Sibyl
A Practical Internet Route Oracle

Ítalo Cunha¹, Pietro Marchetta², Matt Calder³, Yi-Ching Chiu³
Brandon Schlinker³, Bruno Machado¹, Antonio Pescapè²
Vasileios Giotsas⁴, Harsha Madhyastha⁵, Ethan Katz-Bassett³
The Art of Network Troubleshooting

1. Operator has routing problem
2. Runs traceroute to the destination
3. Needs to identify where the problem is
The Art of Network Troubleshooting

1. Operator has routing problem
2. Runs traceroute to the destination
3. Needs to identify where the problem is
   "Are routes through GTT in Seattle experiencing problems?"
   "Are routes through Level3 in LA experiencing problems?"
1. Operator has routing problem
2. Runs traceroute to the destination
3. Needs to identify where the problem is

"Are routes through GTT in Seattle experiencing problems?"

"Are routes through Level3 in LA experiencing problems?"
Operators Have Complex Questions

Someone suggests problem is on link between NTT and GTT in Seattle. Which routes use that link and could be impacted?
Operators Have Complex Questions

Someone suggests problem is on link between NTT and GTT in Seattle. Which routes use that link and could be impacted?

If route to destination D is impacted, which of my providers have a route that avoids that link?

Providers in my region with routes that avoid that link?
2. Runs traceroute to the destination

3. Needs to identify where the problem is

"Are routes through GTT experiencing problems?"
"Are routes through Level3 experiencing problems?"

Seeing similar issues between a few US, Asia and EU sites. What I'm seeing is what appears to be a congested peering connection between GTT and NTT in Seattle as that is what all traceroutes have in common.

Our graphs indicate this started Jan 1st, around ~ 15:00 utc.

Example traceroute from Seattle (ntt) to Vancouver (gtt):

```
HOST: rtr1-re0.sea
1. xe-0-0-0-34.r04.sttlwa01.us.  0.0%  60  4.1  8.3  0.8  23.9  6.5
2. ae3.sea22.ip4.gtt.net    20.0%  60  44.6  41.9  33.6  60.3  6.0
3. xe-1-2-0.van10.ip4.gtt.net  18.3%  60  48.9  51.6  38.0  129.1  17.3
4. opendns-gw.ip4.gtt.net    20.0%  60  46.6  46.6  38.3  65.1  6.9
5. rtr1.yvr.opendns.com     28.3%  60  44.8  47.5  38.6  74.2  8.9
```
Problems with the Current Approach

1. Takes time

2. Requires help from others
   a. Needs operator with right VP to respond
   b. Destination to measure may not be obvious

3. Limits analyses
The Art of Internet Measurement

1. Researcher wants to study, for example, boomerang routes
Researchers Have Complex Questions

What routes from the US to the US detour through Canada?
Researchers Have Complex Questions

What routes from the US to the US detour through Canada?

Eastward routes detouring through Canada?

Routes from Southern US through Canada? Mexico?

Routes from Africa to Africa through Europe?
1. Researcher wants to study, for example, boomerang routes
2. Collect (a lot of) traceroute measurements
1. Researcher wants to study, for example, boomerang routes
2. Collect (a lot of) traceroute measurements
The Art of Internet Measurement

1. Researcher wants to study, for example, boomerang routes.

2. Collect (a lot of) traceroute measurements.

PlanetLab lacks diversity. Other platforms cannot make exhaustive measurements.
“The number one go-to tool is traceroute.”
NANOG Troubleshooting Tutorial, 2009.
NANOG Traceroute Tutorial, 2014.
“The number one go-to tool is traceroute.”
NANOG Troubleshooting Tutorial, 2009.
NANOG Traceroute Tutorial, 2014.

But traceroute only answers one simple question:

“What is the path from vantage point $s$ to destination $d$?”
“The number one go-to tool is traceroute.”
NANOG Troubleshooting Tutorial, 2009.
NANOG Traceroute Tutorial, 2014.

But traceroute only answers one simple question:

“What is the path from vantage point $s$ to destination $d$?”

