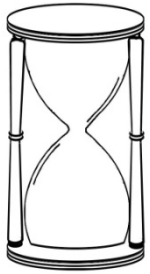
Lecture 5A

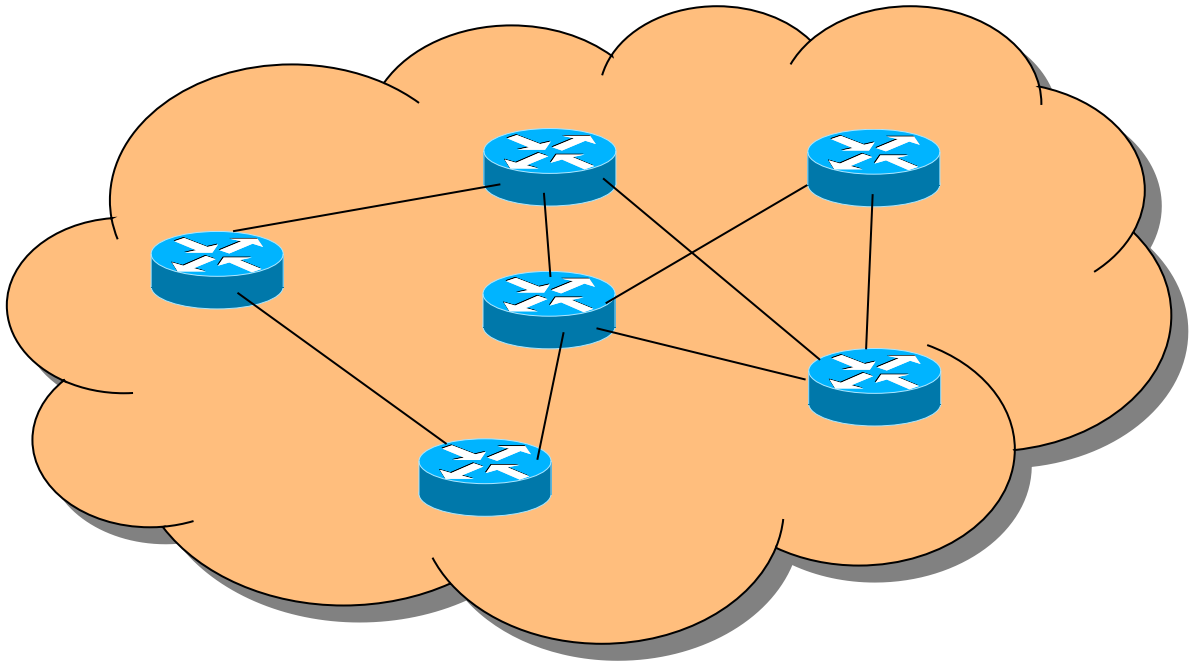
Network and Transport Layer Protocols and ARP/RARP

The success of TCP / IP
"Rough Consensus – Working Code"

- Make sure it works!
  - First try some test implementations before you finalize on a design
- Keep it simple.
  - Leave out unnecessary features
  - Occam's razor: explain the phenomena by the simplest hypothesis possible
- Make clear choices (choose one!), exploit modularity (protocol stack)
- Expect heterogeneity (keep the network layer simple) – the hourglass
- Avoid static options and parameters – negotiate!
- Look for a good design – it need not be perfect
- Be strict when sending packets (rigorously comply with standards), but be tolerant when receiving
- Think about scalability, performance and cost



THE NETWORK LAYER IN TCP/IP



The IP addressing scheme makes the internals of the cloud transparent

The Internet is composed of independent networks called Autonomous Systems (AS).

