Séminaire STORE

Parsimonious Active Monitoring and Overlay Routing

Olivier Brun

LAAS–CNRS, Toulouse, France

February 23rd, 2017
Outline

Introduction

SMART – An open-source software for overlay routing

Shortest Path Discovery Problem

Learning-based Routing in an Adversarial Environment

Learning-based Routing as a POMDP

Conclusion
Introduction

- Classic measurement papers have shown that Internet routing results in paths that are sub-optimal with respect to a number of metrics.
- Experiment with 20 nodes of the NLNOG Ring.

✔ The IP route is optimal only in 50% of cases.

✔ Average gap to min latency is 31%.

<table>
<thead>
<tr>
<th></th>
<th>IP</th>
<th>OPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moscow/Dublin</td>
<td>180</td>
<td>81</td>
</tr>
<tr>
<td>Singapore/Paris</td>
<td>322</td>
<td>153</td>
</tr>
</tbody>
</table>
Introduction

- **Routing overlays** were proposed as a method for improving performance, without the need to re-engineer the underlying network.
- Overlay nodes monitor the quality of the IP routes between themselves and cooperate to route messages.

Self-healing and self-optimizing application-layer virtual network.
Introduction

- **All-pairs probing**
  - ✓ In an overlay of $n$ nodes, there are $O(n^2)$ links to monitor.
  - ✓ Monitoring the quality of all overlay links is excessively costly, and impairs scalability.

- **Problem:** how to design parsimonious monitoring strategies enabling to achieve near-optimal routing?
Introduction

- We consider a single origin/destination pair.

- How to discover an optimal route by probing only a small subset of possible paths?
  - Shortest path discovery problem
  - Learning-based routing in an adversarial environment
  - Learning-based routing as a POMDP
SMART

An open-source software for overlay routing
SMART – Self-MAAnaging RouTing overlay

- Open source software for deploying self-healing and self-optimizing overlays over a sizable population of nodes
Proxy

- **Monitoring Agent**: it monitors the quality of the Internet paths between the local cloud and the other clouds (latency, bandwidth, loss rate).

- **Routing Agent**: It controls the monitoring agent so as to discover an optimal path with a minimum monitoring effort.

- **Forwarding Agent**: It forwards each incoming packet to its destination using source routing.
Packet Routing/Forwarding

src vm

src proxy

relay proxy

dst proxy

dst vm

src task

monitor

router

forwarder

dst task

ra

ta

ra

ta
Shortest Path Discovery Problem
Shortest Path Discovery Problem

- **Input:** a complete graph of \( n \) nodes whose edge lengths are unknown but can be discovered by querying an oracle.

- **Goal:** discover a shortest path from \( s \) to \( t \) by querying the minimum number of edges.

- **Online algorithm** with an approximation ratio of 2

- **Negative results:**
  - Any algorithm needs to query at least \( n - 1 \) edges
  - For any algorithm, there exists a bad instance for which the number of queries is \( O(n^2) \)
Learning-based Routing in an Adversarial Environment
Adversarial setting

- A decision algorithm $\mathcal{A}$ is given as input $N$ paths from the origin to the destination, indexed from 1 to $N$.

- For example, the paths of at most $k$ hops

$$N = \sum_{j=0}^{k-1} \frac{(n-2)!}{(n-2-j)!}$$

For each round $t = 1, 2, \ldots$

1. a cost $\ell_i(t) \in [0, 1]$ is assigned to each path $i$, but it is not revealed to the algorithm.

2. then, the algorithm chooses a subset $Q(t) \subset \{1, 2, \ldots, N\}$ of $K$ paths, observe their costs, and sends a message over a path $i^*(t) \in \arg\min_{i \in Q(t)} \ell_i(t)$. 
Adversarial setting

- **Cumulative cost** of the algorithm over $T$ rounds is defined as

\[
L_T(A) = \sum_{t=1}^{T} \min_{i \in Q(t)} \ell_i(t),
\]  

(1)

whereas the cumulative cost of path $i$ is $L_T(i) = \sum_{t=1}^{T} \ell_i(t)$.

