CS 640: Introduction to Computer Networks

Aditya Akella

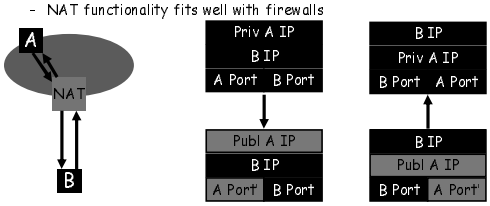
Lecture 12 -
IP-Foo

The Road Ahead

- NAT
- IPv6
- Tunneling / Overlays
- Network Management

Network Address Translation

- NAT maps
  (private source IP, source port) onto
  (public source IP, unique source port)
  - reverse mapping on the way back
  - destination host does not know that this process is happening
- Very simple working solution
  - NAT functionality fits well with firewalls



Types of NATs

- Bi-directional NAT: 1 to 1 mapping between internal and external addresses.
  - E.g., 128.237.0.0/16 -> 10.12.0.0/16
  - External hosts can directly contact internal hosts
  - Why use?
    - Flexibility: Change providers, don't change internal addrs.
    - Need as many external addresses as you have *hosts* - can use sparse address space internally.
- "Traditional" NAT: Unidirectional
  - Basic NAT: Pool of external addresses
    - Translate source IP address (+checksum,etc) only
  - Network Address Port Translation (NAPT): What most of us use at home
    - Translate ports
      - E.g., map (10.0.0.5 port 5555 -> 18.31.0.114 port 22) to (128.237.233.137 port 5931 -> 18.31.0.114 port 22)
    - Lets you share a single IP address among multiple computers

NAT Considerations

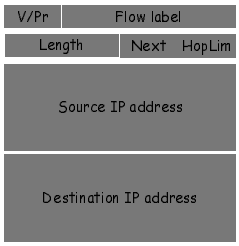
- NAT has to be consistent during a session.
  - Set up mapping at the beginning of a session and maintain it during the session
  - Recycle the mapping at the end of the session
    - May be hard to detect
    - Use DHCP (at home)
    - Usually static, though
- NAT only works cleanly for certain applications.
  - Some applications (e.g. ftp) pass IP information in payload
  - Need application level gateways to do a matching translation
    - Dirty!!

NAT Considerations

- NAT is loved and hated
  - Breaks a lot of applications.
  - Inhibits new applications like p2p.
  - Little NAT boxes make home networking simple.
  - Saves addresses (Address reuse)
  - Makes allocation simple.

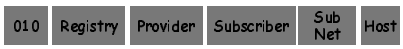
IP v6

- "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 reality in 2005: We're about 50% used.
    - CIDR
    - Tighter allocation policies; voluntary IP reclamation
    - NAT!!
  - Big worry: Millions of IP devices.
    - Small devices, Cell phones, toasters, pants…
- 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.



Back to Switching

- Common case: Switched in silicon ("fast path")
  - Most actions
- Special cases: Handed to CPU ("slow path", or "process switched")
  - Fragmentation
  - TTL expiration (traceroute)
  - IP option handling
  - Considered evil: slows routers down; avenue for attacks

IPv6 Header Cleanup

- No checksum
  - Why checksum just the IP header?
    - Efficiency: If packet corrupted at hop 1, don't waste downstream b/w
      - Useful when corruption frequent, b/w expensive
    - Today: Corruption rare, b/w cheap
- Different options handling
  - IPv4 options: Variable length header field. 32 different options.
    - Rarely used
    - 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

- 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
  - Routers don't deal with it any more

Migration from IPv4 to IPv6

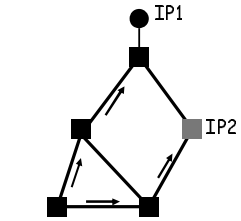
- Interoperability with IPv4 is necessary for gradual deployment.
- Two complementary mechanisms:
  - Dual stack operation: IP v6 nodes support both address types
  - Tunneling: tunnel IP v6 packets through IP v4 clouds
- Alternative is to create "IPv6 islands", e.g. enterprise networks, private interconnections,...
  - Use NAT to connect to the outside world
  - NAT translates addresses and also translate between IPv4 and IPv6 protocols

IPv6 Discussion

- IPv4 Infrastructure got better
  - Address efficiency
  - Co-opted IPv6 ideas: IPSec, diffserv, autoconfiguration via DHCP, etc.
- Massive challenge
  - Huge installed base of IPv4-speaking devices
  - Tussle
    - Who's the first person to go IPv6-only?
- Slow but steady progress in deployment
  - Most hosts & big routers support
  - Long-term: The little devices will probably force IPv6

Tunneling

- Force a packet to go via 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 overlays



Data	IP1	IP2

IP-in-IP Tunneling

- 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
  - Just like payload

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 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
- Dealing with NATs
  - Good or bad?