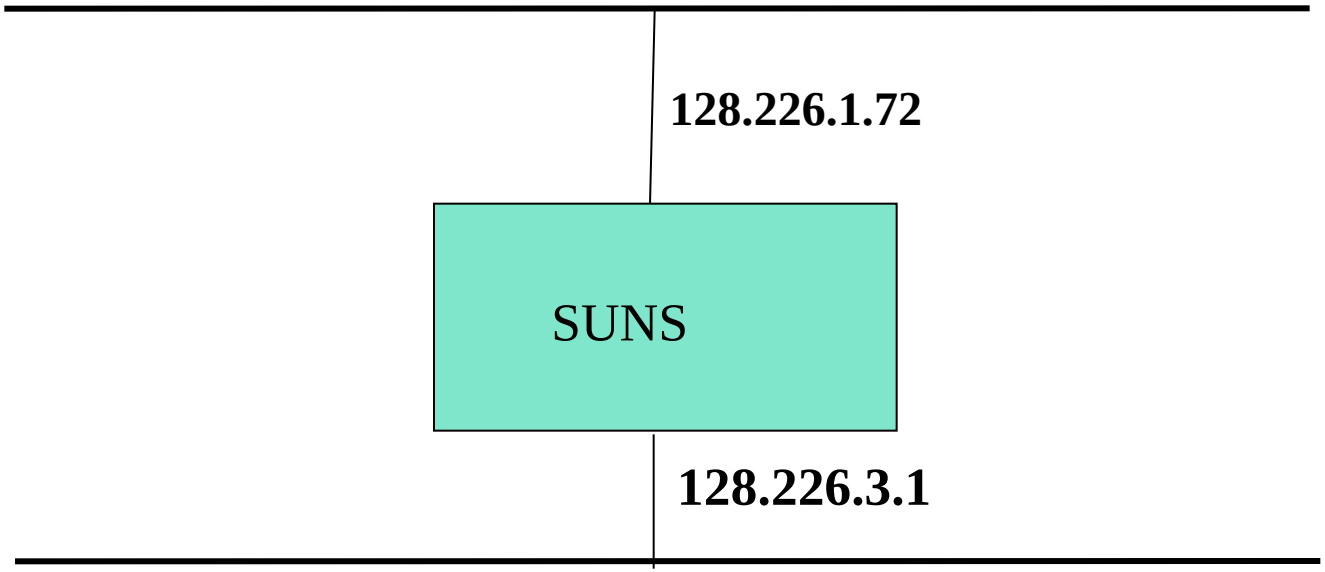
Autonomous System

- A set of networks managed by a single authority
- Defined routing policy within the set of networks
- Has a defined set of IP prefixes
- An AS network has a unique ASN (AS number) and is used by external routing protocols (BGP)
- Key Aspects:
  - Single routing policy
  - Unique ASN identification number
  - BGP primarily used between different AS's
  - Routing independence within the AS
  - Single Organization for Management

