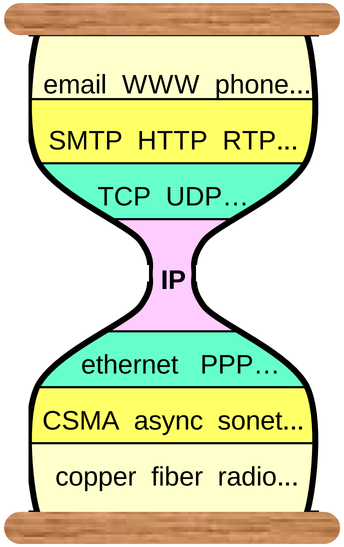
15-441 Computer Networking

Internetworking

Review: Internet Protocol (IP)

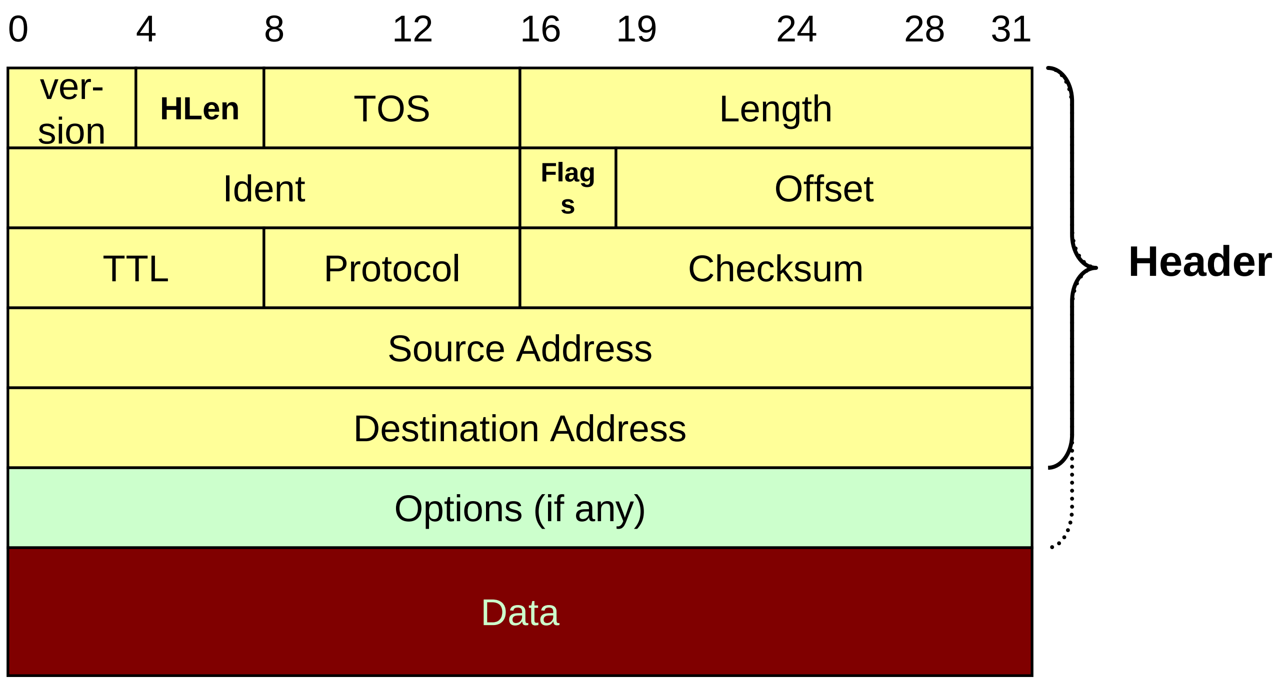
- Hour Glass Model
- Create abstraction layer that hides underlying technology from network application software
- Make as minimal as possible
- Allows range of current & future technologies
- Can support many different types of applications



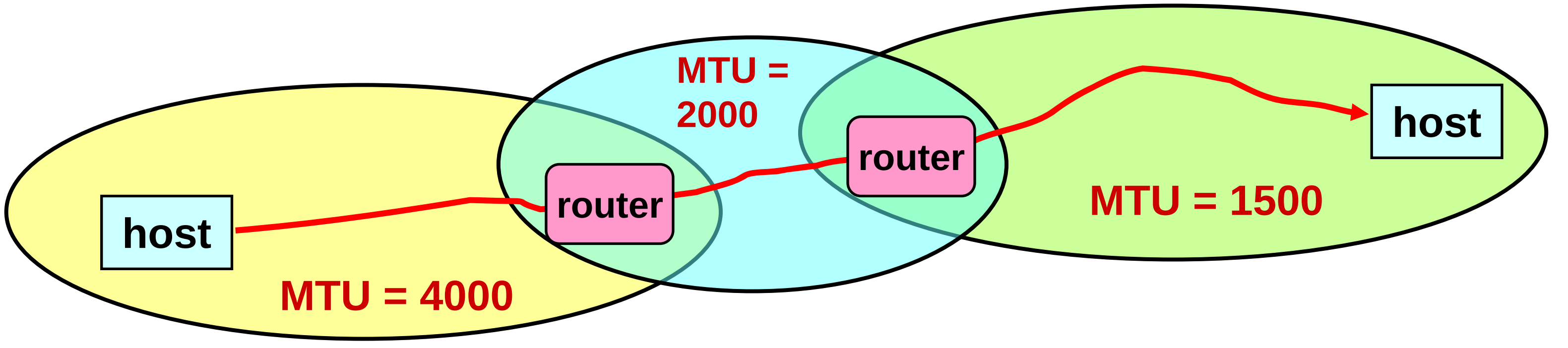
Review: IP Protocol

- What services does it provide?
- What protocol mechanisms to implement the services?

IPv4 Packet Format



IP Fragmentation



- Every network has own Maximum Transmission Unit (MTU)
  - Largest IP datagram it can carry within its own packet frame
    - E.g., Ethernet is 1500 bytes
  - Don't know MTUs of all intermediate networks in advance
- IP Solution
  - When hit network with small MTU, fragment packets

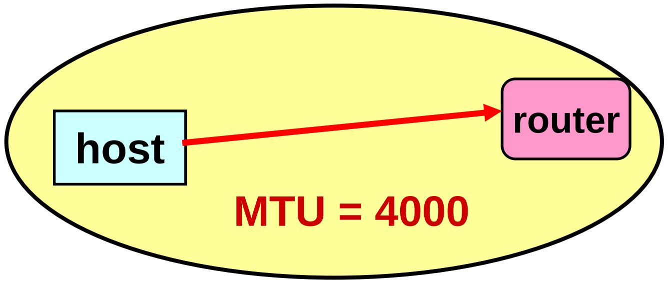
Reassembly

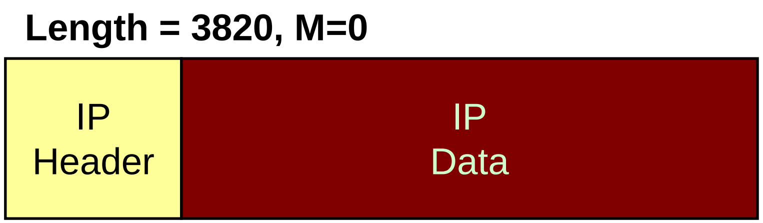
- Where to do reassembly?
  - End nodes or at routers?
- End nodes
  - Avoids unnecessary work where large packets are fragmented multiple times
  - If any fragment missing, delete entire packet
- Dangerous to do at intermediate nodes
  - How much buffer space required at routers?
  - What if routes in network change?
    - Multiple paths through network
    - All fragments only required to go through destination

