ANYCAST and MULTICAST
READING: SECTION 4.4

COS 461: Computer Networks
Spring 2011

Mike Freedman
http://www.cs.princeton.edu/courses/archive/spring11/cos461/
Outline today

• IP Anycast
  – N destinations, 1 should receive the message
  – Providing a service from multiple network locations
  – Using routing protocols for automated failover

• Multicast protocols
  – N destinations, N should receive the message
  – Examples
    • IP Multicast and IGMP
    • SRM (Scalable Reliable Multicast)
    • PGM (Pragmatic General Multicast)
unicast

broadcast

anycast

multicast

http://en.wikipedia.org/wiki/Multicast
Limitations of DNS-based failover

- Failover/load balancing via multiple A records

<table>
<thead>
<tr>
<th>Domain</th>
<th>TTL</th>
<th>Type</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.cnn.com">www.cnn.com</a></td>
<td>300</td>
<td>A</td>
<td>157.166.255.19</td>
</tr>
<tr>
<td><a href="http://www.cnn.com">www.cnn.com</a></td>
<td>300</td>
<td>A</td>
<td>157.166.224.25</td>
</tr>
<tr>
<td><a href="http://www.cnn.com">www.cnn.com</a></td>
<td>300</td>
<td>A</td>
<td>157.166.226.26</td>
</tr>
<tr>
<td><a href="http://www.cnn.com">www.cnn.com</a></td>
<td>300</td>
<td>A</td>
<td>157.166.255.18</td>
</tr>
</tbody>
</table>

- If server fails, service unavailable for TTL
  - Very low TTL: Extra load on DNS
  - Anyway, browsers cache DNS mappings 😞

- What if root NS fails? All DNS queries take > 3s?
Motivation for IP anycast

• Failure problem: client has resolved IP address
  – What if IP address can represent many servers?

• Load-balancing/failover via IP addr, rather than DNS

• IP anycast is simple reuse of existing protocols
  – Multiple instances of a service share same IP address
  – Each instance announces IP address / prefix in BGP / IGP
  – Routing infrastructure directs packets to nearest instance of the service
    • Can use same selection criteria as installing routes in the FIB
  – No special capabilities in servers, clients, or network
IP anycast in action

Announce 10.0.0.1/32

Client

Router 1

Router 2

192.168.0.1

10.0.0.1

Server Instance A

Announce 10.0.0.1/32

Router 3

192.168.0.2

Router 4

10.0.0.1

Server Instance B
IP anycast in action

Routing Table from Router 1:

<table>
<thead>
<tr>
<th>Destination</th>
<th>Mask</th>
<th>Next-Hop</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.0.0</td>
<td>/29</td>
<td>127.0.0.1</td>
<td>0</td>
</tr>
<tr>
<td>10.0.0.1</td>
<td>/32</td>
<td>192.168.0.1</td>
<td>1</td>
</tr>
<tr>
<td>10.0.0.1</td>
<td>/32</td>
<td>192.168.0.2</td>
<td>2</td>
</tr>
</tbody>
</table>
IP anycast in action

DNS lookup for http://www.server.com/
produces a single answer:

www.server.com. IN A 10.0.0.1
IP anycast in action

Routing Table from Router 1:

<table>
<thead>
<tr>
<th>Destination</th>
<th>Mask</th>
<th>Next-Hop</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.0.0</td>
<td>/29</td>
<td>127.0.0.1</td>
<td>0</td>
</tr>
<tr>
<td>10.0.0.1</td>
<td>/32</td>
<td>192.168.0.1</td>
<td>1</td>
</tr>
<tr>
<td>10.0.0.1</td>
<td>/32</td>
<td>192.168.0.2</td>
<td>2</td>
</tr>
</tbody>
</table>
IP anycast in action

Routing Table from Router 1:

<table>
<thead>
<tr>
<th>Destination</th>
<th>Mask</th>
<th>Next-Hop</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.0.0</td>
<td>/29</td>
<td>127.0.0.1</td>
<td>0</td>
</tr>
<tr>
<td>10.0.0.1</td>
<td>/32</td>
<td>192.168.0.1</td>
<td>1</td>
</tr>
<tr>
<td>10.0.0.1</td>
<td>/32</td>
<td>192.168.0.2</td>
<td>2</td>
</tr>
</tbody>
</table>
IP anycast in action

Routing Table from Router 1:

<table>
<thead>
<tr>
<th>Destination</th>
<th>Mask</th>
<th>Next-Hop</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.0.0</td>
<td>/29</td>
<td>127.0.0.1</td>
<td>0</td>
</tr>
<tr>
<td>10.0.0.1</td>
<td>/32</td>
<td>192.168.0.1</td>
<td>1</td>
</tr>
<tr>
<td>10.0.0.1</td>
<td>/32</td>
<td>192.168.0.2</td>
<td>2</td>
</tr>
</tbody>
</table>
IP anycast in action

From client/router perspective, topology could as well be:

Routing Table from Router 1:

<table>
<thead>
<tr>
<th>Destination</th>
<th>Mask</th>
<th>Next-Hop</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.0.0</td>
<td>/29</td>
<td>127.0.0.1</td>
<td>0</td>
</tr>
<tr>
<td>10.0.0.1</td>
<td>/32</td>
<td>192.168.0.1</td>
<td>1</td>
</tr>
<tr>
<td>10.0.0.1</td>
<td>/32</td>
<td>192.168.0.2</td>
<td>2</td>
</tr>
</tbody>
</table>
Downsides of IP anycast

• Many Tier-1 ISPs ingress filter prefixes > /24
  – Publish a /24 to get a “single” anycasted address:
    Poor utilization

• Scales poorly with the # anycast groups
  – Each group needs entry in global routing table

• Not trivial to deploy
  – Obtain an IP prefix and AS number; speak BGP
Downsides of IP anycast

- Subject to the limitations of IP routing
  - No notion of load or other application-layer metrics
  - Convergence time can be slow (as BGP or IGP converge)

- Failover doesn’t really work with TCP
  - TCP is stateful: if switch destination replicas, other server instances will just respond with RSTs
  - May react to network changes, even if server online

- Root nameservers (UDP) are anycasted, little else
Multicast protocols
Multicasting messages

• Simple application multicast: Iterated unicast
  – Client simply unicasts message to every recipient
  – **Pros:** simple to implement, no network modifications
  – **Cons:** $O(n)$ work on sender, network

• Advanced overlay multicast (“peer-to-peer”)
  – Build receiver-driven tree
  – **Pros:** Scalable, no network modifications
  – **Cons:** $O(\log n)$ work on sender, network; complex to implement

• IP multicast
  – Embed receiver-driven tree in network layer
  – **Pros:** $O(1)$ work on client, $O(#$ receivers$)$ on network
  – **Cons:** requires network modifications; scalability concerns?
IP multicast in action

Routing Table from Router 1:

<table>
<thead>
<tr>
<th>Destination</th>
<th>Mask</th>
<th>Next-Hop</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.0.0</td>
<td>/29</td>
<td>127.0.0.1</td>
<td>0</td>
</tr>
<tr>
<td>239.1.1.1</td>
<td>/32</td>
<td>192.168.0.1</td>
<td>1</td>
</tr>
<tr>
<td>239.1.1.1</td>
<td>/32</td>
<td>192.168.0.2</td>
<td>2</td>
</tr>
</tbody>
</table>
IP Multicast

• Simple to use in applications
  – Multicast “group” defined by IP multicast address
    • IP multicast addresses look similar to IP unicast addr
    • 224.0.0.0 to 239.255.255.255 (RPC 3171)
      – 265 M multicast groups at most
  – Best effort delivery only
    • Sender issues single datagram to IP multicast address
    • Routers delivery packets to all subnetworks that have a receiver “belonging” to the group

• Receiver-driven membership
  – Receivers join groups by informing upstream routers
  – Internet Group Management Protocol (v3: RFC 3376)
IGMP v1