IP Addresses

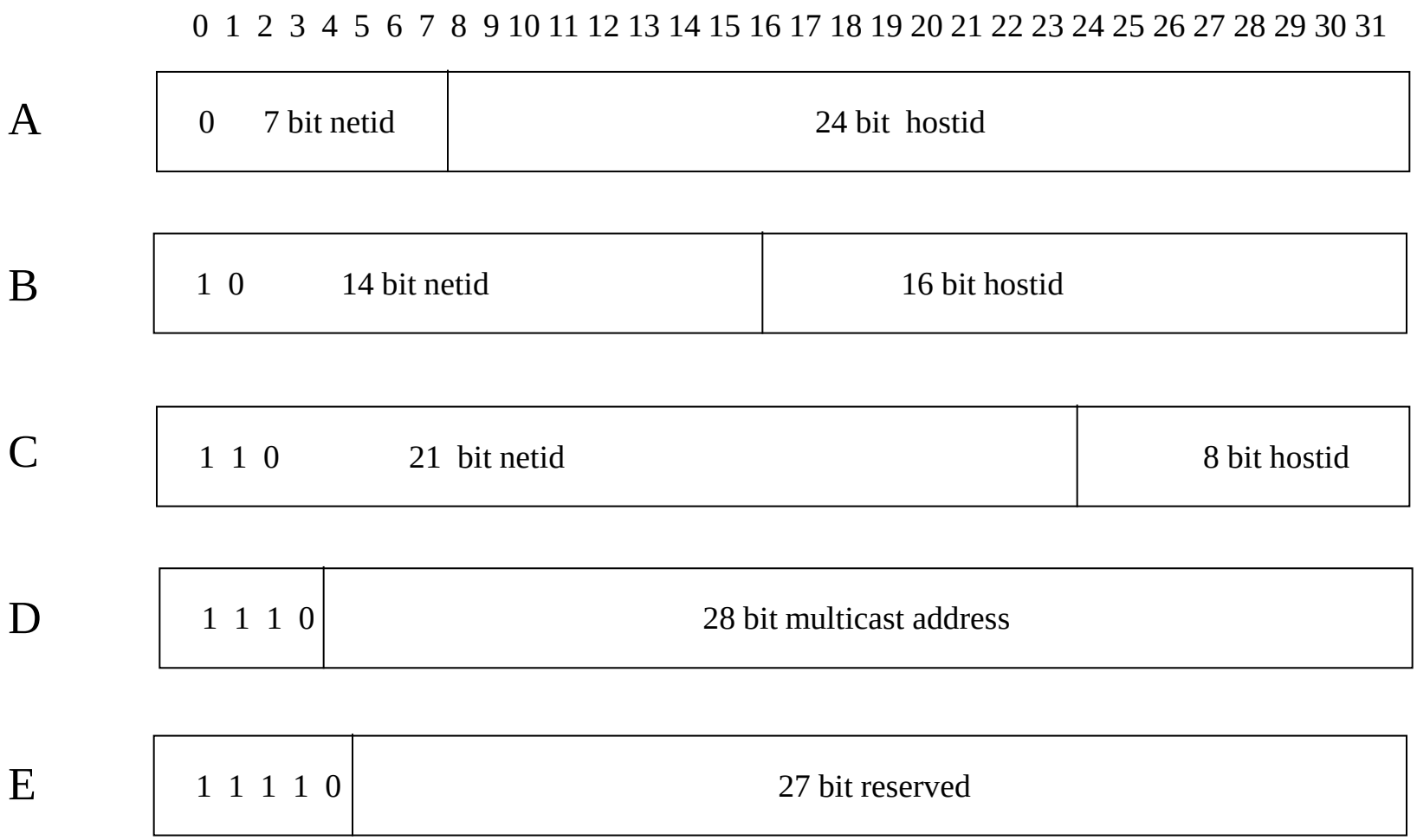
- The IP address is the network address of an adaptor connected to a host. Thus multi-homed hosts have multiple IP addresses
  - The address is a 32-bit value (4 bytes)
  - Dotted decimal notation separates the bytes and encodes each byte as decimal numbers: 192.5.48.1
  - Addresses originally (“classful”) had a two-level structure: netid and hostid to represent the network and the host respectively. The network id was used to route to the network at which point the host id was sufficient to get to the correct host.

Multi-homed host



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
1	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	128.226.3.1

Class Structure of IP Addresses



What does 224.0.0.17 indicate? What other encoding uses a similar idea to separate classes

Some Special Addresses

- Notation
  - 0 means all 0's in corresponding field
  - -1 means all 1's in corresponding field

• This host, this net	netid = 0	host id = 0
• This net	netid = 0	host id = id
• Loopback	netid =127	host id =x.y.z
• Limited broadcast	netid = -1	host id = -1
• Net directed broadcast	netid = id	host id = -1

Subnetting

- Even class B (let alone class A) networks were often “too large” for practical use
- 16-bit host id meant that over 64,000 hosts could be addressed but it is more likely that the network should be broken up into several smaller subnets such as CS, EE, etc.
- Solution: do away with the class structure and instead explicitly define the netid portion and the host portion
- Use a 32-bit network “subnet mask” to define the net plus subnet part (all 1’s in the bit positions)
- External routing routes using a netid to route to the appropriate AS. Within the AS, routing uses more precise netid subnet information (again through the mask)
- Subnetting (and supernetting, etc) is doing away with the original class hierarchy, thus the term classless routing.
- Routers now try to route to the lowest level subnet possible

Subnetting Example

- class B address: 128.226.0.0 at a university
- Let the 3rd digit be the subnetid (8 bits) and the 4th digit be the host id on the subnet (8 bits)
- Thus, subnets are of the form 128.226.subnetid.hostid
- Actually, the difference between the original netid and the subnet id is no longer relevant and the portion that is the net or subnet part is simply the netid or the *prefix*
- For example, a router on the research LAN subnet could have addresses 128.226.1.72 (on the campus wide backbone) and 128.226.3.1 (on the research subnet)
- Subnet mask is 255.255.255.0.
- In general a subnet mask can be an arbitrary subset of 1's in the 32-bit mask but it is generally a contiguous set of 1's from the first bit.

