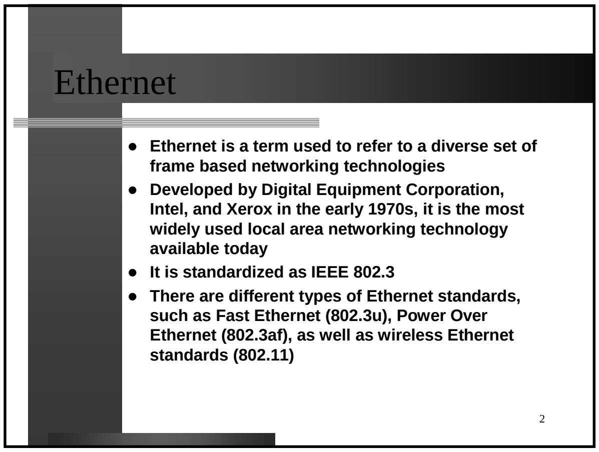
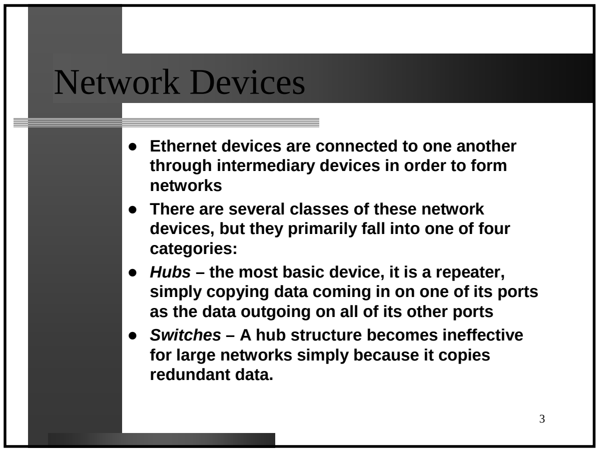
Ethernet

Babak Kia
Adjunct Professor
Boston University
College of Engineering



Ethernet

- Ethernet is a term used to refer to a diverse set of frame based networking technologies
- Developed by Digital Equipment Corporation, Intel, and Xerox in the early 1970s, it is the most widely used local area networking technology available today
- It is standardized as IEEE 802.3
- There are different types of Ethernet standards, such as Fast Ethernet (802.3u), Power Over Ethernet (802.3af), as well as wireless Ethernet standards (802.11)



Network Devices

- Ethernet devices are connected to one another through intermediary devices in order to form networks
- There are several classes of these network devices, but they primarily fall into one of four categories:
- *Hubs* – the most basic device, it is a repeater, simply copying data coming in on one of its ports as the data outgoing on all of its other ports
- *Switches* – A hub structure becomes ineffective for large networks simply because it copies redundant data.

Network Devices

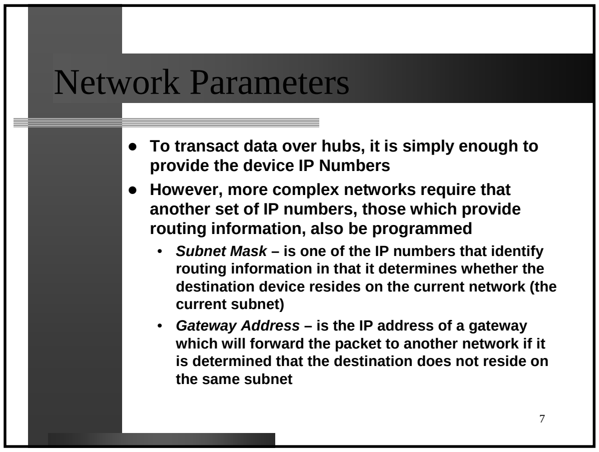
  - A switch is essentially a smart hub, only providing relevant data at a destination port
  - A switch can connect Ethernet, Token Ring, Fiber Channel, or other types of packet switched network protocols together to form a heterogeneous network
  - This has several complications, the primary one being that different standards use different MTU sizes
- *Routers* – At a higher level of complexity, the routers act as switches between networks
  - Routers are different from switches in that they connect networks together, whereas switches are used to connect devices together on a local LAN

Network Devices

- *Gateways* – At even a higher level of complexity are the gateways
  - They are used to interconnect networks at a higher level by mapping addresses from one network to another
  - They are also used to perform the required protocol conversions from one network to another
- IP works irrespective of the existence of these devices

