15-441 Computer Networking

Lecture 8 –

IPv6, NAT, VPNs, Tunnels

ATM and MPLS

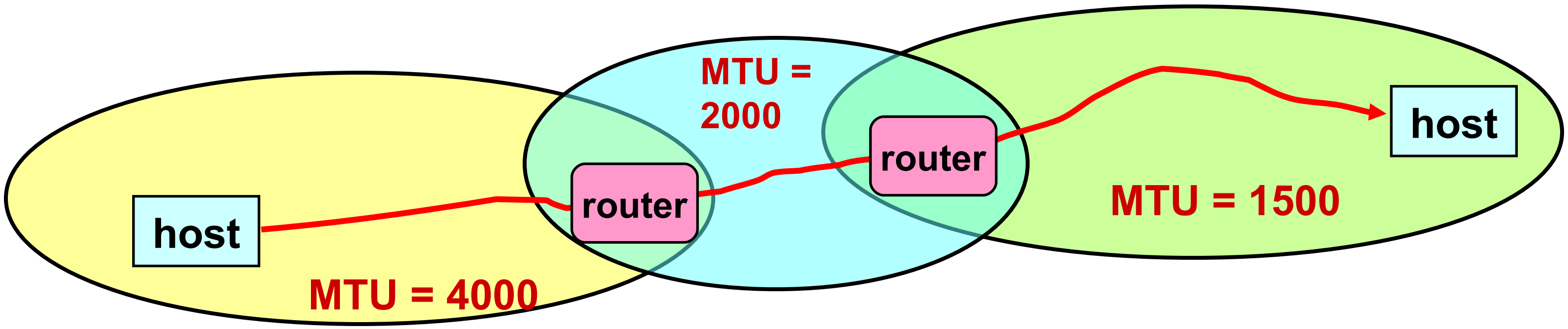
Outline

- ICMP/MTU Discovery
- IPv6
- NAT
- Tunnels
- ATM and MPLS

Internet Control Message Protocol (ICMP)

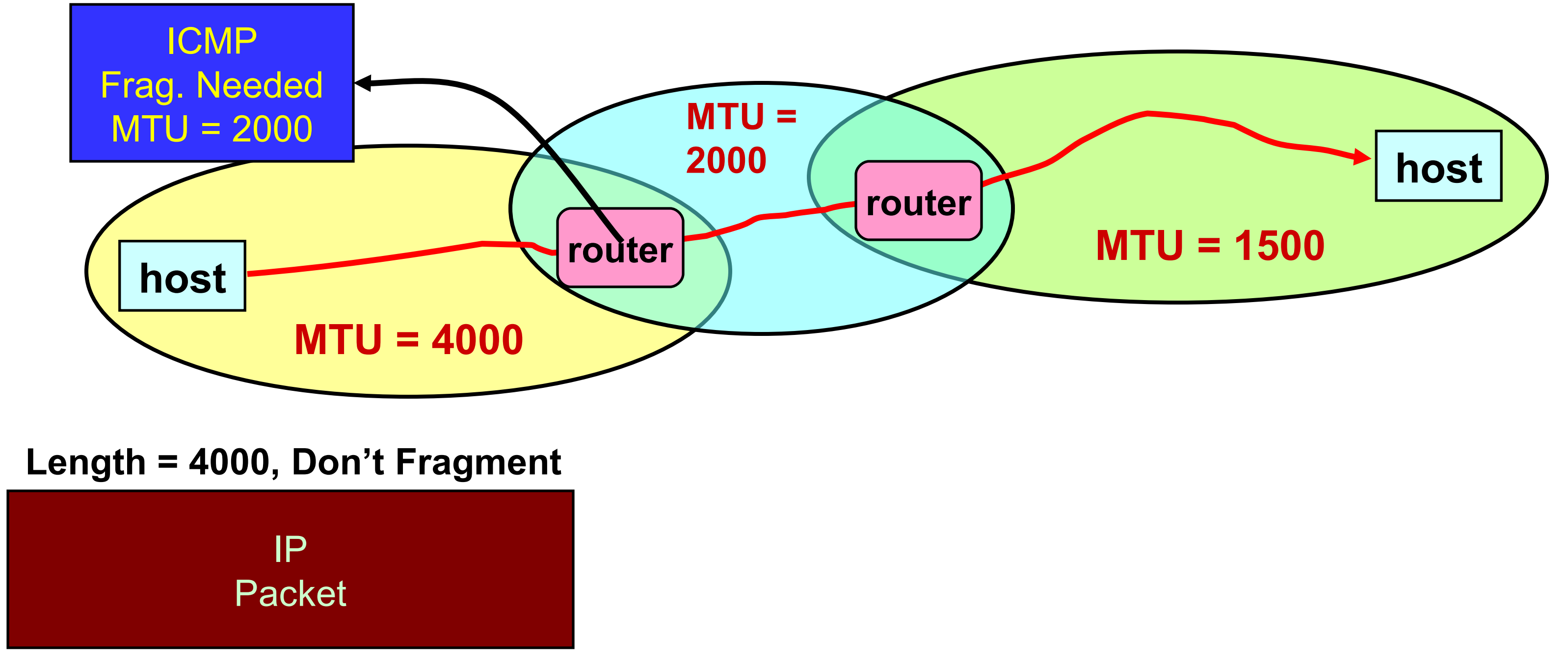
- Short messages used to send error & other control information
- Examples
  - Ping request / response
    - Can use to check whether remote host reachable
  - Destination unreachable
    - Indicates how packet got & why couldn’t go further
  - Flow control
    - Slow down packet delivery rate
  - Redirect
    - 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

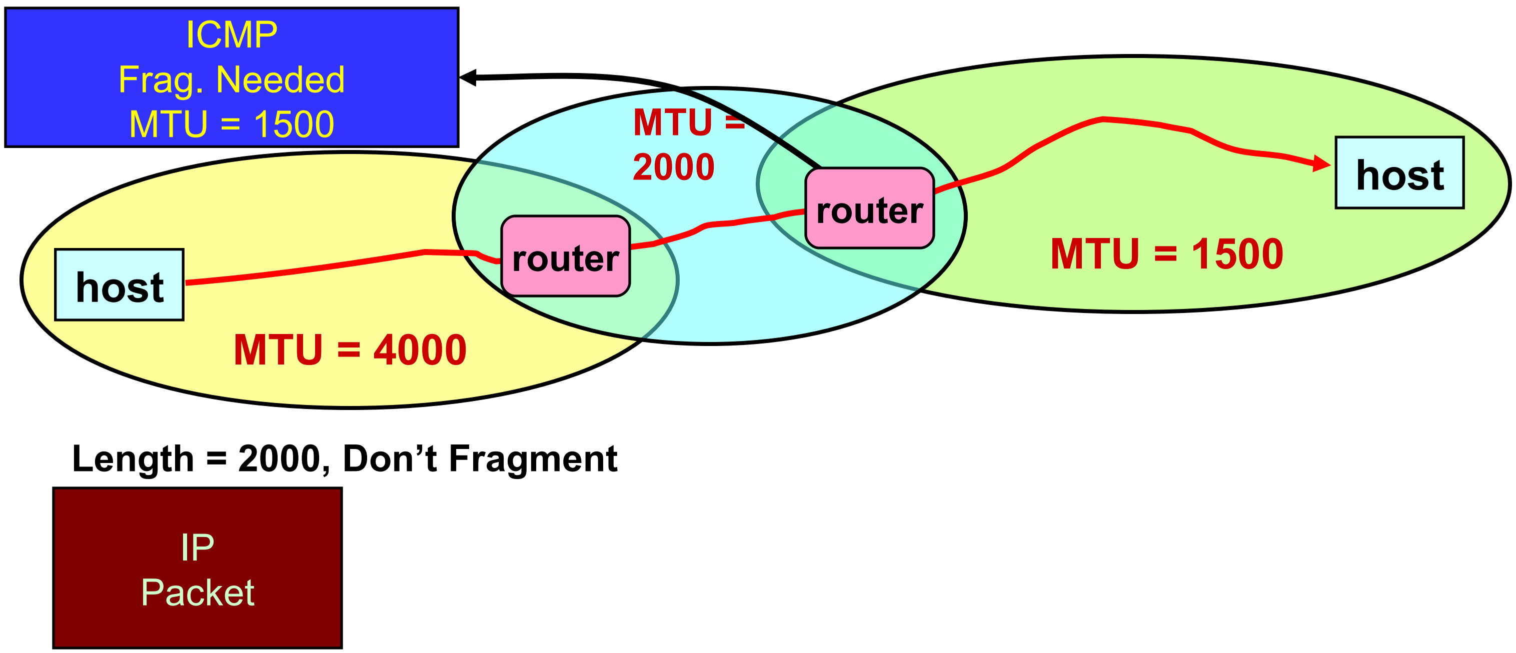


- Typically send series of packets from one host to another
- Typically, all will follow same route
  - Routes remain stable for minutes at a time
- Makes sense to determine path MTU before sending real packets
- Operation
  - Send max-sized packet with “do not fragment” flag set
  - If encounters problem,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

