APT: An Architecture for Practical Transit Core Separation

Michael Meisel
work with Dan Jen, Dan Massey, Lan Wang, Beichuan Zhang, and Lixia Zhang
APT is a Map & Encap Scheme

- Map & encap is one category of separation scheme
- Encapsulation is used in the transit core
- Needs a mapping service
- (LISP is a map & encap scheme, too)
Encap and Decap at Border Routers

- Encapsulating Routers (ERs)
  - Use mappings to encapsulate packets
  - Caches recently used mappings
- Decapsulating Routers (DRs)
  - Strip encapsulation header
- Generally both functions in one device (EDR)
What’s a Mapping?

- Maps an edge network’s prefix(es) to its transit core attachment points
  - Attachment points are providers’ border routers
  - This is a one-to-many mapping per prefix
- No reachability info, just topological info
Map & Encap Example

- **ER**: Source (1.1.1.4)
- **X**: Intermediate Node
- **DR**: Destination (a.b.c.d)

Packet flows from **ER** to **X** to **DR**.
Packet Arrives at ISP
Packet Encapsulated in Transit Core Header
Packet Delivered across Transit Core

s

ER

a.b.c.d
1.1.1.4

X

X

DR

Edge Networks
Transit Core
Packet Decapsulated

s
ER

X
X

X
X

d
DR

1.1.1.4

Edge Networks

Transit Core
Packet Delivered

s

ER

X

X

X

d

1.1.1.4

Edge Networks

Transit Core
The APT Philosophy: “Do No Harm”

- Make no changes at edge sites
- Maintain current network performance
- Minimize the amount of new infrastructure

End result
- ISPs can deploy APT unilaterally
- Cost is aligned with benefit
Make No Changes at Edge Sites

- LISP encapsulates packets at edge sites
- APT encapsulates packets at ISPs
Maintaining Current Network Performance with Default Mappers

- Default mappers store the full mapping table
- Each APT ISP has at least one
  - But no more than a few
- ERs cache only recently used mappings
  - Cache miss? Ask your DM
  - All mapping lookups occur intra-domain
APT Example

- **Site1**
  - ER1
  - X
  - 3.3.3.7

- **Site2**
  - DR1
  - X

- **ISP1**
  - ISP3
  - DR2
  - X

- **ISP2**
  - ISP3
  - DR2
  - X

- **ISP3**
  - ER1
  - X

- **Default Mappers**
  - M1
  - M2

- **Edge Networks**
  - Transit Core

- **Default Mappers**
  - M1
  - M2
Mapping Not in Cache

<table>
<thead>
<tr>
<th>Edge prefix</th>
<th>DR Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1.0/24</td>
<td>x.x.x.x</td>
</tr>
<tr>
<td>2.2.0.0/24</td>
<td>y.y.y.y</td>
</tr>
</tbody>
</table>

Cache Miss!
Encap with the Default Mapper
Anycast Address

Site1
ER1

ISP1
M1

Site2
DR1

ISP2
M2

ISP3
DR2

Edge Networks
Transit Core

3.3.3.7
Default Mapper Decaps the Packet

[Diagram showing network topology with nodes labeled Site1, Site2, ISP1, ISP2, ISP3, ER1, DR1, DR2, M1, M2, and X symbols indicating connections.]
Edge prefix is Multihomed

<table>
<thead>
<tr>
<th>Edge prefix</th>
<th>DR Addr</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>3.3.3.0/24</td>
<td>DR1</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>DR2</td>
<td>20</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Default Mapper Selects a Mapping

<table>
<thead>
<tr>
<th>Edge prefix</th>
<th>DR Addr</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>3.3.3.0/24</td>
<td>DR1</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>DR2</td>
<td>20</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

