15-441/641: Computer Networks
The Internet Protocol

Fall 2019
Profs **Peter Steenkiste** & Justine Sherry

https://computer-networks.github.io/fa19/

Carnegie Mellon University

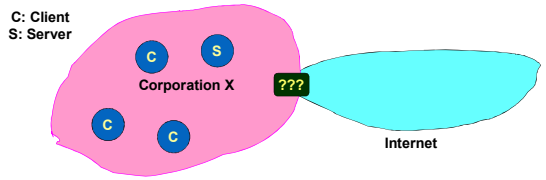
Outline

- The IP protocol
  - IPv4
  - IPv6
- IP in practice
  - Network address translation
  - Tunnels
  - ARP

How have we made it so far with IPv4?

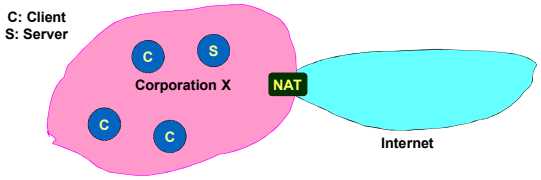
- Original IP Model: Every host has unique IP address
- This has very attractive properties …
  - Any host can communicate with any other host
  - Any host can act as a server: just advertise IP and port number
- … but the system is open – complicates security
  - Any host can attack any other host
  - It is easy to forge packets: just use invalid source address
- … and it places pressure on the address space
  - Every host requires "public" IP address
  - There are at most 4.2 billion IPv4 addresses!

How about a Magic Box?



- Not enough IP addresses for every host in organization
  - Increasingly hard to get large address blocks
- Security
  - Don't want every machine in organization known to outside world
  - Want to control or monitor traffic in / out of organization

Not All Hosts are Equal!

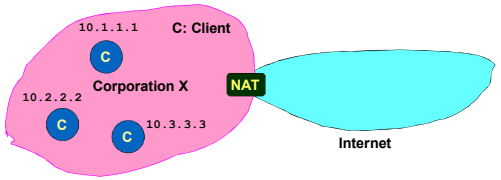


- Most machines within organization are used by individuals
  - They always act as clients
- Only a small number of machines act as servers for the organization
  - E.g., mail server, web, ..
  - All traffic to outside passes through firewall

(Most) machines within organization do not need public IP addresses!

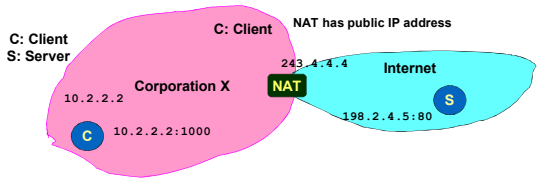
Reducing Address Use: Network Address Translation

- Within organization: assign each host a private IP address
  - IP address blocks 10/8 & 192.168/16 are private
  - Used for routing within the organization by IP protocol
  - Can do subnetting, ..



- The NAT translates between public and private IP addresses as packets travel to/from the Internet
  - It does not let any packets from internal nodes "escape"
  - Outside world does not need to know about internal addresses

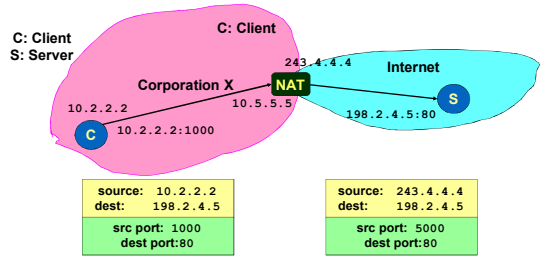
NAT: Opening Client Connection



- 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

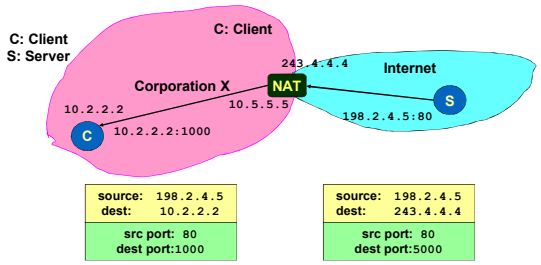
NAT: Client Request



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

Int Addr	Int Port	NAT Port
10.2.2.2	1000	5000

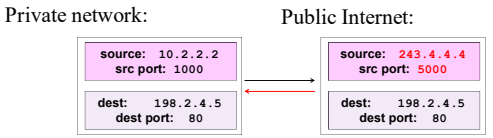
NAT: Server Response



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

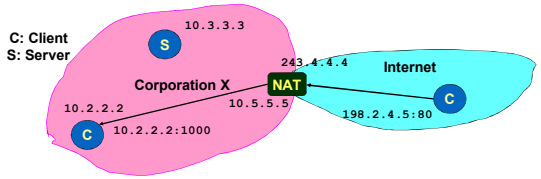
Int Addr	Int Port	NAT Port
10.2.2.2	1000	5000

