Design and implementation of a Routing Control Platform

Matthew Caesar, Donald Caldwell, Nick Feamster, Jennifer Rexford, Aman Shaikh, Jacobus van der Merwe
1. Provide internal reachability (IGP)
2. Learn routes to external destinations (eBGP)
3. Distribute externally learned routes internally (iBGP)
4. Select closest egress (IGP)
What’s wrong with Internet routing?

• Full-mesh iBGP doesn’t scale
  – # sessions, control traffic, router memory/cpu
  – Route-reflectors help by introducing hierarchy
    • but introduce configuration complexity, protocol oscillations/loops

• Hard to manage
  – Many highly configurable mechanisms
  – Difficult to model effects of configuration changes
  – Hard to diagnose when things go wrong

• Hard to evolve
  – Hard to provide new services, improve upon protocols
What’s causing these problems?
- Each router has limited visibility of IGP and BGP
- No central point of control/observation
- Resource limitations on legacy routers

Solution: compute routes from central point, remove protocols from routers
RCP in a single ISP

- **Better scalability**: reduces load on routers
- **Easier management**: configuration from a single point
- **Easier evolvability**: freedom from router software
RCP architecture

Routing Control Platform (RCP)

Route Control Server (RCS)

Available BGP routes

Selected BGP routes

Path cost matrix

BGP Engine

IGP Viewer (NSDI ’04)

BGP updates

... BGP updates

... BGP updates

... IGP link-state advertisements

RCP architecture

Routing Control Platform (RCP)

Route Control Server (RCS)

Available BGP routes

Selected BGP routes

Path cost matrix

BGP Engine

IGP Viewer (NSDI ’04)

BGP updates

... BGP updates

... BGP updates

... IGP link-state advertisements
Challenges and contributions

- **Reliability**
  - **Problem**: single point of failure
  - **Contribution**: simple replication of RCP components

- **Consistency**
  - **Problem**: inconsistent decisions by replicas
  - **Contribution**: guaranteed consistency without inter-replica protocol

- **Scalability**
  - **Problem**: storing all routes increases cpu/memory usage
  - **Contribution**: can support large ISP in one computer

→ Building this system is feasible
Potential consistency problem

- Need to ensure routes are consistently assigned
  - Even in presence of failures/partitions
Consistent assignment
Single RCP, single partition

- **Solution:** Assign all routers along the shortest IGP path the same exit router
  - Ensures forwarding loops don’t arise
Consistent assignment
Single RCP, multiple partitions

Solution: Only use state from router’s partition in assigning its routes
– Ensures next hop is reachable
Consistent assignment
Multiple RCPs, multiple partitions

- **Solution:** RCPs receive same IGP/BGP state from each partition they can reach
  - IGP provides complete visibility and connectivity
  - RCS only acts on partition if it has complete state for it

→ No consistency protocol needed to guarantee consistency in steady state
Scalability solution

• Eliminate redundancy
  – Store only a single copy of each BGP route

• Accelerate lookup
  – Quickly find routers whose routes changed

• Avoid recomputation
  – Compute routes once for groups of routers
  – Don’t recompute if relative ranking of egress routers unchanged
RCS data structures

Global route table
(stores copies of routes)

RIB-Out shadow tables
(points to currently used route for each router)

Egress lists
(points to routes that use each egress)

BGP routes →

← Prefixes
(stores copies of routes)

BGP updates
(from egress routers)

rtr1  rtr2  rtr3

BGP updates
(to routers)

IGP updates
Example of egress list operation

D’s egress list

3  C

4  A

7  B

Diagram:

A —— 4 —— B
    |    |
    v    v
7    |
    |
B —— 7 —— C
    |
    v
3

D

4  7  3

A

B

C

D
Example of egress list operation

D’s egress list

A  B  C

A  B  C

D
Example of egress list operation

D’s egress list

5
3

4

A

B

7

A

B

C

D

4

7

3.5
Example of egress list operation

D’s egress list

4 A
5 C
7 B

A -- 4 -- B
7

D -- 3, 5 -- C
Example of egress list operation

D’s egress list

A

B

C

D

A

B

C

D

1

3

4

7
RCS data structures

Global route table (stores copies of routes)

BGP routes →

Prefixes

BGP updates (from egress routers)

RIB-Out shadow tables (points to currently used route for each router)

rtr1  rtr2  rtr3

Prefixes

BGP updates (to routers)

Egress lists (points to routes that use each egress)

rtr1  rtr2

g  g  g

3 3

IGP updates
Performance evaluation

- BGP and OSPF logs from Tier-1 ISP backbone
  - collected on Aug 1 2004, ~500 routers
- Metrics: memory usage, update processing time
- Measurement techniques:
  - Whitebox (instrument code with timers)
  - Blackbox (workload generator on separate machine)
    - no-queuing (one update at a time)
    - real-time (allow updates to queue)
- 3.2 Ghz P4, 4GB memory, Linux 2.6.5
Results: RCS memory usage

- 5,000 prefixes
- 50,000 prefixes
- all (203,000) prefixes

State for entire ISP in 2.5 gigabytes
BGP change processing time

All BGP updates processed within 30ms
IGP change processing time

High delay due to bursty path cost changes
Towards decoupling BGP from IGP

- **Problem**: Single link change can affect many paths
  - Transient delay/loss, traffic shift, and eBGP updates
- **Solution**: Decouple egress point ranking and cost
  - Experiment: process only reachability-affecting events
IGP change processing time

New approach reduces processing time
Conclusions

• RCP improves routing
  – Correct, scalable route distribution
  – Eases management and evolvability

• RCP is feasible
  – Reliability, scalability, deployability, consistency

• Many open problems:
  – How to simplify network management
  – How to enable new services
  – RCP cooperation between ISPs