Supporting Legacy Applications over Routing Overlays

Karthik Lakshminarayanan
UC Berkeley
(with Jayanthkumar Kannan, Ayumu Kubota, Ion Stoica and Klaus Wehrle)
Motivation

• Overlays have received much attention, yet…
  – no common interface
  – not many native applications
  – not many deployments
  – chicken-and-egg problem?

• Our solution
  – interface legacy applications with overlay

• Overlay model
  – routing overlays, e.g. RON, Detour, OverQoS
  – each end-host has an overlay ID
Design Goals

• Allow users to express preferences
  – which overlay should be used for the flow
  – preferences specific to overlays, e.g. middleboxes to be used

• Transparency to legacy applications
  – oblivious to legacy applications

• Provide flexible deployment options
  – co-located with applications
  – run on a remote host
Address Virtualization

- Capture packets and tunnel over the overlay
- DNS names to express preferences flexibly
<table>
<thead>
<tr>
<th>Application tasks</th>
<th>Proxy tasks</th>
<th>Proxy state</th>
</tr>
</thead>
<tbody>
<tr>
<td>gethostbyname(name)</td>
<td>1. Name lookup</td>
<td>Address book</td>
</tr>
<tr>
<td>address virtualization component</td>
<td>2. Virtual address negotiation</td>
<td>home.foo.i3 (\rightarrow id1)</td>
</tr>
<tr>
<td>overlay handler component</td>
<td>3. Overlay control protocol</td>
<td>bar.i3 (\rightarrow id2)</td>
</tr>
<tr>
<td>virtual address IP_{ab}</td>
<td>4. Data tunneling</td>
<td></td>
</tr>
<tr>
<td>send(IP_{ab}, data)</td>
<td></td>
<td>Overlay state</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IP_{ab} (\rightarrow id1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>id1 (\rightarrow IP_{ab})</td>
</tr>
</tbody>
</table>
Transparency

- Goal: oblivious to legacy applications
- IP addresses are global
  - SIP-based applications
- IP addresses are permanent
  - apps cache address
- IP addresses in packet headers are unmodified end-to-end
  - ftp, H.323
Flexible Deployment Options

• Local proxy at both ends
  – IP ➞ ➞ IP communication over overlay

• Remote server proxy
  – useful for legacy servers
  – proxy inserts triggers, performs flow setup on behalf on legacy servers

• Remote client proxy
  – useful for legacy clients
  – DNS server returns address of remote client proxy which performs flow setup
Application Scenarios

Insert middle-box apps

i3

remote client proxy

remote server proxy

local proxy

local proxy

local proxy

local proxy

CNN.com

Communicate with i3-unaware hosts via i3

Transparent access across Firewall/NAT

mobility
Limitations

• Incorrect DNS caching by apps
  – caching despite zero TTL
• Semantics of addresses not completely preserved
  – virtual addresses cannot be shared between hosts
  – applications based on SIP might not work
• Not all routing overlays can be supported
Resources

• Implemented for Linux and Windows platforms
• http://i3.cs.berkeley.edu/