- The **Normalized regret** of the algorithm $A$ w.r.t. the best path is

\[
R_T(A) = \frac{1}{T} \left( L_T(A) - \min_{i=1,\ldots,N} L_T(i) \right).
\]

(2)

- **Goal**: design an algorithm $A$ such that $R_T(A) \to 0$ as $T \to \infty$. 

Learning-based Routing in an Adversarial Environment

- Similar to a multi-armed bandits problem
  - Objective of the gambler is to maximize his reward (regret minimization)
  - The gambler does not know the expected reward of each arm but can learn by successively choosing different arms
- Well-studied problem in different contexts
  - Several applications: clinical trials, ad placement on webpages, etc.
  - Several variants: stochastic environment, adversarial environment, game, etc.
Learning-based Routing in an Adversarial Environment

- Randomized algorithm based on EXP3

  ✓ Mixture of the uniform distribution and a distribution which assigns to each path a probability mass exponential in the estimated cumulative gain for that path:

  \[ p_i(t) = (1 - \gamma) \frac{w_i(t)}{\sum_{j=1}^{N} w_j(t)} + \gamma \frac{1}{N} \]

  ✓ Asymptotically the same average (per round) end-to-end performance as the best path.

  \[ R_T(\text{EXP3}) \leq \frac{11}{2} \sqrt{\frac{N \log(N/\delta)}{T}} + \frac{\log(N)}{2T} \]

  with probability at least 1 − δ, for any 0 < δ < 1.
Latency Minimization: NLNog ring

- We restrict ourselves to paths with at most one intermediate overlay node
  - SMART probes only 5 overlay links per measurement epoch (among 342 links in total)
  - SMART uses the optimal 2-hop routes in 96% of the cases (gap to opt. latency is 0.39%)

<table>
<thead>
<tr>
<th>IP route</th>
<th>Average RTT (ms)</th>
<th>SMART</th>
</tr>
</thead>
<tbody>
<tr>
<td>Melbourne/Gibraltar</td>
<td>390.0</td>
<td>274.7</td>
</tr>
<tr>
<td>Narita/Santiago</td>
<td>406.7</td>
<td>254.5</td>
</tr>
<tr>
<td>Moscow/Dublin</td>
<td>179.9</td>
<td>81.9</td>
</tr>
<tr>
<td>Honk Kong/Calgary</td>
<td>267.1</td>
<td>131.8</td>
</tr>
<tr>
<td>Singapore/Paris</td>
<td>322.3</td>
<td>154.9</td>
</tr>
<tr>
<td>Tokyo/Haifa</td>
<td>322.6</td>
<td>180.8</td>
</tr>
</tbody>
</table>
Latency Minimization (2)

RTT (ms) Japan/Chile over 5 days

Zoom over the first 90 minutes
We restrict ourselves to paths with at most one intermediate overlay node:

- SMART probes only 5 overlay links per measurement epoch (among 72 links in total)
- SMART uses the optimal 2-hop routes in 70% of the cases (gap to opt. latency is 6.6%)

### Average Throughput (Mbps)

<table>
<thead>
<tr>
<th>IP route</th>
<th>SMART</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dublin/Sydney</td>
<td>11.5</td>
</tr>
<tr>
<td>Singapore/Sao Paulo</td>
<td>12.8</td>
</tr>
<tr>
<td>Sydney/Virginia</td>
<td>8.5</td>
</tr>
<tr>
<td>Virginia/Singapore</td>
<td>7.4</td>
</tr>
<tr>
<td>Virginia/Sydney</td>
<td>6.9</td>
</tr>
<tr>
<td>Virginia/Tokyo</td>
<td>10.3</td>
</tr>
</tbody>
</table>
Throughput Maximization (2)

Throughput from Sydney to Virginia over 4 days

Throughput from Sydney to Virginia (zoom over the first 3 hours)

Throughput (Mbps) from Virginia to Sydney over 4 days

Throughput (Mbps) from Virginia to Sydney (zoom over the first 3 hours)
Live Experiment with SMART

- A client in Tokyo repeatedly downloads 100 MB files from a server in Sao Paulo
  - Every 4 mn, a file is downloaded via a Proxy in Oregon
  - A few seconds later, it is downloaded via the IP route
Learning-based Routing as a POMDP
Stochastic setting

- Analysis of latency data collected over NLNog and RIPE Atlas shows that RTT time series can be modeled as Markov chains or Hidden Markov Models (HMM).

