15-441 Computer Networking

IPv6 and NATs

Peter Steenkiste

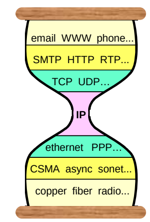
Lecture by David Andersen

Fall 2010

www.cs.cmu.edu/~prs/15-441-F10

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

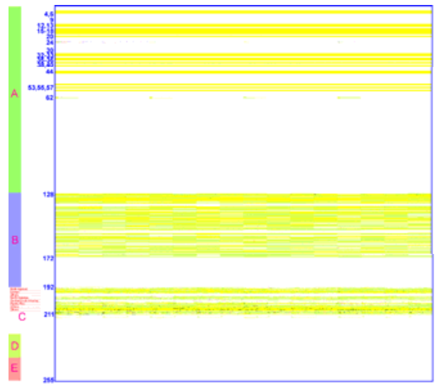
0	4	8	16	19	24–31
ver-sion	**HLen**	TOS	Length		
Ident			**Flags**	Offset	
TTL		Protocol	Checksum		
Source Address					
Destination Address					
Options (if any)					
Data					

Header

IP Address Problem (1991)

- Address space depletion
  - In danger of running out of classes A and B
  - Why?
    - Class C too small for most domains
    - Very few class A – very careful about giving them out
    - Class B – greatest problem
- Class B sparsely populated
  - But people refuse to give it back
- http://tech.slashdot.org/story/10/01/24/2139250/IPv4-Free-Pool-Drops-Below-10-10008-Allocated?art_pos=26

IP Address Utilization ('97)



http://www.caida.org/outreach/resources/learn/ipv4space/ -- broken

5

Outline

- NAT
- IPv6
- Tunneling and VPNs

6

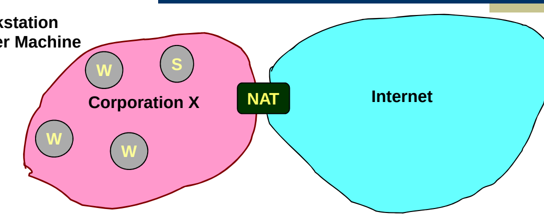
Altering the Addressing Model

- 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

7

Private Network Accessing Public Internet

W: Workstation
S: Server Machine

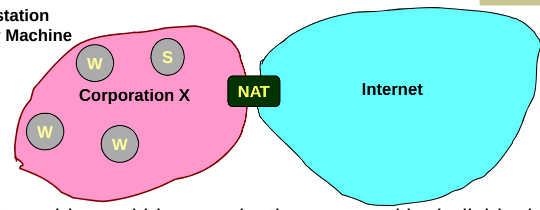


- Don't have enough IP addresses for every host in organization
- Security
  - Don't want every machine in organization known to outside world
  - Want to control or monitor traffic in / out of organization

8

Reducing IP Addresses

W: Workstation
S: Server Machine

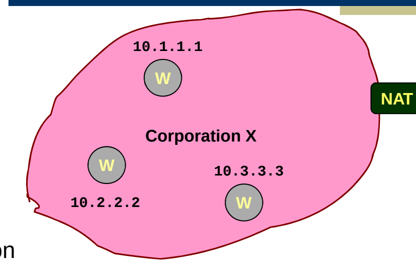


- Most machines within organization are used by individuals
  - “Workstations”
  - For most applications, act as clients
- Small number of machines act as servers for entire organization
  - E.g., mail server
  - All traffic to outside passes through firewall

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

Network Address Translation (NAT)

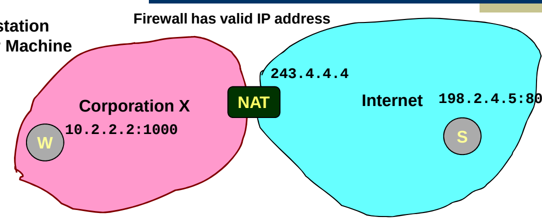
W: Workstation



- 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

W: Workstation
S: Server Machine

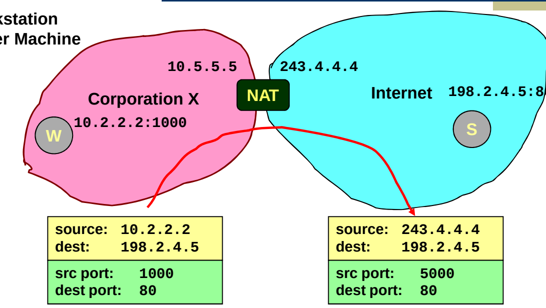


- Client 10.2.2.2 wants to connect to server 198.2.4.5:80
  - OS assigns ephemeral port (1000)
- Connection request intercepted by firewall
  - Maps client to port of firewall (5000)
  - Creates NAT table entry