We need a new tool to answer our questions
Sibyl

Supports high-level queries over Internet routes
Goal: Serve High-Level Routing Queries

Users submit high-level queries

Sibyl

Queries

Traceroutes that match queries
Query Internet Routes Using Regular Expressions

Symbols representing boolean expressions that IP addresses must satisfy

Level3&LA    GTT&Seattle
Query Internet Routes Using Regular Expressions

Symbols representing boolean expressions that IP addresses must satisfy

\[ \text{Level3&LA} \quad \text{GTT&Seattle} \]

Routes traversing Level3 in LA and GTT in Seattle?

\[ ^ \cdot \ast \text{Level3&LA} \cdot \ast \text{GTT&Seattle} \cdot \ast \$ \]
Regular Expressions Capture Operational Questions

Someone suggests problem is on link between NTT and GTT in Seattle. Which routes use that link and could be impacted?

```
^ .* NTT&Seattle GTT&Seattle .* $
```
Someone suggests problem is on link between NTT and GTT in Seattle. Which routes use that link and could be impacted?

\[
\text{^ \.* NTT&Seattle GTT&Seattle \.* $}
\]

If route to destination D is impacted, which of my providers have a route that avoids that link?

\[
\text{^ USC (?!NTT GTT&Seattle)\.* D $}
\]

Providers in my region with routes that avoid that link?

\[
\text{^ \.* LA (?!NTT GTT&Seattle)\.* D $}
\]
Regular Expressions Capture Research Questions

What routes from the US to the US detour through Canada?

\(^ \text{US}\ast \text{Canada}\ast \text{US}\ast \) $

Eastward routes detouring through Canada?

\(^ \text{EastCoast} \ast \text{Canada} \ast \text{WestCoast} \) $

Routes from Southern US through Canada? Mexico?

\(^ \text{SouthernUS} \ast (\text{Canada} | \text{Mexico}) \ast \text{US} \) $

Routes from Africa to Africa through Europe?

\(^ \text{Africa} \ast \text{Europe} \ast \text{Africa} \) $
How do we achieve route coverage to satisfy complex and diverse queries?

Users submit high-level queries

Sibyl

Traceroutes that match queries

Measure Paths
Combining Platforms Provides Great Coverage

Vantage points in 100% of very large networks and 50% of small and regional networks
Combining Platforms Provides Great Coverage

Vantage points in 100% of very large networks and 50% of small and regional networks

<table>
<thead>
<tr>
<th>Measurement Platform</th>
<th>Vantage Points</th>
<th>Vantage Point ASes</th>
</tr>
</thead>
<tbody>
<tr>
<td>PlanetLab</td>
<td>~400</td>
<td>~250</td>
</tr>
<tr>
<td>Traceroute servers</td>
<td>~500</td>
<td>~500</td>
</tr>
<tr>
<td>RIPE Atlas</td>
<td>~9000</td>
<td>~2700</td>
</tr>
</tbody>
</table>
Vantage Points Are Resource Constrained

Want to use network coverage to answer queries, but cannot issue all measurements

<table>
<thead>
<tr>
<th>Measurement Platform</th>
<th>Vantage Points</th>
<th>Vantage Point ASes</th>
<th>Traces per Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>PlanetLab</td>
<td>~400</td>
<td>~250</td>
<td>16,000K</td>
</tr>
<tr>
<td>Traceroute servers</td>
<td>~500</td>
<td>~500</td>
<td>145K</td>
</tr>
<tr>
<td>RIPE Atlas</td>
<td>~9000</td>
<td>~2700</td>
<td>35K</td>
</tr>
</tbody>
</table>
Vantage Points Are Resource Constrained

CDF of Destinations

Number of ASes Seen toward Each Destination

Total coverage
Resource Constraints Limit Visibility

![Graph showing CDF of destinations against number of ASes seen.
- Total coverage
- Rate limits

Number of ASes Seen toward Each Destination vs. CDF of Destinations]
Resource Constraints Limit Visibility

Goal: satisfy queries as if we have total coverage, despite rate limits

CDF of Destin

Rate limits

Total coverage

Number of ASes Seen toward Each Destination
Goal: satisfy queries as if we have **total coverage**, despite **rate limits**

Approach: carefully choose measurements

**Total coverage**

**Rate limits**

Number of ASes Seen toward Each Destination
Too many possible measurements. How do we prune the set of measurements to consider?

