iPlane Nano: Path Prediction for Peer-to-Peer Applications

Harsha V. Madhyastha, Ethan Katz-Bassett, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani

University of California, San Diego, University of Washington, and University of Massachusetts, Amherst
Motivation

• Example application: P2P CDN
  – Content replicated across geographically distributed set of end-hosts
    • RedSwoosh (Akamai)
    • Kontiki (BBC’s iPlayer)
  – Every client needs to be redirected to replica that provides best performance

• Problem (also for BitTorrent, Skype, ...):
  – Internet performance neither constant nor queriable
Need for Performance Prediction

• Current Best Practice:
  – Each application measures the Internet independently

• Desired Solution:
  – Ability for end-hosts to predict performance
  – Infrastructure shared across applications
## Need for iPlane Nano

<table>
<thead>
<tr>
<th>Network Coordinates</th>
<th>Predicted Information</th>
<th>Cost to Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>− Limited to latency</td>
<td>+ Lightweight distr. system</td>
<td></td>
</tr>
</tbody>
</table>
iPlane Nano: Overview

- Server-side: Use iPlane’s measurements but store and process differently
  - Key idea: Replace atlas of paths with atlas of links → from $O(n^2)$ to $O(n)$ representation

Size of Atlas = $O(\#\text{Vantage points} \times \#\text{Destinations} \times \text{Avg. Path Length})$
iPlane Nano: Overview

• **Server-side:** Use iPlane’s measurements but store and process differently
  - **Key idea:** Replace atlas of paths with atlas of links \( \rightarrow \) from \( O(n^2) \) to \( O(n) \) representation

• **Client-side:** Application library
  - **Download atlas** and help disseminate atlas
  - **Service queries locally** with prediction engine
Challenge: Loss of Routing Info

- Routing policy information encoded in routes is lost.
- Need to extract routing policy from measured routes and represent compactly.
Routing Policy: Strawman Approach

- Common aspects of Internet routing applied
  - Shortest AS path + valley-free + early-exit
- Poor AS path prediction accuracy obtained
  - Too many valley-free shortest AS paths

![Graph showing fraction with correct predicted AS path](image)
1. Inferring AS Filters

- Every path is not necessarily a route
  - ASes filter propagation of route received from one neighbor to other neighbors

- Filters inferred from measured routes
  - Record every tuple of three successive ASes observed in any measured route
  - Store \((AS_1, AS_2, AS_3)\) to imply \(AS_2\) forwards routes received from \(AS_3\) on to \(AS_1\)
Applying Inferred AS Filters

- AS filters help discard paths not policy-compliant
- Still have multiple policy-compliant paths
2. Inferring AS Preferences

• For every measured route, alternate paths are determined in link-based atlas

• Divergence of paths indicates preference
  - \( AS_1 \rightarrow AS_2 \rightarrow AS_3 \) ... on measured route
  - Alternate paths imply \( AS_1 \) prefers \( AS_2 \) over \( AS_5 \) and \( AS_2 \) prefers \( AS_3 \) over \( AS_6 \)
Challenge: Routing Asymmetry

- Undirected edges used to compute route \((S \rightarrow D)\)
  - Assuming symmetric routing
- But, more than half of Internet routes asymmetric
3. Handling Routing Asymmetry

- Client library includes measurement toolkit
  - Traceroutes to random prefixes at low rate
  - Uploads to central server

- Each client’s measurements assimilated into atlas distributed to all clients

- Directed path computed for route prediction
  - Fall back to undirected path if not found
Improved Path Predictions

- AS path prediction accuracy with iPlane Nano almost as good as with iPlane!

![Bar chart showing AS path prediction accuracies and Atlas sizes.]

- iPlane: 81%
- Strawman: 30%
- Link-based: 70%

Atlas size:
- iPlane: 2 GB
- iPlane Nano: 6.6 MB (1.4 MB daily update)
From Routes to Properties

• To estimate end-to-end path properties between arbitrary $S$ and $D$
  – Use atlas to predict route
  – Combine properties of links on predicted route

<table>
<thead>
<tr>
<th>Latency</th>
<th>Sum of link latencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss-rate</td>
<td>Probability of loss on any link</td>
</tr>
</tbody>
</table>

• Ongoing challenge: Measuring link properties
Improving P2P Applications

• Used *iPlane Nano* to improve three apps
  – P2P CDN
    • Choose replica with best performance
  – VoIP
    • Choose detour node to bridge hosts behind NATs
  – Detour routing for reliability
    • Choose detour nodes with disjoint routes to route around failure

• Refer to paper for VoIP and detour routing experiments
Improving P2P CDN

- Clients: 199 PlanetLab nodes
- Replicas: 10 random Akamai nodes per client
- 1MB file downloaded from “best” replica

![Graph showing cumulative fraction of clients vs. relative inflation in download time.]

Download time = 2 x Optimal
Conclusions

• Implemented iPlane Nano
  – Practical solution for scalably providing predictions of Internet path performance to P2P applications
  – Compact representation of routing policy to predict route and path properties between arbitrary end-hosts

• Demonstrated utility in improving performance of P2P applications

• Step towards determining minimum information required to capture Internet performance
Thank You!

- iPlane Nano's atlas and traces gathered by iPlane updated daily at
  
  http://iplane.cs.washington.edu

- Send me email if you want to use iPlane Nano's or iPlane's predictions