Int Addr	Int Port	NAT Port
10.2.2.2	1000	5000

NAT: Client Request

W: Workstation
S: Server Machine

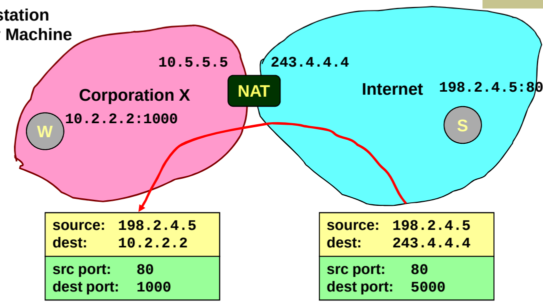


Int Addr	Int Port	NAT Port
10.2.2.2	1000	5000

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

NAT: Server Response

W: Workstation
S: Server Machine

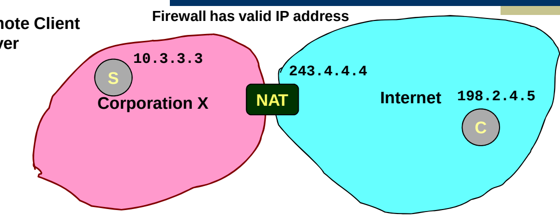


Int Addr	Int Port	NAT Port
10.2.2.2	1000	5000

- 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?

NAT Considerations

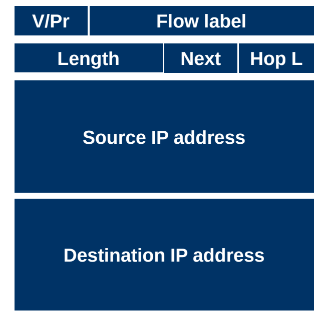
- NAT has to be consistent during a session.
  - Set up mapping at the beginning of a session and maintain it during the session
    - Recall 2nd level goal 1 of Internet: Continue despite loss of networks or gateways
    - What happens if your NAT reboots?
  - Recycle the mapping that the end of the 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.

Outline

- NAT
- Tunneling and VPNs
- IPv6

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.

21

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

22

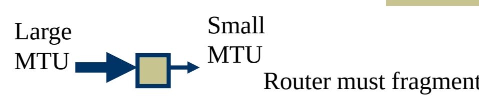
IPv6 Header Cleanup

- 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

23

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.

24

Migration from IPv4 to IPv6

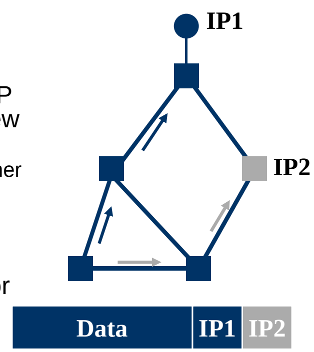
- Interoperability with IP v4 is necessary for gradual deployment.
- Alternative mechanisms:
  - Dual stack operation: IP v6 nodes support both address types
  - Translation:
    - Use form of NAT to connect to the outside world
    - NAT must not only translate addresses but also translate between IPv4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

Outline

- NAT
- IPv6
- Tunneling and VPNs

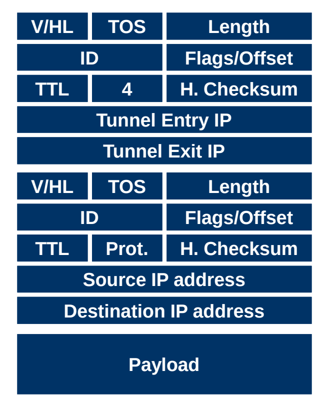
Tunneling

- Force a packet to go to a specific point in the network.
  - Path taken is different from the regular routing
- Achieved by adding an extra IP header to the packet with a new destination address.
  - Similar to putting a letter in another envelope
  - preferable to using IP source routing option
- Used increasingly to deal with special routing requirements or new features.
  - Mobile IP,..
  - Multicast, IPv6, research, ..