Users submit high-level queries

Sibyl

Generate candidates

Combine multiple platforms

Measure Paths

Traceroutes that match queries

Queries

PlanetLab

RIPE Atlas

Traceroute servers
^.* A .* B .*$
\[^.*\ A\ .*\ B\ .*\]$
Vantage Point

Destinations

Vantage Point

^.* A .* B .*$
Many possible splices!
How to estimate which candidate measurements will match queries?

Users submit high-level queries

Queries → Generate candidates → Match likelihood

Sibyl

Combine multiple platforms

Traceroutes that match queries

Measure Paths

PlanetLab

RIPE Atlas

Traceroute servers
1. Predict all paths for candidate measurements
1. Predict all paths for candidate measurements
2. Compute likelihood of prediction being correct
Computing Likelihood that Prediction is Correct

Input → set of predicted paths between source and destination

Output → likelihood of each prediction being correct

Approach → Compute route features and use machine learning
Computing Likelihood that Prediction is Correct

Input $\rightarrow$ set of predicted paths between source and destination

Output $\rightarrow$ likelihood of each prediction being correct

Approach $\rightarrow$ Compute route features and use machine learning

- Most important features include
  - AS peering relationships
  - Route length
1. Predict all paths for candidate measurements
2. Compute likelihood of prediction being correct
3. Compute likelihood actual path matches

Path matches query
Path doesn’t match query
1. Predict all paths for candidate measurements
2. Compute likelihood of prediction being correct
3. Compute likelihood actual path matches

\[ ^.* \text{Path matches query} \]
\[ \text{Path doesn't match query} \]
Match Likelihood Estimation Is Accurate

![Graph showing the relationship between Estimated Likelihood of Matching and Fraction of Candidates Matching the Queries.](image)
Match Likelihood Estimation Is Accurate

Chosen measurements match queries
Users submit high-level queries

Sibyl

Queries

Generate candidates

Match likelihood

Optimize budget

Update topology

Measure Paths

Combine multiple platforms

Optimize budget use to maximize expected utility over all queries

Traceroutes that match queries

PlanetLab

RIPE Atlas

Traceroute servers
Users submit high-level queries

Queries

Generate candidates

Match likelihood

Optimize budget

Update topology

Measure Paths

Traceroutes that match queries

Combined multiple platforms

Optimistically update knowledge base

Sibyl

Optimize budget use to maximize expected utility over all queries

PlanetLab

RIPE Atlas

Traceroute servers
Evaluation

Evaluated Sibyl over multiple rounds. Each round:

- System has fixed, limited probing budget to allocate
- Generate random (but satisfiable) queries
- Evaluate the fraction of queries Sibyl can satisfy
How Accurate Is Sibyl?

PlanetLab only matches 52% of queries.
How Accurate Is Sibyl?

Adding random measurements helps little (56%). PlanetLab only matches 52% of queries.
How Accurate Is Sibyl?

Need intelligence to effectively match (75%)
Adding random measurements helps little (56%)
PlanetLab only matches 52% of queries
Conclusions

- Supports high-level queries over Internet routes
- Combines platforms to improve coverage and budget
- Smartly chooses which measurements to issue
  - Overcome probing budget constraints
Future Work

- Improve path prediction and likelihood estimation
- Long-term budget allocation
  - Balance satisfying current queries vs benefit to serve future queries
  - Adapt probing rate as a function of query load
- Unify queries over historical and live data
  - “Give me paths that looked like X but now look like Y”
Query Language
We Have Complex Monitoring Systems

iSpy: diverse routes to a given prefix?

Reverse traceroute: routes through Level3 to monitoring node?