Classless Interdomain Routing (CIDR)

- A prefix number indicates the number of bits of the mask.
- Given 128.226.0.0 how do we indicate that the network prefix part is not 16 bits but actually 20 bits? We simply write it as 128.226.0.0/20
- This is the same as saying the network mask is: 11111111111111111111000000000000 or 255.255.240.0
- Note that the size of a block (host portion) is a power of 2. For example, with 12 bits available for the host portion we have $2^{12} = 4096$ host machines possible.
- We have discussed subnetting. Supernetting is aggregating addresses using the same idea “in reverse.”

Supernetting Example

- Block of 8192 addresses available from 194.24.0.0. Therefore the host portion available is 13 bits. Note that the fixed part of the netid starts with 19 bits: 11000010.00011000.000
- Suppose Cambridge U. is assigned 2048 addresses. Thus the CIDR address for this network has 11 bits for the host and 21 bits for the prefix and is 194.24.0.0/21. Note that the highest address is thus 194.24.00000111.11111111 or 194.24.7.255.
- Suppose Oxford U. next wants 4096 addresses from this range. It cannot be assigned from 194.24.8.0. *Why?* Because the host range must have 12 bits and thus the lowest address after the given addresses is 194.24.16.0/20
- Next Edinburgh asks for 1024 addresses and is given the address 194.24.8.0/22.
- Available is 194.24.12.0/22

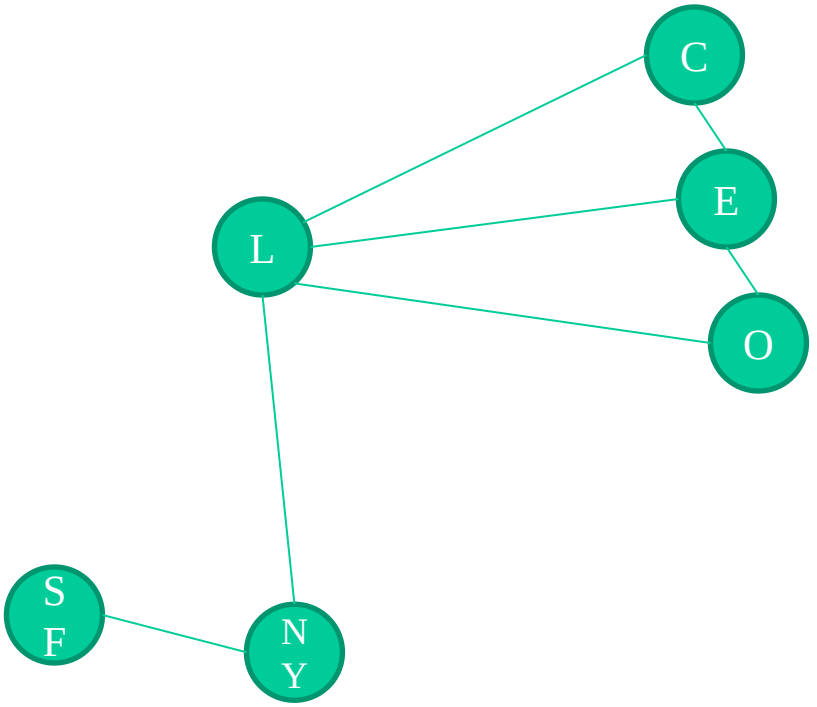
Supernetting Example

- Cambridge 194.24.0.0/21 194.24.0.0 -- 194.24.7.255
- Edinburgh 194.24.8.0/22 194.24.8.0 -- 194.24.11.255
- (Available) 194.24.12.0/22 194.24.12.0 -- 194.24.15.255
- Oxford 194.24.16.0/20 194.24.16.0 -- 194.24.31.255

- Suppose London can route to all of these network. How should London tell, for example, New York, what addresses it can do the routing for, to make it easiest for New York? Clearly, it can route for the original 8192 block which is 194.24.0.0/19
- This is call supernetting or combining prefixes in an aggregated entity.