Fragmentation Related Fields

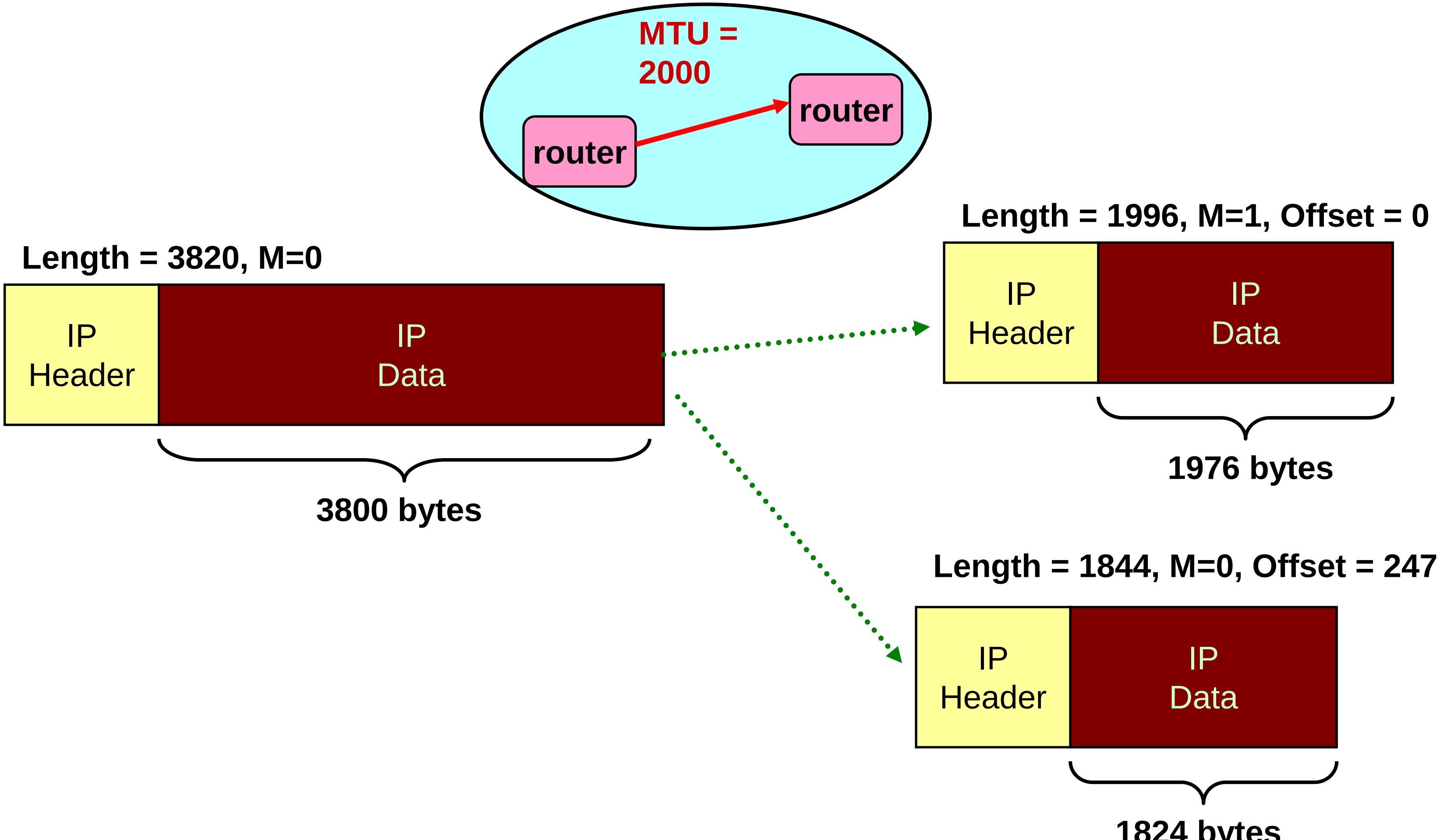
- Length
  - Length of IP fragment
- Identification
  - To match up with other fragments
- Flags
  - Don’t fragment flag
  - More fragments flag
- Fragment offset
  - Where this fragment lies in entire IP datagram
  - Measured in 8 octet units (13 bit field)

IP Fragmentation Example #1

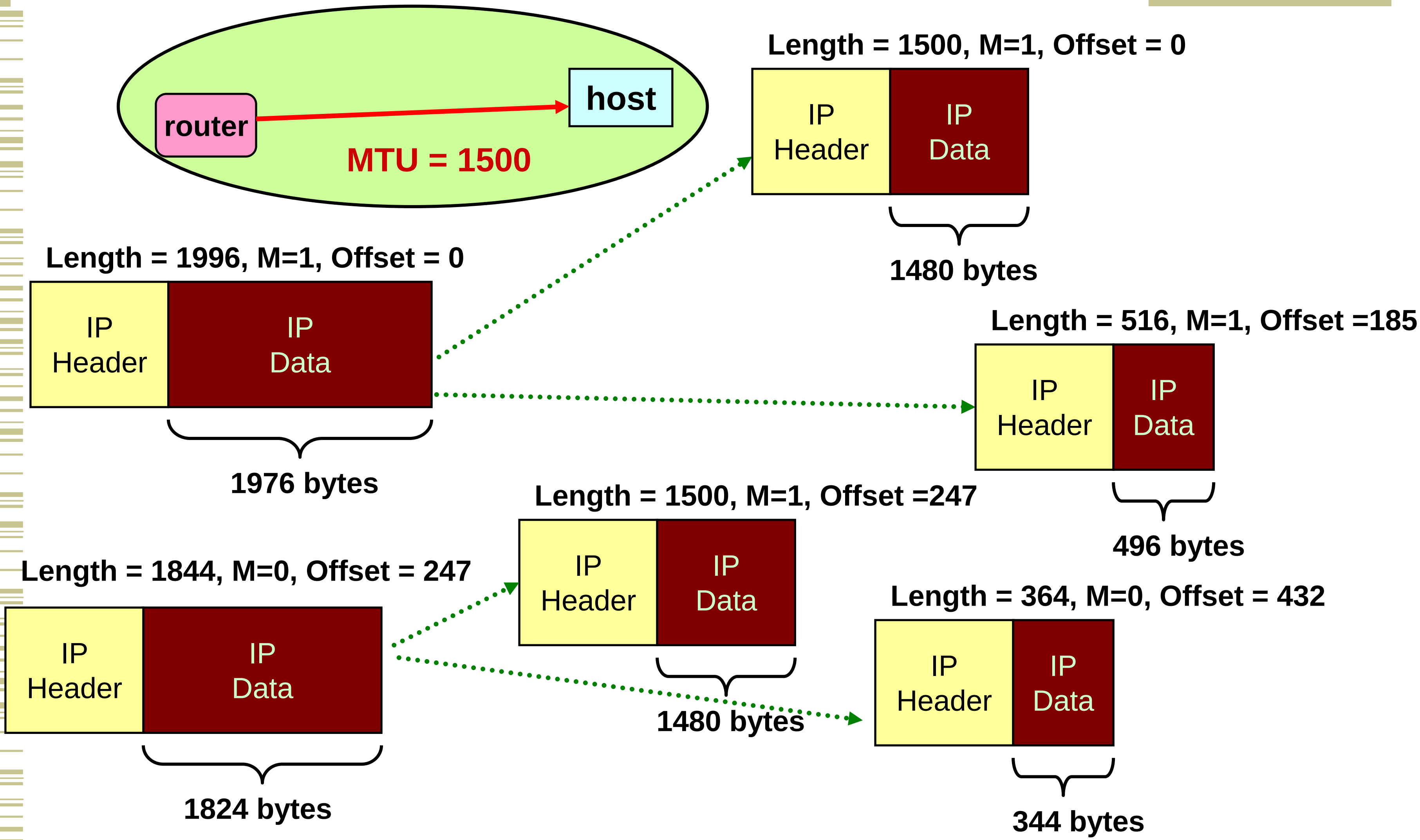




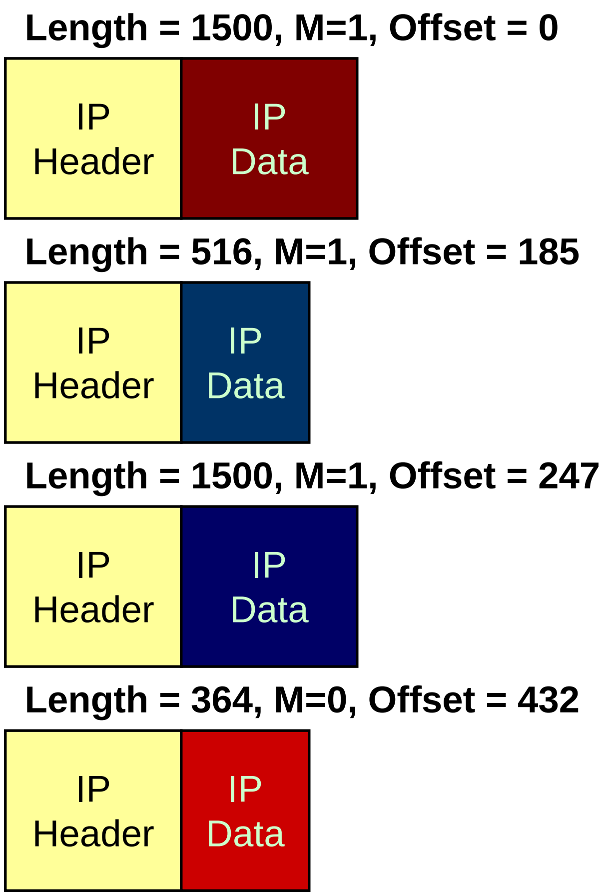
IP Fragmentation Example #2



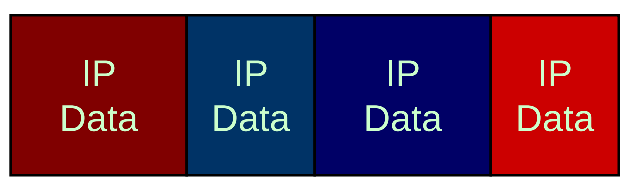
IP Fragmentation Example #3



IP Reassembly



- Fragments might arrive out-of-order
  - Don’t know how much memory required until receive final fragment
- Some fragments may be duplicated
  - Keep only one copy
- Some fragments may never arrive
  - After a while, give up entire process