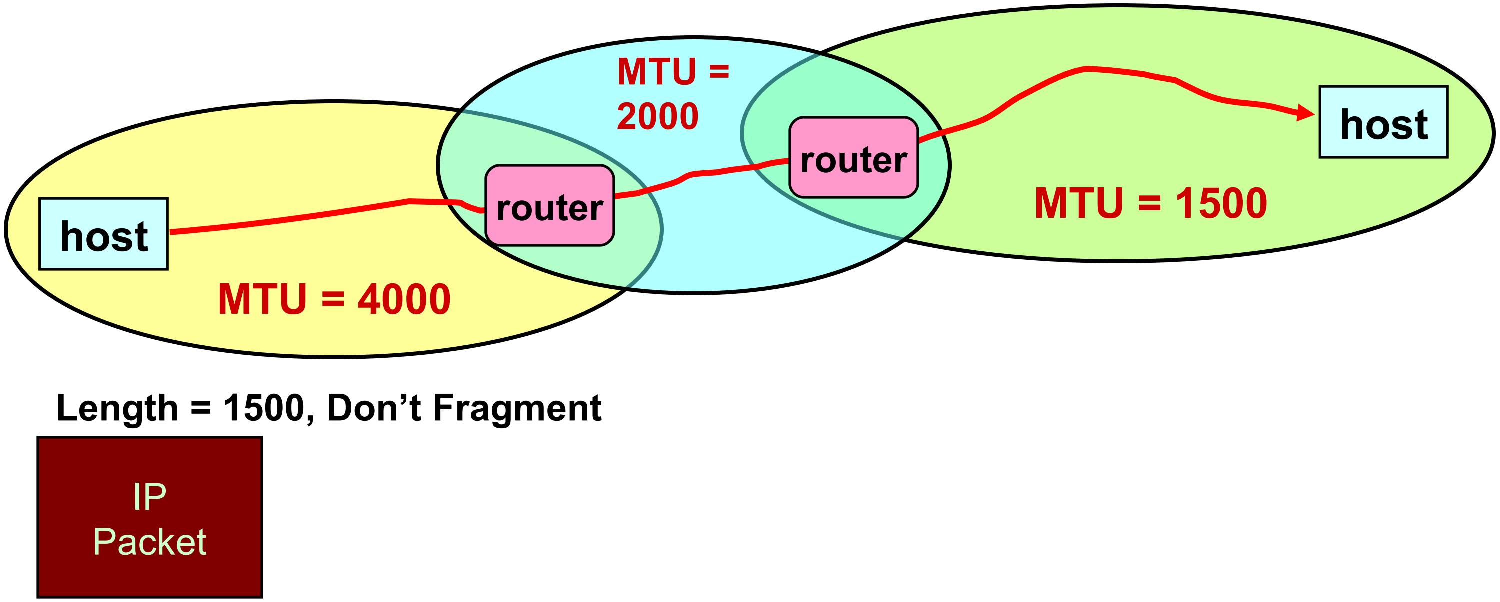
IP MTU Discovery with ICMP



IP MTU Discovery with ICMP



IP MTU Discovery with ICMP



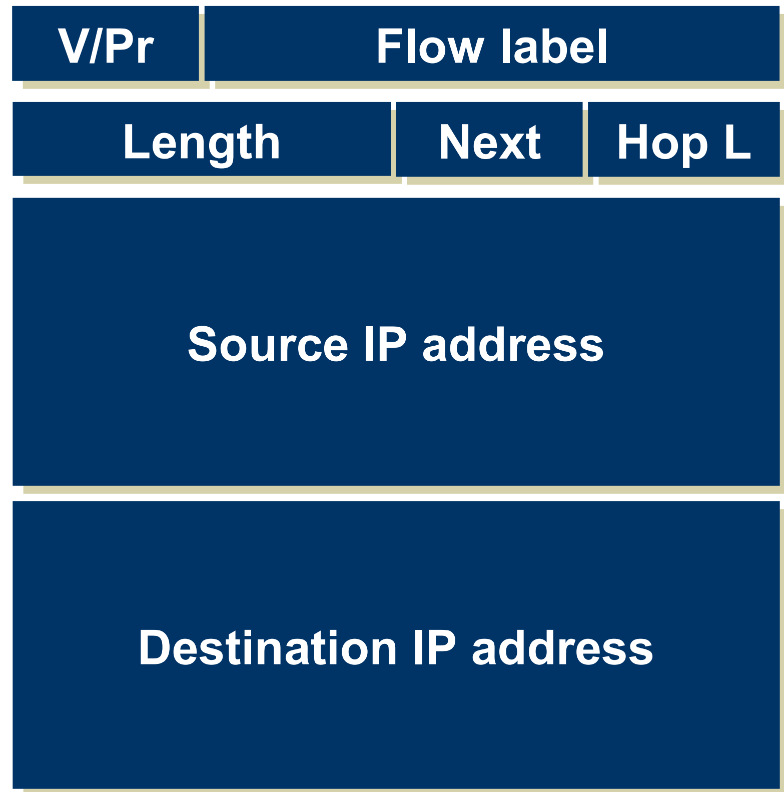
- When successful, no reply at IP level
  - "No news is good news"
- Higher level protocol might have some form of acknowledgement

Outline

- ICMP/MTU Discovery
- IPv6
- NAT
- Tunnels
- ATM and MPLS

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 Autoconfiguration

- Serverless ("Stateless"). No manual config at all.
  - Only configures addressing items, NOT other host things
    - If you want that, use DHCP.
- Link-local address
  - 1111 1110 10 :: 64 bit interface ID (usually from Ethernet addr)
    - (fe80::/64 prefix)
  - Uniqueness test ("anyone using this address?")
  - Router contact (solicit, or wait for announcement)
    - Contains globally unique prefix
    - Usually: Concatenate this prefix with local ID → globally unique IPv6 ID
- DHCP took some of the wind out of this, but nice for "zero-conf" (many OSes now do this for both v4 and v6)

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

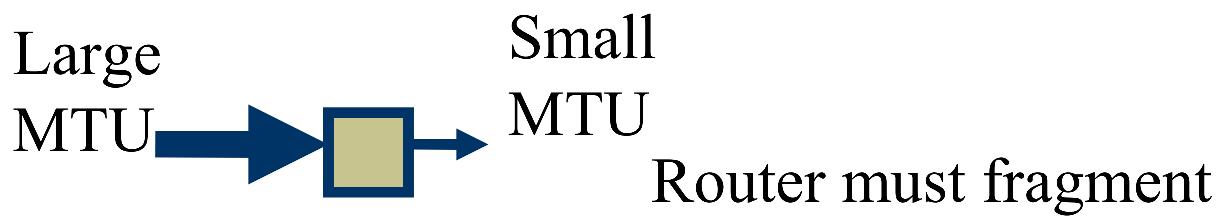
- Different options handling
- IPv4 options: Variable length header field. 32 different options.
  - Rarely used
  - No development / many hosts/routers do not support
    - Worse than useless: Packets w/options often even get dropped!
  - Processed in “slow path”.
- IPv6 options: “Next header” pointer
  - 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 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 Fragmentation Cleanup

- IPv4:



- 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.

- Alternative mechanisms:
  - Dual stack operation: IP v6 nodes support both address types
  - Translation:
    - Use form of NAT to connect to the outside world
    - NAT must not only translate addresses but also translate between IPv4 and IPv6 protocols
  - **Tunneling**: tunnel IP v6 packets through IP v4 clouds

Outline

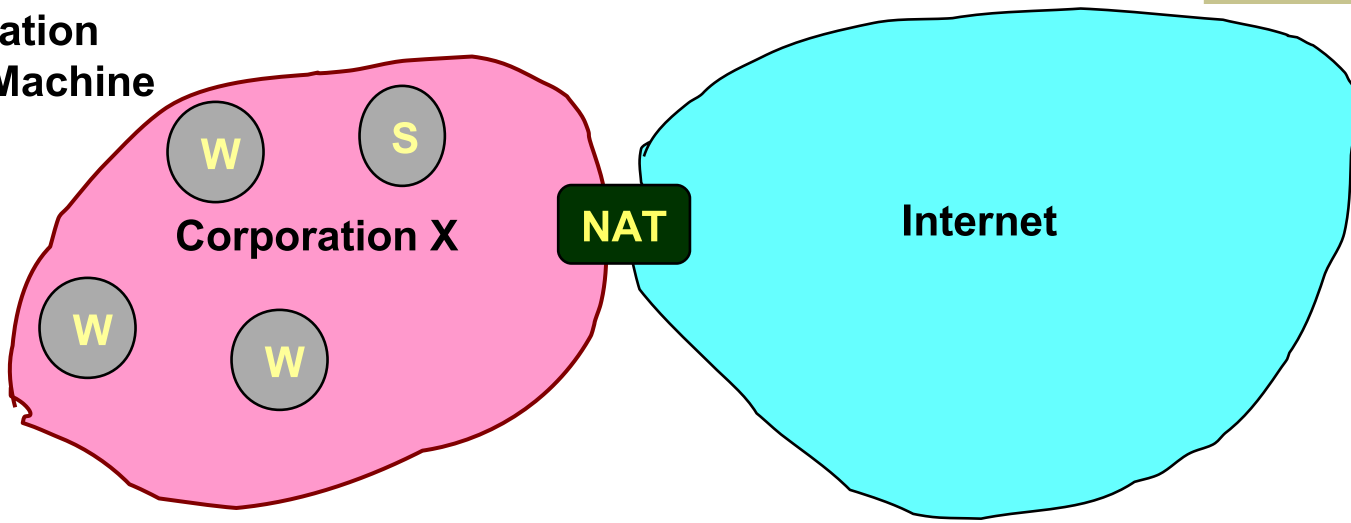
- ICMP/MTU Discovery
- IPv6
- NAT
- Tunnels
- ATM and MPLS

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

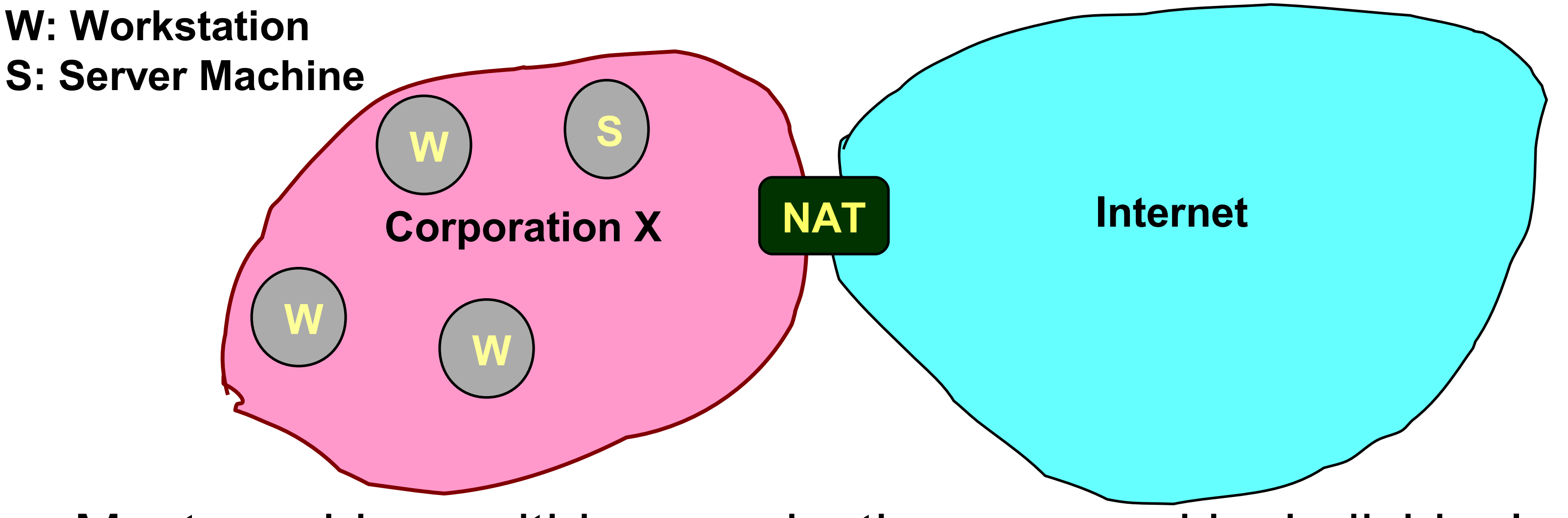
Private Network Accessing Public Internet

W: Workstation
S: Server Machine



- 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

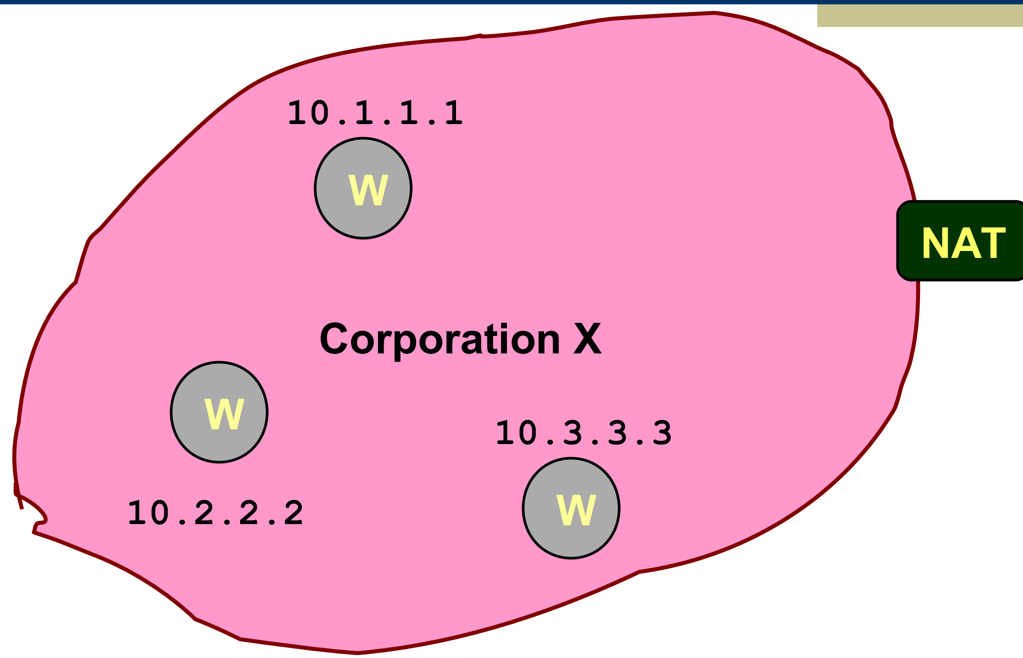


- Most machines within organization are used by individuals
  - “Workstations”
  - For most applications, act as clients
- Small number of machines act as servers for entire organization
  - E.g., mail server
  - All traffic to outside passes through firewall

