Fibbing: Central Control over Distributed Routing

Olivier Tilmans
UCLouvain

IRTF Open Meeting
Nov. 14, 2016

Joint work with
S. Vissicchio (UCL), L. Vanbever (ETH Zürich) and J. Rexford (Princeton)
Fibbing
Fibbing
Control routers’ FIB, lying to routers
Consider this example network.
Consider this example network.
Link-state Interior Gateway Protocols (IGPs) exchange reachability information to infer the topology of the network.
The intra-domain traffic flows along the shortest path on the shared topology.
IGPs cause operator to follow a *descriptive* management process.

Operator

- Express requirements
IGPs cause operator to follow a **descriptive** management process.

**Operator**
- Express requirements
- Computes paths
- Figures out how to implement them

**Protocol configuration operation**
IGPs cause operator to follow a *descriptive* management process.

**Operator**
- Express requirements
- Computes paths
- Figures out how to implement them

**Protocol configuration operation**

**Distributed Control-Plane**
- Derives FIB entries
- Installs FIB entries
Software-Defined Networking (SDN) enables *declarative* management.
Software-Defined Networking (SDN) enables *declarative* management.
Software-Defined Networking (SDN) enables *declarative* management.

SDN Controller
- Compute paths
- Derives FIB entries
- install FIB entries

Operator
- Express requirements

Well-defined API
**SDN sacrifices** the robustness and scalability of distributed protocols.

<table>
<thead>
<tr>
<th></th>
<th>Traditional</th>
<th>SDN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manageability</td>
<td>low</td>
<td>high</td>
</tr>
<tr>
<td>Flexibility</td>
<td>low</td>
<td>highest</td>
</tr>
<tr>
<td>Scalability</td>
<td>by design</td>
<td>ad hoc</td>
</tr>
<tr>
<td>Robustness</td>
<td>high</td>
<td>low</td>
</tr>
</tbody>
</table>
The networking world has two paradigm, based on opposed principles.

<table>
<thead>
<tr>
<th></th>
<th>Traditional</th>
<th>SDN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manageability</td>
<td>low</td>
<td>high</td>
</tr>
<tr>
<td>Flexibility</td>
<td>low</td>
<td>highest</td>
</tr>
<tr>
<td>Scalability</td>
<td>by design</td>
<td>ad hoc</td>
</tr>
<tr>
<td>Robustness</td>
<td>high</td>
<td>low</td>
</tr>
</tbody>
</table>
We propose a middleground approach, named *Fibbing*.

<table>
<thead>
<tr>
<th></th>
<th>Traditional</th>
<th>Fibbing</th>
<th>SDN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manageability</td>
<td>low</td>
<td>high</td>
<td>high</td>
</tr>
<tr>
<td>Flexibility</td>
<td>low</td>
<td>high</td>
<td>highest</td>
</tr>
<tr>
<td>Scalability</td>
<td>by design</td>
<td>by design</td>
<td>ad hoc</td>
</tr>
<tr>
<td>Robustness</td>
<td>high</td>
<td>high</td>
<td>low</td>
</tr>
</tbody>
</table>
Fibbing: Central Control over Distributed Routing

1. Controlling distributed protocols
2. Case study: surviving flash crowds
3. Fibbing today’s networks
4. Food for thoughts
Fibbing uses an hybrid control plane.
Fibbing centralizes high level routing decisions.

Operator
- Express requirements

Fibbing Controller
- Compute paths

Distributed Control-Plane
- Derives FIB entries
- install FIB entries

IGP messages
Fibbing keeps the route installation distributed.

Operator

- Express requirements

Fibbing Controller

- Compute paths

Distributed Control-Plane

- Derives FIB entries
- install FIB entries
We study which IGP messages to inject.

Operator

- Express requirements

Fibbing Controller

- Compute paths

Distributed Control-Plane

- Derives FIB entries
- Install FIB entries

IGP messages
Operators specify paths that must be enforced.
The controller injects one IGP message adding a fake node and links.

requirements
(A, R1, R2, C, blue)

node fA (blue),
link (fA, A, 2),
map (fA, A) to (A, R1)
IGP flooding propagates the information.

requirements
(A, R1, R2, C, blue)

node fA (blue),
link (fA, A, 2),
map (fA, A) to (A, R1)
The Fibbing message *augments* the topology.

requirements
(A, R1, R2, C, blue)
	node fA (blue),
link (fA, A, 2),
map (fA, A) to (A, R1)
Augmented topologies translate into new control-plane paths.

requirements
(A, R1, R2, C, blue)

node fA (blue),
link (fA, A, 2),
map (fA, A) to (A, R1)
Augmented topologies translate into new data-plane paths.

