Middleboxes and Tunneling

Mike Freedman

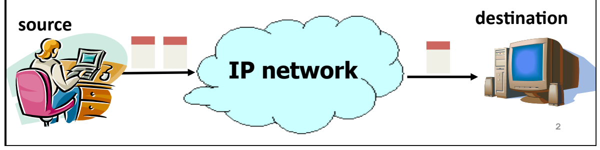
COS 461: Computer Networks

Lectures: MW 10-10:50am in Architecture N101

http://www.cs.princeton.edu/courses/archive/spr13/cos461/

Internet Ideal: Simple Network Model

- Globally unique identifiers
  - Each node has a unique, fixed IP address
  - … reachable from everyone and everywhere
- Simple packet forwarding
  - Network nodes simply forward packets
  - … rather than modifying or filtering them



Internet Reality

- Host mobility
  - Host changing address as it moves
- IP address depletion
  - Multiple hosts using the same address
- Security concerns
  - Detecting and blocking unwanted traffic
- Replicated services
  - Load balancing over server replicas
- Performance concerns
  - Allocating bandwidth, caching content, …
- Incremental deployment
  - New technology deployed in stages

Middleboxes

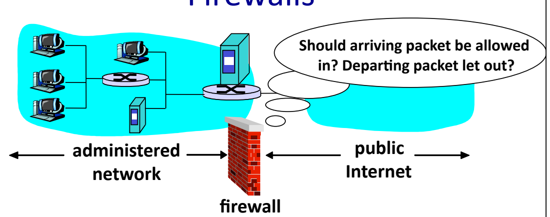
- Middleboxes are intermediaries
  - Interposed between communicating hosts
  - Often without knowledge of one or both parties
- Myriad uses
  - Address translators
  - Firewalls
  - Traffic shapers
  - Intrusion detection
  - Transparent proxies
  - Application accelerators

"An abomination!"
- **Violation of layering**
- **Hard to reason about**
- **Responsible for subtle bugs**

"A practical necessity!"
- **Solve real/pressing problems**
- **Needs not likely to go away**

Firewalls

Firewalls



- Firewall filters packet-by-packet, based on:
  - Source and destination IP addresses and port #'s
  - TCP SYN and ACK bits; ICMP message type
  - Deep packet inspection on packet contents (DPI)

Packet Filtering Examples

- Block all packets with IP protocol field = 17 and with either source or dest port = 23.
  - All incoming and outgoing UDP flows blocked
  - All Telnet connections are blocked
- Block inbound TCP packets with SYN but no ACK
  - Prevents external clients from making TCP connections with internal clients
  - But allows internal clients to connect to outside
- Block all packets with TCP port of Quake

7

Firewall Configuration

- Firewall applies a set of rules to each packet
  - To decide whether to permit or deny the packet
- Each rule is a test on the packet
  - Comparing headers, deciding whether to allow/deny
- Order matters
  - Once packet matches a rule, the decision is done

8

Firewall Configuration Example

- Alice runs a network in 222.22.0.0/16
- Wants to let Bob's school access certain hosts
  - Bob is on 111.11.0.0/16
  - Alice's special hosts on 222.22.22.0/24
- Alice doesn't trust Trudy, inside Bob's network
  - Trudy is on 111.11.11.0/24
- Alice doesn't want any other Internet traffic

9

Firewall Configuration Rules

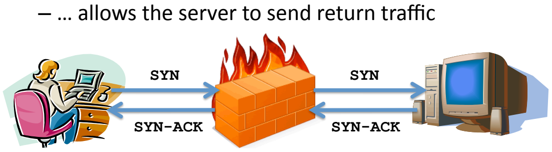
1. Allow Bob's network in to special destinations
   - Permit (src=111.11.0.0/16, dst = 222.22.22.0/24)
2. Block Trudy's machines
   - Deny (src = 111.11.11.0/24, dst = 222.22.0.0/16)
3. Block world
   - Deny (src = 0.0.0.0/0, dst = 0.0.0.0/0)

- Order?
  - (A) 3, 1
  - (B) 3, 1, 2
  - (C) 1, 3
  - (D) 1, 2, 3
  - (E) 2, 1, 3

10

Stateful Firewall

- Stateless firewall:
  - Treats each packet independently
- Stateful firewall
  - Remembers connection-level information
  - E.g., client initiating connection with a server
  - … allows the server to send return traffic



11

A Variation: Traffic Management

- Permit vs. deny is too binary a decision
  - Classify traffic using rules, handle classes differently
- Traffic shaping (rate limiting)
  - Limit the amount of bandwidth for certain traffic
- Separate queues
  - Use rules to group related packets
  - And then do weighted fair scheduling across groups

12

Clever Users Subvert Firewalls

- Example: filtering dorm access to a server
  - Firewall rule based on IP addresses of dorms
  - … and the server IP address and port number
  - Problem: users may log in to another machine
- Example: filtering P2P based on port #s
  - Firewall rule based on TCP/UDP port numbers
    - E.g., allow only port 80 (e.g., Web) traffic
  - Problem: software using non-traditional ports
    - E.g., write P2P client to use port 80 instead