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

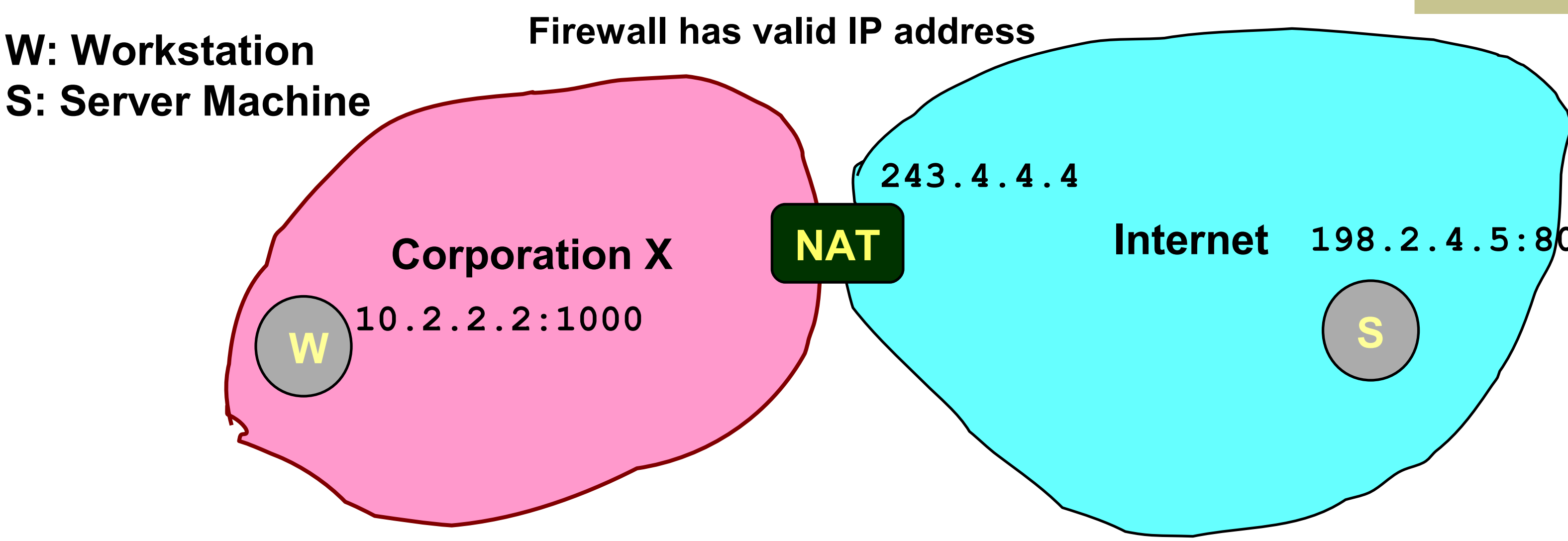
Network Address Translation (NAT)

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

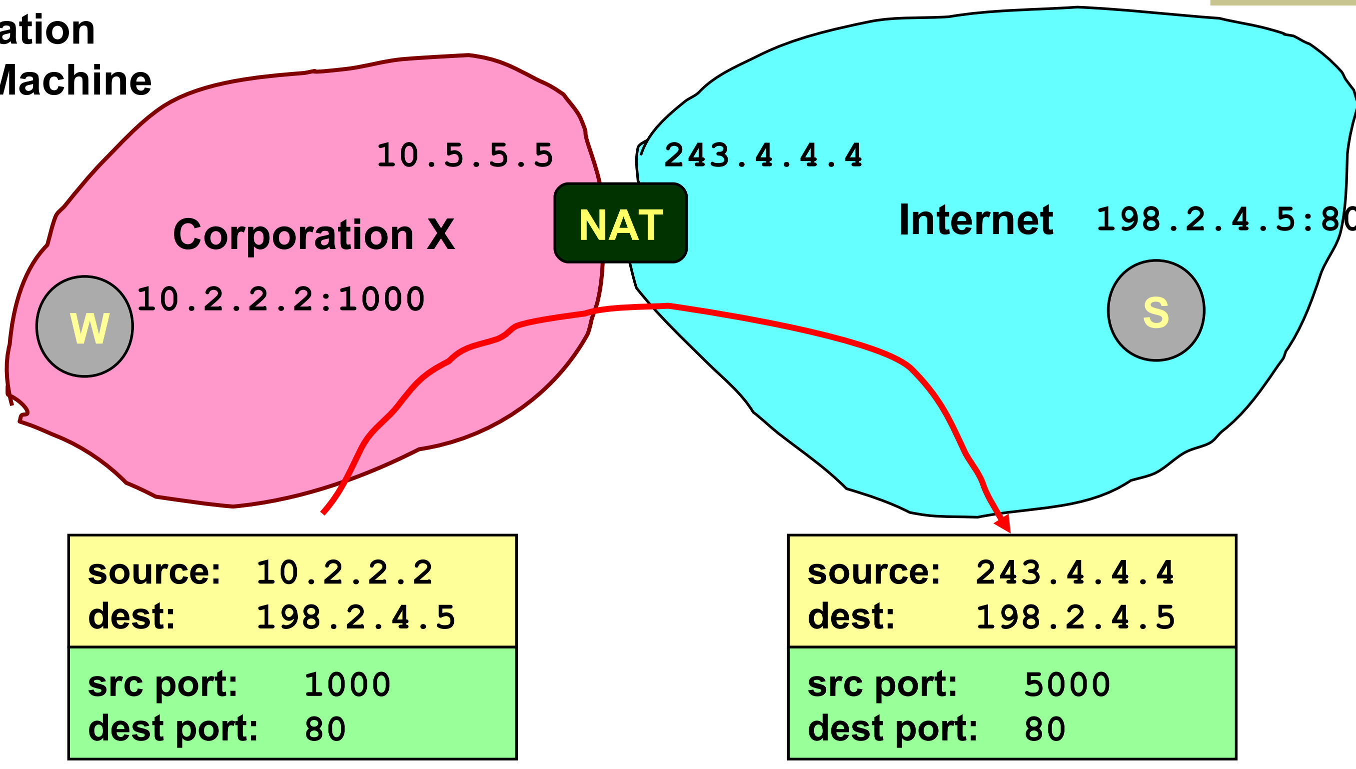


- Client 10.2.2.2 wants to connect to server 198.2.4.5:80
  - OS assigns ephemeral port (1000)
- Connection request intercepted by firewall
  - Maps client to port of firewall (5000)
  - Creates NAT table entry

Int Addr	Int Port	NAT Port
10.2.2.2	1000	5000

NAT: Client Request

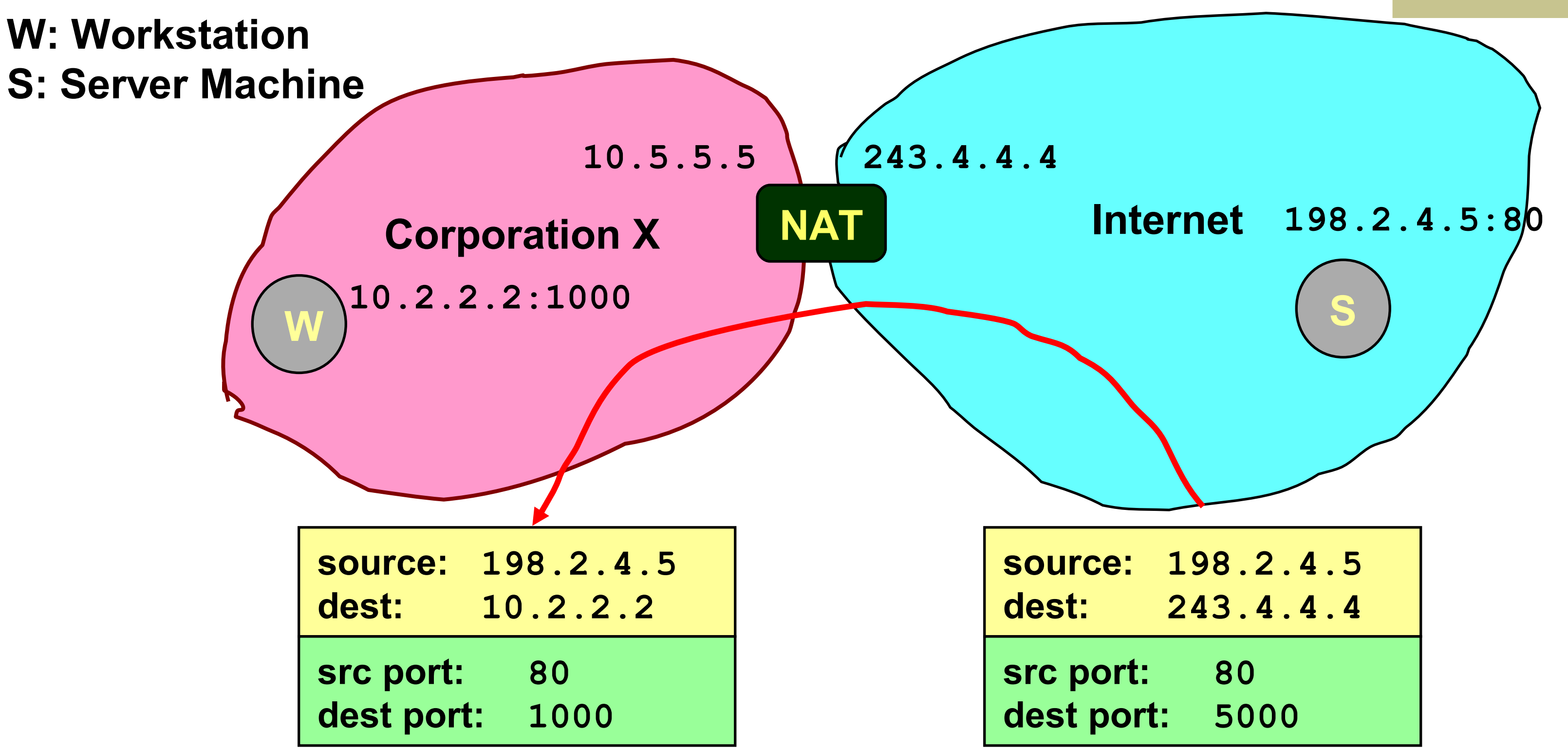
W: Workstation
S: Server Machine



Int Addr	Int Port	NAT Port
10.2.2.2	1000	5000

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

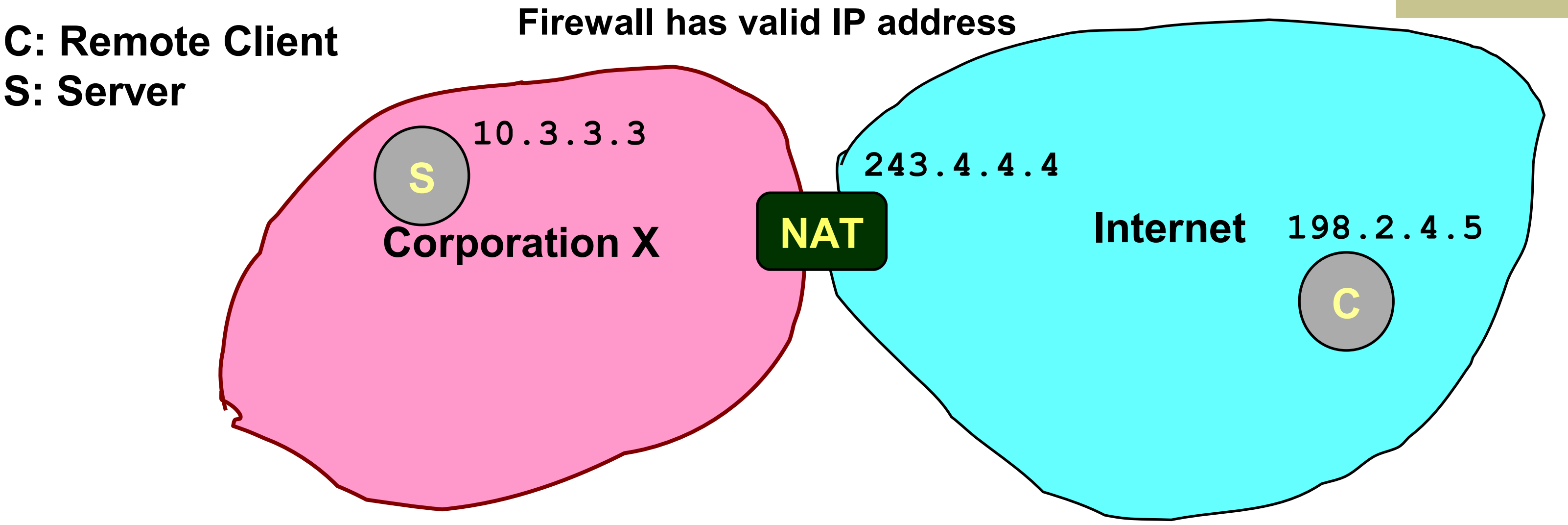
NAT: Server Response



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



- 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?