Fragmentation and Reassembly Concepts

- Demonstrates many Internet concepts
- Decentralized
  - Every network can choose MTU
- Connectionless
  - Each (fragment of) packet contains full routing information
  - Fragments can proceed independently and along different routes
- Best effort
  - Fail by dropping packet
  - Destination can give up on reassembly
  - No need to signal sender that failure occurred
- Complex endpoints and simple routers
  - Reassembly at endpoints

Fragmentation is Harmful

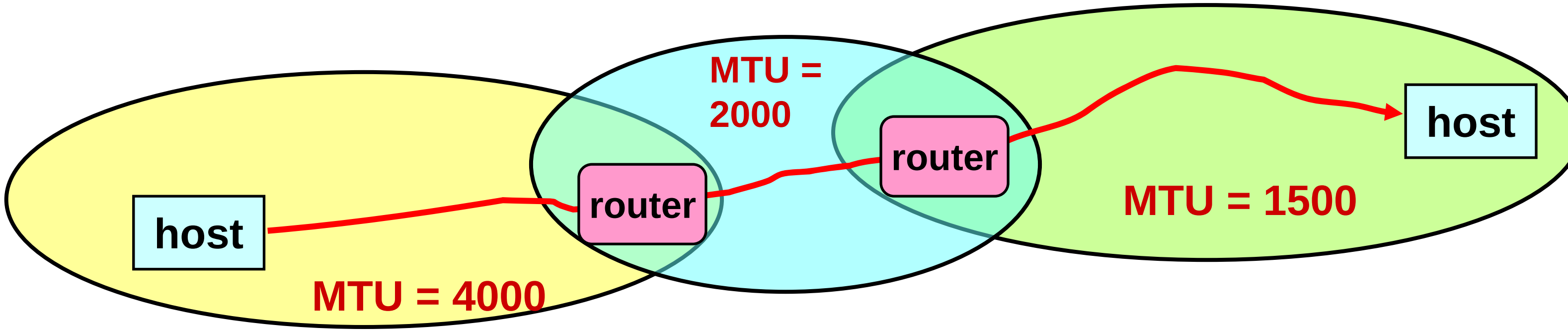
- Uses resources poorly
  - Forwarding costs per packet
  - Best if we can send large chunks of data
  - Worst case: packet just bigger than MTU
- Poor end-to-end performance
  - Loss of a fragment

- Path MTU discovery protocol → determines minimum MTU along route
  - Uses ICMP error messages
- Common theme in system design
  - Assure correctness by implementing complete protocol
  - Optimize common cases to avoid full complexity

Internet Control Message Protocol (ICMP)

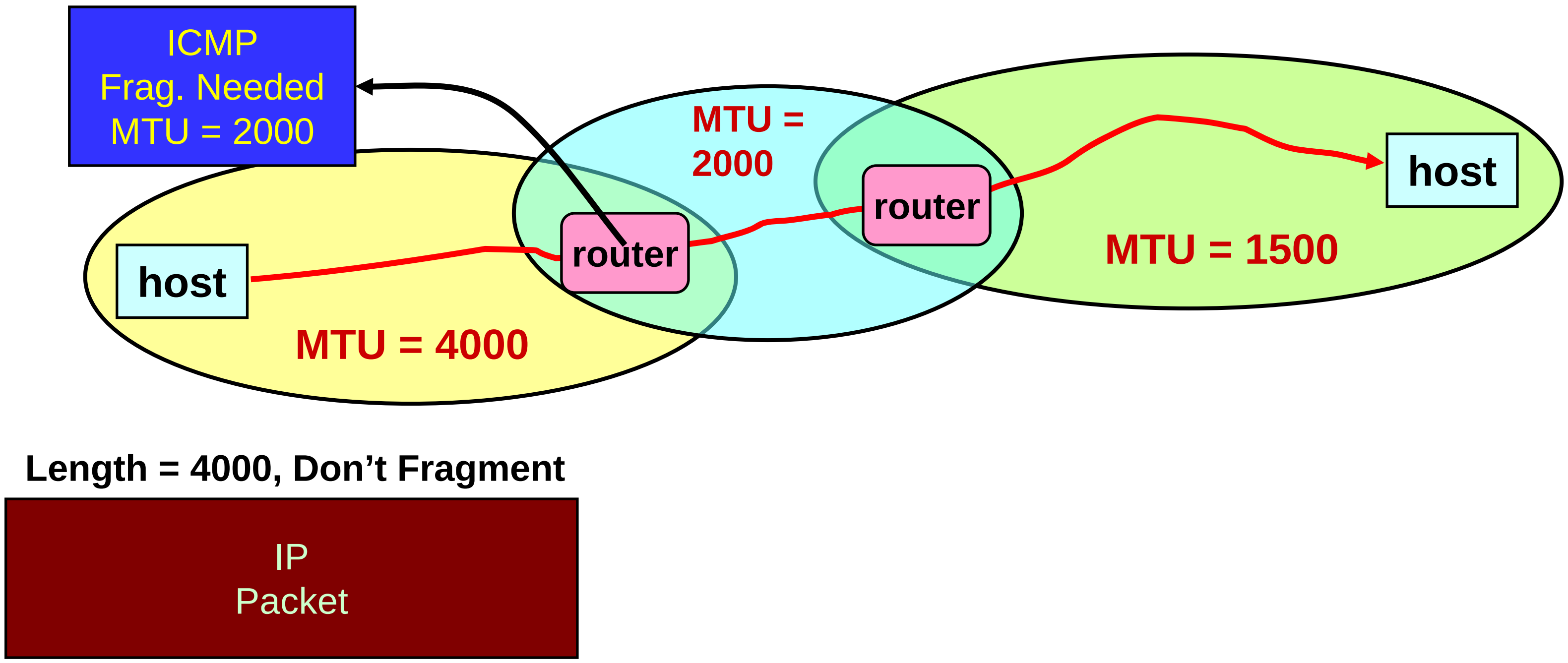
- Short messages used to send error & other control information
- Examples
  - Ping request / response
    - Can use to check whether remote host reachable
  - Destination unreachable
    - Indicates how packet got & why couldn't go further
  - Flow control
    - Slow down packet delivery rate
  - Redirect
    - Suggest alternate routing path for future messages
  - Router solicitation / advertisement
    - Helps newly connected host discover local router
  - Timeout
    - Packet exceeded maximum hop limit

IP MTU Discovery with ICMP

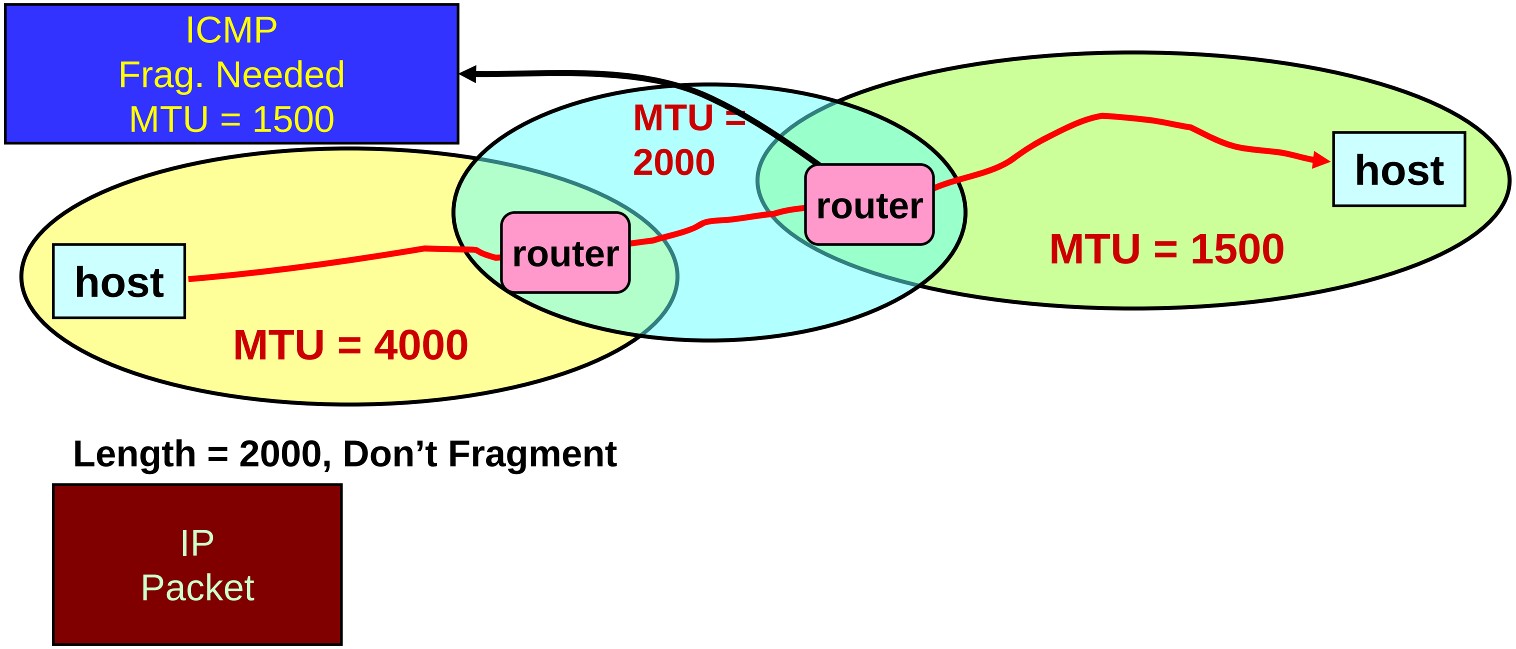


- Typically send series of packets from one host to another
- Typically, all will follow same route
  - Routes remain stable for minutes at a time
- Makes sense to determine path MTU before sending real packets
- Operation
  - Send max-sized packet with “do not fragment” flag set
  - If encounters problem, ICMP message will be returned
    - “Destination unreachable: Fragmentation needed”
    - Usually indicates MTU encountered