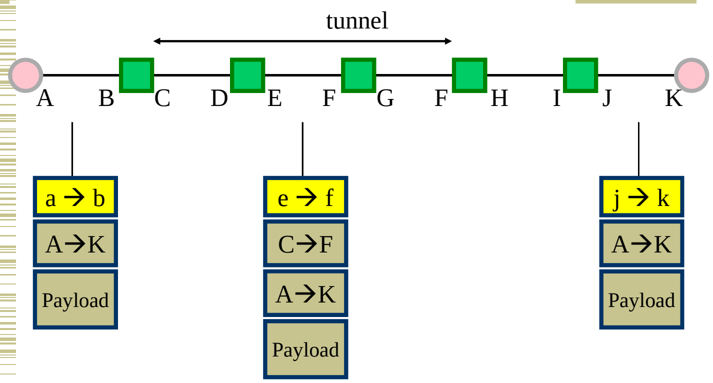


IP-in-IP Tunneling

- 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



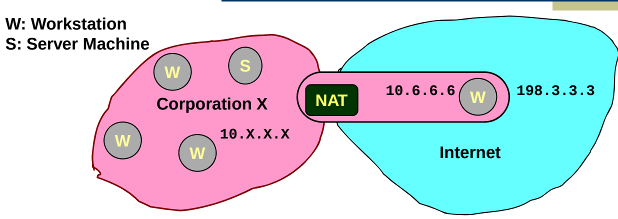
29

Tunneling Applications

- Virtual private networks.
  - Connect subnets of a corporation using IP tunnels
  - Often combined with IP Sec
- 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

30

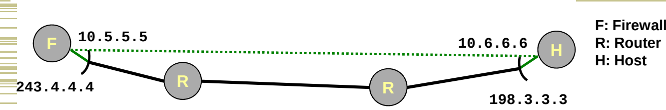
Extending Private Network



- 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

31

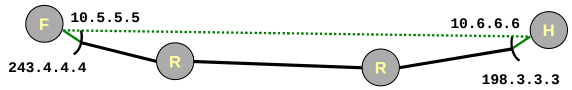
Supporting VPN by Tunneling



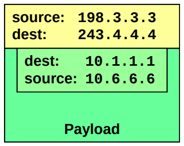
- Concept
  - Appears as if two hosts connected directly
- Usage in VPN
  - Create tunnel between road warrior & firewall
  - Remote host appears to have direct connection to internal network

32

Supporting VPN by Tunneling

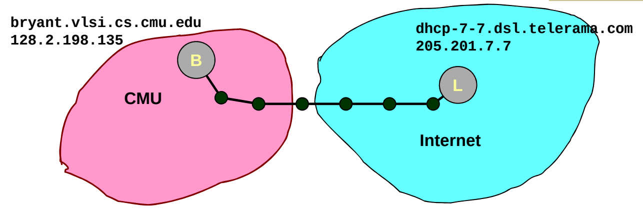


- Host creates packet for internal node 10.1.1.1
- Entering Tunnel
  - Add extra IP header directed to firewall (243.4.4.4)
  - Original header becomes part of payload
  - Possible to encrypt it
- Exiting Tunnel
  - Firewall receives packet
  - Strips off header
  - Sends through internal network to destination



33

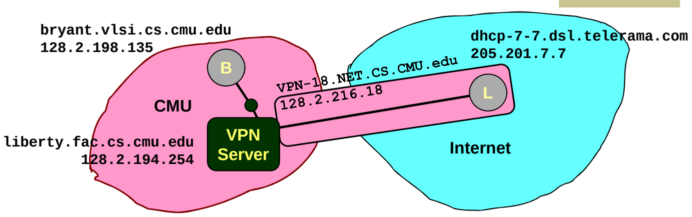
CMU CS VPN Example



- 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)`

34

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

35

Overlay Networks

- A network “on top of the network”.
  - E.g., initial Internet deployment
    - Internet routers connected via phone lines
      - An overlay on the phone network
  - Tunnels between nodes on a current network
- Examples:
  - The IPv6 “6bone”, the multicast “Mbone” (“multicast backbone”).
- But not limited to IP-layer protocols…
  - Can do some pretty cool stuff:

36

Overlay Networks 2

- Application-layer Overlays
  - Application Layer multicast
    - Transmit data stream to multiple recipients
  - Peer-to-Peer networks
    - Route queries (Gnutella search for "britney spars")
    - Route answers (Bittorrent, etc. -- project 2)
  - Anonymizing overlays
    - Route data through lots of peers to hide source
      - (google for "Tor" "anonymous")
  - Improved routing
    - Detect and route around failures faster than the underlying network does.
- Overlays provide a way to build interesting services / ideas without changing the (huge, hard to change) IP infrastructure.
- Design Q: When are overlays good?
  - Functionality between small(er) group of people w/out requiring global state/changes/etc.

37

Important Concepts

- 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
  - Cleanup of various v4 flaws
  - Larger addresses

38

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.

39