NAT Considerations

- NAT has to be consistent during a session.
  - Set up mapping at the beginning of a session and maintain it during the session
    - Recall 2nd level goal 1 of Internet: Continue despite loss of networks or gateways
    - What happens if your NAT reboots?
  - Recycle the mapping that the end of the session
    - May be hard to detect
- NAT only works for certain applications.
  - Some applications (e.g. ftp) pass IP information in payload
  - Need application level gateways to do a matching translation
  - Breaks a lot of applications.
    - Example: Let's look at FTP
- NAT is loved and hated

\- Breaks many apps (FTP)

\- Inhibits deployment of new applications like p2p (but so do firewalls!)

\+ Little NAT boxes make home networking simple.

\+ Saves addresses. Makes allocation simple.

Important Concepts

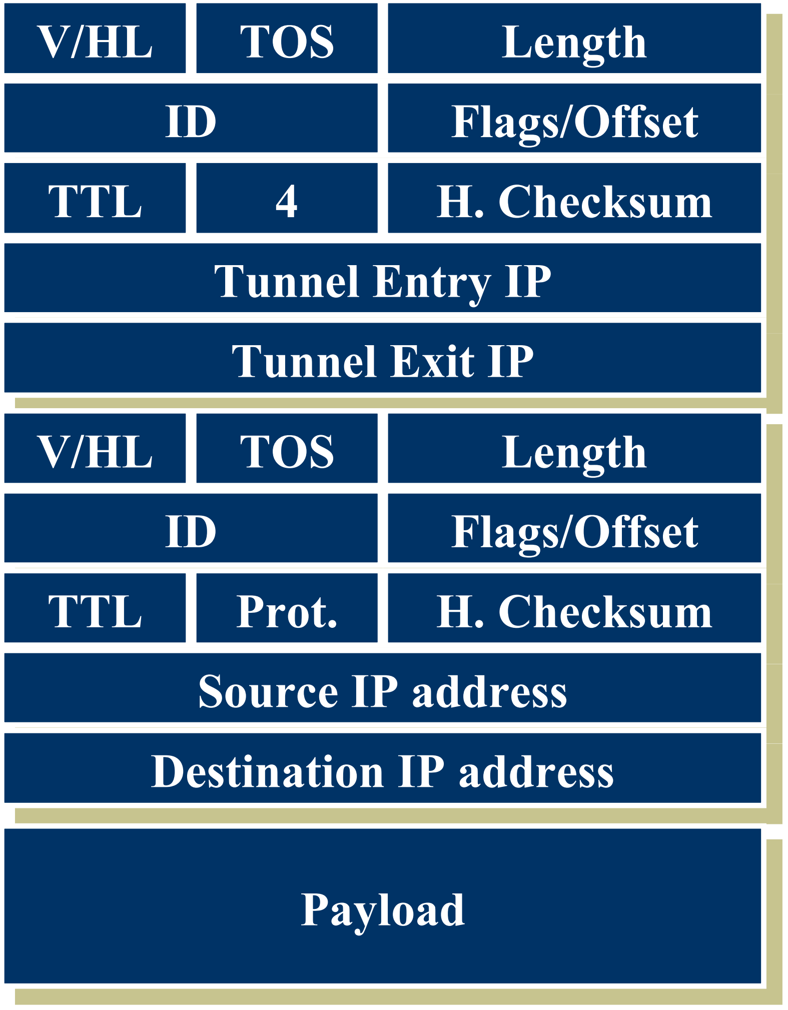
- Base-level protocol (IP) provides minimal service level
  - Allows highly decentralized implementation
  - Each step involves determining next hop
  - Most of the work at the endpoints
- ICMP provides low-level error reporting
- IP forwarding → global addressing, alternatives, lookup tables
- IP addressing → hierarchical, CIDR
- IP service → best effort, simplicity of routers
- IP packets → header fields, fragmentation, ICMP

Outline

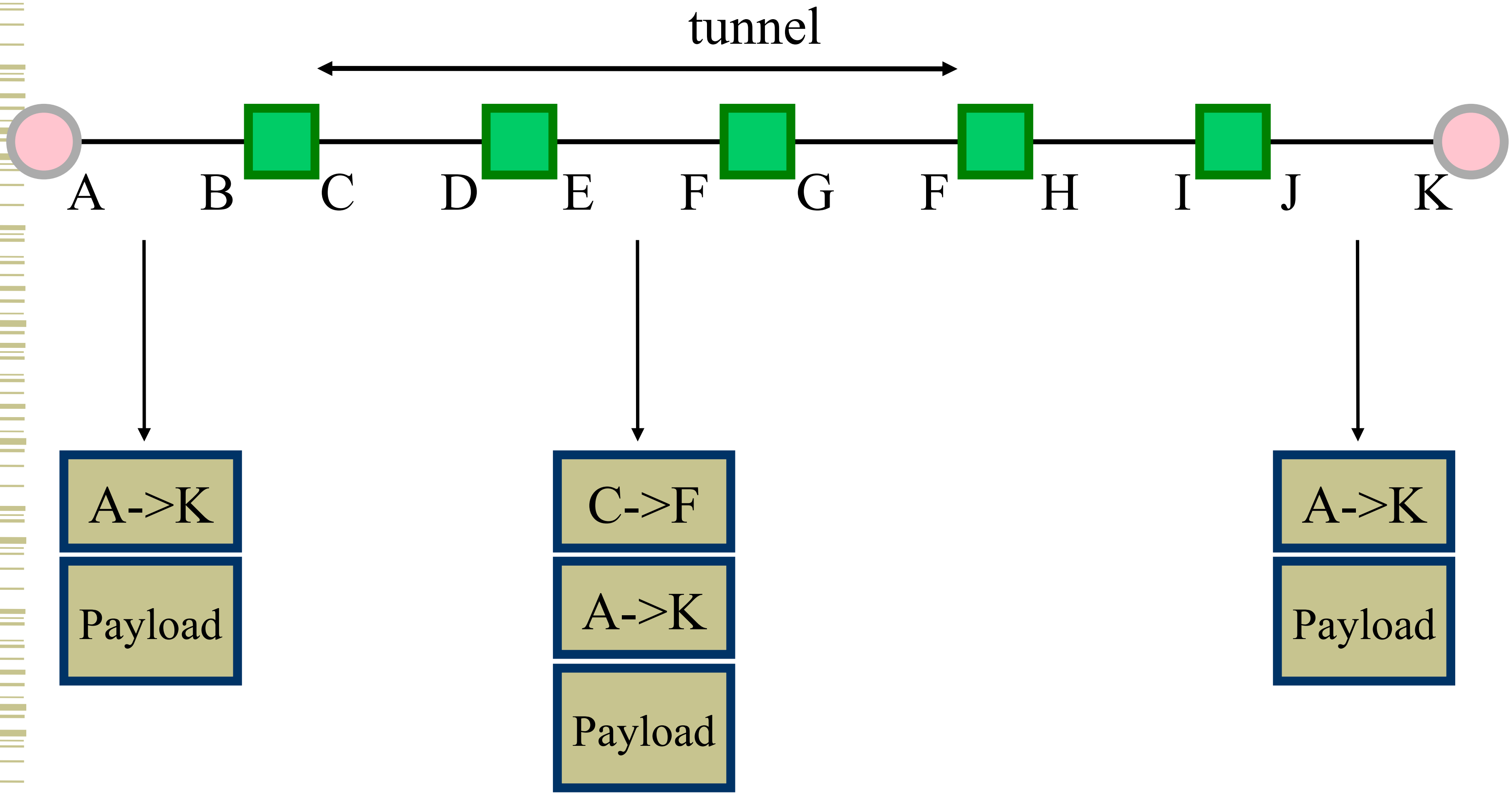
- ICMP/MTU Discovery
- IPv6
- NAT
- Tunnels
- ATM and MPLS

IP-in-IP Tunneling

- Described in RFC 1993.
- IP source and destination address identify tunnel endpoints.
- Protocol id = 4.
  - IP
- Several fields are copies of the inner-IP header.
  - TOS, some flags, ..
- Inner header is not modified, except for decrementing TTL.



Tunneling Example



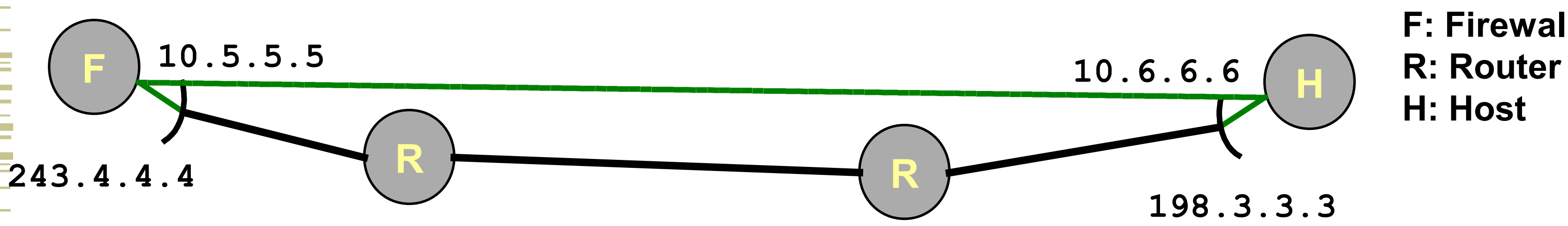
Tunneling Considerations

- Performance.
  - Tunneling adds (of course) processing overhead
  - Tunneling increases the packet length, which may cause fragmentation
    - BIG hit in performance in most systems
    - Tunneling in effect reduces the MTU of the path, but end-points often do not know this
- Security issues.
  - Should verify both inner and outer header
  - E.g., one-time flaw: send an ip-in-ip packet to a host. Inner packet claimed to come from “trusted” host. Bypass firewalls.

Tunneling Applications

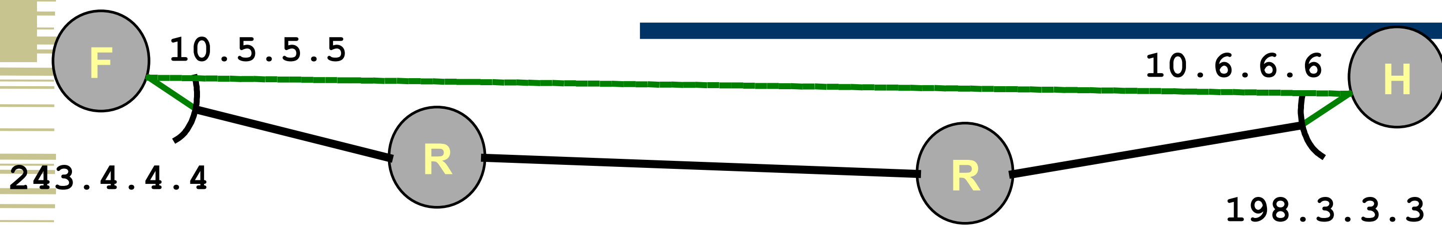
- Virtual private networks.
  - Connect subnets of a corporation using IP tunnels
  - Often combined with IP Sec
    - (Amusing note: IPSec itself an IPv6 spinoff that was backported into IPv4)
- Support for new or unusual protocols.
  - Routers that support the protocols use tunnels to “bypass” routers that do not support it
  - E.g. multicast
- Force packets to follow non-standard routes.
  - Routing is based on outer-header
  - E.g. mobile IP

Supporting VPN by Tunneling

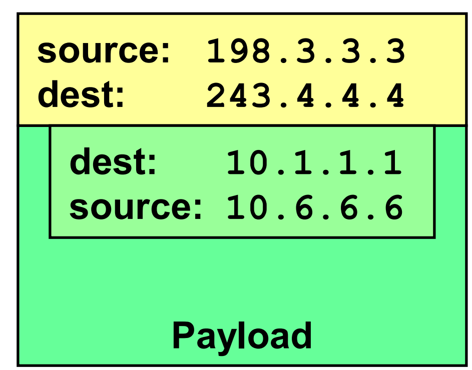


- Concept
  - Appears as if two hosts connected directly
- Usage in VPN
  - Create tunnel between road warrior & firewall
  - Remote host appears to have direct connection to internal network

