# Central Control over Distributed Routing

fibbing.net

Stefano Vissicchio

UCLouvain

SIGCOMM

18th August 2015

Joint work with

O. Tilmans (UCLouvain), L. Vanbever (ETH Zurich) and J. Rexford (Princeton)

SDN is based on antithetical architecture with respect to traditional networking

Traditional
(e.g., IGP, distributed MPLS)

SDN
(e.g., OpenFlow, Segment Routing)

# Centralization improves network management, but *sacrifices* robustness of distributed protocols

| | Traditional | SDN |
|---|---|---|
| Manageability | low | **high** |
| Flexibility | low | **highest** |
| Scalability | **by design** | ad hoc |
| Robustness | **high** | low |

# We propose Fibbing, a hybrid SDN architecture

Fibbing

central control over link-state IGPs

# Fibbing *combines* advantages of SDN and traditional networking

| | Traditional | **Fibbing** | SDN |
|---|---|---|---|
| |  |  |  |
| Manageability | low | **high** | **high** |
| Flexibility | low | **high** | **highest** |
| Scalability | **by design** | **by design** | ad hoc |
| Robustness | **high** | **high** | low |

# Fibbing *combines* advantages of SDN and traditional networking

**Fibbing**


same as SDN

| | |
|---|---|
| Manageability | **high** |
| Flexibility | **high** |
| Scalability | **by design** |
| Robustness | **high** |

