Administrivia

- Second Midterm 11/19
- Project extension to 11/19 (not very helpful with midterm and homework due in the same week)
- Last time: started lookup
- Today
  - Finish IPnG discussion
  - Start Multicast
IPNL paper

- Can we retro-fit instead of re-engineer?

- Given a limited or faulty design, is it better to tear it down and restart or to try and fix it?
  - Should you fix a clunker or buy a new car?

- NAT already used to delay the most critical problem with IPv4 – address depletion
  - We discussed why NAT is considered a hack, but are there legitimate benefits to it?

- Only a quick discussion of this paper; need to move on to multicast
Why NAT may not Evil

- Isolates site address from global address space
  - More secure
  - External changes don't affect site and vice versa
    * Changing provider becomes easy (only NAT box changes)
    * Internal IP reassignment easier
  - What if two NAT'd networks merged?
- Is global addressability really needed?
  - Dominant applications are client/server based
    - only server needs to be addressable
    - Is this true with new applications such as peer to peer?
- Delays (eliminates?) address depletion, without having to re-engineer the net
  - Is the cost of maintaining NAT really less than switching an network over to IPv6?
IPNL Paper

• Motivation: Can a “suitable” Internet architecture be developed using an extension to NAT?

• “suitable”
  – All hosts have long-lived, globally routable addresses (if they so choose) that serve to also identify the host.
  – Routers are stateless – no per connection state
  – A network’s address prefix is assigned independently of where the network attaches to the Internet
  – Packets cannot be easily hijacked by rogue or misconfigured hosts

• what is an extension vs. a brand new protocol
  – Isolate changes to hosts and NAT boxes, no changes to routers
General Idea

- Create another layer on top of IP (IPNL) that is routed using “NAT boxes” (IPNL routers)
  - Uses FQDN as a unique end to end identifier
  - Extends the available IP address space by defining an private IP address space
  - Original IP address becomes “high order” part of the full address
- A Realm is behind one (or more?) frontdoors
  - Internal IP routers route based on the private address only
- To an IPNL router, the internal IP “realm” becomes like a switched non broadcast link layer, with IP being the address family
Topology
More specifically

- Multiple Realms (zones) behind a front door (16-bit realm identifier)
  - Internal NL routers dynamically maintain information to reach realms behind the same front door
  - DNS zone associated with every realm
- Routing possible by FQDN or by IPNL address
Routing by FQDN

- Realm associated with one or more DNS zones (exclusive)
- DNS must be aware of visiting hosts too
- An internal router keeps explicit paths to zones behind the same nl-router
Routing

- If source and destination don't share the same front door
  - Route by default to the front door
  - Front door routes through middle realm to destination front door (uses global DNS)

- Routing also possible by IPNL address (need address resolution, a little complicated)
Discussion

- Less drastic solutions than IPv6 possible to overcome the address depletion problems
- Does it address the shortcomings of NAT?
- Does it address the shortcomings of IPv4?
  - What about the other functionality that is allowed by IPv6?
- IPNL is an academic solution, but its representative of similar solutions that are getting traction (more in a little bit)
- Which solution will win?
  - Nobody really knows...
Discussion

- IPNL is an academic solution, but generally idea is being pushed using even less drastic changes
  - Provide peer-to-peer addressability on top of NAT

- One suggestion that is getting more traction than IPNL is...
  - NUTSS (NAT, URI, Tunnel, SIP, STUN)
  - Basically, uses SIP URI names as global names, uses STUN to associate name with current NAT box/port

- Another suggestion is Teredo (IPv6 over UDP over IPv4)
Getting through NAT

App: I'm steve
UDP: 1234
IP: 1.1.1.1

App: I'm bob
UDP: 5678
IP: 2.1.1.1

1.1.1.1

10.1.1.1

steve

Application Server

2.1.1.1

10.1.1.1

bob

10.1.1.2
Getting through NAT (2)
NUTSS
Multicast

- Multicast is sending messages to a set of receivers potentially scattered all over the internet
  - Can you think of applications?

- Multicast approach: just unicast to the destinations, will this work?

- Multicast approach: broadcast to everyone, drop the packet if you don't need it

- What do we want in a multicast implementation?
Multicast – Issues

- Use Internet efficiently
  - Send messages only to intended receivers
  - Use best paths
  - Consolidate messages on common links
  - Scalable solutions

- General approach: recipients are anonymous
  - Identified by group addresses
  - Membership is dynamic

- What cooperation is needed from intermediate routers?