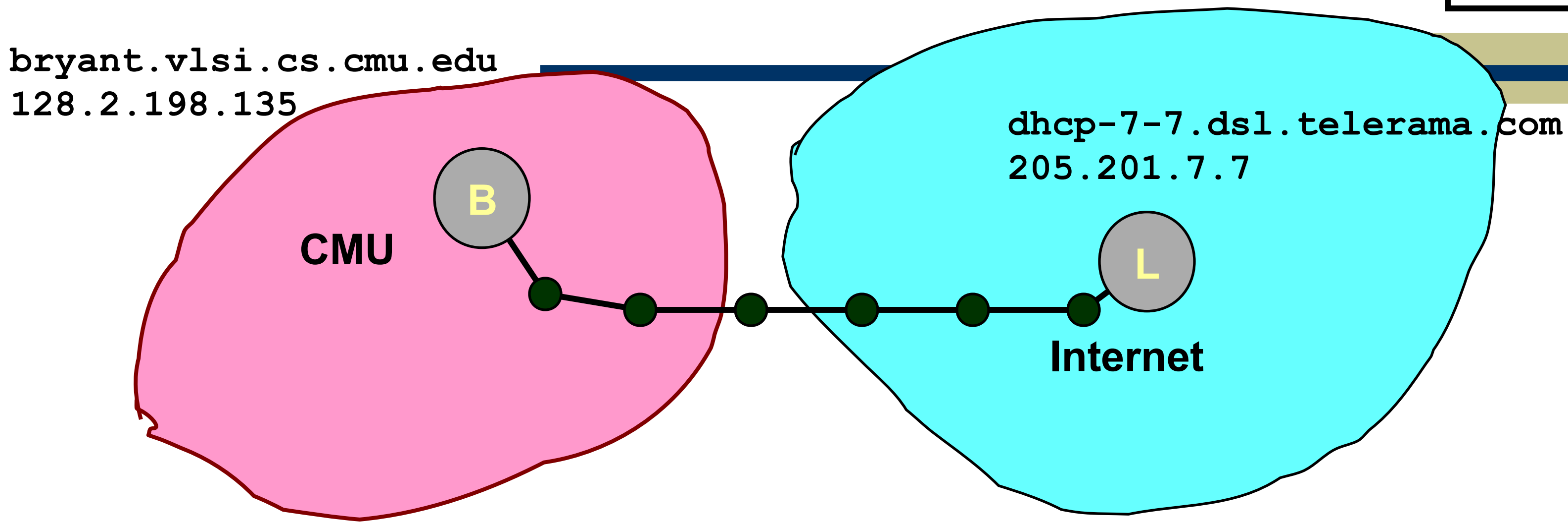
Implementing Tunneling



- 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

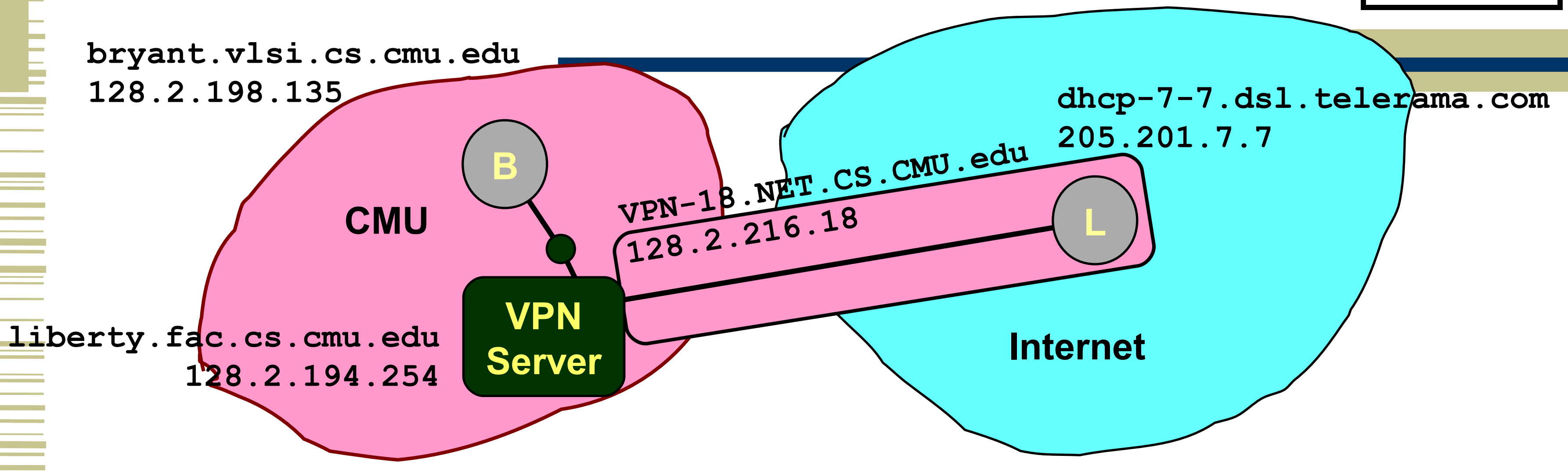


- Operation
  - Running echo server on CMU machine 128.2.198.135
  - Run echo client on laptop connected through DSL from non-CMU ISP

- 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)

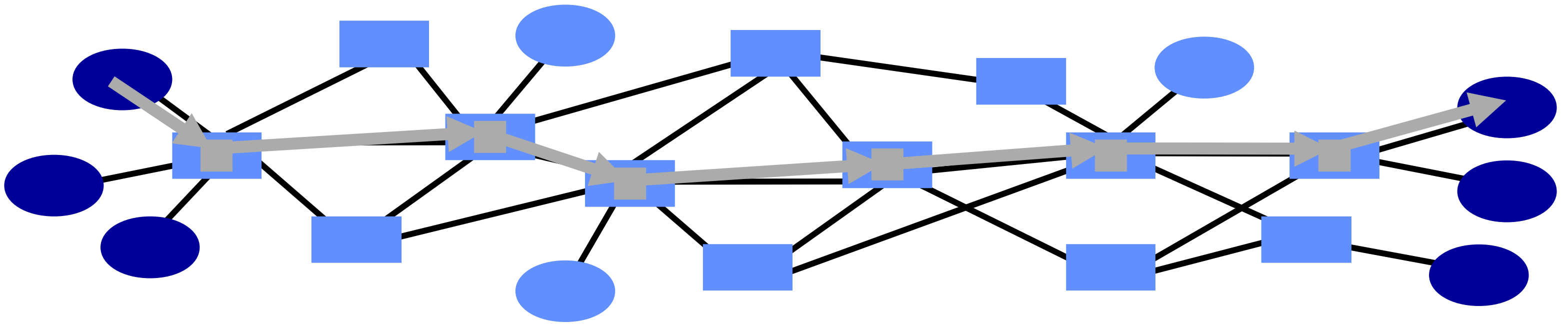
- Effect
  - For other hosts in CMU, packets appear to originate from within CMU

Outline

- ICMP/MTU Discovery
- IPv6
- NAT
- Tunnels
- ATM and MPLS

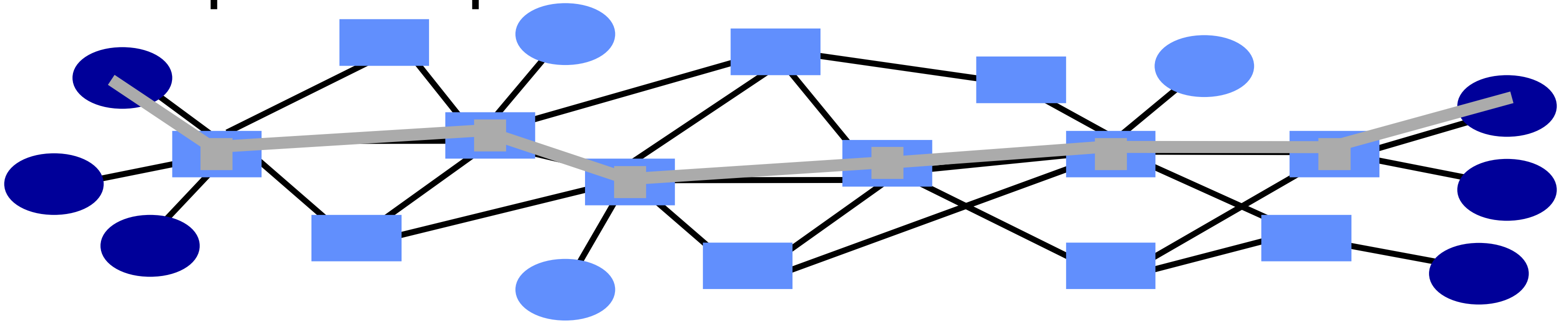
Packet Switching

- Source sends information as self-contained packets that have an address.
  - Source may have to break up single message in multiple
- Each packet travels independently to the destination host.
  - Routers and switches use the address in the packet to determine how to forward the packets
- Destination recreates the message.
- Analogy: a letter in surface mail.



Circuit Switching

- Source first establishes a connection (circuit) to the destination.
  - Each router or switch along the way may reserve some bandwidth for the data flow
- Source sends the data over the circuit.
  - No need to include the destination address with the data since the routers know the path
- The connection is torn down.
- Example: telephone network.

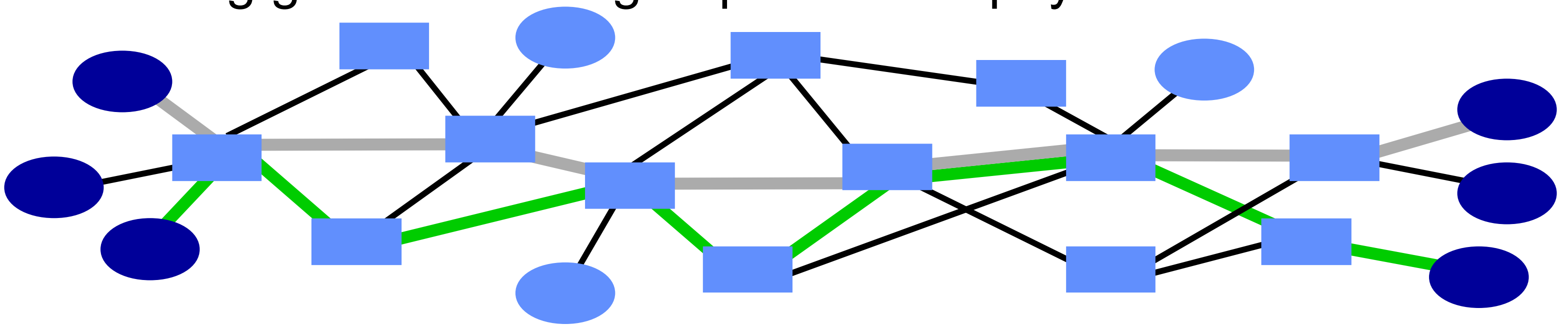


Circuit Switching Discussion

- Consider traditional circuits: on each hop, the circuit has a *dedicated* wire or slice of bandwidth.
  - Physical connection - clearly no need to include addresses with the data
- Advantages, relative to packet switching:
  - Implies guaranteed bandwidth, predictable performance
  - Simple switch design: only remembers connection information, no longest-prefix destination address look up
- Disadvantages:
  - Inefficient for bursty traffic (wastes bandwidth)
  - Delay associated with establishing a circuit
- Can we get the advantages without (all) the disadvantages?