![Round Trip Times Moscow-Dublin](image)

- Calibration phase for parameter estimation from measured RTT data

Olivier Brun  
STORE Seminar, February 20th, 2017  
24 / 34
Stochastic setting

For each round $t = 1, 2, \ldots$

1. **State evolution:** the state $x_i(t)$ of each path $i$ evolves according to a DTMC over state space $\mathcal{S}_i$ with transition matrix $P_i = (p_i(u, v))$. The delay in state $x$ is $\ell_i(x)$

2. **Monitoring decision:** the player observes the states of a subset $Q(t)$ of paths, each at some cost $c$

3. **Routing decision:** the player sends a message over path $r(t)$ and incurs cost $\ell_{r(t)}(x_{r(t)}(t))$. 
Stochastic setting

Goal: minimise the expected sum of transmission delays and monitoring costs

\[ \mathbb{E} \left\{ \sum_{t=0}^{\infty} \beta^t (\ell_r(t) (x_r(t)(t)) + c |Q(t)|) \right\}, \]

where \( \beta < 1 \) is a given positive discount factor

Tradeoff between the cost of monitoring paths, which brings more up-to-date state information, and the higher probability of experiencing high transmission delay
Partially Observable Markov Decision Process

- **Sufficient statistics:** all the information available at time $t$ is summarized by the vector $s = (s_1, s_2, \ldots, s_N)$, where $s_i = (y_i, \tau_i)$
  - $y_i$ is the last observed state of path $i$
  - $\tau_i$ is the age of this observation

- **Belief** on the state of path $i$: $[p_i^{(\tau_i)}(y_i, 1), p_i^{(\tau_i)}(y_i, 2), \ldots]$

- **Transition probabilities:** given the set $Q$ of monitored paths,

\[
\pi_Q(s, s') = \prod_{i=1}^{N} \pi_Q^{(i)}(s_i, s_i')
\]

where the information on link $i$ evolves from

- $(y, \tau)$ to $(y, \tau + 1)$ with prob. $1$ if $i \not\in Q$,
- $(y, \tau)$ to $(w, 1)$ with prob. $p_i^{(\tau)}(y, w)$ otherwise.
Optimal Policy

- **Optimal routing decision:** send the message on the path with minimum expected delay given the available information

\[
D(s) = \min \left( \min_{i \not\in Q(t)} \mathbb{E}\ell_i(x_i(t)), \min_{i \in Q(t)} \ell_i(y_i) \right)
\]

- **Optimal monitoring decision** (Bellman equation)

\[
J^*(s) = \min_A \left\{ c|Q| + \sum_{s'} \pi_Q(s, s') \left( D(s') + \beta J^*(s') \right) \right\}
\]

- **Value Iteration Algorithm**
Example for two links

\[ Q = \{2\} \quad y_1 = 0, \ y_2 = 0 \]

\[ Q = \{1\} \quad y_1 = 1, \ y_2 = 0 \]

\[ Q = \{1\} \quad Q = \{2\} \quad Q = \{1, 2\} \quad y_1 = 0, \ y_2 = 1 \]

\[ y_1 = 1, \ y_2 = 1 \]

Olivier Brun
Threshold Policies

- Monitor a link iff its success probability (probability that it is the minimum delay link) is in between $\epsilon$ and $\Delta$ (e.g., $\epsilon = 0.3$, $\Delta = 0.8$).
Conclusion
Conclusion

- Internet routing works reasonably well most of the times, but routing overlay can yield spectacular improvements over native IP routing in some cases.
- A trade-off between the quality of the routes discovered and the monitoring effort to discover them is required.
- Probing does not cover all possible paths but only a few paths which have been observed to be of good quality, exploring also at random other paths whose quality might have improved recently.
- Future work will focus on the analysis of threshold policies in the stochastic setting.
Questions?