Control-Plane

Data-Plane

requirements
(A, R1, R2, C, blue)

node fA (blue),
link (fA, A, 2),
imap (fA, A) to (A, R1)
Chaining multiple fake nodes enables to program complex paths.

requirements
(A, R1, R4, C, blue)

node fA (blue),
link (fA, A, 2),
map (fA, A) to (A, R1)
Chaining multiple fake nodes enables to program complex paths.

requirements
(A, R1, R4, C, blue)

node fR1 (blue), link (fR1, R1, 2), map (fR1, r1) to (R1, R4)
Chaining multiple fake nodes enables to program complex paths.

requirements
(A, R1, R4, C, blue)
Augmented topologies can be reduced to optimize the number of fake nodes.

Naive augmentation

Reduced augmentation

requirements (A, R1, R4, C, blue)
Fibbing preserves the scalability of IGPs.

- We can compute augmented topologies in $O(ms)$
  Ensures quick reaction to changes

- We can *reduce* augmented topologies in $O(s)$
  Ensures limited control-plane overhead
Fibbing leverages the robustness of IGPs.

- Fast failure detection and recovery

- Survive controller failure

- Support fail-close and fail-open semantics
Fibbing can enforce any set of loop-free paths, on a per destination basis.
Fibbing: Central Control over Distributed Routing

1. Controlling distributed protocols
2. Case study: surviving flash crowds
3. Fibbing today’s networks
4. Food for thoughts
Flash crowds cause service disruption.

- Video delivery services require good network performance
Flash crowds cause service disruption.

- Video delivery services require good network performance

- Protecting the services against flash crowds is challenging:
  1. Traditional traffic engineering techniques perform poorly;
  2. Over-provisioning is expensive.
Fibbing reduces the need for over-provisioning by enabling real-time traffic engineering.

Experiment setup

- Network with 2 video streaming servers
- Multiple clients are competing for bandwidth
- The network controller is able to detect flash crowds
The initial IGP configuration has a bottleneck towards router C.

Control-Plane

Data-Plane

Overloaded link

Link load

Video server

Video client
Fibbing can program on-demand ECMP to spread the load
Fibbing can program on-demand ECMP to spread the load.

Control-Plane

Data-Plane
Fibbing controls the splitting ratios across equal-cost paths.

**Control-Plane**

- A
- B
- C
- R1
- R2
- R3
- R4

**Data-Plane**

- D1
- D2
- S1
- S2
As the demand increases, the Fibbing controller adds more paths to spread the load.

- We initially have 1 video stream from S1 to D1.
- At time $t = 14s$, we start 30 new streams from S1 to D1.
- At time $t = 35s$, we start 30 streams from S2 to D2.
Fibbing: Central Control over Distributed Routing

www.fibbing.net

1. Controlling distributed protocols
2. Case study: surviving flash crowds
3. Fibbing today’s networks
4. Food for thoughts
We have a working Fibbing controller prototype

- The controller maintains an OSPF adjacency to one router
- Topology discovery using the adjacency
- Tested against IOS, NX-OS, JunOS
Fake nodes can be injected using LSA types 5/7

- Leverages the forwarding address field
  Advertise reachability towards prefix, with cost, using specified IP next hop

- The controller multiplexes multiple virtual routers
  $\mathcal{N}$ successive fake nodes towards the same prefix require $\mathcal{N}$ different router-ids
Using T5/7 LSAs has (almost) no overhead on routers and is fast.

- No measurable impact on SPF duration
- 10,000 LSAs eat 14.5 MB of DRAM
- 900 μs to push one fibbed route to the FIB
Using T5/7 LSAs comes at a price

- Different expressivity model
- Can only affect prefixes from other T5/T7 LSAs
- Does not work with IS-IS!
Fibbing: Central Control over Distributed Routing

www.fibbing.net

1. Controlling distributed protocols
2. Case study: surviving flash crowds
3. Fibbing today’s networks
4. Food for thoughts
Centrally modifying the shared topology is powerful

- Gives *some* control over BGP/MPLS-LDP
- Simplify configurations through exception-based routing
- Enables optimal, real-time TE
What would be the right abstraction?
Tell me lies, tell me sweet little lies
— Fleetwood Mac

Olivier Tilmans
olivier.tilmans@uclouvain.be