OPCA: Robust Interdomain Policy Routing and Traffic Control

Sharad Agarwal

Chen-Nee Chuah, Randy H. Katz

\{sagarwal,randy\}@eecs.berkeley.edu, chuah@ece.ucdavis.edu.
Outline

- Introduction
  - BGP primer
  - Problem statement
  - Prior work: inadequate solutions

- OPCA
  - Overview
  - Completed components, protocol
  - Evaluation
BGP Introduction

- Internet composed of
  >13,000 domains (ASes) using BGP
  - E.g. MIT, BBN
  - Exchange reachability in BGP
    - But not internal topology
Internet composed of >13,000 domains (ASes) using BGP
- E.g. MIT, BBN
- Exchange reachability in BGP
  - But not internal topology
BGP Shortcomings

- Congestion or failure
  - Seen at destination
- Cannot influence source
  - Convergence slow
  - No explicit control

Network Diagram:
- MIT
- Level3
- BBN
- Concentric
- Stanford

Network Routes:
- 171.64.0.0/16

Sharad Agarwal – p.4/24
Multihoming

- Multihomed stub ASes increasing
  - Two benefits

Failover:
- Primary provider + redundant access links
- However, limited by BGP: ~15 minutes

Traffic load balancing:
- Outgoing traffic
  - Use smart BGP route selection
    - Rexford, Routescience, etc.
- Incoming traffic
  - Not possible today ... (sort of)
  - Can pollute BGP with weird routes
  - Local optimizations have global ramifications
  - Can't scale, not effective enough
Multihoming

- Multihomed stub ASes increasing
  - Two benefits

- Failover
  - Primary provider + redundant access links
  - However, limited by BGP : ~15 minutes
Multihoming

- Multihomed stub ASes increasing
  - Two benefits
- Failover
  - Primary provider + redundant access links
  - However, limited by BGP: ~15 minutes
- Traffic load balancing
  - Outgoing traffic
    - Use smart BGP route selection
      - Rexford, Routescience, etc.
Multihoming

- Multihomed stub ASes increasing
  - Two benefits
- Failover
  - Primary provider + redundant access links
  - However, limited by BGP: ~15 minutes
- Traffic load balancing
  - Outgoing traffic
    - Use smart BGP route selection
    - Rexford, Routescience, etc.
  - Incoming traffic
    - Not possible today ... (sort of)
    - Can pollute BGP with weird routes
    - Local optimizations have global ramifications
    - Can’t scale, not effective enough
Problem Statement

Goal

- Improve fail over time from ~15 minutes
- Improve time to shift incoming traffic between paths
  - Current BGP techniques offer no control
Problem Statement

Goal
- Improve fail over time from ~15 minutes
- Improve time to shift incoming traffic between paths
  - Current BGP techniques offer no control

Constraints
- Coexist with deployed IGP/EGP
- Allow incremental deployment
  - Incremental replacement of BGP
- Detect & avoid oscillations, divergence due to conflicts
- Be scalable
Prior Work

- Limit prefix length, NOPEER, flap limiting
  - Don’t solve underlying issue
Prior Work

- Limit prefix length, NOPEER, flap limiting
  - Don’t solve underlying issue
- MPLS / DiffServ based Intra-domain TE solutions
  - Would follow BGP routes
  - We don’t expect open MPLS clouds everywhere
Prior Work

- Limit prefix length, NOPEER, flap limiting
  - Don’t solve underlying issue

- MPLS / DiffServ based Intra-domain TE solutions
  - Would follow BGP routes
  - We don’t expect open MPLS clouds everywhere

- RON, Routing Arbiter, Nimrod
  - Unscalable in our scenario
Outline

- Introduction
  - BGP primer
  - Problem statement
  - Prior work: inadequate solutions

- OPCA
  - Overview
  - Completed components, protocol
  - Evaluation
Challenges

- How to design routing control structure?
Challenges

- How to design routing control structure?
  - Local optimization isn’t enough
  - Locus of control is remote

**Internet**

**AS**
Challenges

- How to design routing control structure?
  - Local optimization isn’t enough
    - Locus of control is remote
  - Global optimization unattainable
    - Computationally complex
    - Link state
      - Scalability is an issue
    - Full disclosure of policies bad
Challenges

- How to design routing control structure?
  - Local optimization isn’t enough
    - Locus of control is remote
  - Global optimization unattainable
    - Computationally complex
    - Link state
      - Scalability is an issue
    - Full disclosure of policies bad
  - Middle ground
    - Logically separate routing control plane
    - Find loci of control
    - Negotiate policy control
    - Adapt to non-responsiveness, network change
OPCA: Architecture

- AS W
- AS X
- AS Y
- AS V

EBGP

Internet

PD PA = Policy Agent
PD = Policy Database
MI = Measurement
MI = Measurement
Directory
Components of OPCA

- Policy database
  - Important ASes (e.g. $$ customers)
  - Local application servers
  - SLAs & pricing constraints