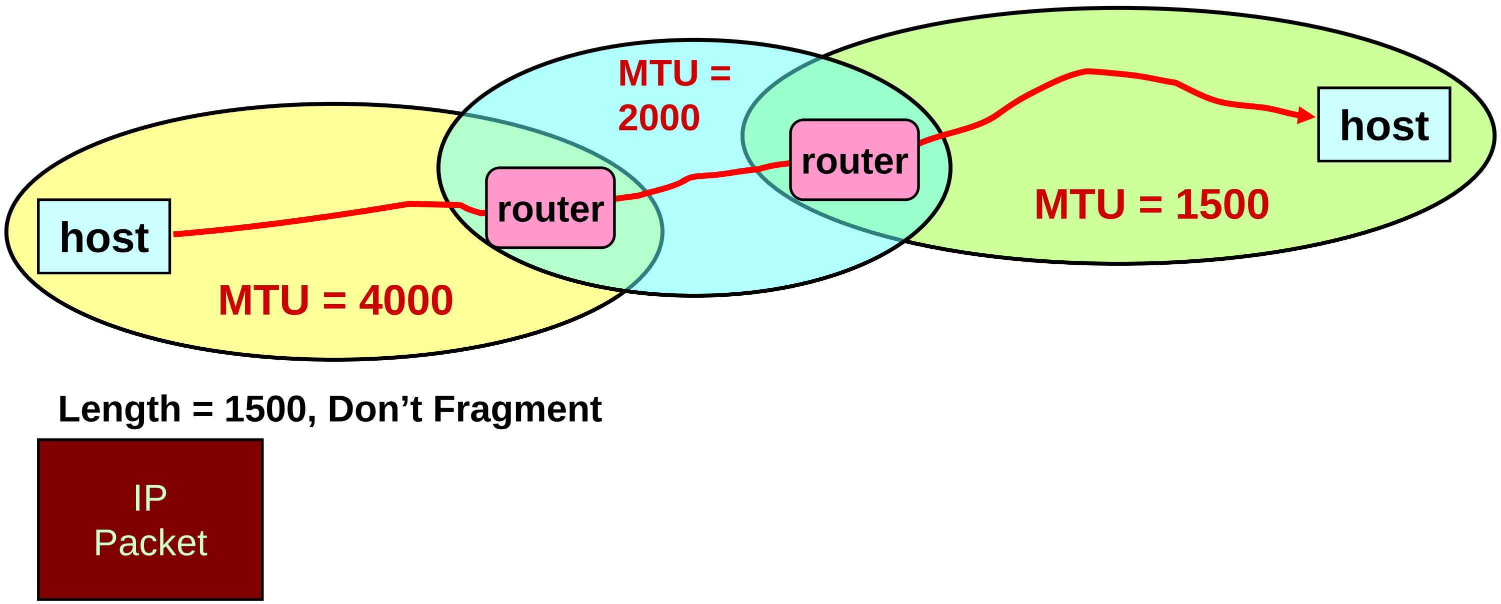
IP MTU Discovery with ICMP



IP MTU Discovery with ICMP



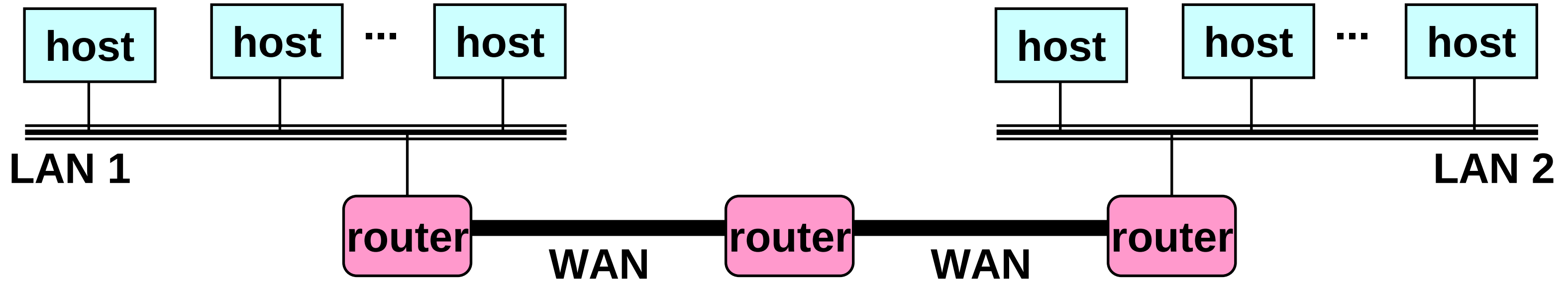
IP MTU Discovery with ICMP



- When successful, no reply at IP level
  - “No news is good news”
- Higher level protocol might have some form of acknowledgement

Private and Public Internet

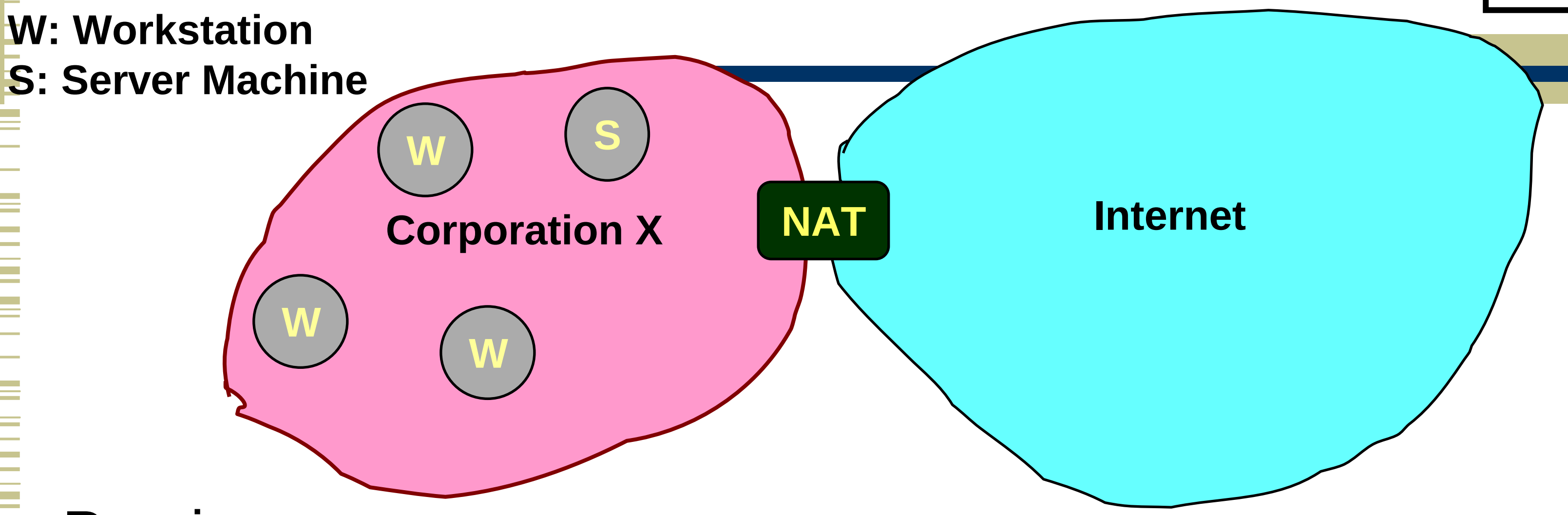
- Both private and public networks can be built on top of IP
  - Internet: public
  - Corporate, military IP networks: private (intranet)
- Users in private networks can access public Internet
- Users can use public Internet to access private



Altering the Addressing Model

- Original IP Model
  - Every host has a unique IP address
- Implications
  - Any host can find any other host
  - Any host can communicate with any other host
  - Any host can act as a server
    - Just need to know host ID and port number
- No Secrecy or Authentication
  - Packet traffic observable by routers and by LAN-connected hosts
  - Possible to forge packets
    - Use invalid source address

