Motivation

- Types of measurements
  - Understand the topology of the Internet
  - Measure performance characteristics

- Tools:
  - BGP Tables
  - Traceroute measurements
  - Probes for latency, bandwidth and loss-rate
Economics of the Internet

Transit relationship:
- Reveal customers to everyone
- Reveal all known paths to customers
- ISP Y will advertise paths to C2 and C3

Peering Relationships

ISP Y peers with ISP Z
- Y informs Z of C2 and C3
- Z informs Y of C4
Peering Relationships

- ISP Y peers with ISP X
  - Y informs X of C2 and C3
  - X informs Y of C1
  - Y does not tell X of the path to C4
  - Y needs to have transit relationship with P to route to C4

Multi-Exit Discriminator

- C has a transit relationship with P
- C needs to send a packet into P's domain
  - Packet origin: Boston, packet destination: SF
  - It has two choices: transit into P earlier (near Boston) or transit later (near SF)
  - Prefer early transit
Routing Preferences

- In general, customer > peer > provider
  - Use LOCAL PREF to ensure this
  - Routing through customer and peer could lower latency and lower traffic costs
- Processing order of path attributes:
  - Select route with highest LOCAL-PREF
  - Select route with shortest AS-PATH
  - For routes learned from same neighbor:
    - Apply multi-exit discriminator

Routing algorithms for the internet

- Link state or distance vector?
- Problems with link state:
  - LS database too large – entire Internet
  - Metric used by routers not the same
  - May expose policies to other AS’s

- Can we use distance-vector algorithms for policy routing?
Distance Vector with Path

- Each routing update carries the entire path
- Loops are detected as follows:
  - When AS gets route check if AS already in path
    - If yes, reject route
    - If no, add self and (possibly) advertise route further
- Advantage:
  - Metrics are local - AS chooses path, protocol ensures no loops
- Hop-by-hop Model
  - BGP advertises to neighbors only those routes that it uses
    - Consistent with the hop-by-hop Internet paradigm
  - e.g., AS1 cannot tell AS2 to route to other AS's in a manner different than what AS2 has chosen (need source routing for that)
Terminology

- Each POP is a physical location where the ISP houses a collection of routers.
- The ISP backbone connects these POPs, and the routers attached to inter-POP links are called backbone or core routers.
- Within every POP, access routers provide an intermediate layer between the ISP backbone and routers in neighboring networks.
Points of Presence and Backbone

Rocketfuel Methodology

- ISPs release "helpful" information:
  - BGP - which prefixes are served
  - Traceroute - what the paths are
  - DNS - where routers are and what they do
- Build detailed maps:
  - Backbone
  - POPs
  - Peering links
Traceroutes

- Publicly available traceroute servers
- Challenge: To build accurate ISP maps using few measurements
- Brute Force Method
  - 784 vantage points to 120,000 allocated prefixes in BGP table
  - Queried every 1.5 minutes: 125 days to complete a map.
- Take all paths and use portions of the paths to determine structure of target AS

Directed probing

- Capitalize on BGP routing information
- Identify traceroutes which transit the ISP network

Goal: map AS 7

Need to consider only those paths that go through AS 7
1) Probes to dependent prefixes traverse AS 7
   Dependent Prefixes: 4.5.0.0/16
2) Probes from insiders to outside in dependent prefixes
3) Up/down traces: AS 11 to 1.2.3.0/24
Path Reductions

Paths to P1 and P2 are through the same AS on the way to the same destination.

Ingress Reduction  Egress Reduction  Next-hop AS Reduction

Reduction Effectiveness

- Brute force: 90-150 million traceroutes required
- BGP directed probes: 0.2-15 million traceroutes required
- Executed after path reduction: 8-300 thousand traceroutes required
Location and Role Discovery

- Where is this router located?
  - use DNS names
    - S1-bb11-nyc-3-0.sprintlink.net is a Sprint router in New York City
  - use connectivity information
    - if a router connects only to routers in Seattle, it is in Seattle

- What role does this router play in the topology?
  - only backbone routers connect to other cities
  - use DNS names
    - s1-gw2-sea-3-1.sprintlink.net is a Sprint gateway router

Alias resolution problem

A well-known problem for Internet mapping.

Because traceroute lists IP’s, we might think that the path to www.cs and the path from www.cs are different.

Alias resolution finds IP’s that belong to the same router.

www.cs.washington.edu
**Level 3**

**Internet Measurements**

Now we have router-level topology in an ISP

More information: link weights?
- Connectivity is not enough to derive best path
- Link weights in an ISP is not public

Inferring link weights provides
- a simple, concise, and useful model of intra-domain routing.
  - Common routing protocols choose least-cost paths using link weights

It extends router-level ISP maps
- Includes connectivity
- Link weights consistent with routing
Problem Formulation

- Input
  - Network topology
  - Routing as a set of chosen paths (lowest weight)
- Output:
  - Assign weights for each link, so that
    - shortest paths match chosen paths
  - Not a unique solution
    - Scaling all weights with same factor
    - Changing the weight of a link within bounds

Basic Solution

- Two key observations
  - Chosen path has less weight than any other path between two nodes
    - If multiple chosen paths exist, they have equal weight
Basic solution

- Chosen paths between A-G
  - ADG and ABEG
- Similar constraints for chosen paths between all other node-pairs
- Linear programming
- Shortcomings?

