Characterizing Selfishly Constructed Overlay Routing Networks

INFOCOM 2004

謝志峰

2004/4/7
Outline

- Introduction
- Overlay Routing Network
- Routing Network Creation Game
- Experiment
- Conclusions
- Future Work
We analyze the characteristics of overlay routing networks generated by selfish nodes playing competitive network construction games. By modeling network formation as a non-cooperative game, each node chooses its overlay neighbors to maximize its benefit and to minimize its linking cost.
Introduction

- They want to have low cost paths to other nodes in the network
  - by establishing more links.
  - not want to establish many links, which may turn out to be costly.

- The outcome of the game is a network topology, which is a Nash equilibrium.

- Nash equilibrium
  - a set of strategies that no player can benefit by changing its strategy, while the other players keep their strategies unchanged
The ellipses are physical nodes and the rectangles are overlay nodes labeled A, B, C, and D.

- In (a) has virtual links AB, BC, and CD.
- In (a), the path length is 6 physical hops.
- When A decides to add a link to D, the resulting network is the network (b).
- A incurs cost to establish this new link, but the distance reduces to 2 physical hops due to the virtual link AD.
Routing Network Creation Game

- The cost model is the most important part of the game.

- The “distance” between two nodes may be represented by other functions than the number of hops (include the cost of the path or the latency).
Routing Network Creation Game

- From a connected random graph, and in each round, each player changes its link configuration to minimize its cost as given.

Algorithm 1 Link Addition for node $i$

- Randomly select node $j$ not in the neighborhood of $i$
- Compute $\text{Cost}_{\text{new}}$ with $j$ included
- if $\text{Cost}_{\text{old}} - \text{Cost}_{\text{new}} > 0$ then
  - Add the link

Algorithm 2 Link Dropping for node $i$

- $\text{NodeToDrop} = -1$
- $\text{MinCost} = \text{Cost}_{\text{old}}$
- for all node $j$ in the neighborhood of $i$ do
  - Compute $\text{Cost}_{\text{new}}$ without $j$
  - if $\text{MinCost} - \text{Cost}_{\text{new}} > 0$ then
    - $\text{MinCost} = \text{Cost}_{\text{new}}$
    - $\text{NodeToDrop} = j$
  - if $\text{NodeToDrop} != -1$ then
    - Drop the link between $i$ and $\text{NodeToDrop}$. 
Routing Network Creation Game

- **Exhaustive search:**
  - The node should verify all possible configurations of the edges existing or not, to all other neighbors.
  - There are $2^{(n-1)}$ different strategies.
  - The time complexity of running the game in this fashion is **exponential** in the number of nodes.
Routing Network Creation Game

- **Randomized local search:**
  - Runs the link state (LS) protocol.
  - Each node periodically performs the link drop and link addition procedures.
  - Randomly selects a node that
    - is not the previously dropped link.
    - is not in the neighborhood.
Routing Network Creation Game

- We randomly choose one node that is not in the neighborhood of node $i$, and fetch its link state and cost $t_j$.
- We add the link if the cost of node $i$ is reduced by linking to node $j$.

\[
\text{Cost}_{\text{old}} - \text{Cost}_{\text{new}} > 0 \iff \sum_{j=0}^{n-1} (d_{\text{old}}(i, j) - d_{\text{new}}(i, j)) > \alpha \cdot t_j
\]
Routing Network Creation Game

- It computes the node’s cost of a new graph when a particular link is dropped.

- It chooses the neighbor that leads to the minimum cost value that is less than the old value.

\[
\text{Cost}_{\text{old}} - \text{Cost}_{\text{new}} > 0 \quad \text{iff} \quad \sum_{j=0}^{n-1} (\text{d}_{G_{\text{new}}}(i, j) - \text{d}_{G_{\text{old}}}(i, j)) < \alpha \ t_j
\]
Routing Network Creation Game

- **Unit-Countout**: Node that initiates the connection pays the total cost of the connect.

- **Exp-Countout**: The linking costs are generated from an exponential distribution of mean 1.0.

- **Unit-Nodedegree**: Cost incurred by a node to create a link depends on the node degree of the node to connect to.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>$d_{ij}(x,j)$</th>
<th>Explored Parameters</th>
<th>Strategy Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>Number of Hops</td>
<td>$\alpha$, Linking Cost</td>
<td>Exhaustive Search</td>
</tr>
<tr>
<td>Realistic</td>
<td>Latency from physical topology</td>
<td>$\alpha$, Max Degree</td>
<td>Randomized Local Search</td>
</tr>
</tbody>
</table>

**TABLE I**

Summary of the Scenarios Investigated

<table>
<thead>
<tr>
<th>Cost Model</th>
<th>Linking Cost ($c_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit-Countout</td>
<td>1</td>
</tr>
<tr>
<td>Exp-Countout</td>
<td>$c_i$</td>
</tr>
<tr>
<td>Unit-Nodedegree</td>
<td>$\text{degree}(j)$</td>
</tr>
</tbody>
</table>

**TABLE II**
Routing Network Creation Game

- Simple Scenario
  - Sample equilibrium graphs for the simple scenario with 20 nodes, for the different cost models and values of $\alpha$.
Routing Network Creation Game

- **Realistic Scenario**
  - We present simulation results for 100 nodes.
  - We varied $\alpha$ from 800 to 10000 and we varied $MaxDegree$ between 10 and 100.
The stretch decreases as $\alpha$ decreases and it decreases as $MaxDegree$ increases.

Specifically, when $\alpha$ is small and $MaxDegree$ is large.

If $\alpha$ is small, the node is likely to add links, if is large, the node is not likely to add links.
Experiment

- Failure and attack tolerance (a) K when 10 of nodes fail, (b) K when 10 of nodes are attacked

![Graphs showing failure and attack tolerance](image)
Conclusions

- We use a non-cooperative game model to evaluate such networks and examine the effects of the underlying topology and different linking cost functions in the resulting Nash equilibria of the game.

- We find that the games can produce widely different networks, from complete graphs to trees with different properties.

- We also find that there is a fundamental tradeoff between these two metrics, and that it can be controlled by restricting the maximum node degree.
Future Work

- We want to examine the game in a dynamic network where the total number of nodes changes over time due to node joins and leaves.

- Another interesting area is to take traffic into consideration.