Suggestions Towards Next Steps in RRG

Christian Vogt

Routing research group meeting at IETF 73. November 2008
Solution Approaches: Host vs. Network

- seeking for a one-size-fits-all solution
  - to enable multi-homing and failover
  - to eliminate renumbering
## Solution Analysis

<table>
<thead>
<tr>
<th></th>
<th>benefits</th>
<th>technical maturity</th>
<th>deployability</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>network-based solutions</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>proxying (LISP, Ivip)</td>
<td>no renumbering multi-homing</td>
<td>longer path</td>
<td>lack of incentives</td>
</tr>
<tr>
<td>translation (Six/One Router)</td>
<td>no renumbering multi-homing if bilateral support</td>
<td>NAT-like effect</td>
<td>clear incentives</td>
</tr>
<tr>
<td><strong>host-based solutions</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID/address split (HIP, ILNP)</td>
<td>easier renumbering multi-homing</td>
<td>well understood</td>
<td>small incentives</td>
</tr>
<tr>
<td>address indirection (Shim6, Six/One)</td>
<td>multi-homing only – requires renumbering</td>
<td>well understood</td>
<td>small incentives</td>
</tr>
</tbody>
</table>

- all solutions with strengths and weaknesses
- solutions can complement each other
Proposal of “Dual Approach”

- multi-homing in hosts
- eliminate renumbering in network

host-based + network-based solutions
- independent of each other
- complementary in benefits
1st Solution Part: Host-Based Multi-Homing

- Application transparency implies deployment hurdle
- API evolution proves desire for transparency unfounded
Hostname-Oriented Stack

- backwards compatibility — not transparency
  - simpler stack architecture
  - easier application programmability (indirection in stack)
Hostname-Oriented Stack

- explicit service names supersede well-known port numbers
- session names = port numbers without service semantics
- regular IP packets on wire
Session Establishment Example

host.left.net has address a.b.c.d

peer.right.net has address v.x.y.z

accept from any to peer.right.net using HTTP
Session Establishment Example

**host.left.net** has address **a.b.c.d**

**peer.right.net** has address **v.x.y.z**

---

**Application**

- **connect** from **host.left.net** to **peer.right.net** using **HTTP**
- **resolve** **peer.right.net**
- **allocate session name**

---

**Stack**

- **source** **a.b.c.d** destination **v.x.y.z**
- session source **101** from **host.left.net** to **peer.right.net** using **HTTP**

---

**Application**

- **accept** from **any** to **peer.right.net** using **HTTP**
- **verify** **peer.right.net**
- **allocate session name**
Session Establishment Example

**host.left.net** has address **a.b.c.d**

**peer.right.net** has address **v.x.y.z**

---

**application**  
**connect** from **host.left.net** to **peer.right.net** using HTTP

**resolve** **peer.right.net**

**allocate session name**

---

**stack**  
source **a.b.c.d** destination **v.x.y.z**

session source **101** from **host.left.net** to **peer.right.net** using HTTP

source **v.x.y.z** destination **a.b.c.d**

session source **20** destination **101** from **peer.right.net** to **host.left.net** using HTTP

---

**application**  
**accept** from **any** to **peer.right.net** using HTTP

**verify** **peer.right.net**

**allocate session name**

---

**application**  
**connect** from **host.left.net** to **peer.right.net** using HTTP

**allocate session name**
Session Establishment Example

```
host.left.net has address a.b.c.d
peer.right.net has address v.x.y.z
```

application

connect from host.left.net to peer.right.net using HTTP

stack

source a.b.c.d destination v.x.y.z
session source 101 from host.left.net to peer.right.net using HTTP

allocate session name

resolve peer.right.net

allocate session name

connected from host.left.net to peer.right.net using HTTP

stack

accept from any to peer.right.net using HTTP

verify peer.right.net

application

source v.x.y.z destination a.b.c.d
session source 20 destination 101 from peer.right.net to host.left.net using HTTP

accepted from host.left.net to peer.right.net using HTTP
Further Considerations

- connection-less protocols
- bootstrapping protocols
- anonymous protocols
- mobility support
- middlebox support
## Multiply Benefits Ease Deployment

<table>
<thead>
<tr>
<th>Feature</th>
<th>for users</th>
<th>for host administrators</th>
<th>for application developers</th>
<th>for OS vendors</th>
<th>for network operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>human-readable hostnames</td>
<td>yes</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>multi-homing support</td>
<td>yes</td>
<td></td>
<td>yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mobility support</td>
<td>yes</td>
<td></td>
<td>yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>no renumbering of hosts</td>
<td></td>
<td></td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>no new layer of indirection</td>
<td></td>
<td></td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>no new infrastructure</td>
<td>yes</td>
<td></td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>addressing functions in stack</td>
<td></td>
<td></td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>better middlebox support</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td>yes</td>
</tr>
</tbody>
</table>
2nd Solution Part: Avoid IPv6 Renumbering

- could simply be unilateral IPv6 prefix translation
  - reachability via 1-to-1 mappings
  - robust via statelessness
  - transparent to applications via hostname-oriented stack

- as easily deployable as NATs
  - no external dependencies
  - local affects only
Conclusion

- dual approach most reasonable
  - exploit strengths of either approach
  - align costs with benefits

- possible dual approach
  - multi-homing + more via hostname-oriented stack
  - no renumbering in networks via prefix translation