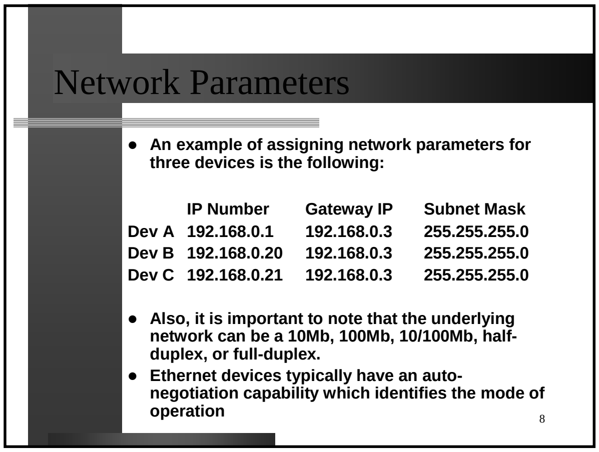
Network Parameters

- A network becomes operational not after physical connections have been established, but after appropriate network parameters have been assigned to all devices
- The most important network parameter for Ethernet networks is the *Internet Protocol*, or the IP number of the device
- IP is a unique string consisting of four numbers in the following format 123.123.123.123, where each group of numbers ranges from 0 to 255
- Every computer on the network must have a unique IP number assigned to it



Network Parameters

- To transact data over hubs, it is simply enough to provide the device IP Numbers
- However, more complex networks require that another set of IP numbers, those which provide routing information, also be programmed
  - *Subnet Mask* – is one of the IP numbers that identify routing information in that it determines whether the destination device resides on the current network (the current subnet)
  - *Gateway Address* – is the IP address of a gateway which will forward the packet to another network if it is determined that the destination does not reside on the same subnet

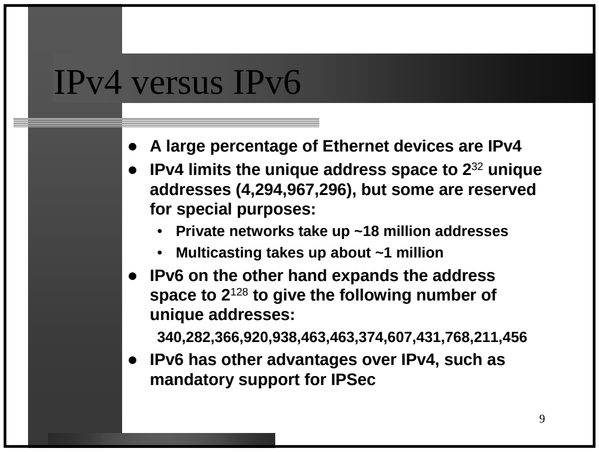


Network Parameters

- An example of assigning network parameters for three devices is the following:

	IP Number	Gateway IP	Subnet Mask
Dev A	192.168.0.1	192.168.0.3	255.255.255.0
Dev B	192.168.0.20	192.168.0.3	255.255.255.0
Dev C	192.168.0.21	192.168.0.3	255.255.255.0

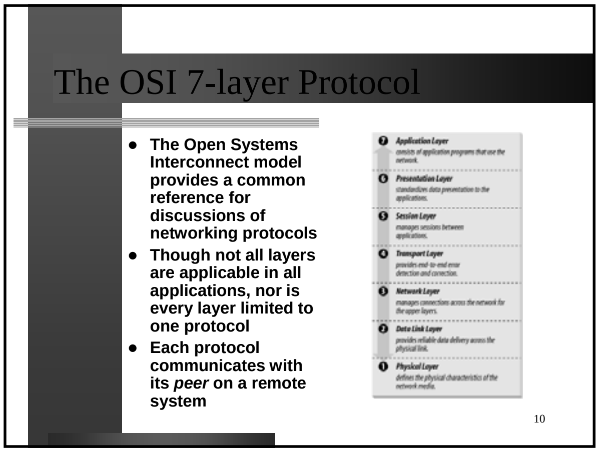
- Also, it is important to note that the underlying network can be a 10Mb, 100Mb, 10/100Mb, half-duplex, or full-duplex.
- Ethernet devices typically have an auto-negotiation capability which identifies the mode of operation



IPv4 versus IPv6

- A large percentage of Ethernet devices are IPv4
- IPv4 limits the unique address space to 2^{32} unique addresses (4,294,967,296), but some are reserved for special purposes:
  - Private networks take up ~18 million addresses
  - Multicasting takes up about ~1 million
- IPv6 on the other hand expands the address space to 2^{128} to give the following number of unique addresses:

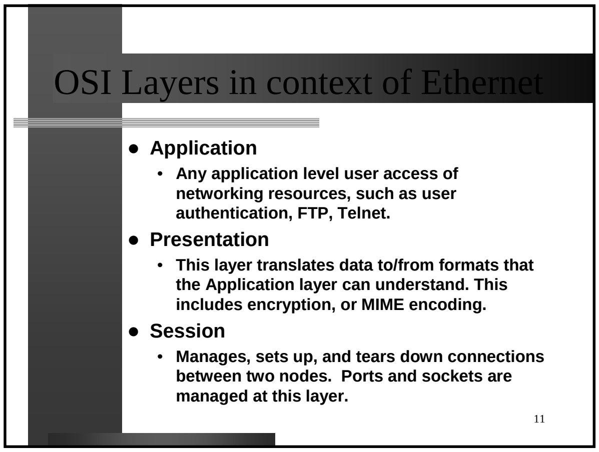
  340,282,366,920,938,463,463,374,607,431,768,211,456
- IPv6 has other advantages over IPv4, such as mandatory support for IPSec