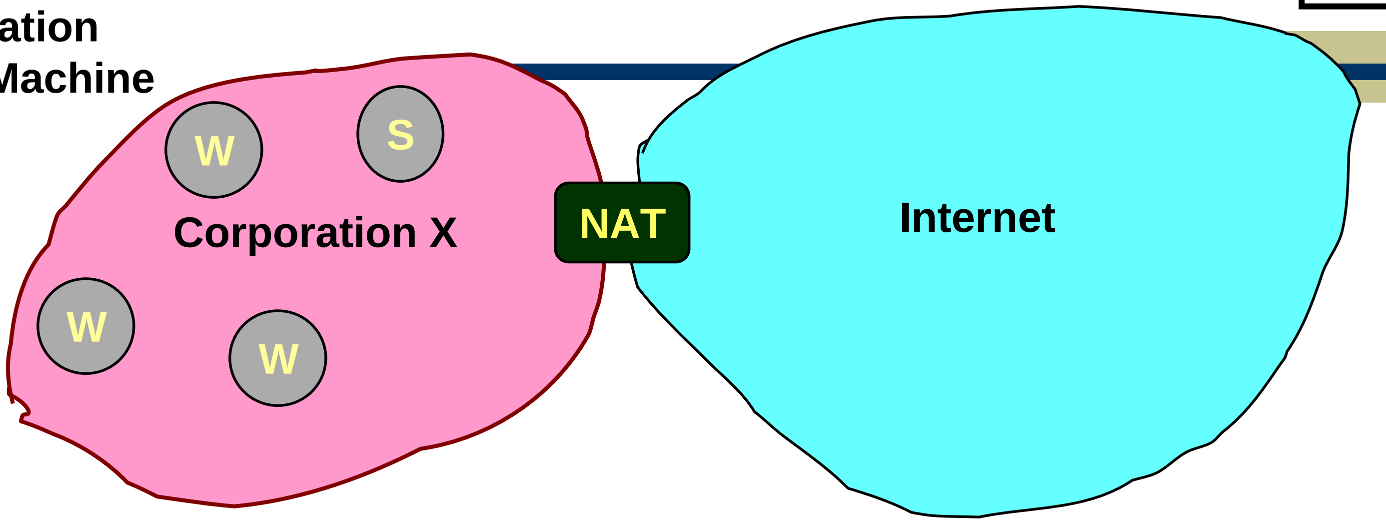
Private Network Accessing Public Internet



- Parsimony
  - Don’t have enough IP addresses for every host in organization
- Security
  - Don’t want every machine in organization known to outside world
  - Want to control or monitor traffic in / out of organization

Reducing IP Addresses

W: Workstation
S: Server Machine

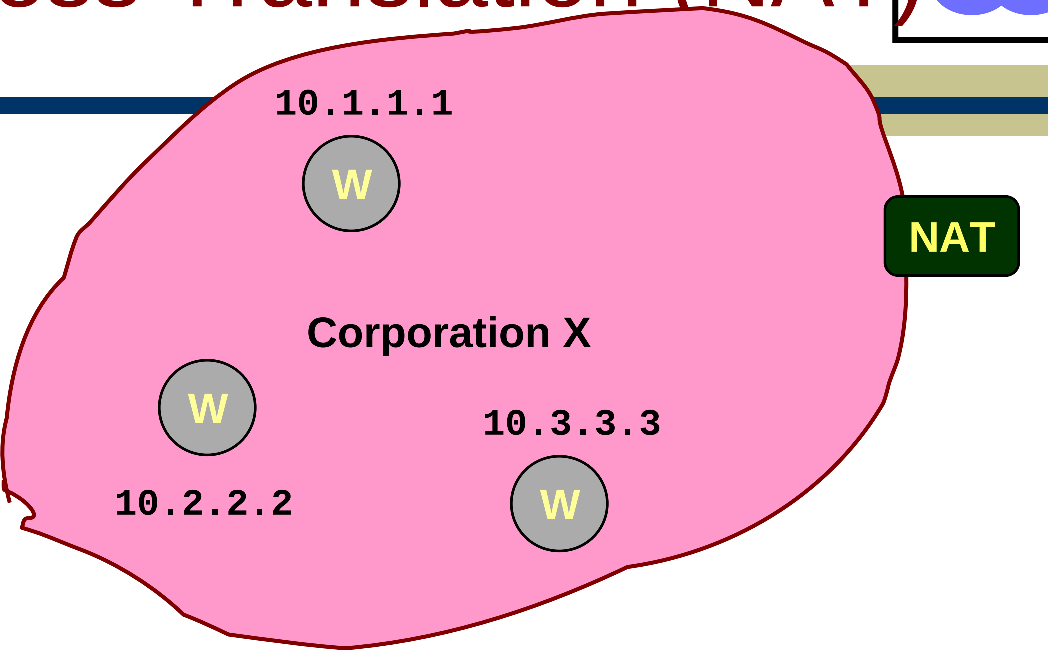


- Most machines within organization are used by individuals
  - “Workstations”
  - For most applications, act as clients
- Small number of machines act as servers for entire organization
  - E.g., mail server
  - All traffic to outside passes through firewall

(Most) machines within organization don’t need actual IP addresses!

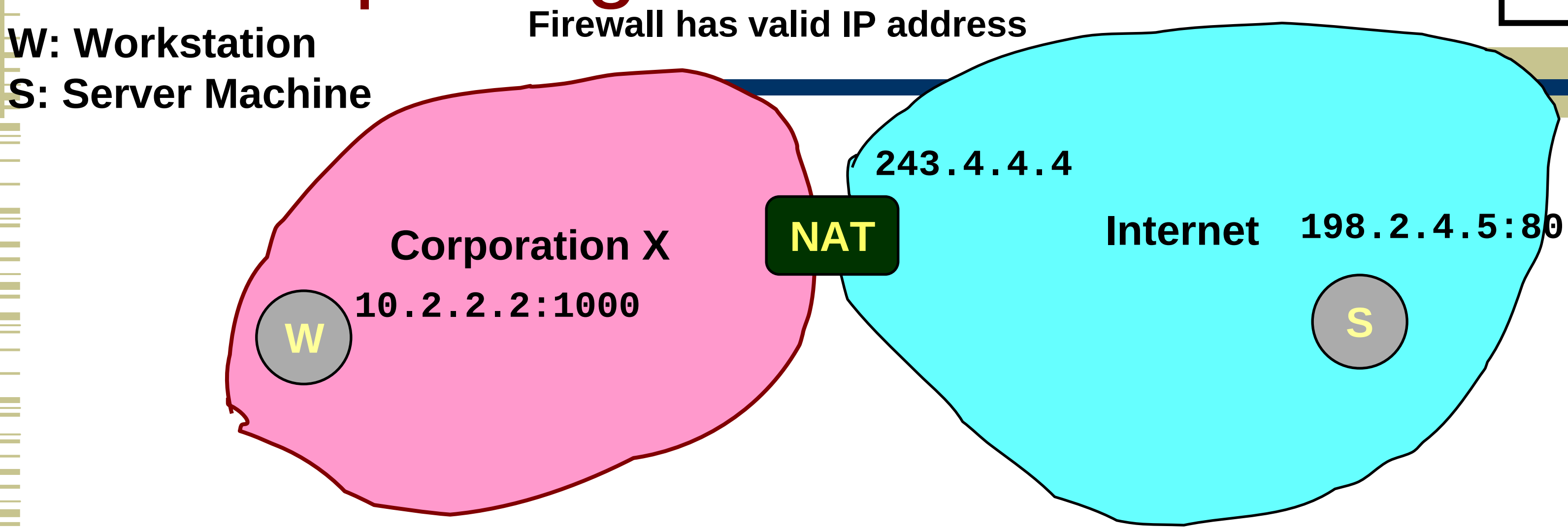
Network Address Translation (NAT)

W: Workstation



- Within Organization
  - Assign every host an unregistered IP address
    - IP addresses 10/8 & 192.168/16 unassigned
  - Route within organization by IP protocol
- Firewall
  - Doesn’t let any packets from internal node escape
  - Outside world doesn’t need to know about internal addresses

