A Framework for Scalable Global IP-Anycast (GIA)

Dina Katabi, John Wroclawski
MIT Laboratory of Computer Science
Presented by Zhenkai Zhu

Anycast
- **Anycast** is a network addressing and routing scheme whereby data is routed to the "nearest" or "best" destination as viewed by the routing topology.

- Performance
- Robustness

Anycast In Real World
- DNS Root Anycast Servers

- C, F, I, J, K, L and M

IP Anycast
- Announce the same IP prefix at different locations
- Rely on global routing to direct traffic

- Pros
  - Simple
  - Completely transparent to clients and routers
  - End-to-End path is automatically efficient

- Cons
  - Violate the form of hierarchical aggregation
  - Cause route table to grow exponentially
  - Need at least a /24 for each anycast service
Application Layer Anycast

- This approach attempts to build a directory system which, queried with a service name and a client address, returns the unicast address of server that is nearest.

- **Cons**
  - The directory service needs to probe the servers to ensure they are alive
  - Need to obtain the distance between the client and different servers
  - Lack of boot-strapping mechanism where by the user access the nearest directory

- **Pros**
  - Do not need to modify the router
  - Can use distance metrics that are available only at application level (e.g. server load).

Design Rationale

- Decoupling IP anycast routing from unicast routing
  - Reduce impact on unicast routing
  - Spend different amount of routing resources on different routes
  - Google vs Weather in France
- Caching at each edge domain the routes to popular anycast groups
  - Learned from Web
- Default unicast routes for unpopular anycast groups
  - The expense for scalability

Address Architecture

- IP Anycast Address
- Anycast Indicator
- Home Domain’s Unicast Prefix
- Group ID
- Default Route
- Home Domain’s Unicast Prefix
- Group ID
- Zeros

- Anycast service provider should have at least one member node at its home domain

Anycast Routing Terminology

- **Domain**
  - Basically an AS
- **Neighborhood**
  - The set of domains that are R or fewer domain hops away
- **Group Classification**
  - Internal: at least one member in the domain
  - External: no member in the domain
  - Popular
  - Unpopular
Joining An Anycast Group

- Ask the first hop router to advertise the group’s address
- Add a new message type to IGMP or Neighbor Discovery Protocol
- Router uses keep-alive mechanism to ensure the availability of the group member
- Need a security mechanism to ensure the host is allowed to join the group

Routing Internal Groups

- Intra-domain routing protocols intrinsically provides the anycast service
- No need to modify distance vector algorithms
- For link state algorithms, large cost should be assigned to the links connecting anycast nodes to the local network

Routing Unpopular Groups

- Three possible ways to deliver the packet
  - Come across a domain with an anycast group member
  - Come across a domain that regards this group as a popular group
  - Hits the home domain

Routing Popular Groups

- On-demand Query-based Routing
  - The border routers in an edge domain decide which groups are popular
  - Periodically, border routers search for nearest members of popular groups by controlled flooding
  - Once the routes are found, border routers cache the routes and tunnel all subsequent packets to the domain where the nearest members reside
Initiating A Search

- At the beginning of each “search interval”, the originating border router (OBR) generates a search message for all popular groups for which there is no learned routes.
- The search is a scoped domain-by-domain broadcast that explores the neighborhood.
- OBR sets a timer and waits for reply.

<table>
<thead>
<tr>
<th>BGP Header</th>
<th>AS Path</th>
<th>Anycast Address 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence Number</td>
<td>OBR’s IP</td>
<td>Anycast Address 2</td>
</tr>
<tr>
<td>TTL</td>
<td></td>
<td>Anycast Address 3</td>
</tr>
<tr>
<td>Total Path Attribute Length</td>
<td></td>
<td>……</td>
</tr>
<tr>
<td>Path Attributes</td>
<td></td>
<td>Anycast Address N</td>
</tr>
<tr>
<td>Network Layer Reachability Information</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Receiving A Search