• Two types of IGMP msgs (both have IP TTL of 1)
  – Host membership query: Routers query local networks to discover which groups have members
  – Host membership report: Hosts report each group (e.g., multicast addr) to which belong, by broadcast on net interface from which query was received

• Routers maintain group membership
  – Host sends an IGMP “report” to join a group
  – Multicast routers periodically issue host membership query to determine liveness of group members
  – Note: No explicit “leave” message from clients
IGMP: Improvements

• IGMP v2 added:
  – If multiple routers, one with lowest IP elected querier
  – Explicit leave messages for faster pruning
  – Group-specific query messages

• IGMP v3 added:
  – Source filtering: Join specifies multicast “only from” or “all but from” specific source addresses
IGMP: Parameters and Design

• Parameters
  – Maximum report delay: 10 sec
  – Membership query internal default: 125 sec
  – Time-out interval: 270 sec = 2 * (query interval + max delay)

• Is a router tracking each attached peer?
  – No, only each network, which are broadcast media

• Should clients respond immediately to queries?
  – Random delay (from 0..D) to minimize responses to queries
  – Only one response from single broadcast domain needed

• What if local networks are layer-2 switched?
  – L2 switches typically broadcast multicast traffic out all ports
  – Or, IGMP snooping (sneak peek into layer-3 contents), Cisco’s proprietary protocols, or static forwarding tables
IP multicast often best effort

• Application protocols on top of UDP
  – Within enterprises
  – Commercial stock exchanges
  – Multimedia content delivery
    • Streaming audio, video, etc.
    • Everybody in group listening/watching same content
    • IPTV
  – Many applications insensitive to loss, and networks managed/provisioned so little/no loss
What if we want reliability?
Challenges for reliable multicast

• Send an ACK, much like TCP?
  – ACK-implosion if all destinations ACK at once
  – Source does not know # of destinations

• How to retransmit?
  – To all? One bad link effects entire group
  – Only where losses? Loss near sender makes retransmission as inefficient as replicated unicast

• Once size fits all?
  – Heterogeneity: receivers, links, group sizes
  – Not all multicast apps need reliability of type offered by TCP. Some can tolerate reordering, delay, etc.
Scalable Reliable Multicast

• Receives all packets or unrecoverable data loss

• Data packets sent via IP multicast
  – ODATA includes sequence numbers

• Upon packet failure
  – ACK’s don’t scale, so...
  – If failures relatively rare, use Negative ACKs (NAKs) instead: “Did not receive expected packet”
  – What if it’s the last packet?
    • Sender issues heartbeats if no real traffic. Receiver knows when to expect (and thus NAK)
Handling failure in SRM

• Receiver multicasts a NAK
  – Or send NAK to sender, who multicasts NAK confirmation (NCF)

• Scale through NAK suppression
  – If received a NAK or NCF, don’t NAK yourself
  – What do we need to do to get adequate suppression?
    • Add random delays before NAK’ing
    • But what if the multicast group grows big?
      – Delay needs to grow \(\rightarrow\) lack of efficiency

• Repair through packet retransmission (RDATA)
  – From initial sender
  – From designated local repairer (DLR – IETF loves acronyms!)
Pragmatic General Multicast (RFC 3208)

• Similar approach as SRM: IP multicast + NAKs
  – ... but more techniques for scalability

• Hierarchy of PGM-aware network elements
  – NAK suppression: Similar to SRM
  – NAK elimination: Send at most one NAK upstream
    • Or completely handle with local repair!
  – Constrained forwarding: Repair data can be
    suppressed downstream if no NAK seen on that port
  – Forward-error correction: Reduce need to NAK

• Works when only sender is multicast-able
Outline today

• IP Anycast
  – N destinations, 1 should receive the message
  – Providing a service from multiple network locations
  – Using routing protocols for automated failover

• Multicast protocols
  – N destinations, N should receive the message
  – Examples
    • IP Multicast and IGMP
    • SRM (Scalable Reliable Multicast)
    • PGM (Pragmatic General Multicast)