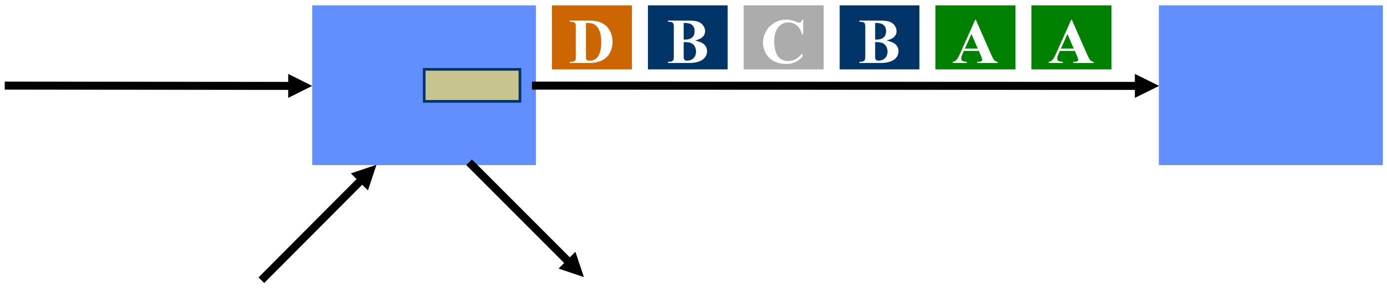
Virtual Circuits

- Each wire carries many “virtual” circuits.
  - Forwarding based on virtual circuit (VC) identifier
    - IP header: src, dst, etc.
    - Virtual circuit header: just a small index number
  - A path through the network is determined for each VC when the VC is established
  - Use statistical multiplexing for efficiency
- Can support wide range of quality of service.
  - No guarantees: best effort service
  - Weak guarantees: delay < 300 msec, …
  - Strong guarantees: e.g. equivalent of physical circuit



Packet Switching and Virtual Circuits: Similarities

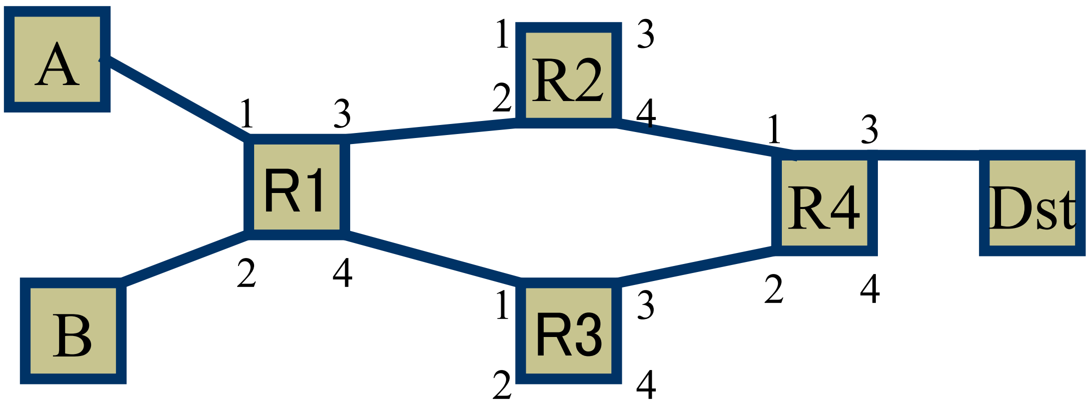
- "Store and forward" communication based on an address.
  - Address is either the destination address or a VC identifier
- Must have buffer space to temporarily store packets.
  - E.g. multiple packets for some destination arrive simultaneously
- Multiplexing on a link is similar to time sharing.
  - No reservations: multiplexing is statistical, i.e. packets are interleaved without a fixed pattern
  - Reservations: some flows are guaranteed to get a certain number of "slots"



Virtual Circuits Versus Packet Switching

- Circuit switching:
  - Uses short connection identifiers to forward packets
  - Switches know about the connections so they can more easily implement features such as quality of service
  - Virtual circuits form basis for traffic engineering: VC identifies long-lived stream of data that can be scheduled
- Packet switching:
  - Use full destination addresses for forwarding packets
  - Can send data right away: no need to establish a connection first
  - Switches are stateless: easier to recover from failures
  - Adding QoS is hard
  - Traffic engineering is hard: too many packets!

Packet switched vs. VC



R1 packet forwarding table:

Dst R2

Different paths to same destination!

(useful for traffic engineering!)

R1 VC table:

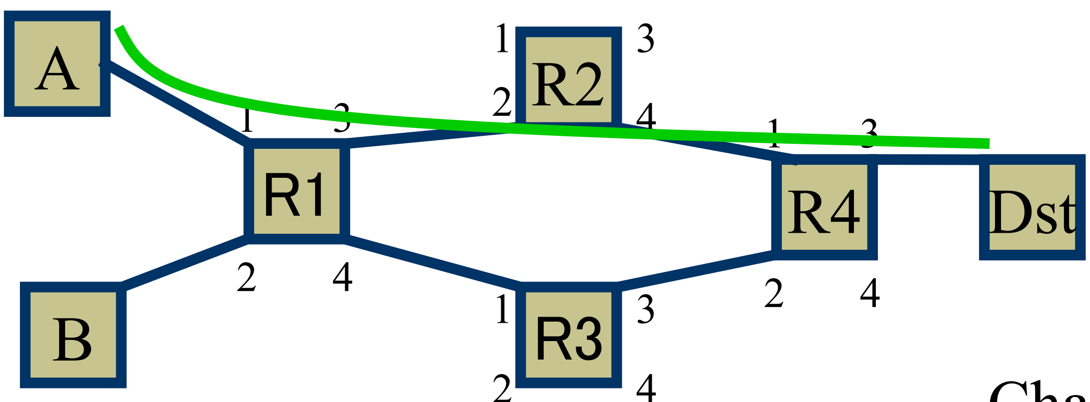
VC 1 R2

VC 2 R3

Virtual Circuit

Payload | VCI

Payload



Challenges:

- How to set up path?
- How to assign IDs??

R1 VC table:

VC 5 R2

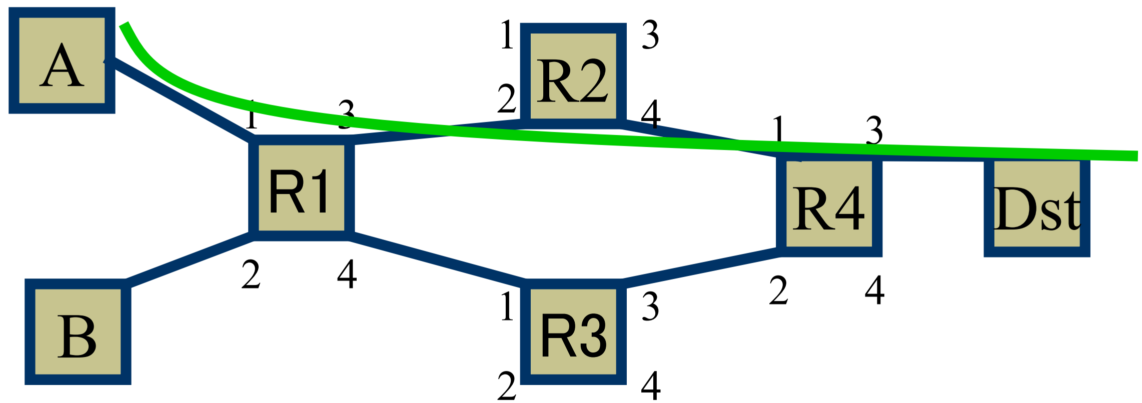
R2 VC table:

VC 5 R4

Connections and Signaling

- Permanent vs. switched virtual connections (PVCs, SVCs)
  - static vs. dynamic. PVCs last “a long time”
    - E.g., connect two bank locations with a PVC
    - SVCs are more like a phone call
  - PVCs administratively configured (but not “manually”)
  - SVCs dynamically set up on a “per-call” basis
- Topology
  - point to point
  - point to multipoint
  - multipoint to multipoint
- Challenges: How to configure these things?
  - What VCI to use?
  - Setting up the path

Virtual Circuit Switching: Label ("tag") Swapping



- Global VC ID allocation -- ICK! Solution: Per-link uniqueness. *Change VCI each hop.*

	Input Port	Input VCI	Output Port	Output VCI
R1:	1	5	3	9
R2:	2	9	4	2
R4:	1	2	3	5

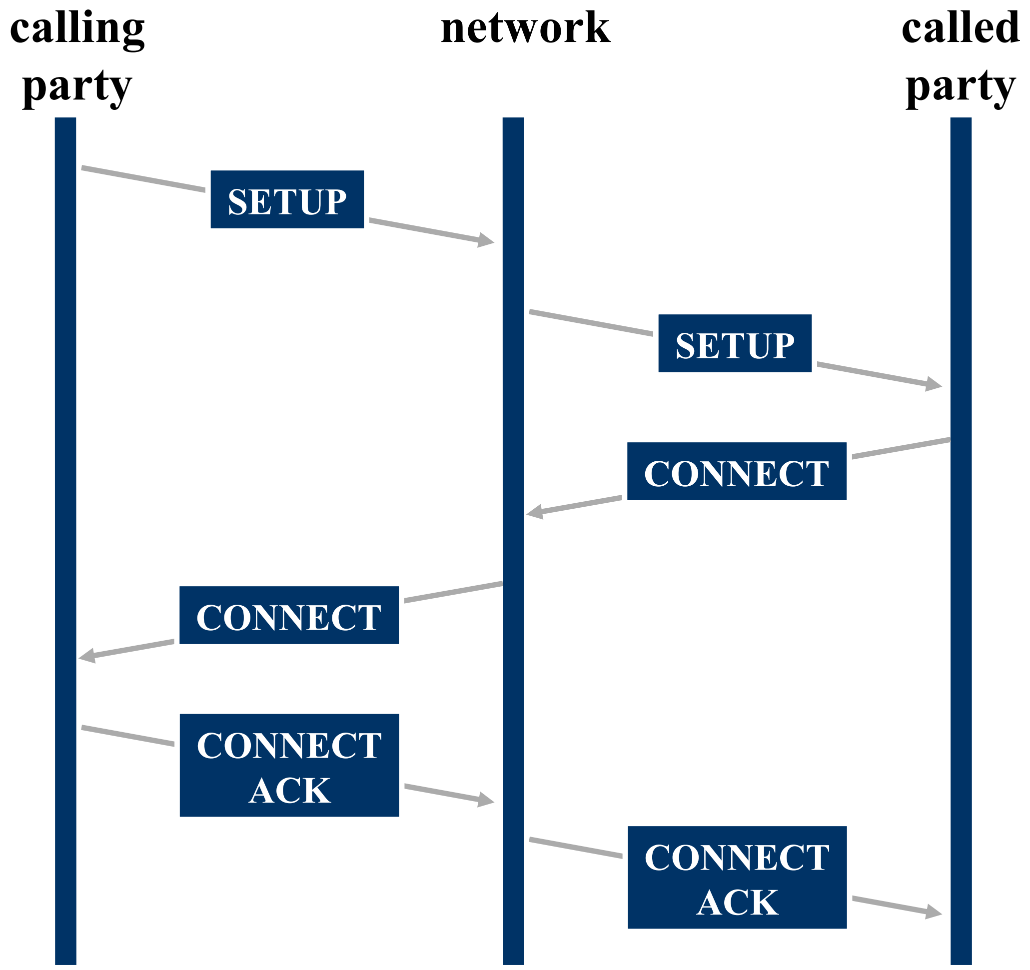
Label (“tag”) Swapping

- Result: Signalling protocol must only find per-link unused VCIs.
  - “Link-local scope”
  - Connection setup can proceed hop-by-hop.
    - Good news for our setup protocols!

PVC connection setup

- Manual?
  - Configure each switch by hand. Ugh.
- Dedicated signaling protocol
  - E.g., what ATM uses
- Piggyback on routing protocols
  - Used in MPLS. E.g., use BGP to set up

SVC Connection Setup



Virtual Circuits In Practice

- ATM: Telco approach
  - Kitchen sink. Based on voice, support file transfer, video, etc., etc.
  - Intended as IP replacement. That didn't happen. :)
  - Today: Underlying network protocol in many telco networks. E.g., DSL speaks ATM. IP over ATM in some cases.
- MPLS: The "IP Heads" answer to ATM
  - Stole good ideas from ATM
  - Integrates well with IP
  - Today: Used inside some networks to provide VPN support, traffic engineering, simplify core.
- Other nets just run IP.
- Older tech: Frame Relay
  - Only provided PVCs. Used for quasi-dedicated 56k/T1 links between offices, etc. Slower, less flexible than ATM.

Asynchronous Transfer Mode: ATM

- Connection-oriented, packet-switched
  - (e.g., virtual circuits).
- Telco-driven. Goals:
  - Handle voice, data, multimedia
  - Support both PVCs and SVCs
  - Replace IP. (didn't happen…)
- Important feature: Cell switching

Cell Switching

- Small, fixed-size cells

  [Fixed-length data][header]
- Why?
  - Efficiency: All packets the same
    - Easier hardware parallelism, implementation
  - Switching efficiency:
    - Lookups are easy -- table index.
  - Result: Very high cell switching rates.
  - Initial ATM was 155Mbit/s. Ethernet was 10Mbit/s at the same time. (!)
- How do you pick the cell size?

