Excuse me, Sir, but can we deliver packets without addresses?

Petri Jokela, Ericsson Research, NomadicLab

With: András Zahemszky, Somaya Arianfar, Pekka Nikander (Ericsson Research, NomadicLab)
and Christian Esteve (University of Campinas)
Outline

LIPSIN: Line Speed Publish/Subscribe Inter-Networking

- Context: background and targets for the project
- How to forward without IP addresses?
  - Our solution for forwarding without globally routable addresses
- Optimizations for better performance
- Implementation
- Summary
Context - Clean Slate Approach

- **DATA as the first class citizen**
  - Users interested in data, not in the hosts
  - Topic based publish/subscribe

- **DDoS problems**
  - Unwanted traffic can be sent against the receiver’s will
  - Target: Data delivery ONLY when explicitly requested

- **Data published once, received multiple times BUT from different locations and at different times**
  - Multicast - also for *timely* separated events
  - Data caching in the network
Context - RTFM architecture

- Rendezvous - matching publish and subscribe events
- Topology - network topology knowledge, path creation
- Forwarding - fast delivery
Topic based pub/sub: How to deliver data?

- **Routing based on Topic ID**
  - $10^{11}$ topics => enormous amount of state in forwarders
  - State need to be changed based on subscriptions
  - => Not scalable

- **How about storing the state in the packet?**
  - Define the path from the source to the destination
    - IP: include all visited IP addresses in a list
      - A long list of IP addresses, and we do not solve the DDoS
    - Without IP: Include all visited nodes in the packet
      - Long list of Node IDs!
    - Compress the list into a Bloom filter!
      - Path not visible
Link IDs and forwarding Bloom filters (zFilters)

- **No names for nodes**
  - Each *link* is identified with a unidirectional Link ID

- **Link IDs**
  - Statistically unique
  - Periodically changing
  - Size e.g. 256 bits
  - Local or centrally controlled

- **Source routing**
  - Include all Link IDs into a Bloom filter
  - Multicasting supported

- **“Stateless”**

\[ A \rightarrow B \]

\[ B \rightarrow C \]

\[ zF: A \rightarrow B \rightarrow C \]
Forwarding Decision

- Forwarding decision based on binary AND and CMP
  - zFilter in the packet matched with all outgoing Link IDs
  - Multicasting: zFilter contains more than one outgoing links
Using Link Identity Tags (LIT)

- Make results better with a simple trick
  - Define $d$ different LITs instead of a single LID
  - LIT has the same size as LID, and also $k$ bits set to 1
  - [Power of choices]

- Route creation and packet forwarding
  - Calculate $d$ different candidate zFilters
  - Select the best performing zFilter, based on some policy

```
<table>
<thead>
<tr>
<th>Host 1: Iface out</th>
<th>Host 2: Iface out</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link ID</td>
<td>Link ID</td>
</tr>
<tr>
<td>LIT 1</td>
<td>LIT 1</td>
</tr>
<tr>
<td>LIT 2</td>
<td>LIT 2</td>
</tr>
<tr>
<td>LIT d</td>
<td>LIT d</td>
</tr>
</tbody>
</table>
```

Candidate zFilter
- zFilter 1
- zFilter 2
- zFilter $d$
Using Link Identity Tags (LIT)

- LIT1
- LIT2
- LITd

BF

Yes/No

n? & =

n BF
Forwarding efficiency

- Simulations with
  - Rocketfuel
  - SNDlib
- Forwarding efficiency with 20 subscribers
  - ~80%
Forwarding efficiency

- Simulations with
  - Rocketfuel
  - SNDlib
- Forwarding efficiency with 20 subscribers
  - ~80%
  - LIT Optimized: 88 %
Virtual Trees

- Popular paths can be merged into virtual trees
  - A single Link ID for the tree
  - Additional state in the forwarding nodes
  - Increase scalability

Virtual B→C→D→E  0 0 1 0 1 0 0 0 1
Implementation

- NS3 simulator
- FreeBSD 7.x : end-host + forwarding
- NetFPGA : Forwarding
- BSD & NetFPGA Implementations available at http://www.psirp.org
Summary

- Link-identity-based source routing
- Stateless small-group multicast and unicast
- Small forwarding table
- Very simple forwarding decision
- Preventing unwanted traffic
  - No possibility to send data by guessing the destination
- Forwarding implemented: both software and hardware