ISP1

ISP2

Site1

Site2

ER1

M1

M2

DR1

3.3.3.7

DR1

Priority

Edge prefix

DR Addr

ISP1

ISP2

Edge Networks

Transit Core
Default Mapper Responds with Mapping and Deliveres Packet

Site1

ISP1

ISP2

ISP3

Site2

DR1

DR2

Edge Networks

Transit Core

Cache Add Message

ISP1

3.3.3.7
Mapping Added to Cache

<table>
<thead>
<tr>
<th>Edge prefix</th>
<th>DR Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1.0/24</td>
<td>x.x.x.x</td>
</tr>
<tr>
<td>2.2.0.0/24</td>
<td>y.y.y.y</td>
</tr>
<tr>
<td>3.3.3.0/24</td>
<td>DR1</td>
</tr>
</tbody>
</table>

Site1

Site2

ISP1

ISP2

ER1

M1

M1

M2

DR1

DR1

3.3.3.7
Packet Decapsulated and Delivered

<table>
<thead>
<tr>
<th>Edge prefix</th>
<th>DR Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1.0/24</td>
<td>x.x.x.x</td>
</tr>
<tr>
<td>2.2.0.0/24</td>
<td>y.y.y.y</td>
</tr>
<tr>
<td>3.3.3.0/24</td>
<td>DR1</td>
</tr>
</tbody>
</table>

Site1

ISP1

ISP2

Site2

ER1

M1

DR1

M2

X

Edge Network

Transit Core
Next Packet

<table>
<thead>
<tr>
<th>Edge prefix</th>
<th>DR Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1.0/24</td>
<td>x.x.x.x</td>
</tr>
<tr>
<td>2.2.0.0/24</td>
<td>y.y.y.y</td>
</tr>
<tr>
<td>3.3.3.0/24</td>
<td>DR1</td>
</tr>
</tbody>
</table>

Site1

Site2

ER1

DR1

ISP1

ISP2

M1

M2

Edge Networks

Transit Core
Mapping Already in Cache

Site1

ER1

X

ISP1

M1

3.3.3.7

ISP2

M1

M2

DR1

Site2

Table:

<table>
<thead>
<tr>
<th>Edge prefix</th>
<th>DR Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1.0/24</td>
<td>x.x.x.x</td>
</tr>
<tr>
<td>2.2.0.0/24</td>
<td>y.y.y.y</td>
</tr>
<tr>
<td>3.3.3.0/24</td>
<td>DR1</td>
</tr>
</tbody>
</table>

Legend:
- Edge Networks
- Transit Core
# Packet Encapsulated by ITR

## Diagram

- **Site1**
- **Site2**

## Table: Edge prefix vs DR Address

<table>
<thead>
<tr>
<th>Edge prefix</th>
<th>DR Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1.0/24</td>
<td>x.x.x.x</td>
</tr>
<tr>
<td>2.2.0.0/24</td>
<td>y.y.y.y</td>
</tr>
<tr>
<td>3.3.3.0/24</td>
<td>DR1</td>
</tr>
</tbody>
</table>

## Edges

- ER1 connects to ISP1.
- DR1 connects to ISP2.
- M1 and M2 are located in Site1 and Site2 respectively.

## Edge Networks and Transit Core

- **Edge Networks**
- **Transit Core**
Packet Delivered Directly to ETR

<table>
<thead>
<tr>
<th>Edge prefix</th>
<th>DR Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1.0/24</td>
<td>x.x.x.x</td>
</tr>
<tr>
<td>2.2.0.0/24</td>
<td>y.y.y.y</td>
</tr>
<tr>
<td>3.3.3.0/24</td>
<td>DR1</td>
</tr>
</tbody>
</table>

- Site1
- Site2
- ER1
- DR1
- ISP1
- ISP2
- M1
- M2

Edge Networks
Transit Core
Minimizing New Infrastructure

- DMs store the entire mapping table locally
- No third party infrastructure for mapping lookups
- So how do DMs get the table?
Mapping Dissemination

- DMs need to learn other ISPs’ mapping info
- DMs form an overlay
- ISPs flood their mapping info through the overlay
In Defense of Flooding

- Extremely low latency for mapping lookups
- “Aren’t you just moving the problem from the routing system to the mapping system?”
  - No!
  - Mapping info doesn’t contain reachability info
  - Only changes when customer-provider relationship changes
  - The dissemination path isn’t important
Aligning Cost with Benefit

- EDRs and DMs are both deployed at ISPs
- An ISP can have as many or few DMs as it likes
- Each ISP maintains its own mapping table
- One ISP can turn on APT on its own
APT Incremental Deployment

By Dan Jen
APT Deployment Incentives

- APT *can* be deployed unilaterally by an ISP
  - But *why* would an ISP want to do this?