- Service Model
  - Reliable? QoS?
Multicast Groups

• Recall that class D IP addresses are reserved for multicast (to be used as group ids)

• Senders transmit to a class D address; routers forward it along to appropriate receivers

• Receivers must express interest in receiving traffic for a multicast group
  – Must join the multicast group
Multicasting on a LAN

• How to implement Multicast on a LAN segment?
• Like broadcast? All nodes receive it?
  – Wasteful
• NIC identifies whether its host is a member of the multicast group
  – How is a group defined?
  – Do we need a hardware multicast address? ARP to translate?
  – No ARP for Ethernet – Direct translation
    * Ethernet Multicast group 01:xx:xx:xx:xx:xx
    * Reserved for IP: 01:00:5e:00:00:00 to 01:00:5e:7f:ff:ff
    * Copy low order 23 bits from IP address to the low order 23 bits above
• Multicast leaking issues in NIC
• How far should a multicast packet propagate?
Reserved Internet Multicast Addresses (IANA)

- Class D addresses are 224.0.0.0 to 239.255.255.255
- 224.0.0.0 to 224.0.0.255 are local link addresses (TTL=1)
  - 224.0.0.1: All systems on this subnet
  - 224.0.0.2: All routers on this subnet
  - 224.0.0.5: OSPF routers
- 224.0.1.0 to 238.255.255.255 are globally scoped addresses
- 239.0.0.0 to 239.255.255.255 are limited scope addresses (within AS)
Extending Beyond a Single Segment

- Groups represented by a class D address

- Internet Group Management Protocol (IGMP – RFC 2236)
  - Used by routers to determine which groups have members on attached networks
  - Implemented on top of IP; it is in the same layer since IP uses it (like ICMP)
  - Supported by all Internet hosts
IGMP Operation

- Idea: Each router polls the hosts on its interfaces for their group memberships

- Each multicast router maintains a list of group memberships for each of its interfaces
  - Router multicasts a general membership query to the all-systems multicast address on each of its interfaces when it starts up and periodically
  - Hosts respond separately for each group of which they are a member
  - When a host joins a group, it multicasts an announcement to the group address (to which the router is listening)
  - When a host leaves a group, it transmits a leave-group message

- Any problems?
Optimizations

• Too many responses to membership query
  – Host waits a random time before responding
  – Responses are multicast on the group address
  – If it receives another group membership response it does not respond

• Too many leave-group messages
  – Send leave message (to all-router multicast group) only if you were the host to respond to the last general-membership message
  * Other hosts know that there is at least one more member so not necessary to leave group
  – Router multicasts a group membership query when it receives the leave-group message

• Reduce number of queries
  – Only one router designated as querier on each LAN segment
Using Multicast Socket Interface

• Typically uses UDP (why?)

• Use setsockopt to access IGMP:
  – `IP_MULTICAST_IF` to set outgoing interface (not needed usually)
  – `IP_ADD_MEMBERSHIP` to join multicast group
  – `IP_DROP_MEMBERSHIP` to leave group
  – `IP_MULTICAST_TTL` to set TTL

• What if multiple processes join the same multicast group?
• How to implement multicast over a large switched network?

• Dumb solution: flood (broadcast) to everyone; ignore if you don’t belong to the group
  – Why dumb?

• Good solution must only send it to appropriate parts of the network
  – Network as a graph – how should a multicast packet be forwarded ideally?
Multicast Performance

• How should the performance of a multicast algorithm be judged?
  – Network load (how many “link level” transmissions)
  – Router state (size of the tables in the router)
  – Router cpu requirement

• Additional metrics
  – Loss probability
  – Delay
  – Join delay ...
Reverse Path Broadcast (RPB)

- When a multicast packet comes in:
  - If it came from the NextHop for the source of the multicast, send it out on all links

- Better than a flood?
  - Broadcasts the packet away from the source, but does not loop back to it

- How does this approach perform in terms of our metrics?
Problems

- Floods the network, even if part of it does not include a member in the group

- Message forwarded twice over a LAN with two routers connected to it
  - solution, elect one of the routers as multicast agent
  - What extra capability is needed?