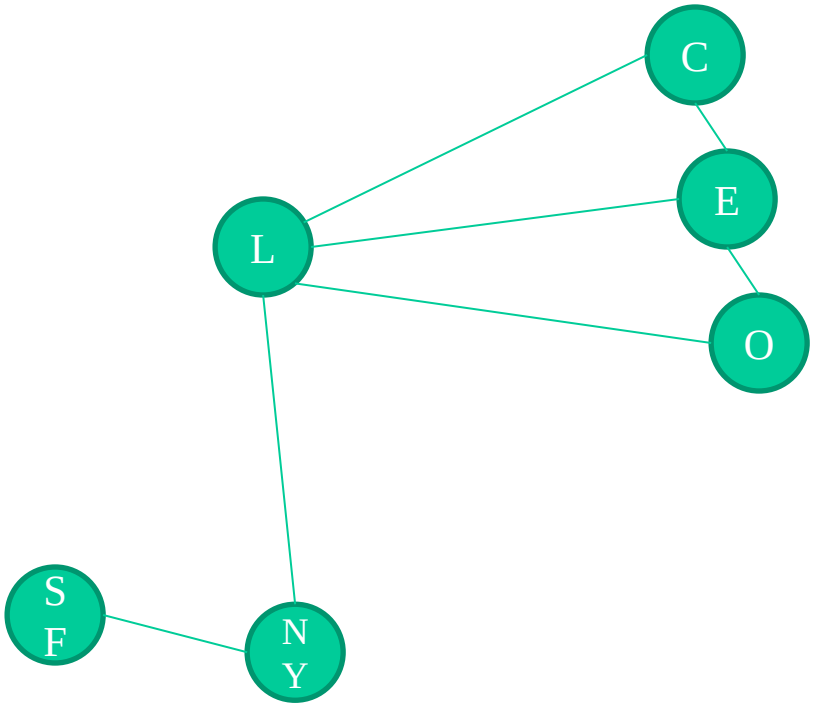
CIDR – Aggregating the prefixes

- From 194.24.0.0 to 194.24.31.255, all to London.
- Aggregate the three entries into one 194.24.0.0/19



CIDR – Using longest matching prefix

- Suppose block 194.24.12.0/22 assigned to San Francisco
- NY Router has entry 194.24.0.0/19
- It now also has entry 194.24.12.0/22
- When routing to a network, we always use the longest match.
- When a packet arrives with address 194.24.15.8, the router checks the routing table and there will be two matches: 194.24.12.0/22 and 194.24.0.0/19. Pick the longest match.



The IP Protocol

- Very simple protocol: basically sends a datagram for best effort delivery.
- Maximum transmission unit (MTU) on a link might require an IP packet to be fragmented to fit in to a frame.
  - IP is responsible for fragmenting and reassembly. The packet is fragmented at the first place it needs to be and reassembled at the destination in case packet is too long to fit in the data portion of the link frame. De facto MTU is 1500 bytes.
- Routing is next hop routing. A router either routes it to the next hop (by sending it to another router) or delivers it to a destination on the local subnet
- The IP protocol handles receipt of an IP packet, determining if this the final destination or not, and sending of an IP packet. Sending is assisted by the routing table and routing algorithm. *How is the table filled?*
- Error processing is defined for various conditions through the Internet Control Message Protocol (ICMP).

Logical Format of an IP Packet

Version 4 bits	IHL 4 bits	Service Type 8 bits	Total length 16 bits	
Identification 16 bits			Flags 3 bits	Fragment offset 13 bits
Time to Live 8 bits		Protocol 8 bits	Header Checksum 16 bits	
Source IP Address 32 bits				
Destination IP Address 32 bits				
IP Options if used plus padding to 4 bytes Variable length multiples of 4 bytes				
Encapsulated Data Variable length, integral number of bytes				

Fields of the IP Packet

- *Version*: the version number of the protocol. Version = 4 for IPv4.
- *Header length*: the length of the header in 4 byte words. Header length = 5 if options are not used.
- *Service type*: *Previously*: 3 bits of precedence (rarely used), 4 bits DTRM representing delay, throughput, reliability, and monetary cost. Last bit 0. Rarely used.