The OSI 7-layer Protocol

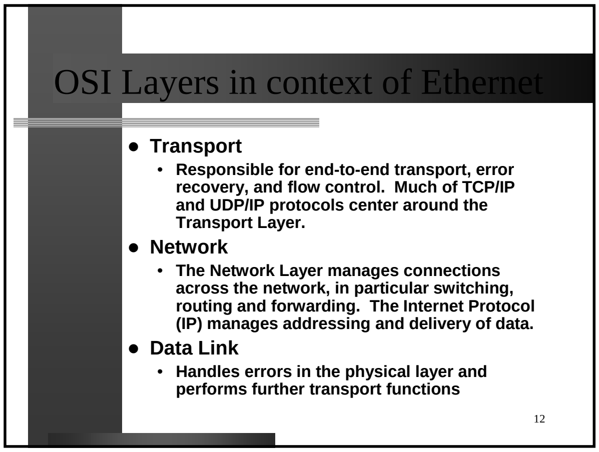
- The Open Systems Interconnect model provides a common reference for discussions of networking protocols
- Though not all layers are applicable in all applications, nor is every layer limited to one protocol
- Each protocol communicates with its *peer* on a remote system

7 Application Layer
consists of application programs that use the network.
6 Presentation Layer
standardizes data presentation to the applications.
5 Session Layer
manages sessions between applications.
4 Transport Layer
provides end-to-end error detection and correction.
3 Network Layer
manages connections across the network for the upper layers.
2 Data Link Layer
provides reliable data delivery across the physical link.
1 Physical Layer
defines the physical characteristics of the network media.



OSI Layers in context of Ethernet

- Application
  - Any application level user access of networking resources, such as user authentication, FTP, Telnet.
- Presentation
  - This layer translates data to/from formats that the Application layer can understand. This includes encryption, or MIME encoding.
- Session
  - Manages, sets up, and tears down connections between two nodes. Ports and sockets are managed at this layer.



OSI Layers in context of Ethernet

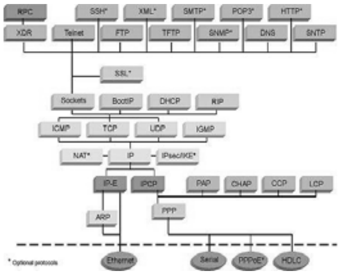
- Transport
  - Responsible for end-to-end transport, error recovery, and flow control. Much of TCP/IP and UDP/IP protocols center around the Transport Layer.
- Network
  - The Network Layer manages connections across the network, in particular switching, routing and forwarding. The Internet Protocol (IP) manages addressing and delivery of data.
- Data Link
  - Handles errors in the physical layer and performs further transport functions

OSI Layers in context of Ethernet

- **Physical**
  - **Performs the physical function of transmitting the bits (light, radio, electrical pulse) over a network layer**
- **Ethernet uses 4 of the 7 OSI layers:**
  - **Application (Telnet, FTP)**
  - **Transport (TCP/UDP)**
  - **Network (IP, ICMP, ARP)**
  - **Network Interface and Hardware (Ethernet, Token Ring, FDDI, X.25, ATM, Wireless, …)**

13

Networking Protocols



14

Address Resolution Protocol (ARP)

- **Address Resolution Protocol, defined by RFC 826 maps a network IP address into an Ethernet MAC address**
- **On a physical network, individual devices are identified using their network addresses, known as the *Media Access Control* (MAC) address, and not an IP addresses which is handled at a higher protocol level**
- **While the IP address of machines can often change, specially if they're moved between different networks, the MAC addresses are unique to devices and are permanently assigned to the hardware**

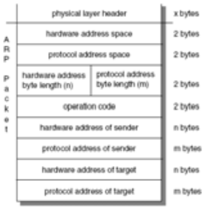
15

Address Resolution Protocol (ARP)

- **ARP works in the following fashion:**
  - **When a device A addresses device B, whose IP number is not in A’s ARP cache, it needs to find out the address and therefore broadcasts an ARP request**
  - **Since all devices on the network accept broadcast packets, when device B recognizes its own address it replies to device A with an ARP reply including its MAC address**
  - **If the ARP packet was transferred through a router or a gateway, routing information is also returned in the ARP reply packet**