ATM Features

- Fixed size cells (53 bytes).
  - Why 53?
- Virtual circuit technology using hierarchical virtual circuits.
- Support for multiple traffic classes by adaptation layer.
  - E.g. voice channels, data traffic
- Elaborate signaling stack.
  - Backwards compatible with respect to the telephone standards
- Standards defined by ATM Forum.
  - Organization of manufacturers, providers, users

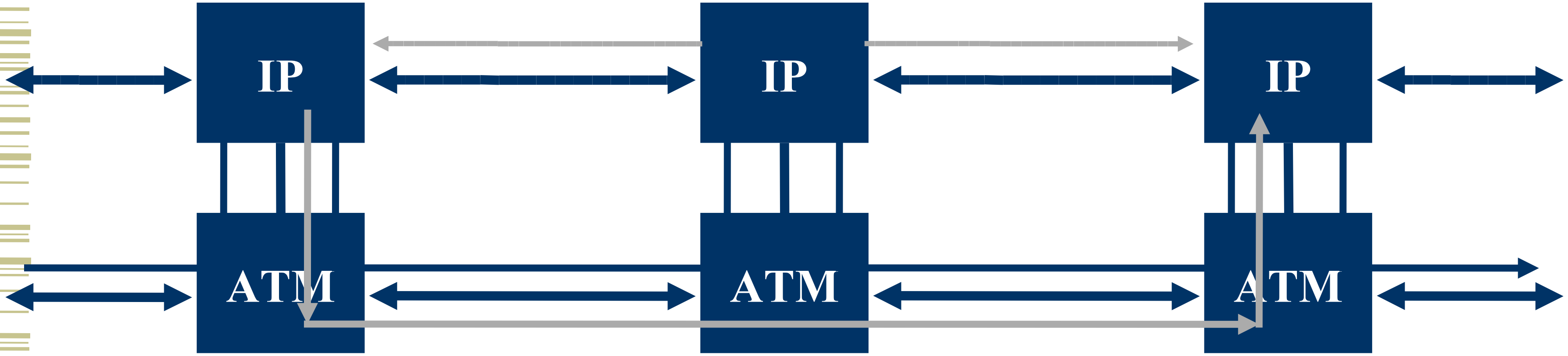
ATM Discussion

- At one point, ATM was viewed as a replacement for IP.
  - Could carry both traditional telephone traffic (CBR circuits) and other traffic (data, VBR)
  - Better than IP, since it supports QoS
- Complex technology.
  - Switching core is fairly simple, but
  - Support for different traffic classes
  - Signaling software is very complex
  - Technology did not match people's experience with IP
    - deploying ATM in LAN is complex (e.g. broadcast)
    - supporting connection-less service model on connection-based technology
  - With IP over ATM, a lot of functionality is replicated
- Currently used as a datalink layer supporting IP.

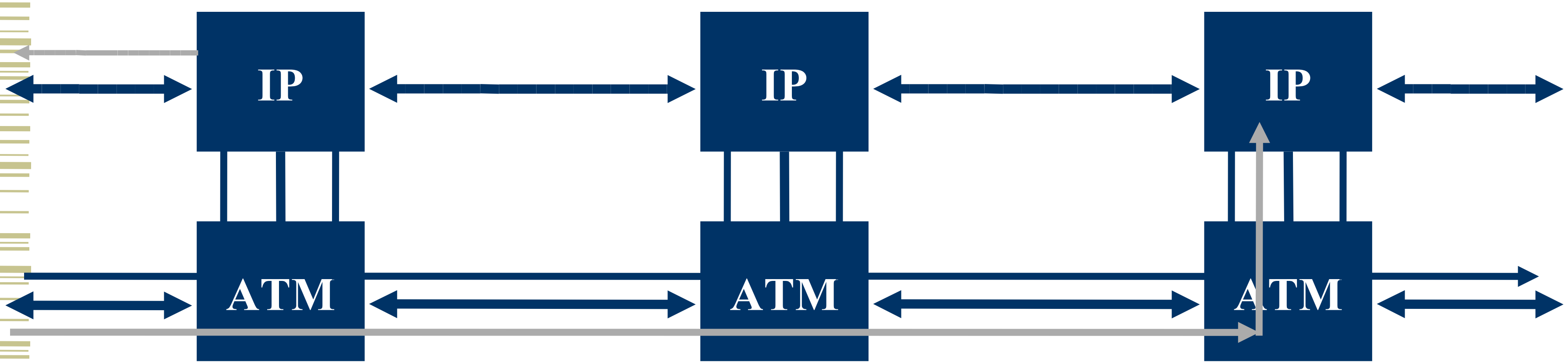
IP Switching

- How to use ATM hardware without the software.
  - ATM switches are very fast data switches
  - software adds overhead, cost
- The idea is to identify flows at the IP level and to create specific VCs to support these flows.
  - flows are identified on the fly by monitoring traffic
  - flow classification can use addresses, protocol types, ...
  - can distinguish based on destination, protocol, QoS
- Once established, data belonging to the flow bypasses level 3 routing.
  - never leaves the ATM switch
- Interoperates fine with “regular” IP routers.
  - detects and collaborates with neighboring IP switches

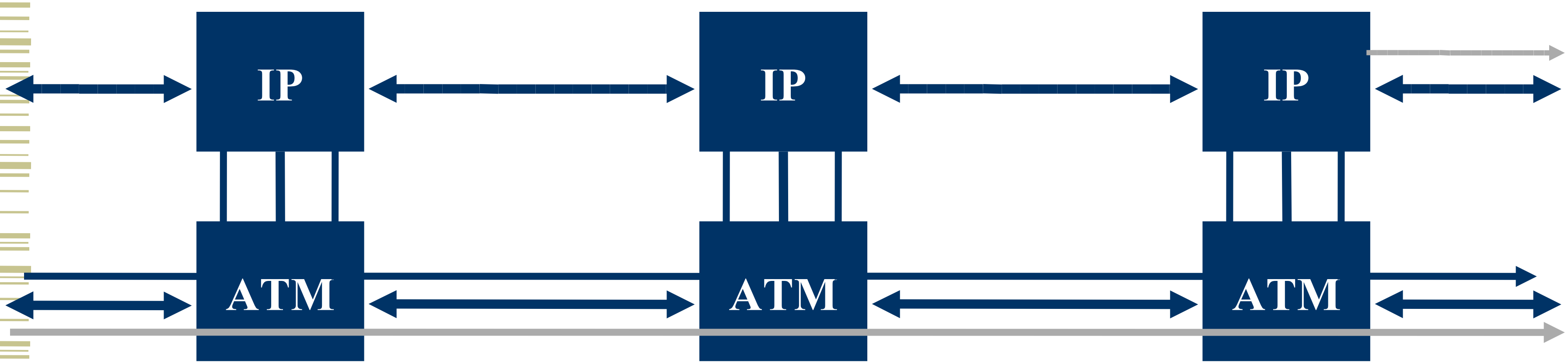
IP Switching Example



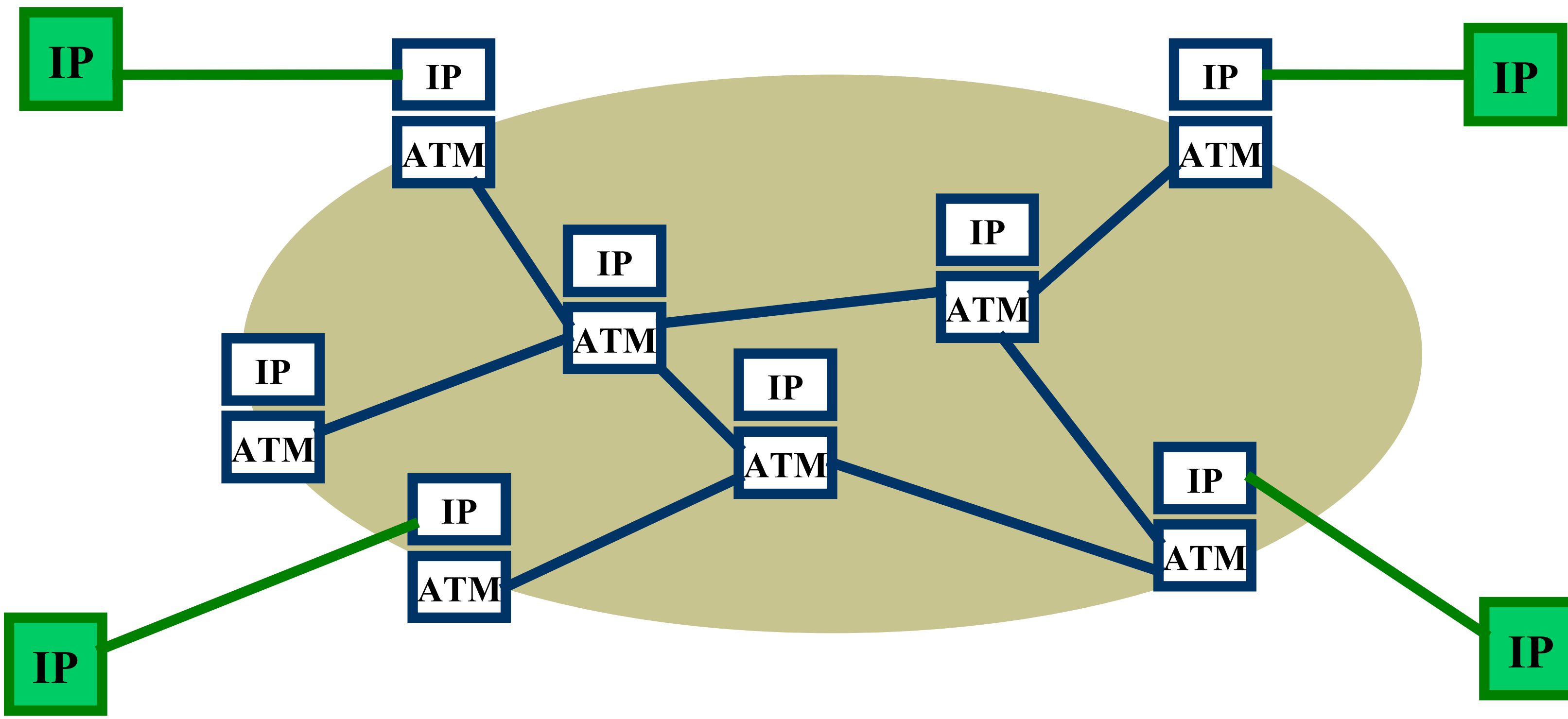
IP Switching Example



IP Switching Example



Another View

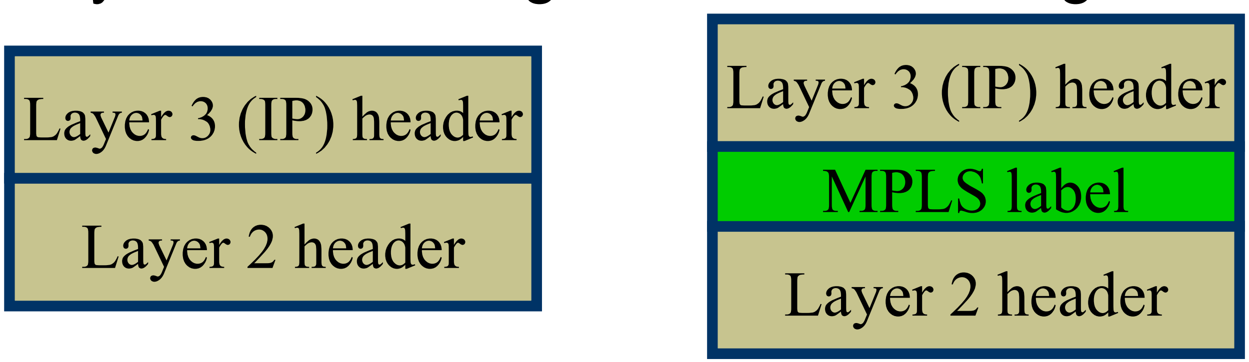


IP Switching Discussion

- IP switching selectively optimizes the forwarding of specific flows.
  - Offloads work from the IP router, so for a given size router, a less powerful forwarding engine can be used
  - Can fall back on traditional IP forwarding if there are failures
- IP switching couples a router with an ATM switching using the GSMP protocol.
  - General Switch Management Protocol
- IP switching can be used for flows with different granularity.
  - Flows belonging to an application .. Organization
  - Controlled by the classifier
- IP switching can be set up quickly, e.g. before a TCP connection starts sending data!