  Now: DiffServ domain – 6 bit DS field, 2 bit ECN field.
- *Total length*: length in bytes of the header plus data. Maximum size is 65,535 bytes.
- *Identification*, *flags*, *fragment offset*: used for fragmentation and reassembly (offset in 8 byte chunks)
- *Time to live (TTL)*: Originally seconds, now usually hop count. Source sets it (often 30 used). Each router must decrement by at least 1. When 0 packet discarded.

Fields of the IP Packet (continued)

- *Protocol*: the higher level identification of the packet or how or by which module the data is to be interpreted
  - 1: Internet Control Message Protocol (ICMP)
  - 6: Transmission Control Protocol (TCP)
  - 8: Exterior Gateway Protocol (EGP)
  - 17: User Datagram Protocol (UDP)
  - 89: Open Shortest Path Protocol (OSPF)
- *Header checksum*: a checksum computed over the header only. Note that this needs to change if header changes.
- *Source IP Address*: address of source. Must be an individual (unicast) address
- *Destination IP address*: address of destination. May be a multicast address
- *IP options*:sequence of 8 bit option code, 8 bit option length, plus rest of information for the option. Options can be for example: record route, loose source routing, strict source routing, timestamp

DiffServ (Differentiated services) & ECN (Explicit congestion notification)

- DiffServ traffic management marks packets to classify traffic into different classes. Thus different classes can be given preferential treatment
- The DS field of 6 bits is used for this classification. The value in this field is called the Code Point value.
  - Default is the normal best effort behavior and has value 000000.
  - Expedited Forwarding (EF) has DSCP of 101110. This is for real-time traffic that requires low delay and low jitter.
  - Other DiffServ classifications have also been defined.
- ECN
  - Uses least significant two bits of the service type field
  - Requires both ends of an upper layer protocol such as TCP to cooperate.
  - Packets can be marked by intermediate routers with 11 to indicate congestion encountered.

Transport Layer Protocols

- Two important ones are UDP and TCP
- End-to-end transmission between an entity on one machine and an entity on the other machine.
- The end point of the connection on each machine is called a *port*. This is a 16 bit number.
- UDP
  - Connectionless “unreliable” service with best effort delivery
  - Packets may be routed on different paths and arrive out of order
  - Packets may be duplicated or dropped
- TCP
  - Reliable connection oriented service
  - Connection can be viewed as implementing a “data pipe” or sequence of bytes
  - Reliable delivery through sequencing and retransmission.
  - Flow control
  - Timers and counters are used to implement the features

UDP

- Basically adds notion of source and destination ports to identify endpoints.
- Computation of the checksum includes a pseudo-header that includes the source and destination IP addresses plus the protocol value in the IP header.
- Treats each packet sent as a separate entity with automatic boundaries.
- Used often for audio / video types of data

UDP Logical Format

source port 16 bits	destination port 16 bits
UDP length (in bytes) 16 bits	checksum 16 bits
Data, variable length, multiple bytes	

Ports

- Endpoints of the transport layer and used by both TCP and UDP
- Port should not be equated with a mailbox as a port can demultiplex packets and a given port in TCP can be shared by multiple connections on the same machine
- Ports can be:
  - Dynamically assigned for the duration of a session
  - Well-known ports for commonly defined services
- UDP and TCP share the same range and services and can use the same number
- Range 1-1023 reserved for well-known ports; range above 5000 are called ephemeral ports and can be used by applications at will; range from 1024-4999 may be restricted by the systems administrator

Some well known ports

- 7 Echo
- 9 Discard
- 17 Quote
- 53 DNS (Domain Name Server)
- 25 SMTP (Simple Mail Transfer Protocol)
- 80 HTTP
- 67 DHCP server
- 68 DHCP client

TCP

- Adds a port to identify end point on a machine
  - Endpoint = (IP address, TCP port #)
  - Pair of endpoints forms a connection
  - Note that even if one endpoint is different, it is a different connection
    - (128.9.0.32, 1184) and (128.10.2.3, 53): a connection to a DNS server
    - (18.26.0.36, 1069) and (128.10.2.3, 25): a connection from a different machine to another service at the same machine
    - (128.2.254.139, 1184) and (128.10.2.3, 53): a different connection to the same DNS server