- Answer: Routing Table Reduction!
Routing Table Size is a Problem

- Currently, all routers in an ISP store every prefix in the Internet (currently 300k) in memory.
- Router hardware improvements can't keep up with growing number of prefixes.
  - [http://www.vaf.net/~vaf/apricot-plenary.pdf](http://www.vaf.net/~vaf/apricot-plenary.pdf)

APT Can Help!
Router Storage Under APT

- 3 types of routers
  - Regular, DRs, and Default Mapper nodes
- All routers store 1 prefix per APT ISP
  - Less than 100 even under universal deployment
- DRs also store a small cache of mappings, and prefixes from directly-connected peers
- Default mapper SYSTEM must also store full mapping table as well as full routing table.
  - But DM system can consist of many nodes
DM Distributed Storage
= Memory Savings

- Assume M prefixes on the Internet, mapped and unmapped.
- Assume N nodes make up the DM system
- Each DM node stores M/N prefixes total, mapped and unmapped.
  - Savings!
- All other routers only store about 100 prefixes.
  - Savings!
IP addressing primer

- 32-bit addresses
- 4 sections, separated by '. '
- Each section represents 8 bits, from 0-255
- Ex: 5.255.8.24
  - 00000101 11111111 00001000 00011000
IP addressing primer

- To represent addresses from 1.2.3.0 to 1.2.3.255, we use 1.2.3.0/24, meaning:
  - All addresses whose first 24 bits are 1.2.3
- Thus we can divide the entire IP address range into 4 equal parts:
  - 0.0.0.0/2
  - 64.0.0.0/2
  - 128.0.0.0/2
  - 192.0.0.0/2
- numbered routers are DRs
- A, B, C, D make up the DM system
APT 1st Mover Savings

- Only A and 2 store this route
- A knows 2 is the egress point for address range
- 2 knows the real external next hop for the prefix
APT 1st Mover Packet Delivery

Diagram:

- Node 1: 0.1.2.3, 128.0.0.0/2
- Node 2: 0.0.0.0/2, 64.0.0.0/2, 192.0.0.0/2

Nodes:
- A
- B
- C
- D

Networks:
- 0.0.0.0/2
- 128.0.0.0/2
- 64.0.0.0/2
- 192.0.0.0/2
APT 1st Mover Packet Delivery

0.0.0.0/2

64.0.0.0/2

128.0.0.0/2

192.0.0.0/2
APT 1\textsuperscript{st} Mover Packet Delivery

0.0.0.0/2

64.0.0.0/2

1.2.3.4

128.0.0.0/2

192.0.0.0/2
APT 1\textsuperscript{st} Mover Packet Delivery

1.2.3.4

A

B

C

D

1.2.3.4

0.0.0.0/2

64.0.0.0/2

128.0.0.0/2

192.0.0.0/2
APT 1st Mover Packet Delivery

- 0.0.0.0/2
- 64.0.0.0/2
- 128.0.0.0/2
- 192.0.0.0/2

1.2.3.4

1

2
APT 1st Mover Packet Delivery
The Stretch Issue

- Note that packets traveling through an APT ISP will travel extra hops before exiting the AS
- Every APT ISP will stretch packets traversing through it. Stretch can add up.
- Dest is customer of non-APT ISP
- All blue ISPs are APT ISPs
The Stretch Issue
2nd Mover Incentive

- A 2nd ISP might want to adopt APT for all the same benefits as 1st mover
- But also, ISP can now avoid “the stretch issue”
  - APT ISP can join overlay
  - Mappings ensure that packets stretched at most once.
  - No stretch on cache hits
- Now, Dest's Provider deploys APT
2\textsuperscript{nd} Mover Incentive

- This assumes a cache miss.
- Cache hit would add no stretch at all.
Thank You!

- Questions?
- Comments?