Client Request Mapping



- NAT manages mapping between two four-tuples
- Mapping must be unique: one to one
- Must respect practical constraints
  - Cannot modify server IP address or port number
  - Client NAT has limited number of IP addresses, often 1
  - Mapping client port numbers is important!
- Mapping must be consistent: the same for all packets in the session

NAT: Enabling Servers



- Use *port mapping* to make servers available
  - Manually configure NAT table to include entry for well-known port
  - External users give address 243.4.4.4:80
  - Requests forwarded to server

Int Addr	Int Port	NAT Port
10.3.3.3	80	80

NAT Benefits

- They significantly reduce the need for public IP addresses
- NATs directly help with security
  - Hides IP addresses used in internal network
  - Basic protection against external attack
    - Does not expose internal structure to outside world
    - Can easily control what packets come in and out of system
    - Can reliably determine whether packet from inside or outside
- And NATs have many additional benefits
  - Easy to change ISP: only NAT box needs to have a public IP address
  - NAT boxes make home networking simple
  - Can be used to map between addresses from different address families, e.g, IPv4 and IPv6

NAT Challenges

- NAT has to be consistent during a session.
  - Mapping (hard state) must be maintained during the session
    - Recall Goal 1 of Internet: Continue despite loss of networks or gateways
  - Recycle the mapping after the end of the session
    - May be hard to detect when a session is really over
- NATs only works for certain applications.
  - Some applications (e.g. ftp) pass IP information in payload - oops
  - Need application level gateways to do a matching translation
- NATs are a problem for peer-peer applications
  - File sharing, multi-player games, … Everyone is a server!
  - Need to "punch" hole through NAT

Principle: Fate Sharing

- "You can lose state information relevant to an entity's connections if and only if the entity itself is lost"
  - Example: OK to lose TCP state if either endpoint crashes
  - The TCP connection is no longer useful anyway!
- It is NOT okay to lose the connection if an unrelated entity goes down
  - Example: if an intermediate router reboots
- NATs violate this principle: if it goes down, all communication session are lost!
  - Unless you add redundancy and put state in persistent storage
- Bad news: many stateful "middleboxes" violate this rule
  - Firewalls, mobility services, … - more on this later
- Good news: today's hardware is very reliable

Outline

- The IP protocol
  - IPv4
  - IPv6
- IP in practice
  - Network address translation
  - Tunnels
  - ARP

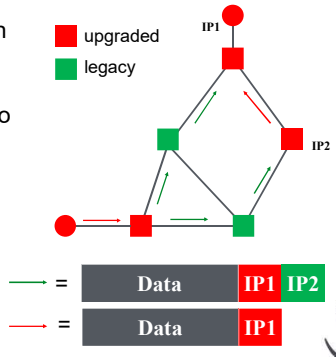
Motivation Tunneling

There are cases where not all routers have the same features

- An experimental IP feature is only selectively deployed – how do we use this feature end-to-end?
  - E.g., IP multicast
- A few are using a protocol other than IPv4 – how can they communicate?
  - E.g., incremental deployment of IPv6
- I am traveling with a CMU laptop - how can I can I keep my CMU IP address?
  - E.g., must have CMU address to use some internal services

Tunneling - Concept

- Force a packet to go to a specific point in the network.
  - Cannot rely on routers on the regular path
- 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 IP source routing
- Used increasingly to deal with special routing requirements or new features.
  - Mobile IP,..
  - Multicast, IPv6, research, ..

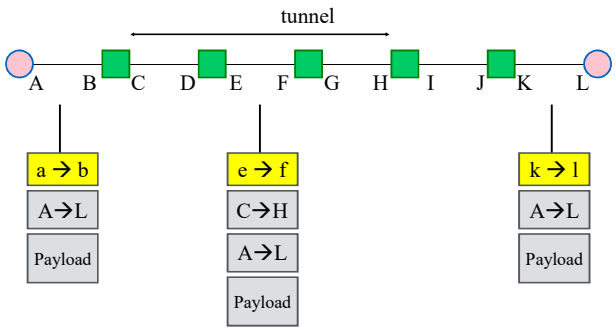


IP-in-IP Tunneling

- Described in RFC 1993.
- IP source and destination address identify tunnel endpoints.
- Protocol id = 4.
  - IPv4
- Several fields are copies of the inner-IP header.
  - TOS, some flags, ..
- Inner header is not modified, except for decrementing TTL.

