Tarpan: a router that supports evolvability

Harshal Sheth, Andrew Sun
Inter-Domain Routing

- Creates paths between destination ISPs and source ones
  - e.g. Netflix to Comcast
- Paths can be used to deliver traffic (from sources to destinations)
- Provided today by BGP
Inter-Domain Routing

- Creates paths between destination ISPs and source ones
  - e.g. Netflix to Comcast
- Paths can be used to deliver traffic (from sources to destinations)
- Provided today by BGP
Inter-Domain Routing

- Creates paths between destination ISPs and source ones
  - e.g. Netflix to Comcast
- Paths can be used to deliver traffic (from sources to destinations)
- Provided today by BGP
Problems with BGP and Proposed Solutions
Problems with BGP and Proposed Solutions

<table>
<thead>
<tr>
<th>Problem</th>
<th>Proposed Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only one possible route</td>
<td>Pathlets: gives greater freedom in route selection by presenting multiple path fragments</td>
</tr>
</tbody>
</table>
# Problems with BGP and Proposed Solutions

<table>
<thead>
<tr>
<th>Problem</th>
<th>Proposed Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only one possible route</td>
<td>Pathlets: gives greater freedom in route selection by presenting multiple path fragments</td>
</tr>
<tr>
<td>Does not account for cost or link quality</td>
<td>Wiser: adds a cost value that could represent either cost or link quality</td>
</tr>
</tbody>
</table>
## Problems with BGP and Proposed Solutions

<table>
<thead>
<tr>
<th>Problem</th>
<th>Proposed Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only one possible route</td>
<td>Pathlets: gives greater freedom in route selection by presenting multiple path fragments</td>
</tr>
<tr>
<td>Does not account for cost or link quality</td>
<td>Wiser: adds a cost value that could represent either cost or link quality</td>
</tr>
<tr>
<td>Trivial to hijack routes</td>
<td>S-BGP: adds authentication to BGP</td>
</tr>
<tr>
<td></td>
<td>BGPsec: digital signatures in advertisements</td>
</tr>
</tbody>
</table>
## Problems with BGP and Proposed Solutions

<table>
<thead>
<tr>
<th>Problem</th>
<th>Proposed Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only one possible route</td>
<td>Pathlets: gives greater freedom in route selection by presenting multiple path fragments</td>
</tr>
<tr>
<td>Does not account for cost or link quality</td>
<td>Wiser: adds a cost value that could represent either cost or link quality</td>
</tr>
<tr>
<td>Trivial to hijack routes</td>
<td>S-BGP: adds authentication to BGP</td>
</tr>
<tr>
<td></td>
<td>BGPsec: digital signatures in advertisements</td>
</tr>
<tr>
<td>BGP is rigid, assumes that all other routers are using BGP</td>
<td>Evolvable routing protocols</td>
</tr>
</tbody>
</table>
Example: Deploying Wiser

- Path costs
  - represents cost of sending on link and/or link saturation
- Cost normalization
  - prevents ISPs from inflating their path costs
- Bidirectional protocol
Example: Deploying Wiser

• Path costs
  • represents cost of sending on link and/or link saturation
• Cost normalization
  • prevents ISPs from inflating their path costs
• Bidirectional protocol
Example: Deploying Wiser

- Path costs
  - represents cost of sending on link and/or link saturation
- Cost normalization
  - prevents ISPs from inflating their path costs
- Bidirectional protocol
Example: Deploying Wiser

- Path costs
  - represents cost of sending on link and/or link saturation
- Cost normalization
  - prevents ISPs from inflating their path costs
- Bidirectional protocol
Example: Deploying Wiser

- Path costs
  - represents cost of sending on link and/or link saturation
- Cost normalization
  - prevents ISPs from inflating their path costs
- Bidirectional protocol
Example: Deploying Wiser

- Path costs
  - represents cost of sending on link and/or link saturation
- Cost normalization
  - prevents ISPs from inflating their path costs
- Bidirectional protocol
Previous Work
Previous Work

- Identified features needed to support evolvability
  - Pass-through support
  - Multi-protocol support
Previous Work

- Identified features needed to support evolvability
  - Pass-through support
  - Multi-protocol support
- Initial implementation sends multi-protocol data out-of-band
  - i.e., Wiser routers must use external service to communicate info
  - Allows routers to be simpler, but hides info from ISPs
Previous Work