Basic aspects of the TCP protocol

- A connection transfers a stream of bytes without boundaries.
- The stream of bytes is divided up into *segments* for transfer under TCP. The size is negotiated during the start up phase. Suggested size is 536 bytes. With a TCP header of 20 bytes, and an IP header of 20 bytes, this results in an IP packet of 576 bytes
- During data transfer, a sliding window scheme and positive acks are used to effect a reliable virtual circuit
- Flow control is done through increasing and decreasing window size because of packets not acked within a timeout period
- Start up phase is often termed a 3 way handshake with the following segment exchanges
  - SYN, SYN-ACK, ACK
- There is also a termination phase with FIN segments

TCP Logical Format

<table>
<tr><td colspan="3">Source port
16 bits</td><td>Destination port
16 bits</td></tr>
<tr><td colspan="4">sequence number
32 bits</td></tr>
<tr><td colspan="4">acknowledgement number
32 bits</td></tr>
<tr><td>Hlen
4 bits</td><td>Reserved
4 bits</td><td>8-bit control
8 bits</td><td>window size
16 bits</td></tr>
<tr><td colspan="3">checksum
16 bits</td><td>Urgent pointer
16 bits</td></tr>
<tr><td colspan="4">Options plus pad, variable length</td></tr>
<tr><td colspan="4">Data, variable length</td></tr>
</table>

The Control Bits

- 8 bits of control, from left to right, if set, indicate:

  CWR (congestion window reduced)

  ECE (ECN echo)

  URG (urgent pointer)

  ACK (acknowledgement)

  PSH (push) *sent data to application immediately*

  RST (reset)

  SYN (sync)

  FIN (finish)

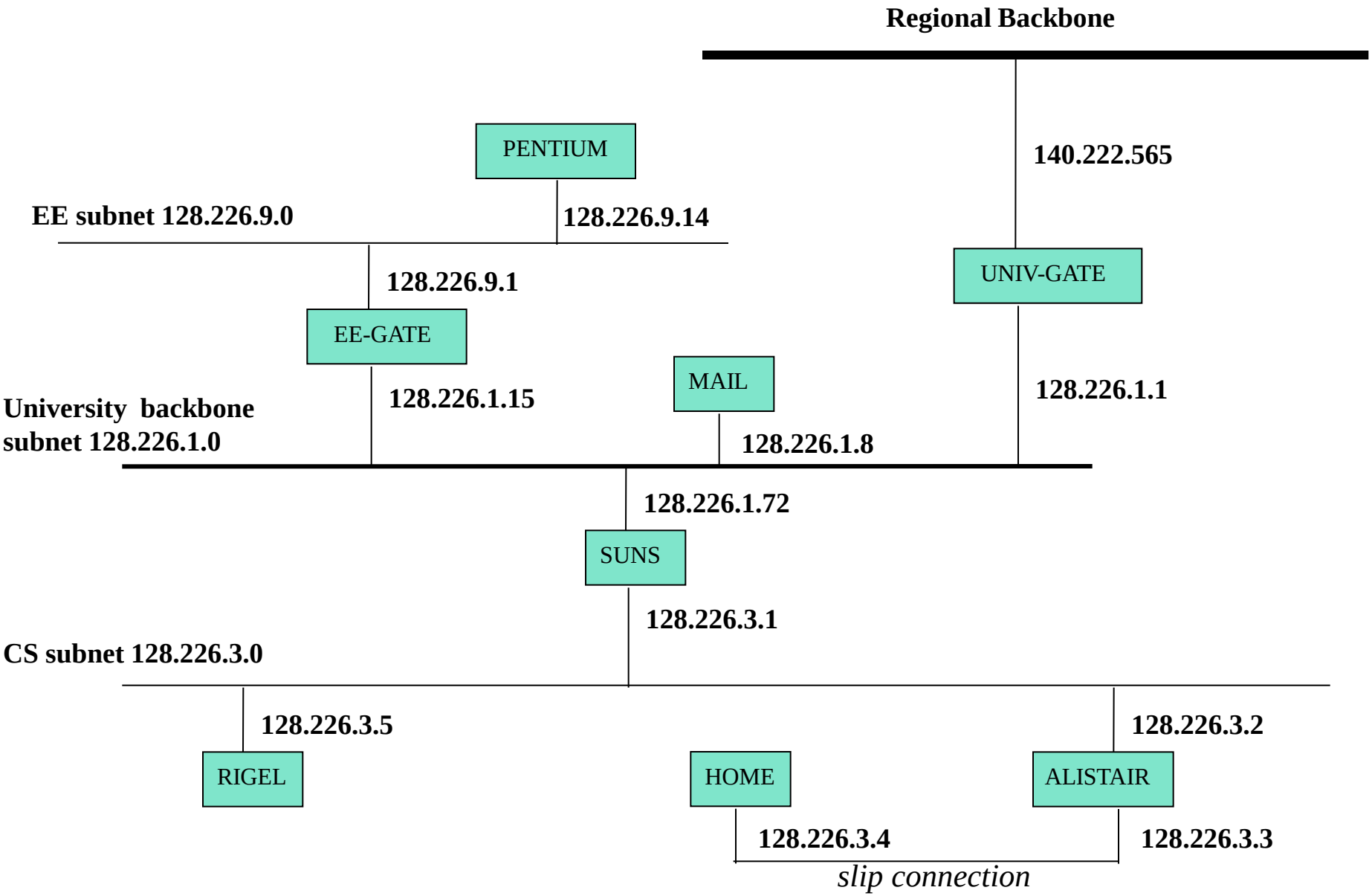
- Three way handshake, A to B
  - A sends SYN packet with SYN set and sequence number x.
  - B syn-acks with ACK set, ack no x + 1, and SYN set with sequence number y.
  - A acks with ACK set, ack number y + 1, and sequence number x + 1.
  - Note that sequence numbers and acks refer to the data bytes that are to be transferred. The first sequence number byte is just for the three way handshake and does not count as data.

