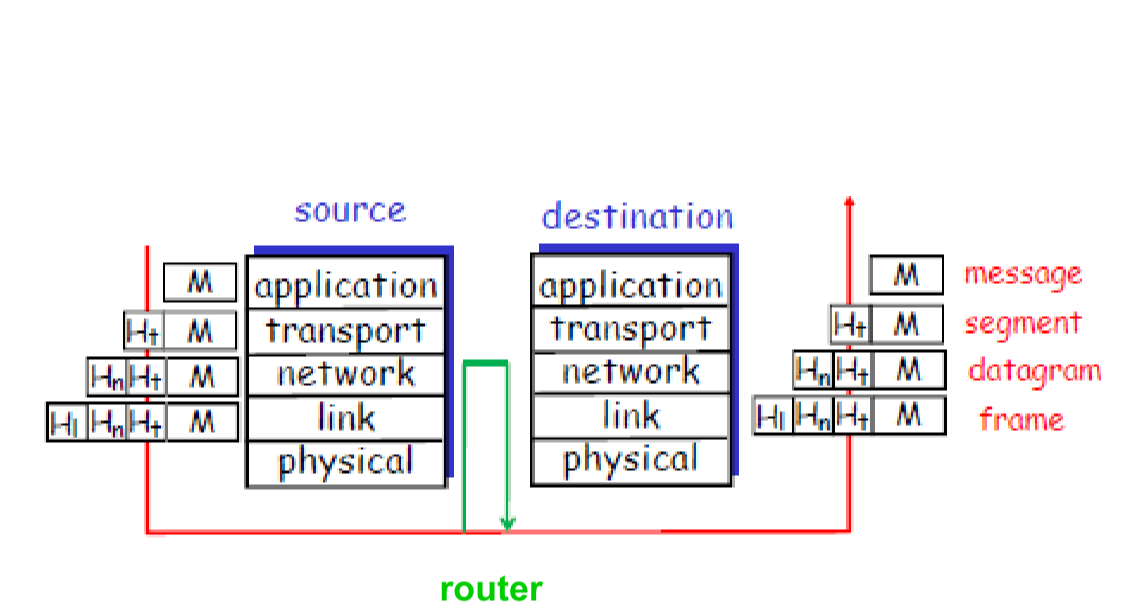
### Peer Processes

* Layer in one machine interacts with layer in another to provide services to layer
* **Peer processes** are the entities that contain the corresponding layers on different machines
  + They utilize the **layer-n protocol**
  + They communicate through exchanging **Protocol Data Units (PDUs)**

