Effect of anycast on K-root

Some early results
K root anycast deployment

- 3 global nodes (BGP transit)
  - LINX
    - ... 5459 25152 25152 i
  - AMS-IX
    - ... 25152 25152 25152 i
  - Tokyo (since 5/2005)
    - ... 25152 25152 25152 25152 i

- ~10 local nodes (announced with no-export)

- Future nodes will be global
  - Miami (live as we speak?)
  - India
  - West coast?
Node structure

- 2 machines running nsd, switches, routers

- Production IP: OSPF load balancing
  - K-root IP address: 193.0.14.129

- Service interfaces
  - Normally firewalled, don’t reply to queries
    - LINX: 193.0.16.1, 193.0.16.2
    - AMS-IX: 193.0.17.1, 193.0.17.2
    - ...

- Management interfaces, ...
Why anycast?

• Reasons for anycasting:
  – Provide resiliency and stability
  – Reduce latency
  – Spread server and network load, contain DOS attacks
  – ...

• Is it effective?
Latency
Latency comparison

• Ideally, BGP should choose the instance with the lowest RTT.
• Does it?

• Measure RTTs from the Internet to:
  – Anycasted IP address (193.0.14.129)
  – Service interfaces of global nodes (not anycasted)

• Compare results

• Just to make sure this is apples to apples:
  – Are AS-paths to service interfaces the same as to production IP?
  – According to the RIS, “mostly yes”
Probe locations: TTM (bias?)
Method

• Send DNS queries from all test-boxes
  – For each K-root IP:
    • Do a “dig hostname.bind”
    • Extract RTT
    • Take minimum value of 5 queries
  – Compare results of anycast IP with those of service interfaces

• $\alpha = \frac{\text{RTT}_K}{\min(\text{RTT}_i)}$
  $\alpha \approx 1$: BGP picks the right node
  $\alpha > 1$: BGP picks the wrong node
  $\alpha < 1$: local node?
Latency comparison

Local nodes: ⭕️ = better RTT ⭙️ = worse RTT
Local worse than global?

$ cat tt89
193.0.14.129 k2.denic 29 k2.denic 30 k2.denic 29 k2.denic 30 k2.denic 29
193.0.16.1 k1.linx 4 k1.linx 3 k1.linx 3 k1.linx 3 k1.linx 3
193.0.16.2 k2.linx 3 k2.linx 3 k2.linx 3 k2.linx 3 k2.linx 4
193.0.17.1 k1.ams-ix 12 k1.ams-ix 11 k1.ams-ix 12 k1.ams-ix 13 k1.ams-ix 13
193.0.17.2 k2.ams-ix 12 k2.ams-ix 13 k2.ams-ix 11 k2.ams-ix 12 k2.ams-ix 13

(This example has since been fixed)

• What’s going on here? Perhaps:
  – Local node announcements don’t necessarily leak
  – But they do get announced to customers
    …and customers of customers
    …where they compete with announcements from global nodes
    …which lose out due to prepending
Latency comparison (global)
Latency: conclusions

• Local nodes “confuse” the situation due to transit and prepending
• But all in all, BGP does a surprisingly good job
  – Even though the AS-paths are of different lengths!
• This contrasts with other work (Ballani & Francis)
  – Perhaps it is because K only has two global nodes
  – Will it get worse when more nodes are deployed?
Load balancing
Usefulness of local nodes

• How much traffic does a local node get?

• Do local nodes take load off the global nodes?

• Where do local queries come from?
  – From the global K nodes?
  – From the other root servers?
Local queries

![Graph showing local queries](image-url)
Local queries (cumulative)
Local vs global

Local vs global queries

- Global nodes
- Local nodes
- Total

~23%
Load balancing: conclusions

- The traffic a local node gets depends on where it is
- Wide variation
- Location must be chosen carefully to maximise usefulness

- Local nodes do take load off the global nodes
  - but not much
- Increase in local traffic does not correspond to decrease in global traffic
  - Traffic mostly seems to come from the other roots
Stability
Instance switches

• Didn’t measure resiliency
  – Pretty much a given: the more servers there are,
    • the more they can withstand
    • the more localised the impact of an attack

• What about stability?
  – The more routes competing in BGP, the more churn
  – Doesn’t matter for single-packet exchanges (UDP)
  – Does matter for TCP queries

• How frequent are instance switches?
Detecting instance switches

• Measure at the server
• Look at instance switches that actually occur

• Procedure:
  – Look at packet dumps
    • At the time, there were only 2 global nodes
  – Extract all port 53/UDP traffic
  – For each IP address, remember where it was last seen
  – If the same IP is seen elsewhere, log a switch

• Caveats:
  – K nodes are only NTP synchronized
Instance switches: results

• 24 hours of data:
  – 527,376,619 queries
  – 30,993 instance switches (~0.006%)
  – 884,010 IP’s seen
  – 10,557 switching IPs (~1.1%)

• What do the switches look like?
Time since last switch
Time since last switch, log-log
Top switching IPs

[Graph showing the distribution of top instance switchers on a log-log scale]
Stability: results

• Instance switches are rare
• Some IPs switch a lot
  – Load balancing?

• Nice power laws, but what do they mean?
• We don’t know yet
• Further analysis needed
Questions?