- Identified features needed to support evolvability
  - Pass-through support
  - Multi-protocol support
- Initial implementation sends multi-protocol data out-of-band
  - i.e., Wiser routers must use external service to communicate info
  - Allows routers to be simpler, but hides info from ISPs

Our work: testing BGP evolvability in-band
Outline

• BGP and its Shortfalls
• Previous Work
• Tarpan
  • Design
  • Implementation
• Evaluation
• Conclusion and Future Work
Tarpan

- Favors in-band communication
- Encodes multiple protocols
- Passes through unknown protocols
- Can cross gulfs
- Inter-operates with non-Tarpan routers
Data Structure

- Includes information from multiple protocols
- Tarpan operates as an extension to BGP
Implementation

- Implemented within Quagga, an open-source network routing suite
  - Quagga itself was a fork of Zebra
- Tarpan API for simply protocol addition
- Protocol Buffers for efficient data transfer
- Interposes on BGP route selection mechanisms
- About 2000 lines of code added or changed
Modifications within Quagga
Modifications within Quagga

Key

- Functionality
  - original Quagga function
  - modified Quagga function
  - Tarpan function

Diagram:
- Advertisement Received
  - bgp_update_receive
    - bgp_attr_parse
      - bgp_tarpan_parse
Modifications within Quagga

Key

- **Functionality**
  - original Quagga function
  - modified Quagga function
  - Tarpan function

Diagram:
- Advertisement Received
  - bgp_update_receive
  - bgp_attr_parse
  - bgp_tarpan_parse

- Advertisement Must Be Sent
  - bgp_update_packet
  - bgp_packet_attribute
  - tarpan_initialize_packet
  - tarpan_update_packet
Modifications within Quagga

Key

Functionality
- original Quagga function
- modified Quagga function
- Tarpan function
Challenges

• Interning
  • Custom memory management scheme
  • Breaks attributes into pieces for memory de-duplication
  • Interns most internal data structures
• Integrating with Quagga’s interning system was major source of frustration
Outline

• BGP and its Shortfalls
• Previous Work
• Tarpan
• Evaluation
  • Wiser Testing
  • Throughput Measurement
  • Large Payload Behavior
• Conclusion and Future Work
Experimental Setup

- Topologies emulated with miniNExT
- Ubuntu virtual machine on Massachusetts Open Cloud
  - 16 vCPUs
  - 64 GB RAM
Wiser Experiment Overview

• Proof of concept to demonstrate that Tarpan functions as intended
• Tarpan was extended to use out-of-band communication to support Wiser’s cost normalization using bidirectional communication
• Implemented within Tarpan
Wiser Testing

• Ensuring proper Wiser functionality
• Manual verification of route selections
Destination

Source

Wiser data

pass-through

bidirectional,
out-of-band
communication

Quagga
BGP

Tarpan
BGP + Wiser

Path Cost
Throughput Setup

- Two virtual switches in miniNExT
- 8 bgpsimple scripts send actual routing tables into the router
- The router is either Tarpan or Quagga, with instrumentation for timing
Throughput Evaluation

- Graphs show inverted throughput (lower is better)

**Tarpan Throughput**

```
Throughput (seconds per 100,000 packets)
```

```
0 3.75 7.5 11.25 15
```

```
Packets Received
```

```
0 4500000 9000000 13500000 18000000
```

**Quagga Throughput**

```
Throughput (seconds per 100,000 packets)
```

```
0 3.75 7.5 11.25 15
```

```
Packets Received
```

```
0 4500000 9000000 13500000 18000000
```

Avg.: 20,787 packets/sec  Avg.: 21,026 packets/sec
Large Payloads

• Test the effect of sending larger payloads to routers
• Modified bgpsimple script that sends a string of certain length in an attribute
Large Payloads

![Graph showing the relationship between processing rate (packets/second) and additional data in packets (bytes).]
Future Work

- Convergence properties when running multiple protocols
- Exploring incremental deployment
- Further performance and memory usage improvements
Summary

• BGP is too rigid - cannot support deployment of new protocols across gulfs
• Tarpan allows new protocols to be deployed across gulfs by sending information in-band with BGP advertisements
• Wiser implemented using Tarpan’s API
• Low performance overhead
Acknowledgements

• Raja Sambasivan - Mentor
• Massachusetts Open Cloud - Large virtual machines for testing
• MIT PRIMES Program