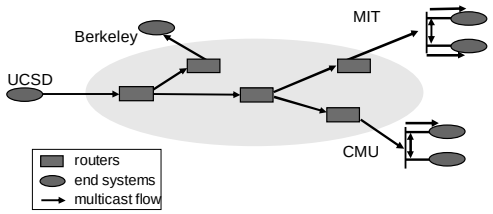
Overlay Networks

- A network "on top of the network".
  - E.g., initial Internet deployment
    - Internet routers connected via phone lines
      - An overlay on the phone network
  - Use 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

Overlay Networks

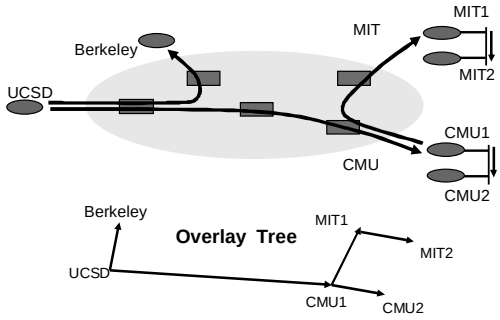
- Application-layer Overlays
  - Application Layer multicast (more later)
    - Transmit data stream to multiple recipients
  - Peer-to-Peer networks
    - Route queries (Gnutella search for "briney spars")
    - Route answers (Bittorrent, etc.)
  - Anonymizing overlays
    - Route data through lots of peers to hide source
      - (google for "Tor" "anonymous")
  - Improved routing
    - Detect and route around failures *faster* than the underlying network does.

IP Multicast



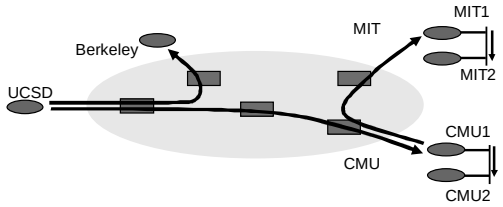
- Highly efficient
- Good delay

End System Multicast



Potential Benefits Over IP Multicast

- Quick deployment
- All multicast state in end systems
- Computation at forwarding points simplifies support for higher level functionality



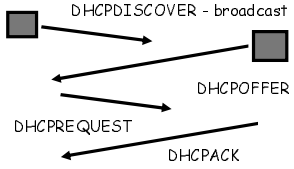
Network Management

- Two sub-issues:
  - Configuration management
    - How do I deal with all of these hosts?!
  - Network monitoring
    - What the heck is going on on those links?

Auto-configuration

- IP address, netmask, gateway, hostname, etc., etc.
  - Type by hand!!!
- IPv4 option 1: RARP (Reverse ARP)
  - Data-link protocol
    - Uses ARP format. New opcodes: "Request reverse", "reply reverse"
  - Send query: Request-reverse [ether addr], server responds with IP
    - Used primarily by diskless nodes, when they first initialize, to find their Internet address
- IPv4 option 2: DHCP
  - Dynamic Host Configuration Protocol
  - ARP is fine for assigning an IP, but is very limited
  - DHCP can provide all the info necessary

DHCP



- DHCPOFFER
  - IP addressing information
  - Boot file/server information (for network booting)
  - DNS name servers
  - Lots of other stuff - protocol is extensible; half of the options reserved for local site definition and use.

DHCP Features

- Lease-based assignment
  - Clients can renew: Servers really should preserve this information across client & server reboots.
- Provide *host* configuration information
  - Not just IP address stuff.
  - NTP servers, IP config, link layer config,…
- Use:
  - Generic config for desktops/dial-in/etc.
    - Assign IP address/etc., from pool
  - Specific config for particular machines
    - Central configuration management

IPv6 Auto-configuration

- Serverless ("Stateless"). No manual config at all.
  - Only configures addressing items, NOT other host things
    - Use DHCP for such things
- 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

Network Monitoring

- Identifying and localizing problems
  - Loss of connectivity, low throughput, ..
- Network operational and infrastructure status
  - Where are the bottlenecks, is it time for an upgrade, redirect traffic, ..
- Trouble-shooting
  - Somebody attacking a subnet, scanning, host initiating an attack
- Common requirements: Must track two sets of info
  - "Static" information: what is connected to what?
  - Dynamic information: what is the throughput on that link?

Common Monitoring Tools

- SNMP
  - Simple Network Management Protocol
    - Device status
      - 5 minute traffic average on outbound links
      - Amount of disk space used on server
      - Number of users logged in to modem bank
      - Etc.
    - Device alerts
      - Line 5 just went down!
  - Netflow
    - Detailed traffic monitoring
      - Break down by protocol/source/etc.
      - ("Who's serving 5 terabytes of briney spars photos??")
      - Usually sampled, coarse-grained

Simple Network Management Protocol (SNMP)

- Protocol that allows clients to read and write management information on network elements
  - Routers, switches, access points
  - Network element is represented by an SNMP agent
- Information is stored in a management information base (MIB)
  - Have to standardize the naming, format, and interpretation of each item of information
  - Ongoing activity: MIB entries have to be defined as new technologies are introduced
  - Lots of MIBs today!
- Different methods of interaction supported
  - Query response interaction: SNMP agent answers questions
  - Traps: agent notifies registered clients of events
- Need security: authentication and encryption.

Next Class

- Mid-term review on 10/18
  - Room 7331
  - 5:30PM
  - Suggestions for topics to review welcome
    - By noon, 10/18
- Mid-term
  - Will cover lectures 1—12 primarily
  - 2 points on 40 from lecture 13
  - Closed-book!
  - 75 mins