Up-and-Down Routing in Mobile Opportunistic Social Networks with Bloom-Filter-Based Hints

Huanyang Zheng and Jie Wu
Dept. of Computer and Info. Sciences
Temple University, USA
Introduction

Mobile opportunistic social networks (MOSN)

- Opportunistic contacts
- Intermittent connectivity
- Instantaneous end-to-end paths may not exist

A scenario

- People walk around with phones that communicate with each other via Bluetooth or WiFi
Introduction

Contact and state information

- Contact information
  - local, but large volume (per node vs. per destination)
- State information
  - costly due to the iterative process

Network structure information of MOSNs

- Nested core-periphery structures (nested hierarchy)

- MIT trace
Up-and-down routing based on nested hierarchy: per node contact with limited state information

- **Up phase**
  - Single-copy routing from source to network core
  - Nested hierarchy
- **Down phase**
  - Multi-copy routing from network core to destination
  - Bloom filter as the routing hint
Introduction

Challenges for traditional hierarchical routings

- Trap in *local maximums* when moving up

- Cannot find the down path efficiently
  - High storage space for *descendants*: each node tracks its child nodes and their child nodes.
Up Phase

Degree hierarchy vs. nested hierarchy
Local Maximum

Local maximums in real dataset (Stanford Large Network Dataset Collection)

AS-733 (autonomous system dataset)
- 6,747 nodes
- 1 local maximum in nested hierarchy (17 levels)
- 8 local maximums in degree hierarchy

p2p-Gnutella08 (Gnutella peer-to-peer network)
- 20,777 nodes
- 3 local maximums in nested hierarchy (20 levels)
- 76 local maximums in degree hierarchy

Nested hierarchy has fewer local maximums!
## Local Maximum

<table>
<thead>
<tr>
<th>CRAWDAD Trace</th>
<th>The Fraction of Contacts Hold by The Most-active 20% Nodes</th>
<th>Total Number of Root Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cambridge/Haggle/Imote/Intel</td>
<td>30.72%</td>
<td>1 node</td>
</tr>
<tr>
<td>Cambridge/Haggle/Imote/Cambridge</td>
<td>51.27%</td>
<td>1 node</td>
</tr>
<tr>
<td>Cambridge/Haggle/Imote/Infocom</td>
<td>29.83%</td>
<td>1 node</td>
</tr>
<tr>
<td>Thlab/Sigcomm2009/Mobiclique/Proximity</td>
<td>43.64%</td>
<td>1 node</td>
</tr>
<tr>
<td>ST_Andrews/Sassy/Mobile</td>
<td>55.14%</td>
<td>1 node</td>
</tr>
</tbody>
</table>
Up Phase

- **Weighted degree** of a node: sum of weights of adjacent links (*total contact frequency*)

- **Effective weighted degree** of a node: weighted degree to unlabeled neighbors

- **Labeling scheme** for nested hierarchy
  - A node labels itself when it has the lowest effective weighted degree among unlabeled neighbors
  - The label is set to be the largest label among its labeled neighbors plus one
Up Phase

- The message is routed towards the root along a DAG
- Single-copy routing to save the forwarding cost
- Switch to the down phase, when first reaching a node that matches (in Bloom filter)
Down Phase

- Each node uses the *Bloom-filter-based routing hint* to record its descendants

- Existence of *false positive* (i.e., a false match)

- The size of Bloom-filter-based routing hint being bounded based on a given false positive rate
Bloom Filters

- Used to test whether an element is a member of a set or not
- A Bloom filter is a bit array of m bits
- k hash functions are used to map an element
- An example (m=5, k=2) of mapping element $e_1$

```
hash \{e_1\}:

1 1
```
Bloom Filters

- Space-efficient at the cost of false positives
- An example of false positive for $e_3$ in $\{e_1, e_2\}$

```
hash \{e_1\}:
1 1

hash \{e_2\}:
1

hash \{e_3\}:
1 1

hash \{e_1, e_2\}:
1 1
```
False positive rate reduces as the level goes up: all child nodes have false positives

Node 4 has a false positive: It claims that node 3 is a descendant
Multi-Copy

- Multi-copy routing serving two objectives
  - Improving delivery ratio by mitigating false positive
  - Reducing down phase delay

- Distributing multiple copies
  - Binary split of copies whenever there is a match

- Bloom filter robustness ratio
  - Ratio of Bloom filter size to number of descendants $d(\alpha-1)^{d-2}$ ($\alpha$: network parameter, $d$: node degree)
  - Keeping robustness level constant at each level
Evaluation Setting

Traces

- Sigcomm trace (76 nodes with $\alpha=2.5$)
- Synthetic trace (100 nodes with average $d=10$, by Barabasi-Albert’s preferential attachment with $\alpha=2.1$, edge weights: 0-0.1)

Algorithms in comparison

- Epidemic (no contact info. with unlimited copies)
- (Binary) Spray and Wait (contact info. per dest.)
- (Binary) Spray and Focus (contact info. per dest.)
- (Modified) Delegation Forwarding (info. per dest. with bounded copies)
Sigcomm Trace

• Data delivery delay and ratio
  • deadline: 500 mins
  • no delivery: deadline as delay
Sigcomm Trace

- Number of forwards
Sigcomm Trace

- Robustness ratio

Overall false positive rate: 38%, 28%, 17%, 10%, 06%, 03%, 02%, 01%, 0.7%
Storage saving percentage: 81%, 72%, 62%, 53%, 44%, 31%, 21%, 10%, 0%
Synthetic Trace

- Data delivery delay and ratio
Synthetic Trace

- Number of forwards
Synthetic Trace

- Robustness ratio

Overall false positive rate: 39%, 24%, 15%, 09%, 06%, 04%, 02%, 01%, 0.8%
Storage saving percentage: 83%, 74%, 65%, 57%, 48%, 39%, 30%, 22%, 13%
Evaluation Summary

- A competitive performance on the data delivery delay and ratio

- Real vs. synthetic traces
  - Real: clustering with more parallel paths
  - Synthetic: multi-hop with fewer parallel paths

- A small diameter does not guarantee a short delay!
Conclusions

Up-and-down routing
• Single-copy up phase and multi-copy down phase
• Nested core-periphery property (nested hierarchy)

Future work
• Bound the number of copies in the down phase
• Coarse grain level
• Deal with multiple local maximums