per-destination full control

some function are distributed

thanks to partial distribution

# Central Control over Distributed Routing

fibbing.net


1 Manageability

2 Flexibility

3 Scalability

4 Robustness

# Central Control over Distributed Routing

fibbing.net

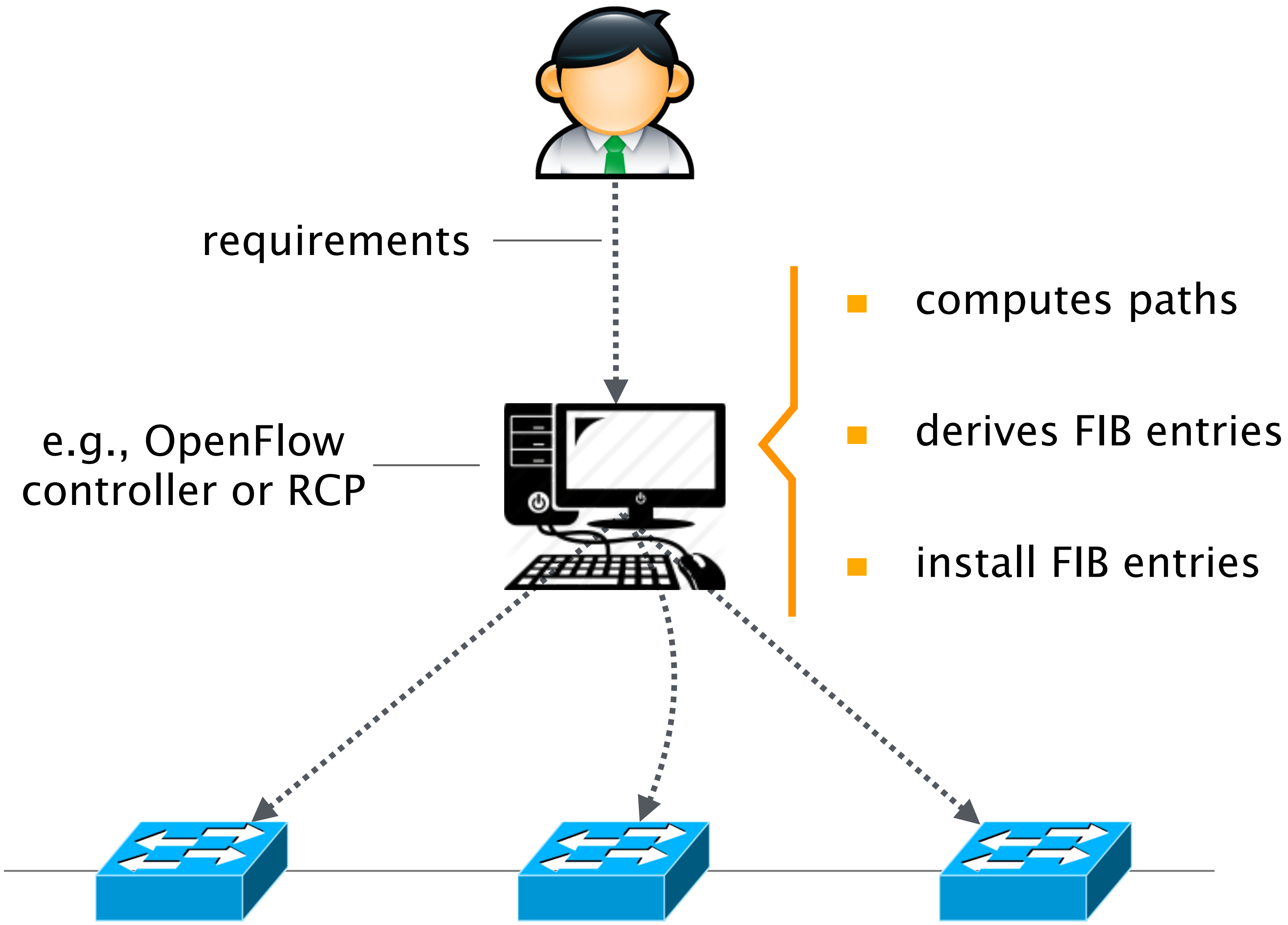

1 Manageability

2 Flexibility

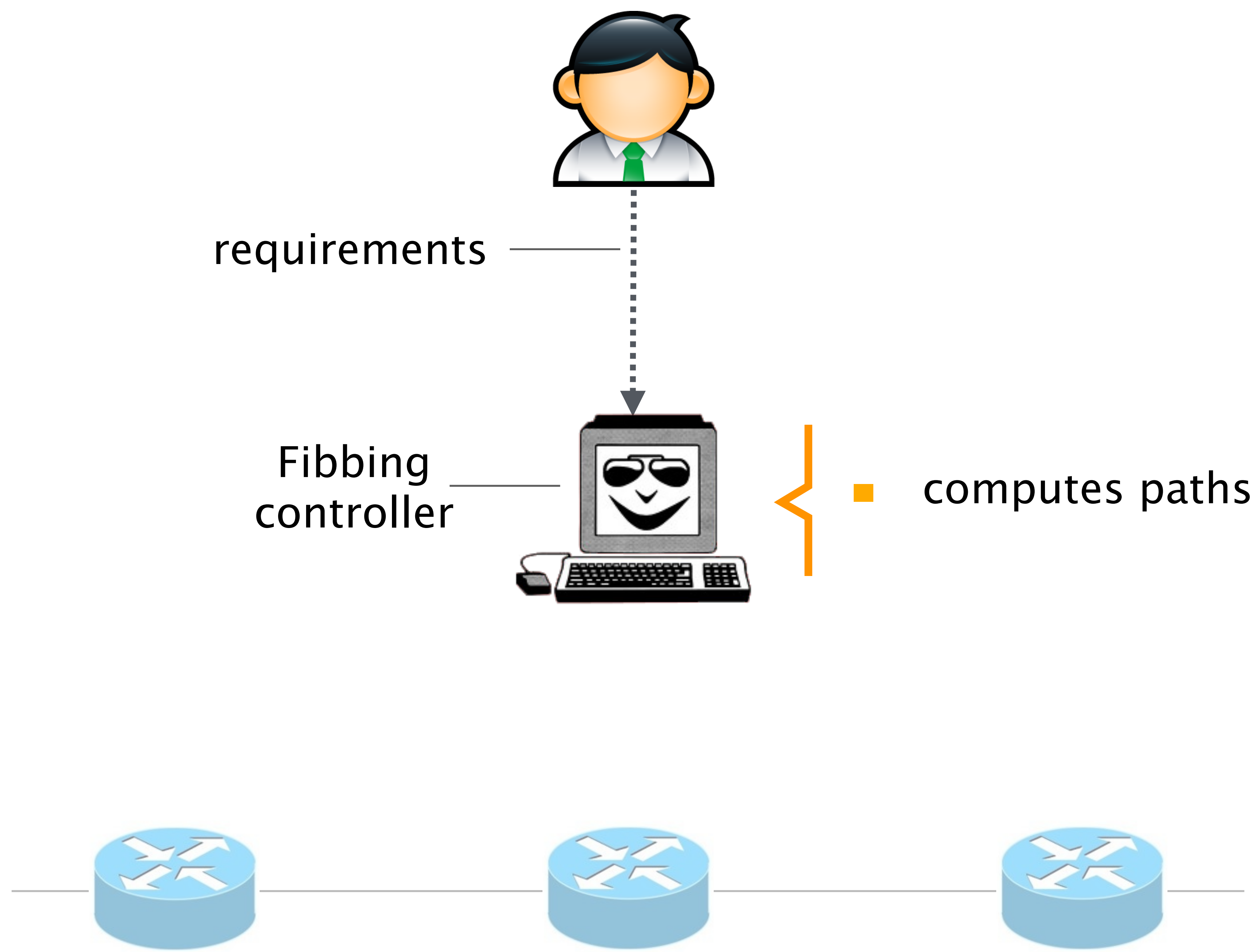
3 Scalability

4 Robustness

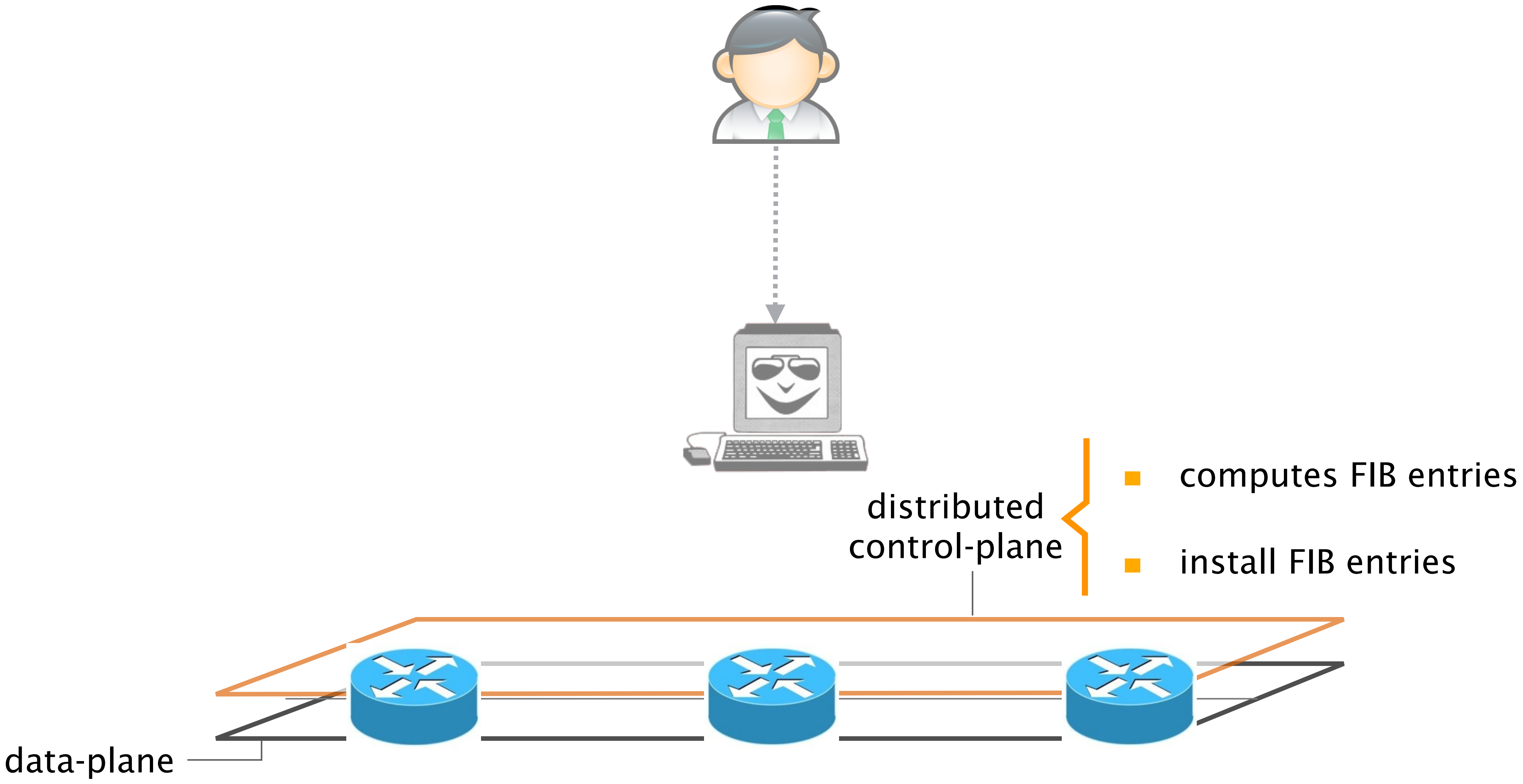
# SDN achieves high manageability by centralizing both computation and installation