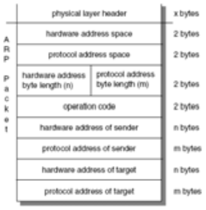
Address Resolution Protocol (ARP)

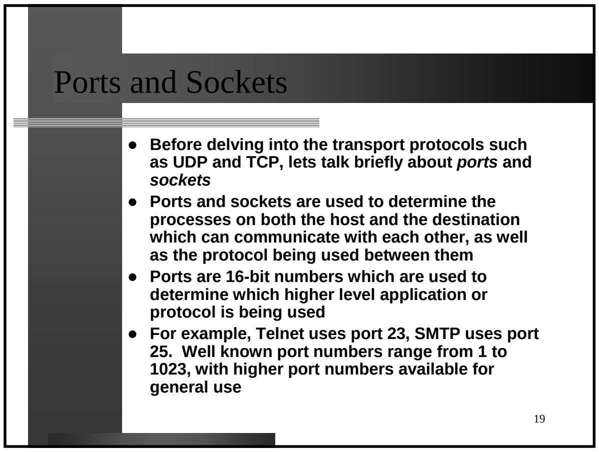
- ***Hardware Address Space*: specifies the type of hardware (Ethernet, or Packet Radio network)**
- ***Protocol Address Space*: Specifies the type of protocol**
- ***Hardware Address Length*: For Ethernet it is 6**
- ***Protocol Address Length*: For IP it is 4**
- ***Operation Code*: Request or Reply**


physical layer header
x bytes
A
R
P
P
a
c
k
e
t
hardware address space
2 bytes
protocol address space
2 bytes
hardware address byte length (n)
protocol address byte length (m)
2 bytes
operation code
2 bytes
hardware address of sender
n bytes
protocol address of sender
m bytes
hardware address of target
n bytes
protocol address of target
m bytes

Address Resolution Protocol (ARP)

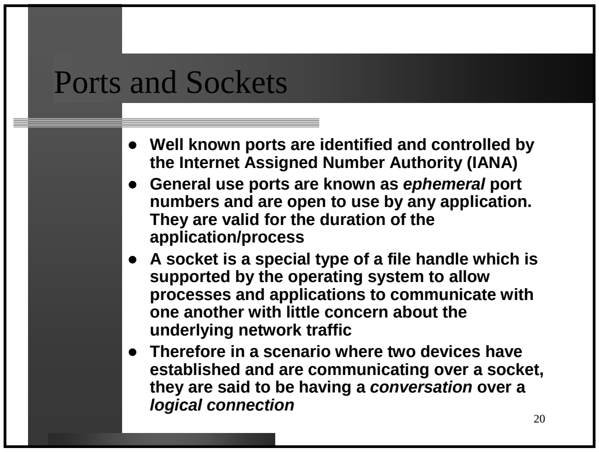
- ***Source/Target Hardware Address*: Contains the physical network hardware address**
- ***Source/Target Protocol Address*: Contains the protocol address – for TCP/IP this is the 32-bit IP address**


physical layer header
x bytes
A
R
P
P
a
c
k
e
t
hardware address space
2 bytes
protocol address space
2 bytes
hardware address byte length (n)
protocol address byte length (m)
2 bytes
operation code
2 bytes
hardware address of sender
n bytes
protocol address of sender
m bytes
hardware address of target
n bytes
protocol address of target
m bytes



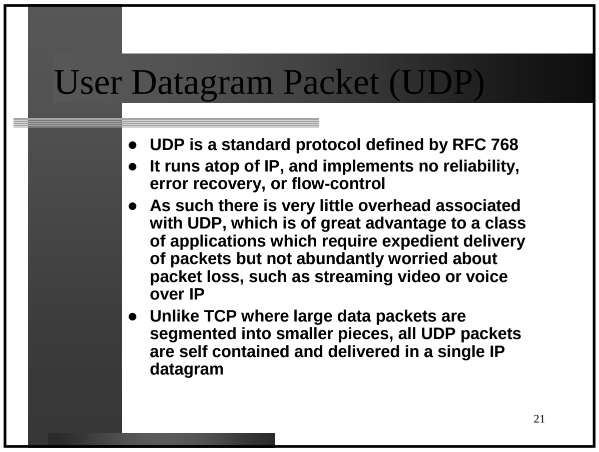
Ports and Sockets

- Before delving into the transport protocols such as UDP and TCP, lets talk briefly about *ports* and *sockets*
- Ports and sockets are used to determine the processes on both the host and the destination which can communicate with each other, as well as the protocol being used between them
- Ports are 16-bit numbers which are used to determine which higher level application or protocol is being used
- For example, Telnet uses port 23, SMTP uses port 25. Well known port numbers range from 1 to 1023, with higher port numbers available for general use