V/HL	TOS	Length
ID		Flags/Offset
TTL	4	H. Checksum
Tunnel Entry IP		
Tunnel Exit IP		
V/HL	TOS	Length
ID		Flags/Offset
TTL	Prot.	H. Checksum
Source IP address		
Destination IP address		
Payload		

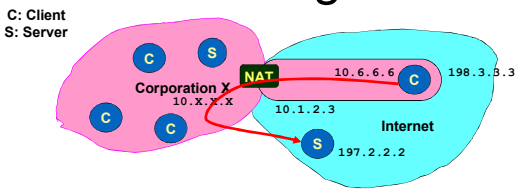
Tunneling Example



Tunneling Applications

- Virtual private networks.
  - Connect subnets of a corporation using IP tunnels
  - Often combined with IP Sec (later)
- Support for new or unusual protocols.
  - Routers that support the protocols use tunnels to “bypass” routers that do not support it
  - E.g. multicast, IPv6 (!)
- Force packets to follow non-standard routes.
  - Routing is based on outer-header
  - E.g. mobile IP (later)

Extending Private Network



V/HL	TOS	Length
ID		Flags/Offset
TTL	4	H. Checksum
198.3.3.3		
10.1.2.3		
V/HL	TOS	Length
ID		Flags/Offset
TTL	Prot.	H. Checksum
10.6.6.6		
197.2.2.2		
Payload		

- Employee works remotely with local address 198.3.3.3
  - Wants to appear as if working internally
- Establishes Virtual Private Network (VPN) – "tunnel"
  - Receives internal address 10.6.6.6 through tunnel
  - Encapsulation forces packets through corporate network
  - Provides access to internal/external services

Outline

- The IP protocol
  - IPv4
  - IPv6
- IP in practice
  - Network address translation
  - Tunnels
  - ARP

IP to MAC Address Translation

- How does one find the Ethernet address of a IP host?
- Address Resolution Protocol - ARP
  - Broadcast search for IP address
    - E.g., "who-has 128.2.184.45 tell 128.2.206.138" sent to Ethernet broadcast (all FF address)
  - Destination responds (only to requester using unicast) with appropriate 48-bit Ethernet address
    - E.g, "reply 128.2.184.45 is-at 0:d0:bc:f2:18:58" sent to 0:c0:4f:d:ed:c6

Caching ARP Entries

- Efficiency Concern
  - Would be very inefficient to use ARP request/reply every time need to send IP message to machine
- Each Host Maintains Cache of ARP Entries
  - Add entry to cache whenever you get ARP response
  - "Soft state": set timeout of ~20 minutes

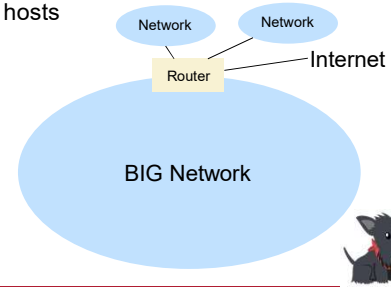
ARP Cache Example

- Show using command “arp -a”

```
Interface: 128.2.222.198 on Interface 0x1000003
  Internet Address      Physical Address      Type
128.2.20.218          00-b0-8e-83-df-50     dynamic
128.2.102.129         00-b0-8e-83-df-50     dynamic
128.2.194.66          00-02-b3-8a-35-bf     dynamic
128.2.198.34          00-06-5b-f3-5f-42     dynamic
128.2.203.3           00-90-27-3c-41-11     dynamic
128.2.203.61          08-00-20-a6-ba-2b     dynamic
128.2.205.192         00-60-08-1e-9b-fd     dynamic
128.2.206.125         00-d0-b7-c5-b3-f3     dynamic
128.2.206.139         00-a0-c9-98-2c-46     dynamic
128.2.222.180         08-00-20-a6-ba-c3     dynamic
128.2.242.182         08-00-20-a7-19-73     dynamic
128.2.254.36          00-b0-8e-83-df-50     dynamic
```

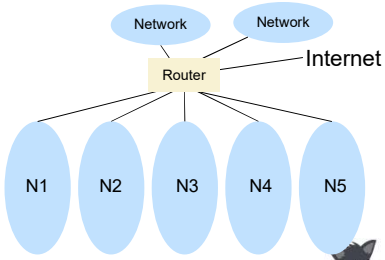
Challenge: Broadcast!

- Overhead scales (roughly) as N^2 for an N host network
  - N host does an ARP broadcast for each (new) destination
  - Each broadcast is delivered to N hosts
- Remember the solution?
- Subnetting!
  - Break up network into networks connected by router
- Not always a good idea
  - Extra complexity, management overhead, cost, …



Subnetting is an Option

- Subnetting!
  - Break up network into networks connected by router
- Limits the scope of ARP requests/responses inside smaller L2 networks
- But not always a good always a good idea
  - Extra complexity, management overhead, cost, …
  - Example: WiFi network



Proxy ARP

- Limit the scope of ARP requests/responses inside an L2
- Proxy ARP makes it look like ne network:
  - Host1 in N1 sends ARP for host 2 in N2
  - Proxy ARP looks up MAC address
    - May require discovery using ARP
  - Responds to host 1’s request
    - Acts as proxy for host 2
  - Also forwards packets from host 1 to host 2 at layer 2
    - Acts as a switch