- Unnecessarily floods routers (once on every incoming link)
  - Possible solution – inform upstream routers which destinations they are your next hop for
Improvement – Truncated Reverse Path Broadcast (TRPB)

- Improve RPB by pruning leaf networks that do not contain a group member
  - How to determine if a network is a leaf?
    * A router can tell from the routing protocol messages (if it receives an update from a router on the same network)
    * Can be done through a modification to RIP
  - If a leaf, how do you figure out if it has members in the group?
    * IGMP
  - Good enough?
Reverse Path Multicasting (RPM)

- Idea: prune full subtrees, not just leaf networks
- Extend TRPB to prune subtrees
  - “on demand” pruning: done only on receipt of message to group
    * If a packet arrives at a router none of whose children have group members, a non-membership report (NMR) packet is sent one hop towards the root
    * if the one-hop-back router receives such messages from all its child routers and none of its incident networks have group members, it sends an NMR packet to its predecessor, etc.
    * NMRs are stored with a finite time to live (TTL) – when TTL expires, multicast transmission resumes
    * if a network subsequently acquires a group member, it sends an NMR-cancel message
- Essentially, this is Distance Vector Multicast Routing Protocol (DVMRP) – RFC 1075
Link State Multicast

- Recall: link-state – exchange information with all routers about immediate neighbors

- Supporting Multicast –
  - Idea: information about what multicast groups that hosts on your network belong to becomes part of the state – how?
  - The “expanded” state information gets exchanged in the update
    * Need to update when groups appear/disappear on a link
  - Each router can figure out the shortest multicast tree depending on the source and forward packets accordingly (a tree per source per group)

- Protocol is known as Multicast OSPF (MOSPF); RFC 1584
Multicast LS Overhead

- Computing and Storing Multicast trees
  - Expensive! need to compute/store for all senders on all groups
  - Optimization: Use a cache of computed multicast trees
    * recomputes on demand (e.g., when the first packet is seen); LRU to expire cache
    * Tradeoff space (cache) for router cpu

- Problem: group volatility triggers flooding
  - Aggregate changes by delaying updates?
Multicast – Discussion

- Would like optimal tree for each source/group
  - Expensive, does not scale gracefully, especially at intermediate routers
  - Optimization: use a multicast route cache for active groups, construct trees as needed. Still need to keep and exchange information

- Dense vs. Sparse Multicast
  - Dense-mode multicast
    * Most networks have members in the group
    * DVMRP and MOSPF handle dense mode well
  - Sparse-mode multicast
    * Few networks have members
    * DVMRP and MOSPF don't perform well
    * Would like:
      - Routers not involved do nothing
      - Distribution trees do not require many resources
      - Scalable solutions
Core Based Trees (CBT)

- Idea, a single distribution tree per multicast group; also called “shared tree” approach

- A multicast message is picked up by a router
  - If it is part of the tree, it floods it along the tree
  - If not, it unicasts to the root of the tree (the core) which floods the tree

- Joining the tree
  - Node informs its router using IGMP of interest in joining the group
    * If the router is not already part of the tree, it unicasts a join request to the “core”
    * As each intermediate router receives the request, it sets up a transient join state
      - If router part of the CBT, it sends a Join ACK; otherwise
      - It forwards the join request on to the core
CBT Discussion

- Overhead is good, for G groups with S sources each, need only G trees, instead of S * G

- Performance possibly bad
  - Data does not follow shortest route (average delay up)
  - Traffic concentration: may congest the tree while alternative paths are idle

- Recall learning bridges – any similarities?
Protocol Independent Multicast (PIM)

• Multicast most expensive when the group is sparse

• PIM distinguishes between sparse and dense multicast

• In the sparse mode
  – Similar to CBT – a shared distribution tree
  – Hosts join/leave groups explicitly using join/prune protocol messages
  – Where to send this message?
    ∗ Every group is assigned a rendezvous point (RP) using a distributed algorithm
  – The RP collects information about members in the group and builds a shared multicast tree
  – It may elect later (“if traffic becomes heavy”) to build a source specific tree that is more optimal

• RFC 2362
PIM Operation

Join

RP
R3 R2 R4
R1 R5

(a)

(b)

RP
R3 R2 R4
R1 R5

(c)

(d)

RP = Rendezvous point

--- Source-specific tree for source R1

--- Shared tree

SUNY-BINGHAMTON – CS528 Fall ’07 LEC. #17
PIM Discussion

• Choice of RP important
  – Single point of failure
  – can cause suboptimal operation

• Problem: both receivers and senders need to know addresses of RPs for every group
  – Configured or can extend IGMP to provide this information

• How do we make it real? Multicast across different AS’s

• Multicast requires router state
  – Breaks the E2E principle
  – Adds significant complexity and serious scaling issues