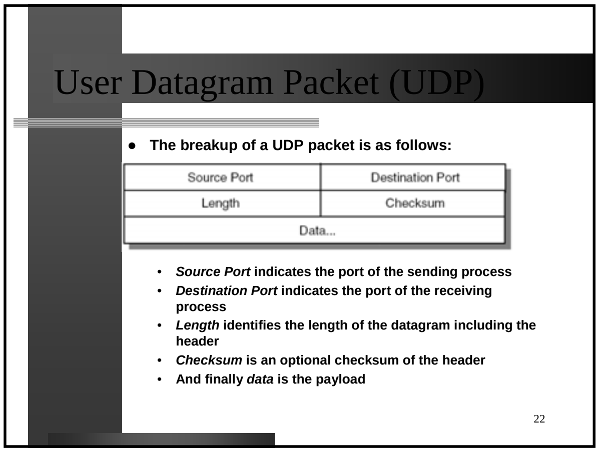
Ports and Sockets

- Well known ports are identified and controlled by the Internet Assigned Number Authority (IANA)
- General use ports are known as *ephemeral* port numbers and are open to use by any application. They are valid for the duration of the application/process
- A socket is a special type of a file handle which is supported by the operating system to allow processes and applications to communicate with one another with little concern about the underlying network traffic
- Therefore in a scenario where two devices have established and are communicating over a socket, they are said to be having a *conversation* over a *logical connection*



User Datagram Packet (UDP)

- UDP is a standard protocol defined by RFC 768
- It runs atop of IP, and implements no reliability, error recovery, or flow-control
- As such there is very little overhead associated with UDP, which is of great advantage to a class of applications which require expedient delivery of packets but not abundantly worried about packet loss, such as streaming video or voice over IP
- Unlike TCP where large data packets are segmented into smaller pieces, all UDP packets are self contained and delivered in a single IP datagram

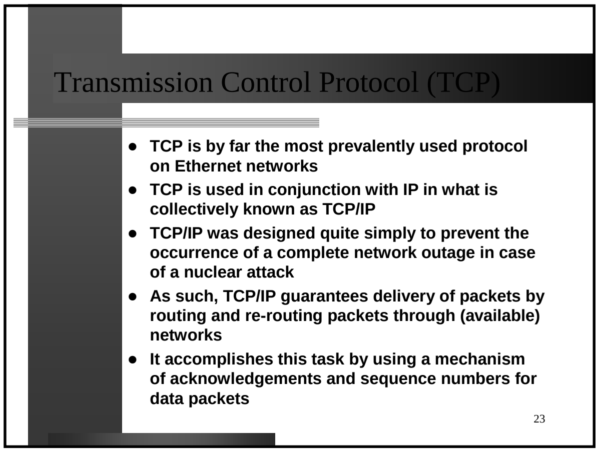


User Datagram Packet (UDP)

- The breakup of a UDP packet is as follows:

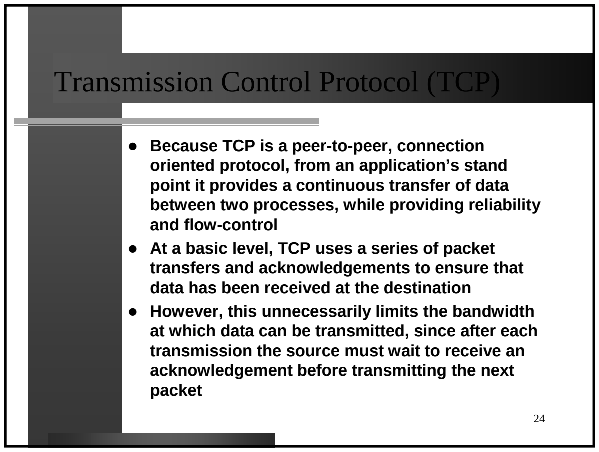
Source Port	Destination Port
Length	Checksum
Data...	

- *Source Port* indicates the port of the sending process
- *Destination Port* indicates the port of the receiving process
- *Length* identifies the length of the datagram including the header
- *Checksum* is an optional checksum of the header
- And finally *data* is the payload



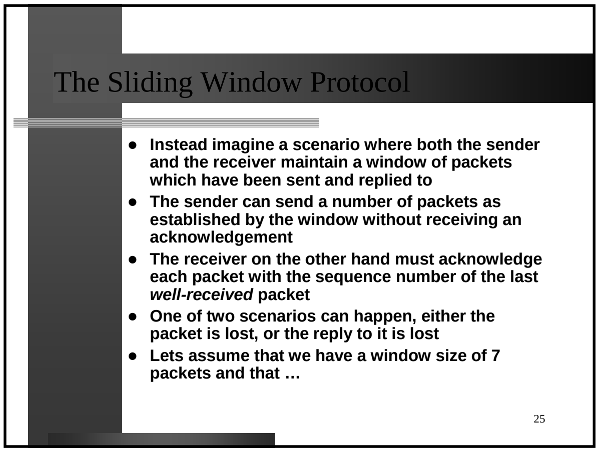
Transmission Control Protocol (TCP)