NAT: Opening Client Connection



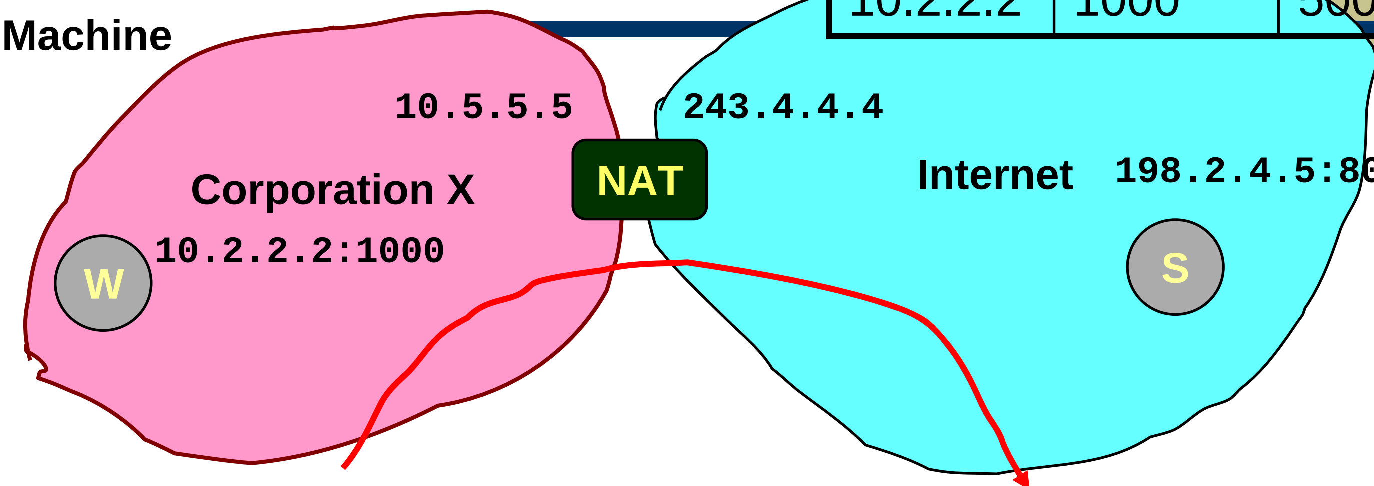
- Client 10.2.2.2 wants to connect to server 198.2.4.5:80
  - OS assigns ephemeral port (1000)
- Connection request intercepted by firewall
  - Maps client to port of firewall (5000)
  - Creates NAT table entry

Int Addr	Int Port	NAT Port
10.2.2.2	1000	5000

NAT: Client Request

Int Addr	Int Port	NAT Port
10.2.2.2	1000	5000

W: Workstation
S: Server Machine



source: 10.2.2.2
dest: 198.2.4.5
src port: 1000
dest port: 80

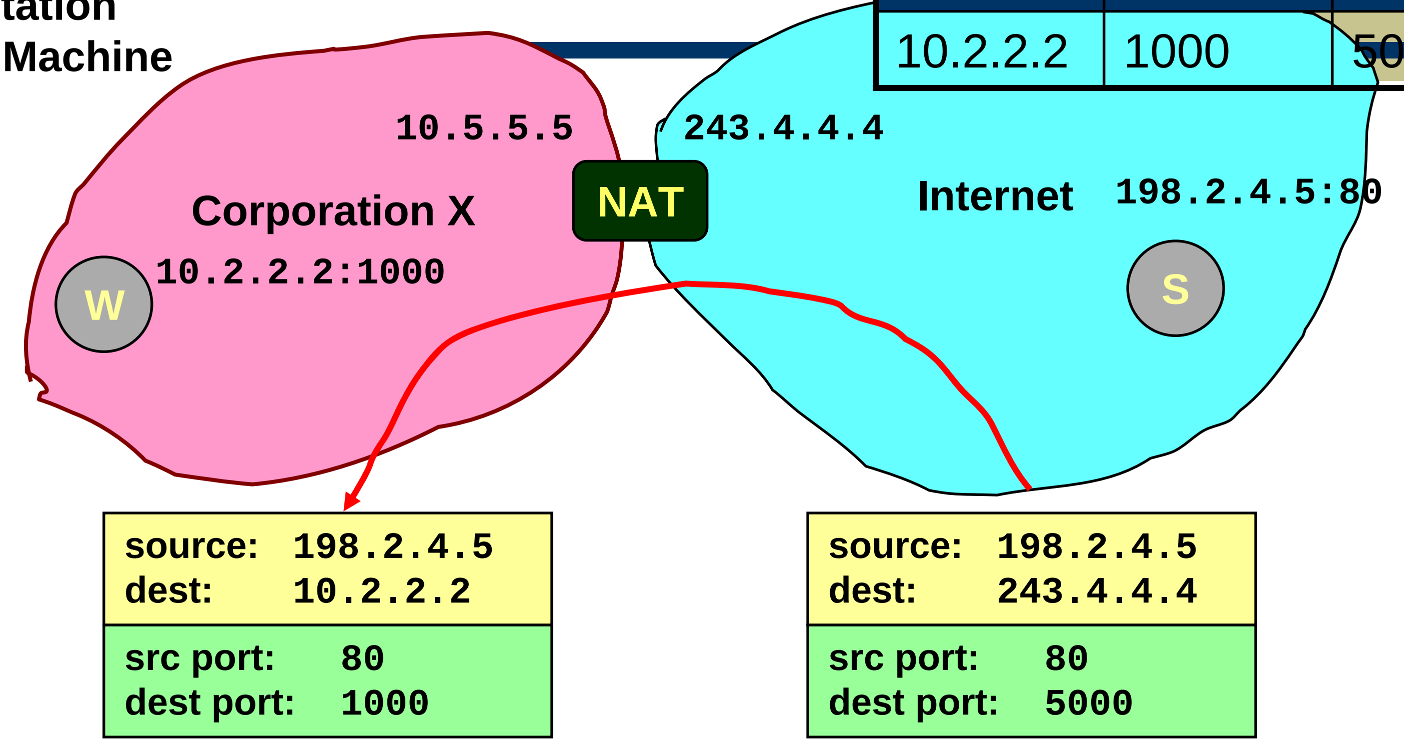
source: 243.4.4.4
dest: 198.2.4.5
src port: 5000
dest port: 80

- Firewall acts as proxy for client
  - Intercepts message from client and marks itself as sender

NAT: Server Response

W: Workstation
S: Server Machine

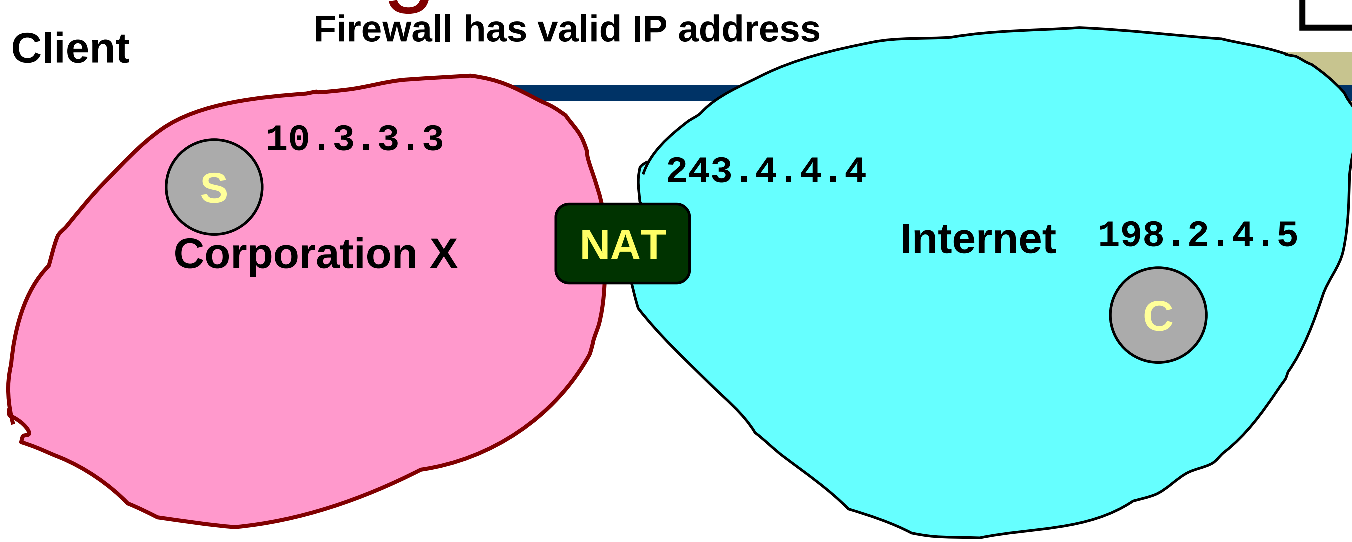
Int Addr	Int Port	NAT Port
10.2.2.2	1000	5000



- Firewall acts as proxy for client
  - Acts as destination for server messages
  - Relabels destination to local addresses