### Services

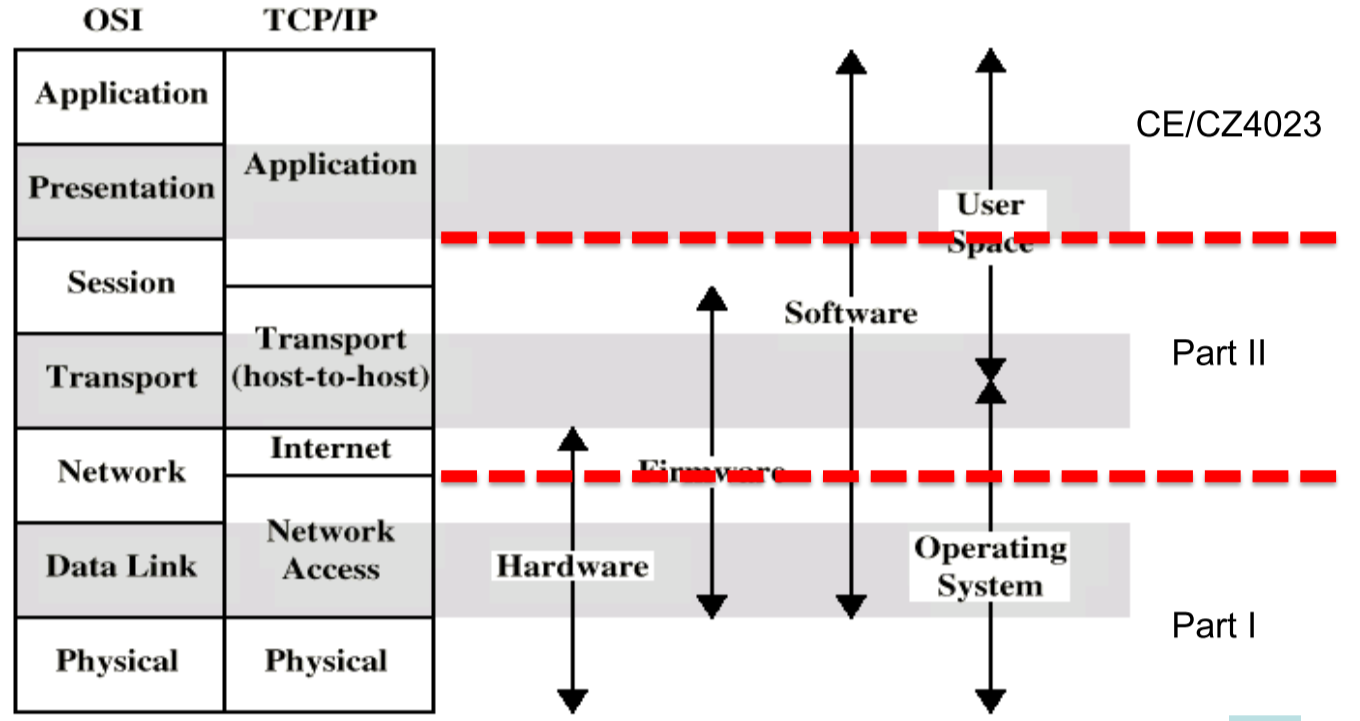
* Layer transfers information by invoking services on layer , which are available from **Service Access Points (SAPs)**
  + Each layer passes data to the layer below, until the physical layer where transfer occurs
  + Data is passed between layers as **Service Data Units (SDUs)**, which encapsulate PDUs

## TCP/IP Model



* **Application**: supporting network applications (FTP, SMTP, HTTP)
* **Transport**: host-to-host data transfer (TCP, UDP)
* **Network**: routing of data from source to destination (IP, routing protocols)
* **Link**: data transfer between neighbouring network elements (PPP, Ethernet)
* **Physical**: bits on the wire
* Each layer takes data from the layer above and adds a header before passing data to layer below

## OSI vs. TCP/IP



# Physical Layer: Network Resilience

## Network Reliability

* The probability that a network performs satisfactorily over a period of time
  + MTBF = mean time between failures
  + MTTF = mean time to failure
  + MTTR = mean time to repair

## Link Failure Probability & Availability

* **Link failure probability** : percentage of time where link does not work
* **Link availability** : percentage of time that link works

## Network Resilience

* Measures how tolerant a network is to faults, expressed based on probability that the network remains connected

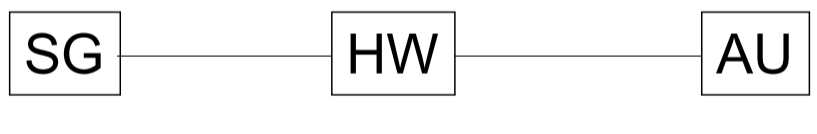
### Single Link



If the probability of a break is 0.05, then:

### Series Links

For links in series, calculate the probability that **all links are working**



If the probability of a break is 0.05, what is the probability that SG cannot communicate with AU?

* P(SG cannot communicate with AU) = 1 – P(SG can communicate with AU)

### Parallel Links

For links in parallel, calculate the probability that **all links are broken**

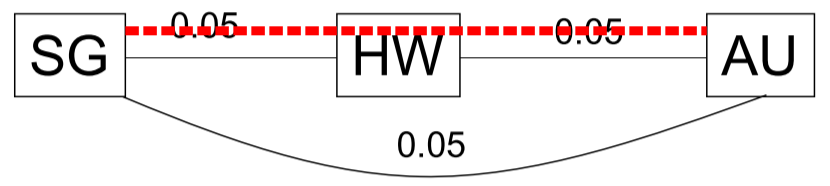


If the probability of a break is 0.05, what is the probability that SG cannot communicate with AU?

* P(SG cannot communicate with AU) = P(both links break)

### Hybrid Paths

For hybrid paths, decompose into paths and multiply the probabilities



If the probability of a break is 0.05, what is the probability that SG cannot communicate with AU?