- After timer expires, OBR checks all received replies and chooses the one with shortest route
- OBR removes anycast addresses for which it has found a route, and decrease the popularity of the remaining ones to reduce the chance of being included in a future search
- OBR puts the routes in the cache, and advertises them to all internal peers

<table>
<thead>
<tr>
<th>Search Message</th>
<th>Internal Peer?</th>
<th>Propagate to Peers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Ignore</td>
</tr>
<tr>
<td></td>
<td>Logged?</td>
<td>Ignore</td>
</tr>
<tr>
<td></td>
<td>Internal Group</td>
<td>Reply to OBR</td>
</tr>
<tr>
<td></td>
<td>Learned Route</td>
<td>Concatenate Route</td>
</tr>
<tr>
<td></td>
<td></td>
<td>in Search with</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Learned Route</td>
</tr>
<tr>
<td>TTL = 0</td>
<td></td>
<td>Reply to OBR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ignore</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Propagate to Peers</td>
</tr>
</tbody>
</table>

Receiving A Reply

- After time expires, OBR checks all received replies and chooses the one with shortest route
- OBR removes anycast addresses for which it has found a route, and decrease the popularity of the remaining ones to reduce the chance of being included in a future search
- OBR puts the routes in the cache, and advertises them to all internal peers

Withdraw And Refreshing

- **Withdraw**
  - Lose connectivity to the nearest member
    - Discover from BGP withdraw message
  - Nearest member crashes or leaves the group
    - Can not be discovered directly
    - Receiving Border Router forwards packets according its best knowledge and replys to OBR an ICMP message
  - Caching space is full

- **Refresh**
  - Ensure the learned route remains the shortest one
  - After a certain interval, send out a search whose TTL is set to one hop less than current route’s length
Scoping a Search

- Set the TTL field to an appropriate value
  - Recommended setting is 2 or 3
- Transit domains control the scope of a search by instructing their border routers not to propagate search message to distant peers

Efficiency

- Let $R = \text{Average (external path in GIA / shortest path)}$
- $\rho$ is percentage of anycast traffic to popular groups
- $R = \rho \cdot R_{\text{popular}/\text{nearest}} + (1 - \rho) \cdot R_{\text{home}/\text{nearest}}$
- Taking into account that a search may fail, let $P_{\text{success}}$ be the success ratio of finding the nearest member
- $R_{\text{popular}/\text{nearest}} = P_{\text{success}} + (1 - P_{\text{success}}) \cdot R'_{\text{home}/\text{nearest}}$
- Thus,
  $$R = \rho \cdot (P_{\text{success}} \cdot I + (1 - P_{\text{success}}) \cdot R'_{\text{home}/\text{nearest}}) + (1 - \rho) \cdot R_{\text{home}/\text{nearest}}$$

Simulation Setup

- Use a set of snapshots of the Internet inter-domain topology based on BGP routing tables
- Randomly choose the home domain for each anycast group
- Randomly assign members of a group to domains
- Model the popularity distribution of anycast groups after the popularity distribution of web servers
- Average life time of a learned route is 30 days
- Prune redundant messages with special rules for highly connected domains
- Do not learn from cached routes in other routers
## Tricks Used

- Split the unicast and anycast routing
  - Reduce the impact on unicast routing
- On-demand Query + Caching
  - Reactive querying instead of proactive advertising
  - Common in ad-hoc routing mechanisms
- Push the work to the edge of Internet
  - Routers at the core do not need to keep anycast information
  - Utilize the unused resource at edge domain
  - Give control to end domain
- Quality of Service
  - Higher popularity $\Rightarrow$ Better service

## Dirt

- Stated Dirt
  - Need to upgrade routers so they can understand the syntax anycast address
  - It’s not trivial
- Hidden Dirt
  - Unpopular anycast groups suffer
  - Still place heavy burden on core internet
    - Searching flooding and refreshing flooding are costly
    - Searching overhead will be much greater than the authors expected
  - Very vulnerable to hijacking if a BR is compromised, and the hijacking is difficult to be noticed