13

Network Address Translation

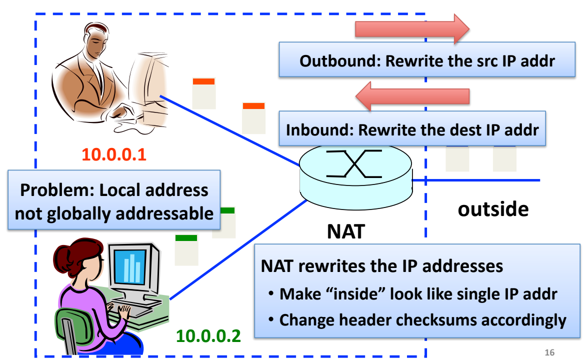
14

History of NATs

- IP address space depletion
  - Clear in early 90s that 2^{32} addresses not enough
  - Work began on a successor to IPv4
- In the meantime…
  - Share addresses among numerous devices
  - … without requiring changes to existing hosts
- Meant as a short-term remedy
  - Now: NAT is widely deployed, much more than IPv6

15

Network Address Translation



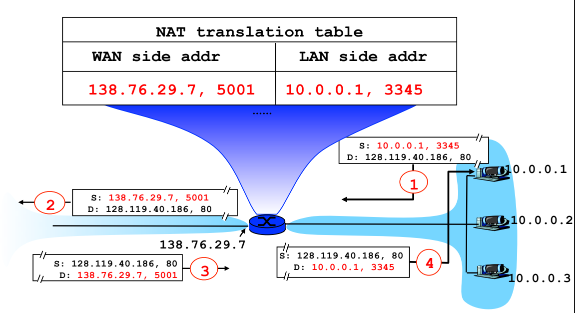
16

Port-Translating NAT

- Two hosts communicate with same destination
  - Destination needs to differentiate the two
- Map outgoing packets
  - Change source address and source port
- Maintain a translation table
  - Map of (src addr, port #) to (NAT addr, new port #)
- Map incoming packets
  - Map the destination address/port to the local host

17

Network Address Translation Example



18

Maintaining the Mapping Table

- Create an entry upon seeing an outgoing packet
  - Packet with new (source addr, source port) pair
- Eventually, need to delete entries to free up #'s
  - When? If no packets arrive before a timeout
  - (At risk of disrupting a temporarily idle connection)
- Yet another example of "soft state"
  - I.e., removing state if not refreshed for a while

19

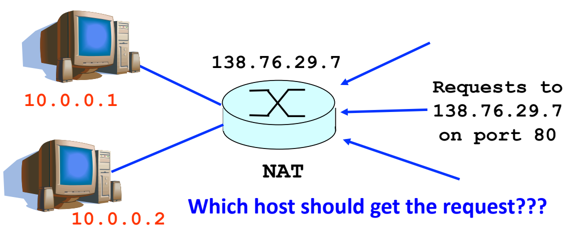
Where is NAT Implemented?

- Home router (e.g., Linksys box)
  - Integrates router, DHCP server, NAT, etc.
  - Use single IP address from the service provider
- Campus or corporate network
  - NAT at the connection to the Internet
  - Share a collection of public IP addresses
  - Avoid complexity of renumbering hosts/routers when changing ISP (w/ provider-allocated IP prefix)

20

Practical Objections Against NAT

- Port #s are meant to identify sockets
  - Yet, NAT uses them to identify end hosts
  - Makes it hard to run a server behind a NAT



Which host should get the request???

- **Explicit config at NAT for incoming conn's**

Principled Objections Against NAT

- Routers are not supposed to look at port #s
  - Network layer should care only about *IP* header
  - … and not be looking at the *port numbers* at all
- NAT violates the end-to-end argument
  - Network nodes should not modify the packets
- IPv6 is a cleaner solution
  - Better to migrate than to limp along with a hack

That's what happens when network puts power in hands of end users!

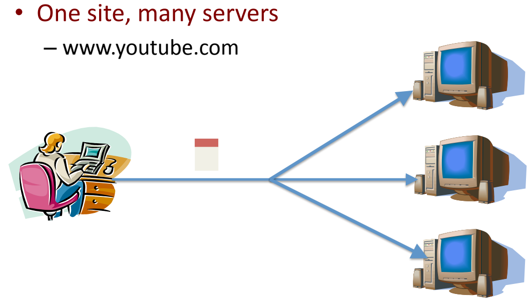
22

Load Balancers

23

Replicated Servers

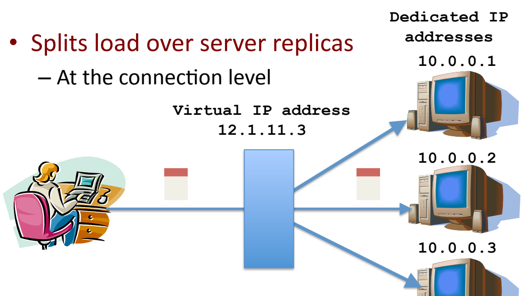
- One site, many servers
  - www.youtube.com