RocketFuel: routes through Verizon between Chicago and LA?
Regular Expressions Support Existing Systems

iSpy: diverse routes to a given prefix?
^ {•*} 184.164.224.0/19 $ by AS

Reverse traceroute: routes through Level3 to monitoring node?
^ •* Level3 .* USC $

RocketFuel: route through Verizon between Chicago and LA?
^ •* Verizon&Chicago •* Verizon&LA •* $
Internal Representation of Regular Expressions

Paths that go through Sprint’s Chicago PoP on the way to Brazil:

\[^.*S.*U\$\] where \(S := \text{Sprint&Chicago}\)

\(U := \text{USC}\)

From NANOG: Problem between Level3 in LA and GTT in Seattle:

\[^.*L.*G.*\$\] where \(L := \text{Level3&LA}\)

\(G := \text{GTT&Seattle}\)
Utility Optimization
Maximize Global Utility

\[ T = \bigcup_{p \in P} T_p \]

The union of the set of traceroutes chosen from each platform
Maximize Global Utility

\[
\text{let } T = \bigcup_{p \in P} T_p \\
\text{max } T : U(T)
\]

The union of the set of traceroutes chosen from each platform

Choose the set of traceroutes that maximizes utility over all queries
Maximize Global Utility

\[
\begin{align*}
\text{let } T &= \bigcup_{p \in P} T_p \\
\max_T \sum_{q \in Q} U_q(T)
\end{align*}
\]

The union of the set of traceroutes chosen from each platform

Choose the set of traceroutes that maximizes utility over all queries
Maximize Global Utility

\[
\text{let } T = \bigcup_{p \in P} T_p \\
\text{max } \sum_{T} \sum_{q \in Q} U_q(T) \\
\text{subject to } |T_p| \leq B_p \forall p \in P
\]

The union of the set of traceroutes chosen from each platform

Choose the set of traceroutes that maximizes utility over all queries

Subject to budget constraints of each platform
Utility functions

Existence queries → find one path that matches a query

\[ \mathbb{E} [f_q(T)] = 1 - \prod_{t \in T} [1 - \mathbb{P}(t \in q)] \]
Utility functions

Existence queries → find one path that matches a query

\[ \mathbb{E} [f_q(T)] = 1 - \prod_{t \in T} [1 - \mathbb{P}(t \in q)] \]

Diversity queries → find all different paths that match a part of the query

\[ \mathbb{E} [f_q(T)] = 1 - \prod_{t \in T} [1 - \mathbb{P}(t \in q)] \]
Utility functions

Existence queries → find one path that matches a query

$$\mathbb{E} [f_q(T)] = 1 - \prod_{t \in T} [1 - \mathbb{P}(t \in q)]$$

Diversity queries → find all different paths that match a part of the query

$$\mathbb{E} [f_q(T)] = \sum_{h} \left\{ 1 - \prod_{t \in T} [1 - \mathbb{P}(t \in q \land h \in t)] \right\}$$
RuleFit and Likelihood Estimation
## Most Important Features

<table>
<thead>
<tr>
<th>Spliced Path Feature</th>
<th>Importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. PoP-level similarity with the other paths</td>
<td>1</td>
</tr>
<tr>
<td>2. PoP-level path length inflation vs iPlane’s top-ranked path</td>
<td>.90</td>
</tr>
<tr>
<td>3. Total number of PoP splice points</td>
<td>.60</td>
</tr>
<tr>
<td>4. Total number of AS splice points</td>
<td>.55</td>
</tr>
<tr>
<td>5. AS splice point type</td>
<td>.52</td>
</tr>
<tr>
<td>6. AS splice point relationship with neighbors</td>
<td>.49</td>
</tr>
<tr>
<td>7. Number of PoPs in iPlane’s top-ranked path</td>
<td>.44</td>
</tr>
<tr>
<td>Other features</td>
<td>≤ .34</td>
</tr>
</tbody>
</table>
RuleFit Jaccard Index Predictions Correlate
SFSA Splicing
Query FSA Splicing to Generate Candidates

\[ ^\cdot B .\cdot $ \]