- TCP is by far the most prevalently used protocol on Ethernet networks
- TCP is used in conjunction with IP in what is collectively known as TCP/IP
- TCP/IP was designed quite simply to prevent the occurrence of a complete network outage in case of a nuclear attack
- As such, TCP/IP guarantees delivery of packets by routing and re-routing packets through (available) networks
- It accomplishes this task by using a mechanism of acknowledgements and sequence numbers for data packets



Transmission Control Protocol (TCP)

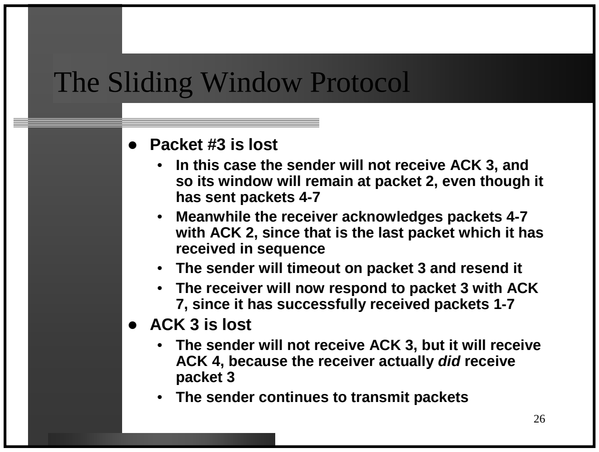
- Because TCP is a peer-to-peer, connection oriented protocol, from an application's stand point it provides a continuous transfer of data between two processes, while providing reliability and flow-control
- At a basic level, TCP uses a series of packet transfers and acknowledgements to ensure that data has been received at the destination
- However, this unnecessarily limits the bandwidth at which data can be transmitted, since after each transmission the source must wait to receive an acknowledgement before transmitting the next packet



The Sliding Window Protocol

- Instead imagine a scenario where both the sender and the receiver maintain a window of packets which have been sent and replied to
- The sender can send a number of packets as established by the window without receiving an acknowledgement
- The receiver on the other hand must acknowledge each packet with the sequence number of the last *well-received* packet
- One of two scenarios can happen, either the packet is lost, or the reply to it is lost
- Lets assume that we have a window size of 7 packets and that …

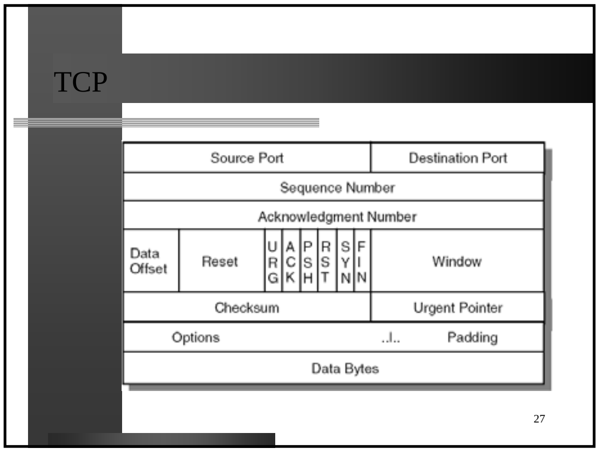
25



The Sliding Window Protocol

- Packet #3 is lost
  - In this case the sender will not receive ACK 3, and so its window will remain at packet 2, even though it has sent packets 4-7
  - Meanwhile the receiver acknowledges packets 4-7 with ACK 2, since that is the last packet which it has received in sequence
  - The sender will timeout on packet 3 and resend it
  - The receiver will now respond to packet 3 with ACK 7, since it has successfully received packets 1-7
- ACK 3 is lost
  - The sender will not receive ACK 3, but it will receive ACK 4, because the receiver actually *did* receive packet 3
  - The sender continues to transmit packets

26



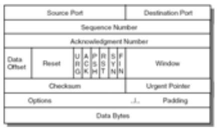
TCP

Source Port								Destination Port
Sequence Number								
Acknowledgment Number								
Data Offset	Reset	U R G	A C K	P S H	R S T	S Y N	F I N	Window
Checksum								Urgent Pointer
Options						..I..		Padding
Data Bytes								

27

TCP

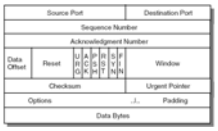
- ***Source Port*: 16-bit source port number**
- ***Destination Port*: 16-bit destination port number**
- ***Sequence Number*: Sequence number of the first data byte in this segment**
- ***ACK Number*: The value of the next sequence number which the receiver is expecting to receive**
- ***Data offset*: Number of 32-bit words in header**