24

Load Balancer

- Splits load over server replicas
  - At the connection level



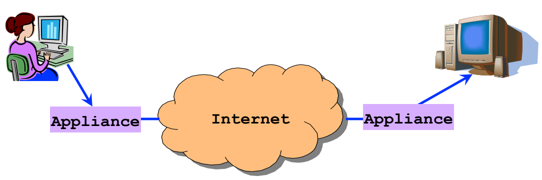
- Apply load balancing policies

25

Wide-Area Accelerators

26

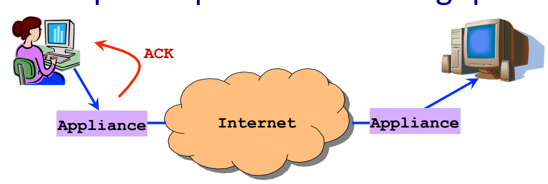
At Connection Point to the Internet



- Improve end-to-end performance
  - Through buffering, compression, caching, …
- Incrementally deployable
  - No changes to end hosts or the rest of the Internet

27

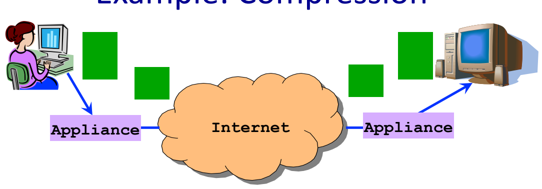
Example: Improve TCP Throughput



- Appliance with a lot of local memory
- Sends ACK packets quickly to the sender
- Overwrites receive window with a large value
- Or, even run a new and improved version of TCP

28

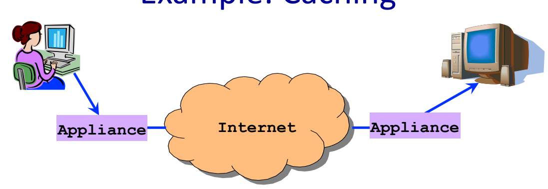
Example: Compression



- Compress the packet
- Send the compressed packet
- Uncompress at the other end
- Maybe compress across successive packets

29

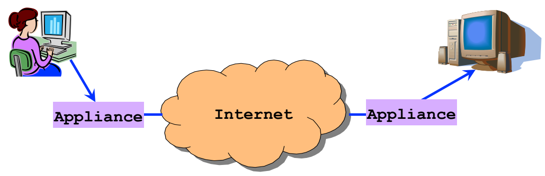
Example: Caching



- Cache copies of the outgoing packets
- Check for sequences of bytes that match past data
- Just send a pointer to the past data
- And have the receiving appliance reconstruct

30

Example: Encryption

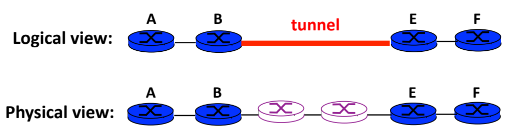


- Two sites share keys for encrypting traffic
- Sending appliance encrypts the data
- Receiving appliance decrypts the data
- Protects the sites from snoopers on the Internet

Tunneling

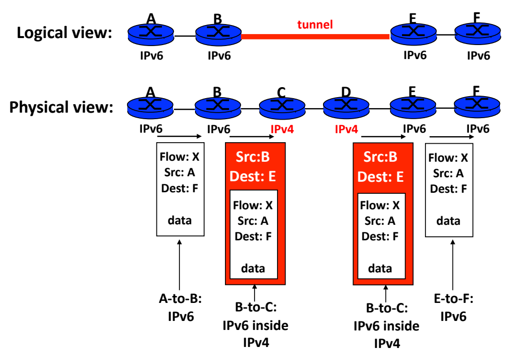
IP Tunneling

- IP tunnel is a virtual point-to-point link
  – Illusion of a direct link between two nodes

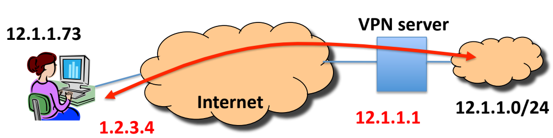


- Encapsulation of the packet inside IP datagram
  – Node B sends a packet to node E
  – ... containing another packet as the payload

6Bone: Deploying IPv6 over IP4



Remote Access Virtual Private Network



- Tunnel from user machine to VPN server
  – A “link” across the Internet to the local network
- Encapsulates packets to/from the user
  – Packet from 12.1.1.73 to 12.1.1.100
  – Inside a packet from 1.2.3.4 to 12.1.1.1

Conclusions

- Middleboxes address important problems
  – Getting by with fewer IP addresses
  – Blocking unwanted traffic
  – Making fair use of network resources
  – Improving end-to-end performance
- Middleboxes cause problems of their own
  – No longer globally unique IP addresses
  – Cannot assume network simply delivers packets