- Measurement infrastructure
  - Already exists in most ASes
  - E-BGP link info, customer-server traffic

- PA Directory
  - 1 or many (e.g. DNS)

- Relationship & Topology Map
  - 1 or many

  - Find likely route, transit / peering relationships
Components of OPCA

- Policy database
  - Important ASes (e.g. $$ customers)
  - Local application servers
  - SLAs & pricing constraints

- Measurement infrastructure
  - Already exists in most ASes
  - E-BGP link info, customer-server traffic
Components of OPCA

- Policy database
  - Important ASes (e.g. $$ customers)
  - Local application servers
  - SLAs & pricing constraints

- Measurement infrastructure
  - Already exists in most ASes
  - E-BGP link info, customer-server traffic

- PA Directory
  - 1 or many (e.g. DNS)

- Relationship & Topology Map
  - 1 or many
  - Find likely route, transit / peering relationships
<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA_locate(AS)</td>
<td>PA to PA directory request for address of PA in remote AS</td>
</tr>
<tr>
<td>PA_locate_reply(AS,ipaddr,port,timeout)</td>
<td>PA directory entry reply</td>
</tr>
<tr>
<td>PA_route(prefix)</td>
<td>PA to PA request for best route</td>
</tr>
<tr>
<td>PA_route_reply(prefix,AS_Path)</td>
<td>PA route reply</td>
</tr>
<tr>
<td>PA_block(prefix,AS1,AS2)</td>
<td>PA to PA request to block all routes for prefix</td>
</tr>
<tr>
<td>PA_block_reply(error_code,prefix,AS1,AS2)</td>
<td>PA block status reply</td>
</tr>
<tr>
<td>PA_select(prefix,AS1,AS2)</td>
<td>PA to PA request to select a particular route</td>
</tr>
<tr>
<td>PA_select_reply(error_code,prefix,AS1,AS2)</td>
<td>PA select status reply</td>
</tr>
</tbody>
</table>

- UDP control messages
  - Reverse path may not be available for session
- Direct PA to PA addressing
  - Don’t want BGP-like propagation
OPP: Protocol Messages

- UDP control messages
  - Reverse path may not be available for session
- Direct PA to PA addressing
  - Don’t want BGP-like propagation

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA_locate(AS)</td>
<td>PA to PA directory request for address of PA in remote AS</td>
</tr>
<tr>
<td>PA_locate_reply(AS,ipaddr,port,timeout)</td>
<td>PA directory entry reply</td>
</tr>
<tr>
<td>PA_route(prefix)</td>
<td>PA to PA request for best route</td>
</tr>
<tr>
<td>PA_route_reply(prefix,AS_Path)</td>
<td>PA route reply</td>
</tr>
<tr>
<td>PA_block(prefix,AS1,AS2)</td>
<td>PA to PA request to block all routes for prefix</td>
</tr>
<tr>
<td>PA_block_reply(error_code,prefix,AS1,AS2)</td>
<td>PA block status reply</td>
</tr>
<tr>
<td>PA_select(prefix,AS1,AS2)</td>
<td>PA to PA request to select a particular route</td>
</tr>
<tr>
<td>PA_select_reply(error_code,prefix,AS1,AS2)</td>
<td>PA select status reply</td>
</tr>
</tbody>
</table>
Example

- X uses X → F → E → B → A
- B → A breaks
  - B’s BGP session resets
  - B sends withdrawal to E & D
  - E receives withdrawal, selects D, announces to F
  - F selects new route through D
  - D sends withdrawal to E
  - E sends withdrawal to F
  - F selects route through C
Example

- X uses X → F → E → B → A
- B → A breaks
  - A notices drop in traffic
  - A’s PA queries RMAP
  - A’s PA queries PA directory
  - A’s PA sends block request to F’s PA
  - F selects route through C
Key Design Factors

- Inherent advantages of OPCA
  - Overhead of OPCA is fixed regardless of # of BGP hops
    - Control messages skip BGP propagation
  - OPCA does not experience per hop router delay
    - Control messages exchanged between PAs
    - Skip router delay, dampening

Avoid policy conflicts
Avoid oscillations
Key Design Factors

- Inherent advantages of OPCA
  - Overhead of OPCA is fixed regardless of # of BGP hops
    - Control messages skip BGP propagation
  - OPCA does not experience per hop router delay
    - Control messages exchanged between PAs
    - Skip router delay, dampening

- But
  - Avoid policy conflicts
  - Avoid oscillations
Outline

- Introduction
  - BGP primer
  - Problem statement
  - Prior work: inadequate solutions

- OPCA
  - Overview
  - Completed components, protocol
  - Evaluation
Evaluation Methodology

- Component analysis
  - Use real topologies, real BGP tables
  - Evaluate individual components
    - RMAP
    - Scalability
Evaluation Methodology