NAT: Enabling Servers

C: Remote Client
S: Server



- Use port mapping to make servers available

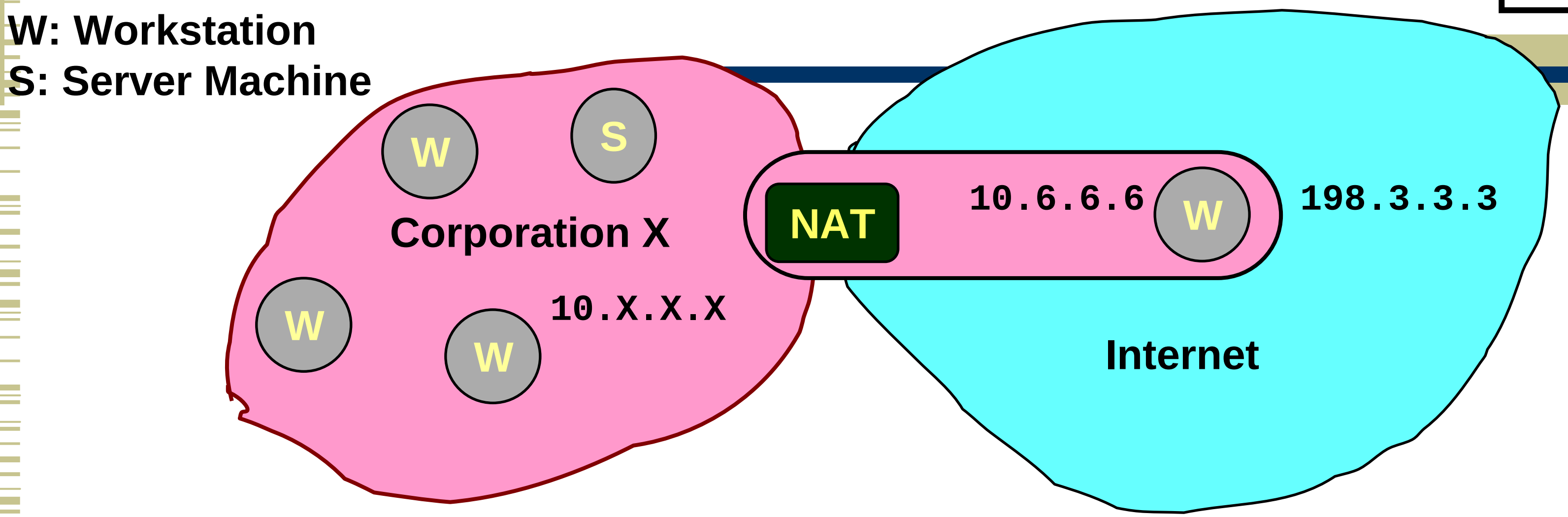
Int Addr	Int Port	NAT Port
10.3.3.3	80	80

  - Manually configure NAT table to include entry for well-known port
  - External users give address 243.4.4.4:80
  - Requests forwarded to server

Properties of Firewalls with NAT

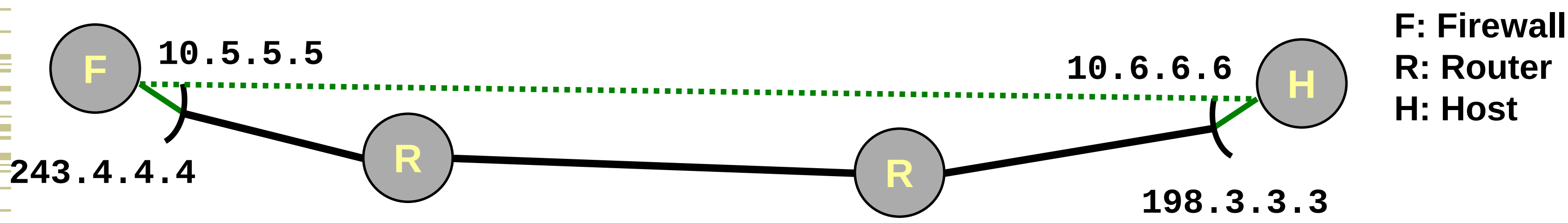
- Advantages
  - Hides IP addresses used in internal network
    - Easy to change ISP: only NAT box needs to have IP address
    - Fewer registered IP addresses required
  - Basic protection against remote attack
    - Does not expose internal structure to outside world
    - Can control what packets come in and out of system
    - Can reliably determine whether packet from inside or outside
- Disadvantages
  - Contrary to the “open addressing” scheme envisioned for IP addressing
  - Hard to support peer-to-peer applications
    - Why do so many machines want to serve port 1214?

Extending Private Network



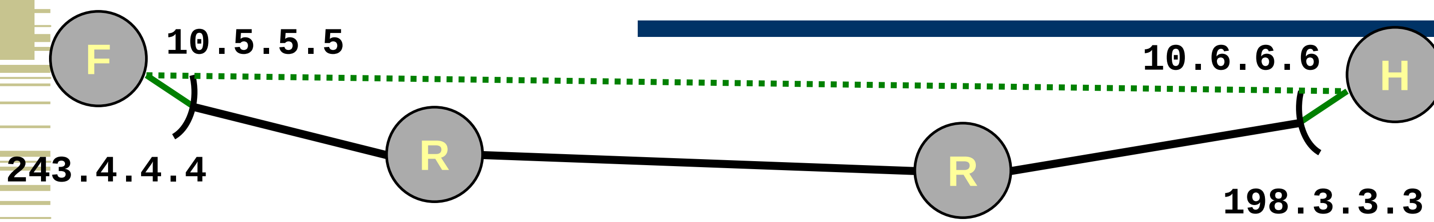
- Supporting Road Warrior
  - Employee working remotely with assigned IP address 198.3.3.3
  - Wants to appear to rest of corporation as if working internally
    - From address 10.6.6.6
    - Gives access to internal services (e.g., ability to send mail)
- Virtual Private Network (VPN)
  - Overlays private network on top of regular Internet

Supporting VPN by Tunneling

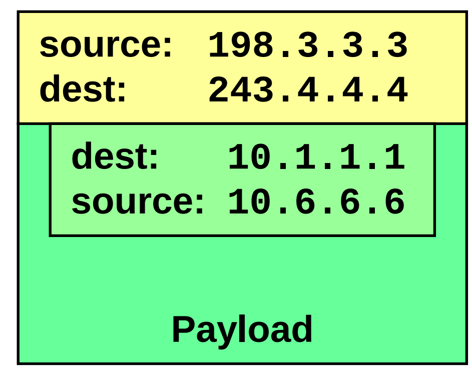


- Concept
  - Appears as if two hosts connected directly
- Usage in VPN
  - Create tunnel between road warrior & firewall
  - Remote host appears to have direct connection to internal network

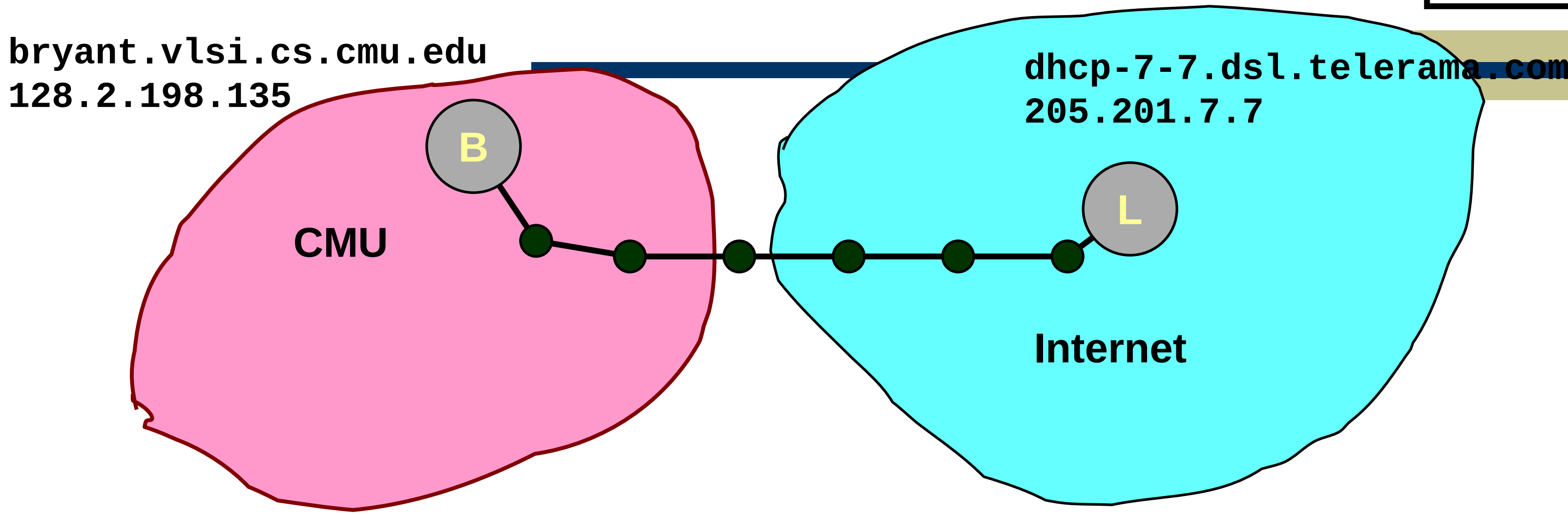
Implementing Tunneling



- Host creates packet for internal node 10.6.1.1.1
- Entering Tunnel
  - Add extra IP header directed to firewall (243.4.4.4)
  - Original header becomes part of payload
  - Possible to encrypt it
- Exiting Tunnel
  - Firewall receives packet
  - Strips off header
  - Sends through internal network to destination