* P(SG cannot communicate with AU) = P(both paths break)

# Data Link Layer (DLL)

## Data Link Layer Services

* **Framing**: encapsulate (network-layer) datagram in a link-layer frame before transmission over link
* **Link Access**: MAC protocol specifying rules by which frames are transmitted onto link
* **Flow Control**: control data flow to ensure all data is received properly
* **Reliable Delivery**: move datagram over link without errors

## Framing

### Byte/Character-Oriented

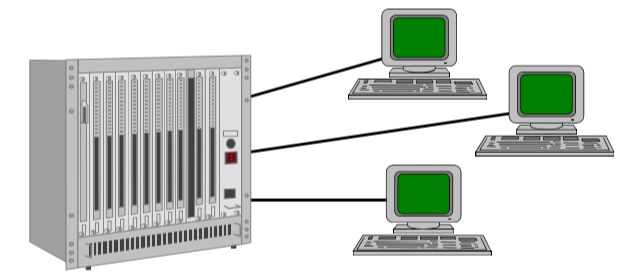
* Information is framed into a fixed 8-bit unit
  + 1 byte = 1 ASCII code
  + Some bytes used for signalling/protocol control
* Useful when technology was primitive

### Bit-Oriented

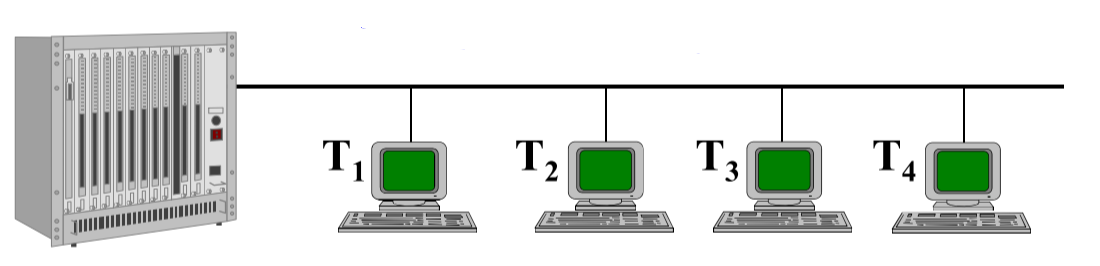
* **Flag** bits are used to frame the data bits
* Header/trailer used to describe frame
  + Some frames used for control
* Used by modern protocols (HDLC, PPP, Ethernet, etc.)

## Link Access

* Determines who gets to transmit when on a link
* Topology: physical arrangement of stations
  + **Point-to-Point**: hosts are directly connected in pairs – multiple direct transmissions



* + **Broadcast**: all hosts connected to single channel – everyone receives all messages

