Overlay Routing (Routing Underlay)
Overlay Networks
End System Multicast
ESM (cont)
Distributed Hash Tables

\[2^{128} - 1 | 0\]

objid

nodeids

September 29, 2003
DHT (cont)

addnode(d46a1c)
Resilient Overlay Networks
RON (cont)

BGP inefficiencies

– Poor Metrics
  ▪ minimize AS hops
– Long failover times
  ▪ measured in minutes
– Manual Load Balancing
  ▪ configuration errors
– Single Path
  ▪ under-utilize alternate paths
RON (cont)

Improvement for roughly 40%
25% shorter latency for 15%
RON (cont)

Fraction of paths measured

Packet Loss Rate

Default
Alternate
Problem

• Discovering efficient topology requires expensive/disturbing network probes

• Single overlay network
  – aggressive probing does not scale (RON)

• Multiple overlay networks
  – Redundant probing to discover the same topological information
  – 1GB-per-day of ping traffic on PlanetLab
    ▪ one ping-per-sec-per-node across 125 nodes
Routing Underlay

- Sits between overlays and the Internet
- Exposes topological information
  - already collected by the Internet (BGP tables)
  - caches active measurements
- Enables cost-effective network probes
  - primitives: interface to shared probes
  - layered architecture: hierarchical probes
Hierarchical Probes

Service Overlay Networks
Use more expensive probes, but in limited scope

Library of Routing Services
Probe efficiently using static data as a hint

Primitives

Topology Probing Kernel
Collect passive data / Cache expensive probes

Raw Topology Information

Expense

Scope
Primitives

- GetGraph (resolution, scope) _ connectivity
- GetPath (resolution, from, to) _ route
- GetDistance (metric, from, to) _ distance

resolution = AS level, router level, ....
scope = entire network, within an ISP, ....
metric = AS hop, router hop, RTT ....
Primitives at AS Level

• Resolution = AS Level
  – GetGraph ➔ AS Peering Graph
  – GetPath ➔ AS Path
  – GetDistance ➔ AS hop count

• Helpers
  – GetPrefixMap ➔ IP to AS translation
  – …
Peering Graph Completeness

The number of distinct edges in BGP table
The number of edges each vantage point contribute

Cumulative number of vantage points

Number of distinct edges included

September 29, 2003
Degree Distribution

Distribution of Outdegrees of ISPs

Edge AS (degree\(\leq 5\)) : 92 %
Regional AS (5<degree\(\leq 100\)) : 7.5%
TransContinental AS (degree>100) : 0.5%
## Degree Distribution

<table>
<thead>
<tr>
<th>Degree (# of peers)</th>
<th>Cumulative Distribution (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>33.26</td>
</tr>
<tr>
<td>2</td>
<td>76.55</td>
</tr>
<tr>
<td>3</td>
<td>86.56</td>
</tr>
<tr>
<td>5</td>
<td>92.26</td>
</tr>
<tr>
<td>10</td>
<td>96.13</td>
</tr>
<tr>
<td>20</td>
<td>98.20</td>
</tr>
<tr>
<td>50</td>
<td>99.30</td>
</tr>
<tr>
<td>99</td>
<td>99.65</td>
</tr>
</tbody>
</table>
# Top 10 players

<table>
<thead>
<tr>
<th>Degree</th>
<th>ASN</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>2554</td>
<td>701</td>
<td>UUNET Technologies, Inc.</td>
</tr>
<tr>
<td>1733</td>
<td>1239</td>
<td>Sprint</td>
</tr>
<tr>
<td>1502</td>
<td>7018</td>
<td>AT&amp;T WorldNet Services</td>
</tr>
<tr>
<td>890</td>
<td>209</td>
<td>Qwest</td>
</tr>
<tr>
<td>798</td>
<td>3561</td>
<td>Cable &amp; Wireless USA</td>
</tr>
<tr>
<td>621</td>
<td>1</td>
<td>Genuity</td>
</tr>
<tr>
<td>589</td>
<td>702</td>
<td>UUNET Technologies, Inc</td>
</tr>
<tr>
<td>589</td>
<td>3549</td>
<td>Global Crossing</td>
</tr>
<tr>
<td>545</td>
<td>3356</td>
<td>Level 3 Communications, LLC</td>
</tr>
<tr>
<td>541</td>
<td>2914</td>
<td>Verio, Inc.</td>
</tr>
</tbody>
</table>