![Diagram showing chosen paths: ADG and ABEG, and their constraints.]

**Fig. 1.** The chosen paths, ADG and ABEG, are shown with solid lines, and the alternate paths with dashed lines.

1. \( w_{ad} + w_{dg} = w_{ab} + w_{be} + w_{eg} \)
2. \( w_{ad} + w_{dg} < w_{ae} + w_{eg} \)
3. \( w_{ad} + w_{dg} < w_{ac} + w_{cf} + w_{fg} \)
4. \( w_{ad} + w_{dg} < w_{ab} + w_{bd} + w_{dg} \)
5. \( w_{ad} + w_{dg} < w_{ad} + w_{de} + w_{eg} \)
6. \( w_{ad} + w_{dg} < w_{ab} + w_{bd} + w_{de} + w_{eg} \)

Reduce Constraints

- Chosen Paths:
  - S-M-N-D, S-A-I, I-B-D
- Weight Constrains
  - SMND < SAI + IBD
  - SAI < SXI
  - IBD < IYD
  - SMND < SXI + IYD will be redundant
    - All alternative paths from S to D through I, except path SAIBD
- Actual number of constraints is much lower than \( n^3 \)
Shortcomings In Measurement Data

- Two shortcomings from using traceroute measurements
  - Some paths might not be shortest path due to transient events (e.g. failures)
  - Not all chosen paths can be observed

Approach to resolve the first shortcoming

- To associate error variables with constraints
  - Associate with error variables for observed paths
    - 1. $W_{ad} + W_{dg} - e_{adg} = W_{ab} + W_{be} + W_{cg} - e_{abeg}$
    - 2. $W_{ad} + W_{dg} - e_{adg} < W_{ac} + W_{cg}$
  - Similar constraints for all node-pairs
  - Using the simplex algorithm to solve the constraint system
  - Minimize the weighted sum of the error variables
Approach to resolve the second shortcoming

- **Pair-wise Completeness**
  - The fraction of vertex-pairs between which at least one path was observed

- **Two techniques to improve pair-wise completeness**
  - Path symmetry
    - Reverse path of a shortest path is shortest
  - Optimal substructure property
    - Any subset of a shortest path is shortest

Applicability

- Applicable for any network with weighted shortest path routing with a single setting of link weights
  - The preferred path between two nodes is dependent only on the weights

- **Data source completeness**
  - Maps collected by Rocketfuel have high pair-wise completeness

<table>
<thead>
<tr>
<th>AS</th>
<th>Name</th>
<th>Rtrs</th>
<th>Links</th>
<th>Paths</th>
<th>Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1221</td>
<td>Telstra (au)</td>
<td>115</td>
<td>153</td>
<td>20K</td>
<td>88%</td>
</tr>
<tr>
<td>1239</td>
<td>Sprint (us)</td>
<td>323</td>
<td>972</td>
<td>214K</td>
<td>54%</td>
</tr>
<tr>
<td>1755</td>
<td>Ebone (eu)</td>
<td>88</td>
<td>161</td>
<td>15K</td>
<td>57%</td>
</tr>
<tr>
<td>3257</td>
<td>Tiscali (eu)</td>
<td>164</td>
<td>328</td>
<td>9K</td>
<td>66%</td>
</tr>
<tr>
<td>3967</td>
<td>Exodus (us)</td>
<td>80</td>
<td>147</td>
<td>27K</td>
<td>68%</td>
</tr>
<tr>
<td>6461</td>
<td>Abovenet (us)</td>
<td>145</td>
<td>376</td>
<td>88K</td>
<td>63%</td>
</tr>
</tbody>
</table>
AS Commercial Relationships

- Provider-customer:
  - customer pays its provider for transit services
- Peer-peer:
  - exchange traffic between customers
  - no exchange of money
- Sibling-sibling:
  - have mutual transit agreement
  - merging ISPs, Internet connection backup
- However, AS relationships are not public!

AS Relationship Graph

AS1 — AS2 — AS3 — AS4 — AS5 — AS6 — AS7

- provider-to-customer edge
- peer-peer edge
- sibling-sibling edge
Route Propagation Rule

- An AS or a set of ASes with sibling relationship does not provide transit services between any two of its providers and peers.
- BGP routing table entries have certain patterns.

Routing Table Entry

<table>
<thead>
<tr>
<th>Network</th>
<th>Next hop</th>
<th>AS Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.24.0/21</td>
<td>134.24.127.3</td>
<td>1740 1 i</td>
</tr>
<tr>
<td></td>
<td>194.68.130.254</td>
<td>5459 5413 1 i</td>
</tr>
<tr>
<td></td>
<td>158.43.133.48</td>
<td>1849 704 702 701 1 i</td>
</tr>
<tr>
<td></td>
<td>193.0.0.242</td>
<td>3333 286 1 i</td>
</tr>
<tr>
<td></td>
<td>144.228.240.93</td>
<td>1239 1 i</td>
</tr>
</tbody>
</table>
Routing Table Entry Patterns

Basic Algorithms

- Heuristics:
  - Top provider has largest degree
  - Based on patterns on BGP routing table entries
    - Consecutive AS pairs on the left of top provider are customer-to-provider or sibling-sibling edges
    - Consecutive AS pairs on the right of top provider are provider-to-customer or sibling-sibling edges
Initialize Consecutive AS Pair Relationship

Maximum degree AS