28

TCP

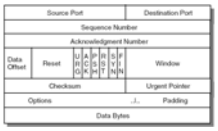
- **Reserved: Reserved for future use, must be zero**
- **URG: Indicates Urgent Pointer field is valid**
- **ACK: Indicates ACK field is valid**
- **PSH: Push function**
- **RST: Reset connection**
- **SYN: Synchronizes the sequence numbers**
- **FIN: No more data from the sender**



29

TCP

- **Window: The number of bytes the receiver can accept**
- **Checksum**



30

Internet Protocol (IP)

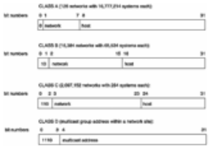
- The Internet Protocol (IP) is the most basic building block of all Internet communication
- It hides the underlying network by providing the higher level applications with a virtual network viewpoint
- In its current form, the IP protocol version 4 (IPv4) as defined in RFC 791
- IP is a *connectionless* protocol, which means that once a packet (or a datagram) has been sent out, it can take any route to reach its destination
- IP is also an *unreliable* protocol, because it contains no mechanism to guarantee deliveries of individual datagrams

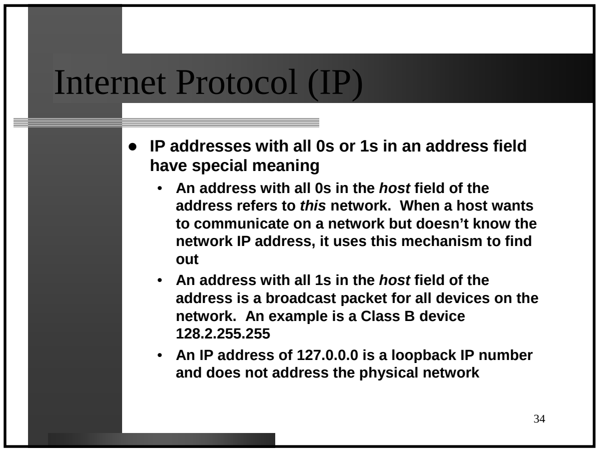
Internet Protocol (IP)

- Since IP is unreliable, it relies on higher level protocols to address the unreliability issues
- IP addresses are represented by 32 bit unsigned numbers, each byte representing one order of hierarchy
- For example, 123.1.2.3 is a valid network IP number (address)
- The mapping between an IP number and a more human-friendly address such as www.apple.com is done by the Domain Name System (DNS)

Internet Protocol (IP)

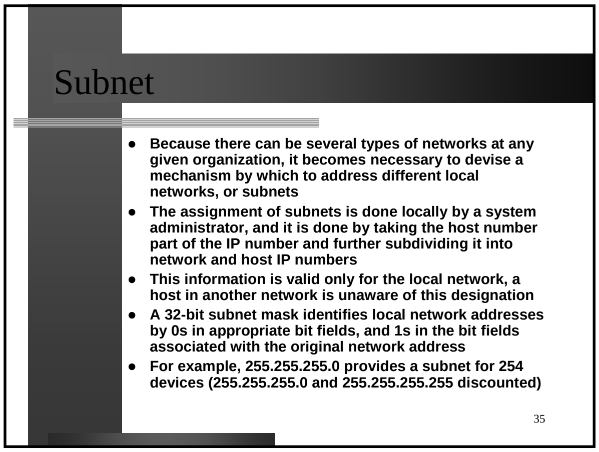
- Class A
  - 24-bit host, 7-bit network ID
- Class B
  - 16-bit host, 14-bit network ID
- Class C
  - 8-bit host, 21-bit network ID
- Class D
  - Reserved for multicasting
- Class E
  - Reserved for future use


CLASS A (126 networks with 16,777,214 systems each)
bit numbers
0 1 7 8 31
0 network host
CLASS B (16,384 networks with 65,534 systems each)
bit numbers
0 1 2 15 16 31
10 network host
CLASS C (2,097,152 networks with 254 systems each)
bit numbers
0 2 3 23 24 31
110 network host
CLASS D (multicast group address within a network site)
bit numbers
0 3 4 31
1110 multicast address



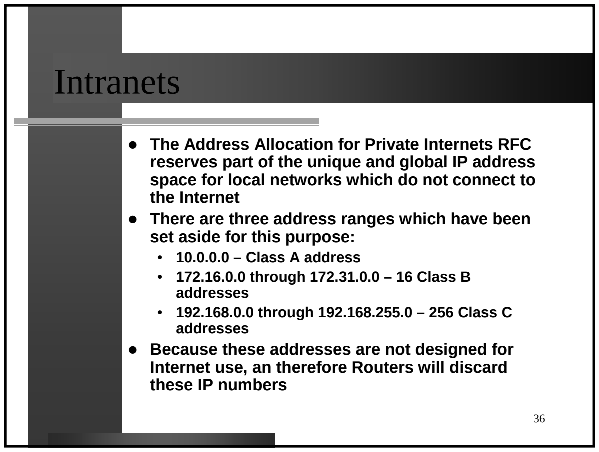
Internet Protocol (IP)

