Discussion: DNS for NIDs

High Performance Data Division

Doug Oucharek
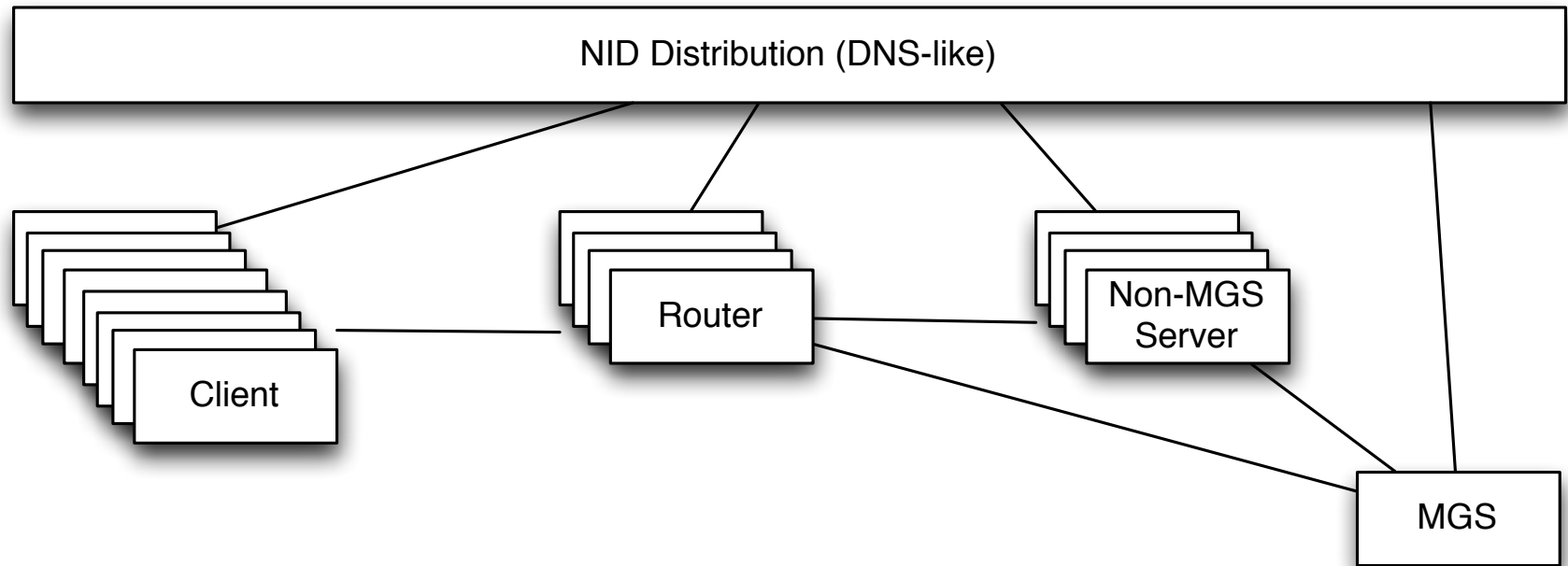
April 18, 2013

The Need

- A more flexible way to map targets to NIDs

- Support for DHCP
  - Needed to run in the cloud
  - Preferable for running in enterprise environments

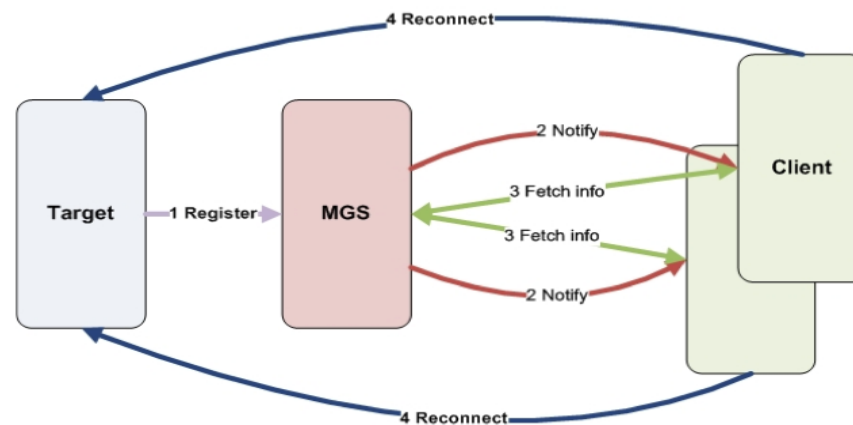- A scalable DNS-like model for NID mapping

Overview

Review: Imperative Recovery

- Accelerates client reconnection

- Version numbered Target-NID table

- MGS notifies clients of change

- Clients retrieve just the changes

- Type of Publisher/ Subscriber model

Target - NID Table

Target	Version	NID
OST0000	57	192.168.1.3@tcp3

Why DHCP does not work today

- Non-MGS servers register their address with MGS but must stick to a static set of NIDs in file system configuration

- Failover NIDs stored statically…currently cannot support DHCP

- MGS and Routers must be static

(intel)

Clients and Non-MGS Servers

Requirement	Suggestions
Don't keep static NID lists	1. Rely on the NIDs in the Target-NID table 2. Configure with Target names mapped to DNS names
Failover NIDs need to be dynamic	See above
Need to retrieve NIDs for a target from a service	1. Could use Target-NID table introduced by Imperative Recovery 2. Use existing Dynamic DNS systems via kernel upcall
Nice to have: Isolate knowledge of NIDs to LNet (give file system NID-Amnesia)	Push target to NID service to LNet. File system only passes target names to LNet, not NIDs.

(intel)

MGS

Requirement	Suggestions
Needs to register NID where all other nodes can find it	1. Publish NID to Dynamic DNS server 2. Respond to broadcast requests for address 3. Publish MGS service via Zeroconf-like system 4. Use a distributed database
If MGS is DNS distributor, needs low-cost redundancy	Use 2-phase commit for registrations

(intel)

NID Distributor

Requirement	Suggestions
Network Service	1. MGS Imperative Recovery 2. Dynamic DNS 3. Database (DaaS)
Redundant	1. If using MGS, see previous slide 2. If using Dynamic DNS, follow standard 3. If database, use replication
Scalable notifications	1. Rely on Imperative Recovery 2. Use a broadcast channel 3. Use a "smart" broadcast like the Gossip protocol
Flexible concept of an "address"	1. Use features in the DNS spec to configure for various address formats 2. On store only strings for NIDs

(intel)

Routers

Requirement	Suggestions
Routers register with DNS distributor like servers	1. Extend Imperative Recovery system to receive router registrations 2. Routers register with Dynamic DNS system
Routing tables become dynamic	1. Have Imperative Recovery generate and distribute routing table with Target-NID table 2. Have routers respond to broadcast requests where they send their configured NIDs to requestor 3. Use target names in existing routing configurations