- Component analysis
  - Use real topologies, real BGP tables
  - Evaluate individual components
    - RMAP
    - Scalability

- Emulation
  - Evaluate existing BGP architecture (ongoing...)
  - Code complete PA, PD (ongoing...)
  - Evaluate OPCA (ongoing...)
RMAP Implemented

- Relationship & Topology Map
- INFOCOM 2002
- “Characterizing the Internet Hierarchy from Multiple Vantage Points”
RMAP Implemented

- Relationship & Topology Map
- INFOCOM 2002
- “Characterizing the Internet Hierarchy from Multiple Vantage Points”

Inferred Relationships for 23,935 AS Pairs

<table>
<thead>
<tr>
<th>Relationship</th>
<th># AS pairs</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider-customer</td>
<td>22,621</td>
<td>94.51%</td>
</tr>
<tr>
<td>Peer-peer</td>
<td>1,136</td>
<td>4.75%</td>
</tr>
<tr>
<td>Unknown</td>
<td>178</td>
<td>0.74%</td>
</tr>
</tbody>
</table>

Distribution of ASes in Hierarchy

<table>
<thead>
<tr>
<th>Level</th>
<th># of ASes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inner core (0)</td>
<td>20</td>
</tr>
<tr>
<td>Transit core (1)</td>
<td>129</td>
</tr>
<tr>
<td>Outer core (2)</td>
<td>897</td>
</tr>
<tr>
<td>Regional ISPs (3)</td>
<td>971</td>
</tr>
<tr>
<td>Customers (4)</td>
<td>8898</td>
</tr>
</tbody>
</table>
Scalability

Multihoming of Customer ASes

- No. of Multihomed Customer ASes
- No. of Doubly Homed Customer ASes
- No. of Triply Homed Customer ASes
- No. of >=4 Homed Customer ASes

Customers

Path Length

No.

01 Jan 1998
01 Jan 1999
01 Jan 2000
01 Jan 2001
01 Jan 2002

No. of Multihomed Customer ASes
Scalability

Different Paths to Innercore
Orthogonal Paths to Innercore (approx)
Scalability

- Not all stub ASes will use OPCA
  - About half can switch between 2 paths
  - To the core of 20 ASes
  - Also need to check orthogonality to 2nd tier
Not all stub ASes will use OPCA
- About half can switch between 2 paths
- To the core of 20 ASes
- Also need to check orthogonality to 2nd tier

May need a hierarchy of PAs inside a tier 1 ISP
- Will need to estimate control traffic
- Calculate rate of routing changes
Evaluation Methodology

- Emulation
  - Build evaluation platform (ongoing...)
    - 9 server setup
    - Dual 1.4Ghz, 1+GB memory
    - Gigabit fiber, gigabit ethernet networks
    - Connected via 52 Gbps Packetengine
    - Multiple SW BGP speakers per server
    - Different BGP session delays
    - Configure arbitrary topology
Evaluation Methodology

- Emulation
  - Build evaluation platform (ongoing...)
    - 9 server setup
    - Dual 1.4Ghz, 1+GB memory
    - Gigabit fiber, gigabit ethernet networks
    - Connected via 52 Gbps Packetengine
    - Multiple SW BGP speakers per server
    - Different BGP session delays
    - Configure arbitrary topology
  - Collect data to feed platform (ongoing...)
    - BGP collector part of Sprint’s internal-BGP network
    - Connects to 130+ routers
    - Store months of routing messages
    - Can be replayed on evaluation platform
Research Issues

Goal
- Reduce fail over time, finer grained traffic balancing

- Measure side effects
  - Table growth, flapping, traffic, scalability

- Deployment
  - Cooperative architecture, like BGP
  - Keep history of uncooperating PAs

- Distribution of PAs
  - Benefits leaf ASes
  - But need PA's in core (at aggregation points)

- Leaf ASes are customers of core
- Large benefits will create pressure

- More participants, better RMAP
Research Issues

- **Goal**
  - Reduce fail over time, finer grained traffic balancing

- **Measure side effects**
  - Table growth, flapping, traffic, scalability
Research Issues

- **Goal**
  - Reduce fail over time, finer grained traffic balancing
  - Measure side effects
    - Table growth, flapping, traffic, scalability

- **Deployment**
  - Cooperative architecture, like BGP
    - Keep history of uncooperating PAs
  - Distribution of PAs
    - Benefits leaf ASes
    - But need PA’s in core (at aggregation points)
      - Leaf ASes are customers of core
      - Large benefits will create pressure
  - More participants, better RMAP
Summary

- **Hypothesis**
  - Available, congestion adaptive connectivity is lacking
  - An overlay control plane can achieve this

- **Many interesting research issues**
  - How to balance local optimization and global optimization
  - Fail over time, load balancing, traffic impact, scalable, deployment, ...

- **Measureable success**
  - Real BGP tables and traffic patterns
  - BGP implementations in emulation testbed