- IP addresses with all 0s or 1s in an address field have special meaning
  - An address with all 0s in the *host* field of the address refers to *this* network. When a host wants to communicate on a network but doesn't know the network IP address, it uses this mechanism to find out
  - An address with all 1s in the *host* field of the address is a broadcast packet for all devices on the network. An example is a Class B device 128.2.255.255
  - An IP address of 127.0.0.0 is a loopback IP number and does not address the physical network



Subnet

- Because there can be several types of networks at any given organization, it becomes necessary to devise a mechanism by which to address different local networks, or subnets
- The assignment of subnets is done locally by a system administrator, and it is done by taking the host number part of the IP number and further subdividing it into network and host IP numbers
- This information is valid only for the local network, a host in another network is unaware of this designation
- A 32-bit subnet mask identifies local network addresses by 0s in appropriate bit fields, and 1s in the bit fields associated with the original network address
- For example, 255.255.255.0 provides a subnet for 254 devices (255.255.255.0 and 255.255.255.255 discounted)

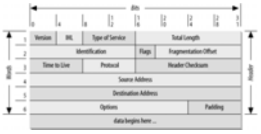


Intranets

- The Address Allocation for Private Internets RFC reserves part of the unique and global IP address space for local networks which do not connect to the Internet
- There are three address ranges which have been set aside for this purpose:
  - 10.0.0.0 – Class A address
  - 172.16.0.0 through 172.31.0.0 – 16 Class B addresses
  - 192.168.0.0 through 192.168.255.0 – 256 Class C addresses
- Because these addresses are not designed for Internet use, an therefore Routers will discard these IP numbers

IP Datagram

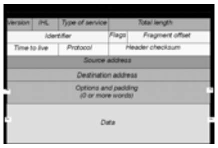
- The unit of transfer in an IP network is called the IP Datagram



- IP automatically breaks apart messages into smaller datagrams, forwards them on a network, and reassembles them at destination
- The maximum length of a datagram is 65,535 bytes, and the minimum size for an unfragmented datagram is 576 bytes

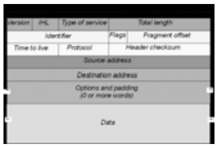
IP Datagram

- *Version*: The IP version number, current IP version number is 4, with version 6 also being in use
- *IHL*: Length of the IP header counted in 32-bit quantities, does not include the data size!
- *Service Type*: Indicates the quality of service requested for this IP datagram (i.e. Routine, priority, etc.)



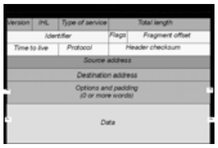
IP Datagram

- ***Total Length:*** Total length of the datagram, including header and data
- ***Identifier***: A unique number provided by sender to help in reassembly of packets
- ***Flags***: Control flags (do not fragment, additional fragments)



IP Datagram

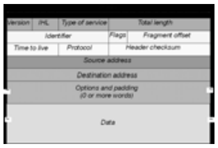
- ***Fragment Offset***: Aids in the reassembly of the full datagram
- ***Time to Live***: The time in seconds that the datagram is allowed to travel. Each router processing the packet decrements this value by one, once it reaches a value of 0 the packet is discarded



Version	IHL	Type of service	Total length	
Identifier			Flags	Fragment offset
Time to live		Protocol	Header checksum	
Source address				
Destination address				
Options and padding (0 or more words)				
Data				

IP Datagram

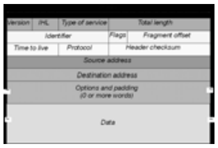
- ***Protocol:*** The higher level protocol which receives this packet (TCP, UDP, etc.)
- ***Header Checksum***: If the header checksum is not matched, the packet is discarded
- ***Source Address***: 32-bit address of the source (host)
- ***Destination Address***: 32-bit address of the destination



Version	IHL	Type of service	Total length	
Identifier			Flags	Fragment offset
Time to live		Protocol	Header checksum	
Source address				
Destination address				
Options and padding (0 or more words)				
Data				

IP Datagram

- ***Options***: Various options covering debugging and traffic measurement
- ***Padding***: If an option is used, the datagram is padded with all zeros up to the next 32-bit boundary
- ***Data***: They payload of the datagram, passed on to the higher protocols



Version	IHL	Type of service	Total length	
Identifier			Flags	Fragment offset
Time to live		Protocol	Header checksum	
Source address				
Destination address				
Options and padding (0 or more words)				
Data				