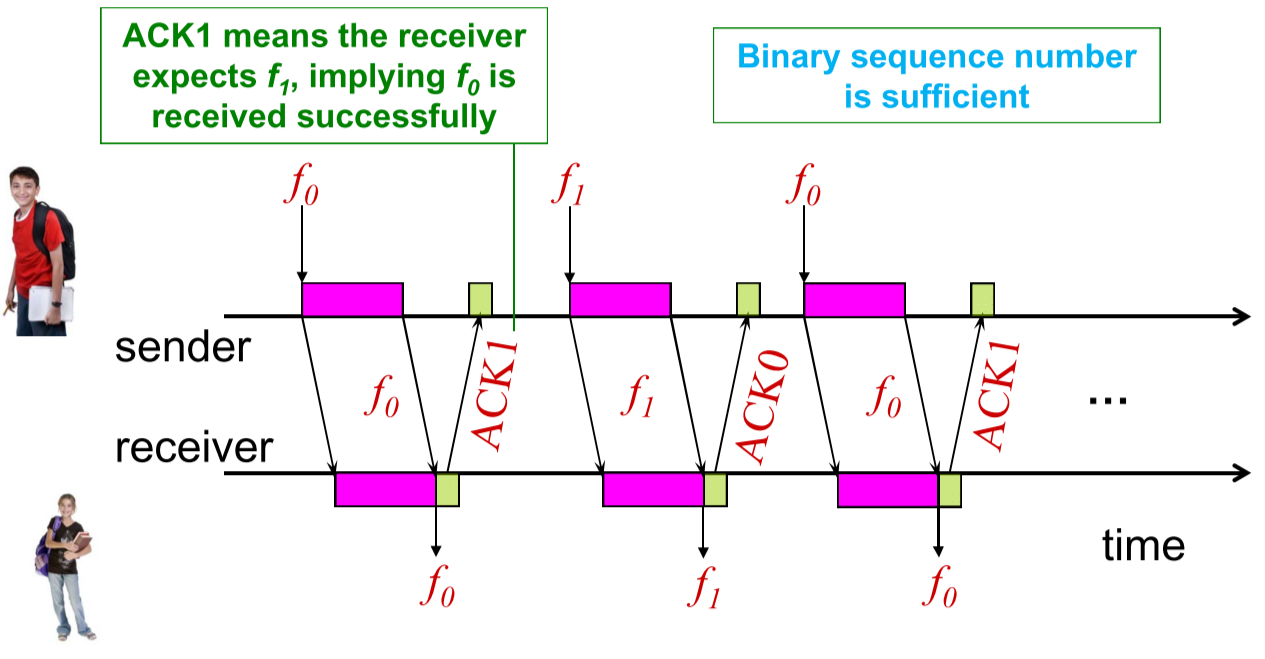


* Duplexity
  + **Half Duplex**: only one party can transmit at a time
  + **Full Duplex**: allows simultaneous transmission and reception between two parties

# Flow Control

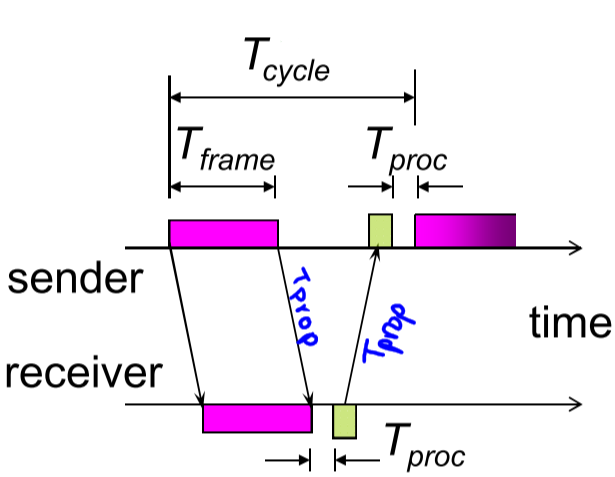
* Ensures that the transmitting party doesn’t overwhelm the receiving party with data

## Stop-and-Wait



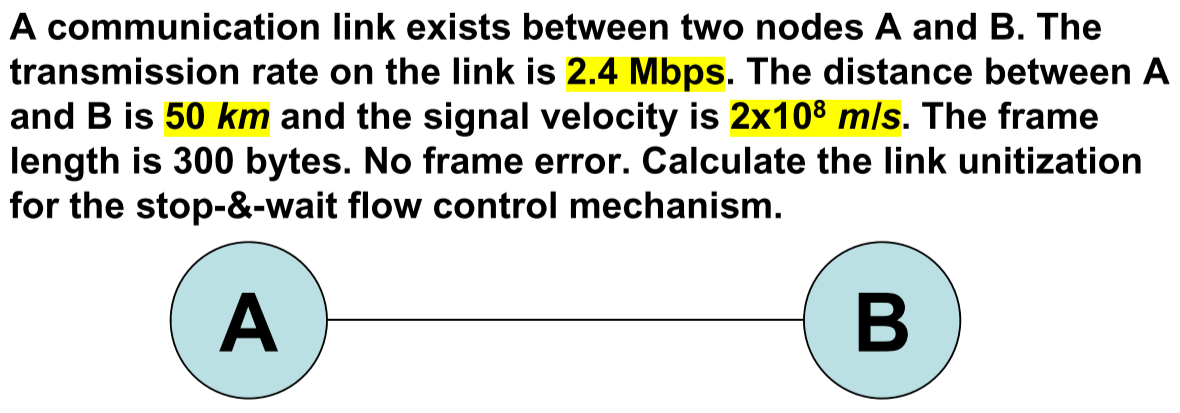
1. A puts data in a frame, sends frame to B
2. A waits for ACK from B
3. B receives data, sends ACK
4. A receives ACK, repeat