Initial

\[ S_1 \rightarrow S_2 \rightarrow S_3 \]
Query FSA Splicing to Generate Candidates

\[
\begin{align*}
&^\cdot * \quad A \quad . * \quad B \quad . * $ \\
\end{align*}
\]

Initial

<table>
<thead>
<tr>
<th>Hop</th>
<th>Transition</th>
<th>Hop</th>
<th>Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>S1</td>
<td>V2</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>S1 \rightarrow S2</td>
<td>X</td>
<td>B</td>
</tr>
<tr>
<td>X</td>
<td>S2</td>
<td>D1</td>
<td>S2</td>
</tr>
</tbody>
</table>

D2
Query FSA Splicing to Generate Candidates

\[
\begin{align*}
^* & \cdot & A & \cdot & B & \cdot & * & $ \\
^* & A & S1 & A & S2 & B & S3
\end{align*}
\]

Initial

<table>
<thead>
<tr>
<th>Hop</th>
<th>Transition</th>
<th>Hop</th>
<th>Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>*</td>
<td>V2</td>
<td>*</td>
</tr>
<tr>
<td>A</td>
<td>S1</td>
<td>X</td>
<td>S2</td>
</tr>
<tr>
<td>X</td>
<td>S2</td>
<td>B</td>
<td>S3</td>
</tr>
<tr>
<td>D1</td>
<td>S2</td>
<td>D2</td>
<td>S3</td>
</tr>
</tbody>
</table>
Query FSA Splicing to Generate Candidates

```
^.* A .* B .*$
```

<table>
<thead>
<tr>
<th>Hop</th>
<th>Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>S1</td>
</tr>
<tr>
<td>V2</td>
<td>S2</td>
</tr>
<tr>
<td>A</td>
<td>S1 → S2</td>
</tr>
<tr>
<td>X</td>
<td>S2</td>
</tr>
<tr>
<td>X</td>
<td>S2</td>
</tr>
<tr>
<td>B</td>
<td>S3 → S2</td>
</tr>
<tr>
<td>D1</td>
<td>S2</td>
</tr>
<tr>
<td>D2</td>
<td>S3</td>
</tr>
</tbody>
</table>
Evaluation
Queries Types Used in Evaluation

● Traverse AS toward destination
  ○ ^ .* A .* B $ 
  ○ ^ [^A]+ A+ .* B $ 

● Traverse link toward destination
  ○ ^ .* A B .* D $ 

● Traverse three different locations in sequence
  ○ ^ .* A .* B .* C .* $
How Much Better Can We Do?

Unlimited budget would allow Sibyl to match 88% of queries.

Sibyl matches 75% of queries.

Current approach matches 52% of queries.

A graph showing the fraction of queries satisfied over rounds. The graph includes lines for Sibyl (No rate limit), Sibyl, and Exhaustive measurements from PlanetLab.
How Much Does Each Module Contribute

Sibyl satisfies 75% of queries.

Could satisfy 84% with unlimited budget.

Each module contributes 5-8% accuracy.
Sibyl Is Effective at Different Granularities

Fraction of Queries Satisfied

Round

Sibyl (AS and country queries)
Sibyl (AS-level queries)
Sibyl (PoP-level queries)
Combining Platforms Helps Satisfy More Queries

Fraction of Queries Satisfied

Round

Sibyl (no rate limits)
Sibyl (PL + RIPE + TS)
PL + RIPE Atlas (RIPE)
PL + Traceroute Servers (TS)
PlanetLab (PL) only
Sibyl Uncovers More Path Diversity

CDF of Queries

Fraction of PoP Diversity Uncovered
Staleness
Patching and Pruning Stale Measurements

New measurements may detect path changes

- Whenever we detect a change toward a destination, update all paths to that destination that overlap
- Whenever we detect a path change from a source, update all paths from that source that overlap
Staleness Has Small Impact
Path Change Properties
Uphill Path Changes Are Less Likely

![Graph showing the CDF of vantage points for different path changes over a fraction of paths that changed after one week. The graph compares PlanetLab uphill changes (red solid line), RIPE uphill changes (blue dashed line), PlanetLab all changes (magenta dashed line), and RIPE all changes (green dotted line).]