Multi Protocol Label Switching - MPLS

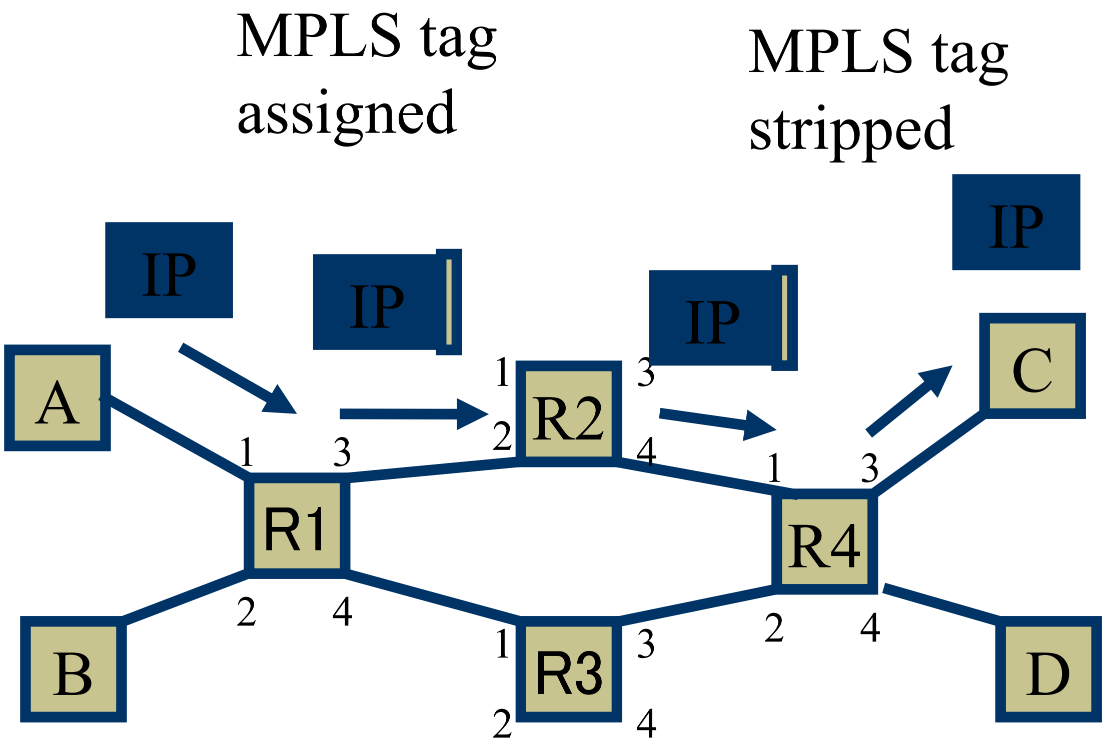
- Selective combination of VCs + IP
  - Today: MPLS useful for traffic engineering, reducing core complexity, and VPNs
- Core idea: Layer 2 carries VC label
  - Could be ATM (which has its own tag)
  - Could be a “shim” on top of Ethernet/etc.:
  - Existing routers could act as MPLS switches just by examining that shim -- no radical re-design. Gets flexibility benefits, though not cell switching advantages



MPLS + IP

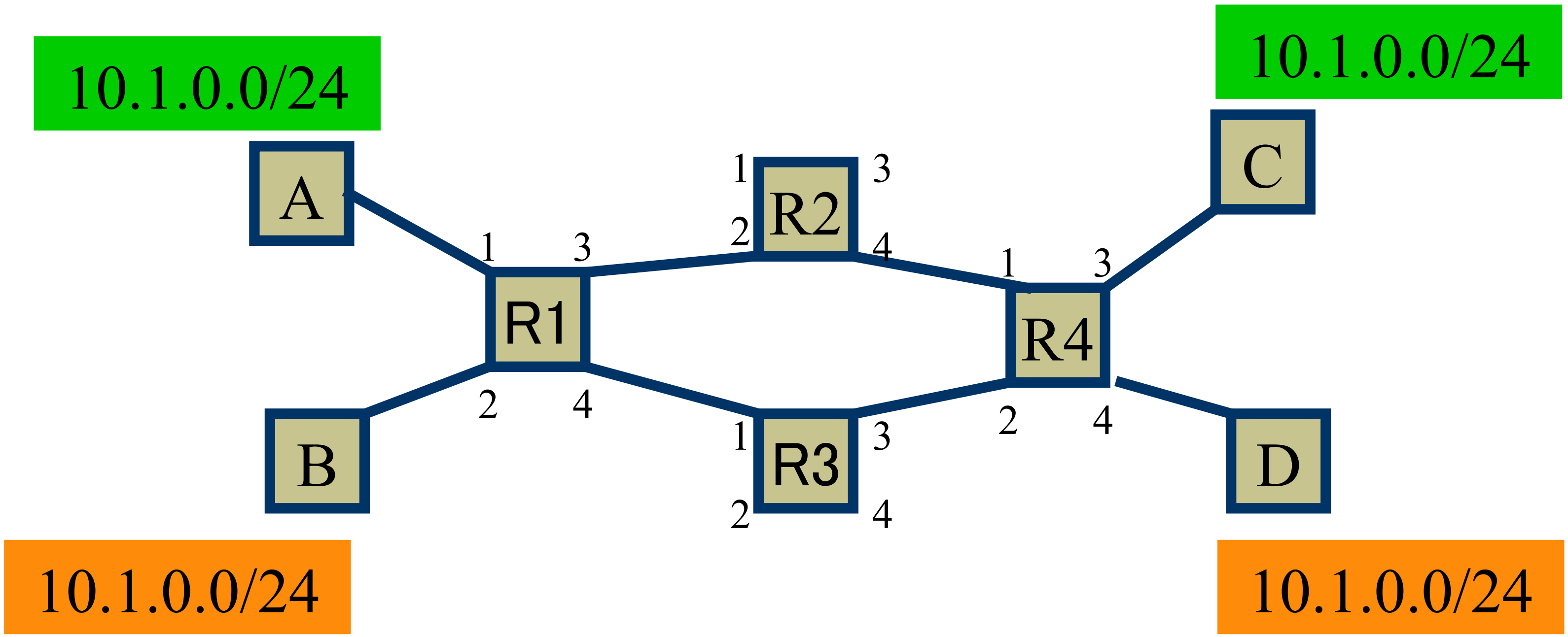
- Map packet onto Forward Equivalence Class (FEC)
  - Simple case: longest prefix match of destination address
  - More complex if QoS of policy routing is used
- In MPLS, a label is associated with the packet when it enters the network and forwarding is based on the label in the network core.
  - Label is swapped (as ATM VCIs)
- Potential advantages.
  - Packet forwarding can be faster
  - Routing can be based on ingress router and port
  - Can use more complex routing decisions
  - Can force packets to followed a pinned route

MPLS core, IP interface



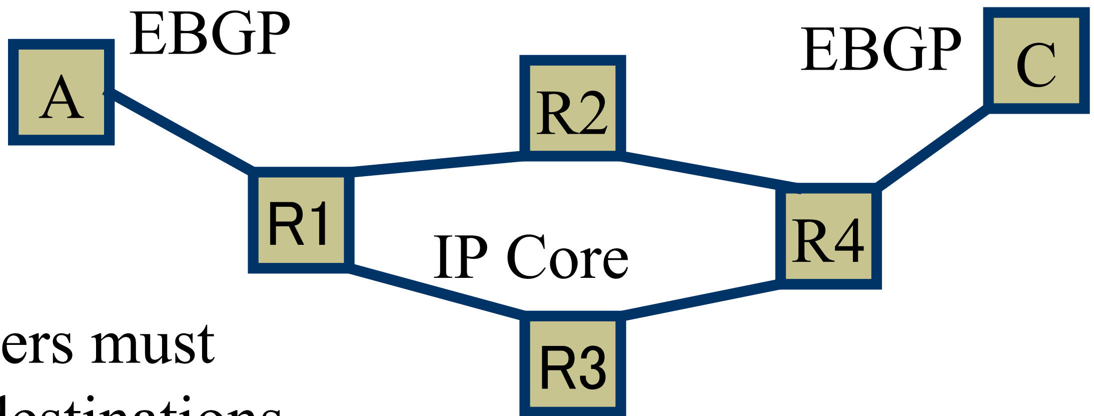
MPLS forwarding in core

MPLS use case #1: VPNs



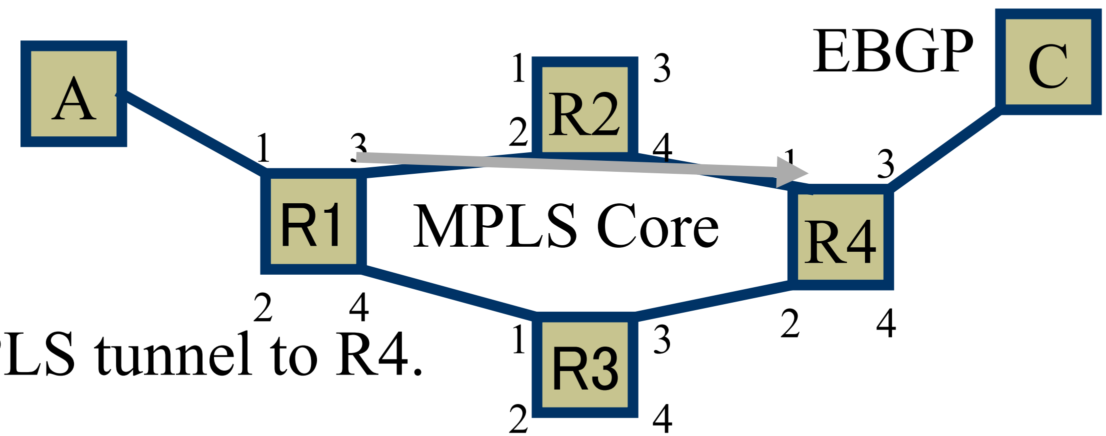
MPLS tags can differentiate green VPN from orange VPN.

MPLS use case #2: Reduced State Core



A-> C pkt

Internal routers must know all C destinations



R1 uses MPLS tunnel to R4.

R1 and R4 know routes, but R2 and R3 don't.

MPLS use case #3: Traffic Engineering

- As discussed earlier -- can pick routes based upon more than just destination
- Used in practice by many ISPs, though certainly not all.

MPLS Mechanisms

- MPLS packet forwarding: implementation of the label is technology specific.
  - Could be ATM VCI or a short extra “MPLS” header
- Supports stacked labels.
  - Operations can be “swap” (normal label swapping), “push” and “pop” labels.
    - VERY flexible! Like creating tunnels, but much simpler -- only adds a small label.

Label	CoS	S	TTL
20	3	1	8

MPLS Discussion

- Original motivation.
  - Fast packet forwarding:
    - Use of ATM hardware
    - Avoid complex "longest prefix" route lookup
    - Limitations of routing table sizes
  - Quality of service
- Currently mostly used for traffic engineering and network management.
  - LSPs can be thought of as "programmable links" that can be set up under software control
  - on top of a simple, static hardware infrastructure

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

Take Home Points

- Costs/benefits/goals of virtual circuits
- Cell switching (ATM)
  - Early high-speed, general-purpose networking
  - Fixed-size small pkts and virtual circuits: Fast hardware
  - Packet size picked for low voice latency and jitter.
- Tag/label swapping
  - Basis for most VCs.
  - Makes label assignment link-local. Understand mechanism.
- MPLS - IP meets virtual circuits; MPLS tunnels used for
  - VPNs,
  - traffic engineering,
  - reduced core routing table sizes

Next Lecture

- A look inside switches and routers
  - What is the “Fabric”?

EXTRA SLIDES

Now for some really bad jokes…

- TTL jokes are short lived
- 10.0.0.1 jokes – best told in private
- IP jokes is that they can arrive out-of-order The most annoying thing about