### Performance



* If we assume that and are negligible, then:  
  + : **Normalized Propagation Delay** (unitless)

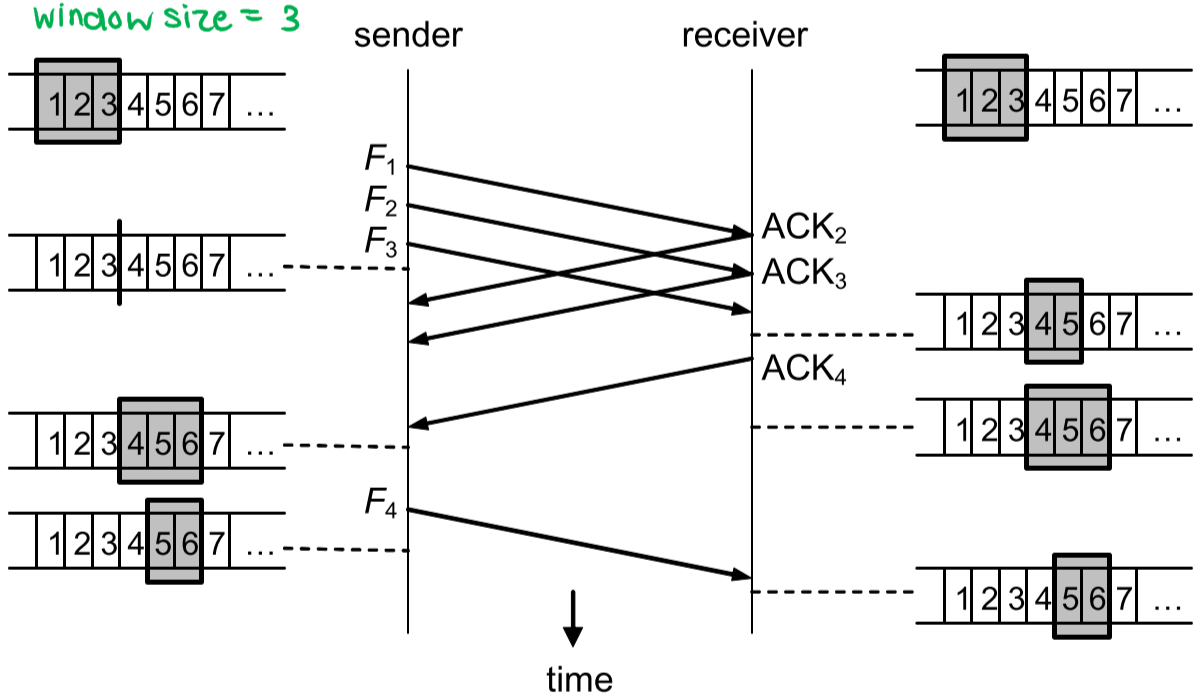
Example:



### Disadvantages

* If frame or ACK is lost (error occurs), long wait time expected
  + Can use timeout, but this is not prime
* If propagation time is long, sender must wait long time to transmit next data
  + Can use buffer – sliding window

## Sliding Window



* Allows multiple frames to be in transit at the same time
* Sender and receiver each have buffer length N
  + Sender can send **maximum N frames** without receiving ACKs
  + Each frame is numbered - ACK indicates number of next expected frame
* Sender:
  + Moves lower bound of buffer when frame sent
  + Moves upper bound of buffer when ACK received
* Receiver:
  + Moves lower bound of buffer when frame received
  + Moves upper bound of buffer when ACK sent
* If frame number bounded by bits, then frames are numbered modulo ( to )
  + Window size N must be chosen such that
* Other features:
  + Receiver can acknowledge frame and stop further transmission by sending **RNR (Receiver Not Ready)** frame, and sending a normal ACK when transmission can continue
  + ACK can be piggy-backed on data frames in the reverse direction

### Performance

* Sliding window performance depends on:
  + Window size
* Two cases:
  + : continuous transmission is possible, so
  + : the window is filled at , and transmission cannot resume until , so