Connection Establishment Example

Host A		Host B
SYN→	<SEQ=100><CTL=SYN>	Receive
Receive	<SEQ=300><ACK=101><CTL=SYN,ACK>	←SYN-ACK
ACK→	<SEQ=101><ACK=301><CTL=ACK>	Receive
Connected→	<SEQ=101><ACK=301><CTL=ACK><DATA>	Receive data

What would be the number of the first byte of Host B's data transmission?

Part of a University Routing Domain



Address Resolution Protocol

- How are Internet addresses mapped to physical addresses? That is, in practice, how are IP addresses "resolved" to MAC addresses.
  - Dynamic binding is used. A low level (link level) protocol is used to determine this mapping.
  - Who knows the physical address of the machine with a given IP address? The machine with that given IP address usually knows. So, we can broadcast on the local sub LAN to ask the appropriate host to inform the requestor of the MAC address of the desired machine.
- SUNS wants to know MAC address of Rigel.
  - SUNS broadcasts an ARP request on sub-LAN 128.226.3.0 asking the host with the IP address of Rigel to reply with its MAC address. On reply SUNS will cache this information in an ARP cache.
  - Since if SUNS is sending to Rigel, it expects Rigel to send to it, and so SUNS includes its own MAC address so Rigel can cache this.
- ARP message sent in a broadcast Ethernet frame with protocol type 0806_{16} to indicate this is an ARP message.

Format of an ARP message

Hardware type 16 bits		Protocol Type 16 bits
HLEN 8 bits	PLEN 8 bits	Operation 16 bits
Sender HA [0-3] 32 bits		
Sender HA [4,5] 16 bits		Sender IP [0,1] 16 bits
Sender IP [2,3] 16 bits		Target HA [0,1] 16 bits
Target HA [2-5] 32 bits		
Target IP [0-3] 32 bits		

Hardware type: For Ethernet physical addresses value is 1

Protocol type: for (higher layer protocol) IP addresses value is 0800_{16}

HLEN: hardware address length in bytes (Ethernet =6); PLEN: protocol length (IPv4 = 4)

Operation: ARP request is 1; ARP reply is 2; RARP request is 3; RARP reply is 4

Reverse Address Resolution Protocol (RARP)
Superseded by the Dynamic Host Configuration Protocol

- A machine knowing its hardware address can determine its IP address from some one who might know such as a server.
  - For example, a diskless machine such as an X terminal might need to use this.
  - Multiple servers might be authorized to reply.
- Both ARP and RARP use as simple request reply message format with a simple protocol
  - Protocol handles duplicate requests, timeouts if no reply received.

Read about DHCP and look at the protocol messages sent and received as well as the information transmitted