September 29, 2003
Library of Routing Services

Service Overlay Networks

Cost Effective Probes

Primitives

Topology Probing Kernel

Raw Topology Information

Expense
Library of Routing Services

• Strategy for efficient probes: Guess & Verify
  – Guess candidate solutions using inexpensive probes over a large scope
  – Verify them with more expensive probes in a limited scope

• Example Library Services
  (1) Nearest neighbors (DHT-based routing)
  (2) Edge-disjoint paths (multi-path routing)
  (3) Physically representative mesh (RON, ESM)
Nearest Neighbors

\[ \text{NodeSet} = \text{NearestNodes}(N,k) \]

- \( N \): a given set of nodes
- \( k \): the number of neighbors
- \( \text{NodeSet} \): a set of \( k \) nodes in \( N \) that are closest to the local overlay node, in terms of latency.
Nearest Neighbors (cont)

Basic idea:

Use weak correlation between latency and AS hops

For a given local node \( u \), …

(1) Sort \( w \)’s in \( N \) according to AS hop count (GetPath(\( u,w \))) into a candidate sequence \( \{w’s\} \)

(2) Invoke GetDistance(\( u,w,ping \)) on the first \( j \) nodes in \( \{w’s\} \), and select the best \( k \) nodes as nearest neighbors (\( j>k \))
Nearest Neighbors (cont)

Among 81 Nodes (PlanetLab + traceroute servers)

Cumulative Distribution

Number of Candidate Nodes

Using AS Path Length
Random
Disjoint Paths

PathSet = DisjointPaths \((u, v, N, k)\)

- \(u, v\): a given pair of overlay nodes
- \(N\): a set of intermediate nodes
- \(k\): the number of disjoint paths

A single-hop indirection \((u, w, v)\) that is edge-disjoint to \((u, v)\) at the AS level
Disjoint Paths (cont)

How often we find disjoint paths using a single hop indirection?

- Examined 1235 paths between multi-homed ASes (from RouteViews; 42 vantage points)
- 93.7% have at least one disjoint path
Disjoint Paths (cont)

Among 1235 paths (42 nodes) from RouteView, 93.7% have at least one disjoint path.
Local node $u$ is looking for alternate paths to $v$ ...

(1) Guess $w$’s likely to produce disjoint paths
   - GetPath($u,v$) and GetGraph

(2) Verify path ($u,w,v$) is disjoint to path ($u,v$)
   - GetPath($u,w$) and GetPath($w,v$)
Disjoint Paths (cont)

Among 1235 paths (among 42 nodes) from RouteViews

Cumulative Distribution (%)

Our method  Random Query  90 percentile

Number of Candidates

Among 1235 paths (among 42 nodes) from RouteViews

random  our method

90 percentile
Disjoint Paths (cont)

Number of Traceroutes to Find Disjoint Paths

Cumulative Distribution (%)

Among 81 Nodes (PlanetLab + traceroute servers)

Number of Candidates

random

inference
Representative Mesh

\[ Mesh = \text{BuildMesh}(N) \]

\( N \) : a set of overlay nodes

\( Mesh \) : a graph that retains only independent edges in the underlying network

Virtual link \((u,v)\) is redundant
Mesh (cont)

Local node $u$ checks virtual links to the other $v$’s.

(1) Guess $w$’s likely to conform to either topology
   - GetPath($u, v$) and GetGraph

(2) Verify the topology
   - GetPath($u, w$) and GetPath($w, v$)
Mesh (cont)

Using A+B
Using A

Peering Graph from RouteViews+Planetlab

Reduction (%) in the number of edges

Number of nodes

algorithm1
algorithm2
BuildMesh on PlanetLab

Mesh with $N=5$ nodes (PU, Duke, ISI, UW, Abilene)
Todo

- Modify RON, ESM, DHT to use underlay
- Discriminate among nodes in transit ASes
- Build sparser mesh(es) on top of AS-based mesh
- Develop better cost/benefit model
- Get more BGP feeds (or fake it)
- Implement finer-grain resolutions
- Implement an “IPvN” service