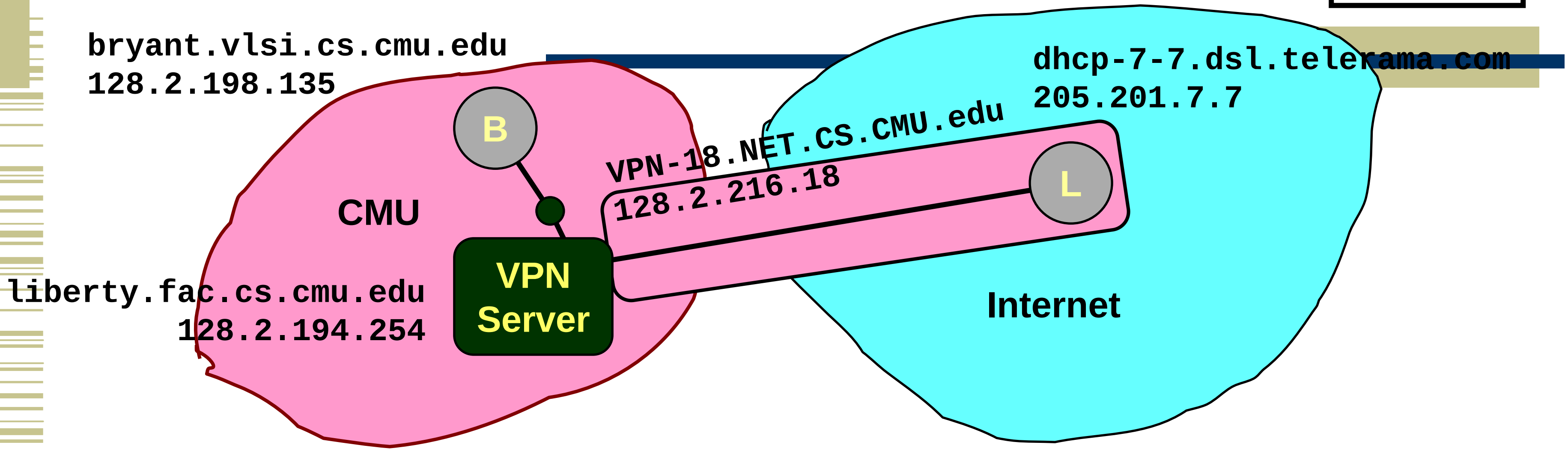
CMU CS VPN Example



- Operation
  - Running echo server on CMU machine 128.2.198.135
  - Run echo client on laptop connected through DSL from non-CMU ISP
- Without VPN

```
server connected to
dhcp-7-
  7.dsl.telerama.com
(205.201.7.7)
```

CMU CS VPN Example



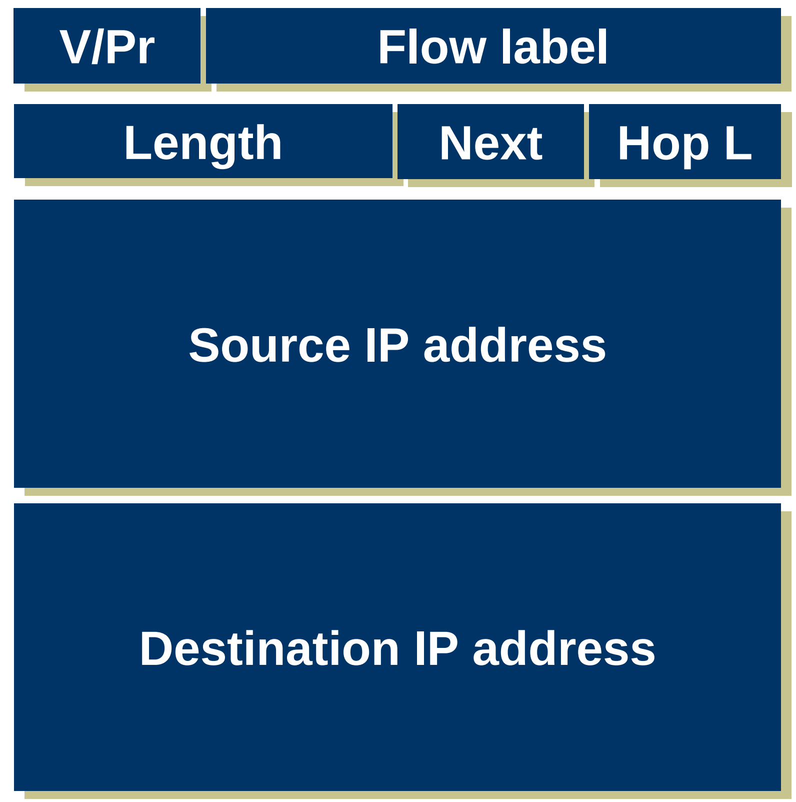
- CS has server to provide VPN services
- Operation
  - Running echo server on CMU machine 128.2.198.135
  - Run echo client on laptop connected through DSL from non-CMU ISP
- With VPN server connected to `VPN-18.NET.CS.CMU.EDU` `(128.2.216.18)`
- Effect
  - For other hosts in CMU, packets appear to originate from within CMU

Important Concepts

- Ideas in the Internet
  - Base-level protocol (IP) provides minimal service level
    - Allows highly decentralized implementation
    - Each step involves determining next hop
    - Most of the work at the endpoints
  - Use ICMP for low-level control functions
- Changes to Addressing Model
  - Have moved away from “everyone knows everybody” model of original Internet
  - Firewalls + NAT hide internal networks
  - VPN / tunneling build private networks on top of commodity network

IPv6

- “Next generation” IP.
- Most urgent issue: increasing address space.
  - 128 bit addresses
- Simplified header for faster processing:
  - No checksum (why not?)
  - No fragmentation (?)
- Support for guaranteed services: priority and flow id
- Options handled as “next header”
  - reduces overhead of handling options



IPv6 Addressing

- Do we need more addresses? Probably, long term
  - Big panic in 90s: “We’re running out of addresses!”
  - Big worry: Devices. Small devices. Cell phones, toasters, everything.
- 128 bit addresses provide space for structure (good!)
  - Hierarchical addressing is much easier
  - Assign an entire 48-bit sized chunk per LAN -- use Ethernet addresses
  - Different chunks for geographical addressing, the IPv4 address space,
  - Perhaps help clean up the routing tables - just use one huge chunk per ISP and one huge chunk per customer.

010	Registry	Provider	Subscriber	Sub Net	Host

IPv6 Cleanup - Router-friendly

- Common case: Switched in silicon ("fast path")
- Weird cases: Handed to CPU ("slow path", or "process switched")
  - Typical division:
    - Fast path: Almost everything
    - Slow path:
      - Fragmentation
      - TTL expiration (traceroute)
      - IP option handling
  - Slow path is evil in today's environment
    - "Christmas Tree" attack sets weird IP options, bits, and overloads router.
    - Developers can't (really) use things on the slow path for data flow.
      - If it became popular, they'd be in the soup!
- Other speed issue: Touching data is expensive. Designers would like to minimize accesses to packet during forwarding.

IPv6 Header Cleanup

- No checksum
- Why checksum just the IP header?
  - Efficiency: If packet corrupted at hop 1, don't waste b/w transmitting on hops 2..N.
  - Useful when corruption frequent, b/w expensive
  - Today: Corruption rare, b/w cheap

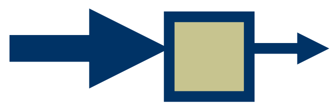
IPv6 Header Cleanup

- Different options handling
- IPv4 options: Variable length header field. 32 different options.
  - Rarely used
  - No development / many hosts/routers do not support
    - Worse than useless: Packets w/options often even get dropped!
  - Processed in “slow path”.
- IPv6 options: “Next header” pointer
  - Combines “protocol” and “options” handling
    - Next header: “TCP”, “UDP”, etc.
  - Extensions header: Chained together
  - Makes it easy to implement host-based options
  - One value “hop-by-hop” examined by intermediate routers
    - Things like “source route” implemented only at intermediate hops

IPv6 Fragmentation Cleanup

- IPv4:

Large MTU



Small MTU

Router must fragment

- IPv6:
  - Discard packets, send ICMP “Packet Too Big”
    - Similar to IPv4 “Don’t Fragment” bit handling
  - Sender must support Path MTU discovery
    - Receive “Packet too Big” messages and send smaller packets
  - Increased minimum packet size
    - Link must support 1280 bytes;
    - 1500 bytes if link supports variable sizes

- Reduced packet processing and network complexity.
- Increased MTU a boon to application writers

- Hosts can still fragment - using fragmentation header. Routers don’t deal with it any more.

Migration from IPv4 to IPv6

- Interoperability with IP v4 is necessary for gradual deployment.
- Two complementary mechanisms:
  - dual stack operation: IP v6 nodes support both address types
  - tunneling: tunnel IP v6 packets through IP v4 clouds
- Alternative is to create IPv6 islands, e.g. corporate networks, ...
  - Use of form of NAT to connect to the outside world
  - NAT must not only translate addresses but also translate between IPv4 and IPv6 protocols