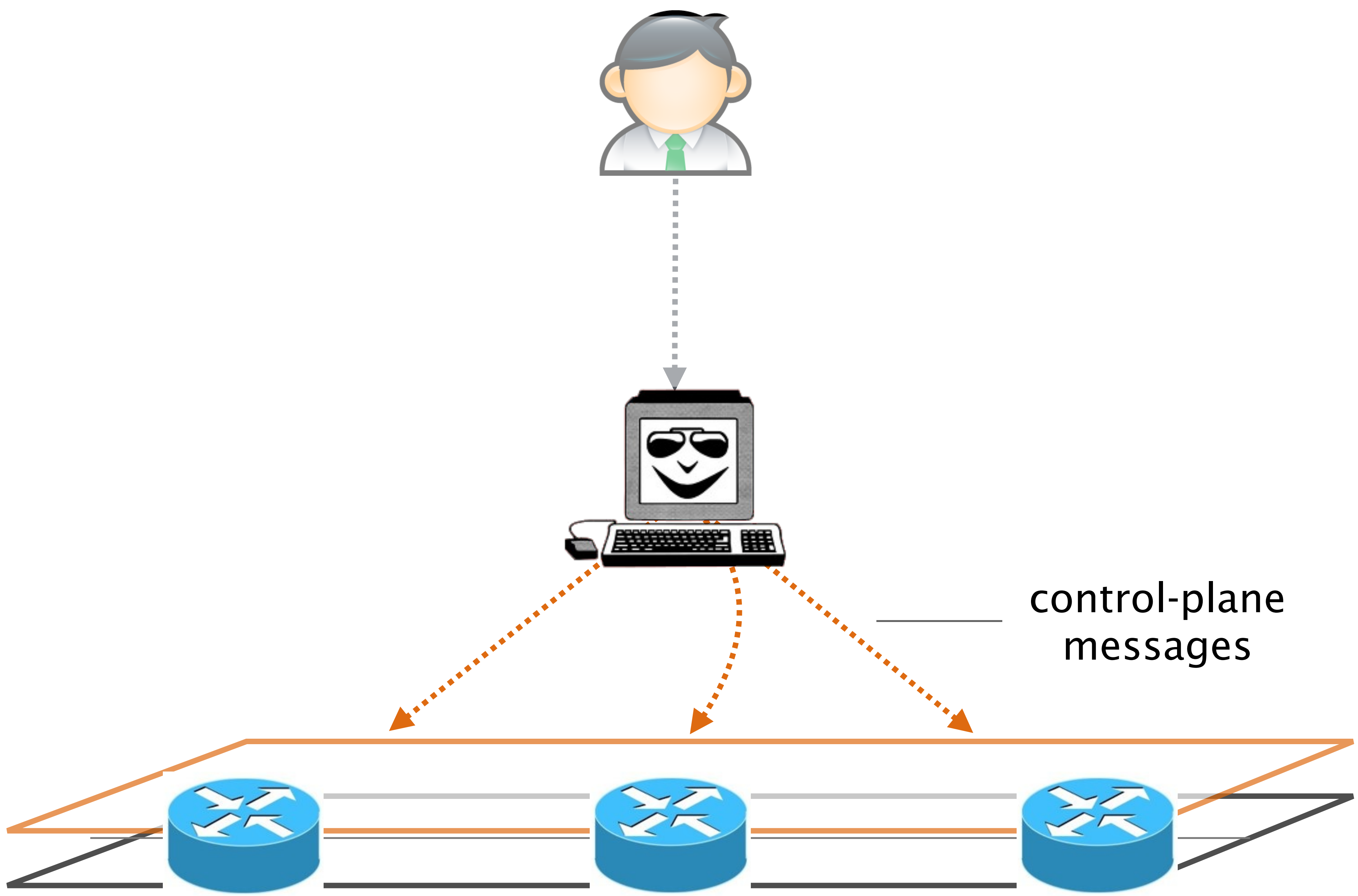
# Fibbing is as manageable as SDN, but centralizes only high-level decisions



# Fibbing keeps installation distributed, relying on distributed protocols



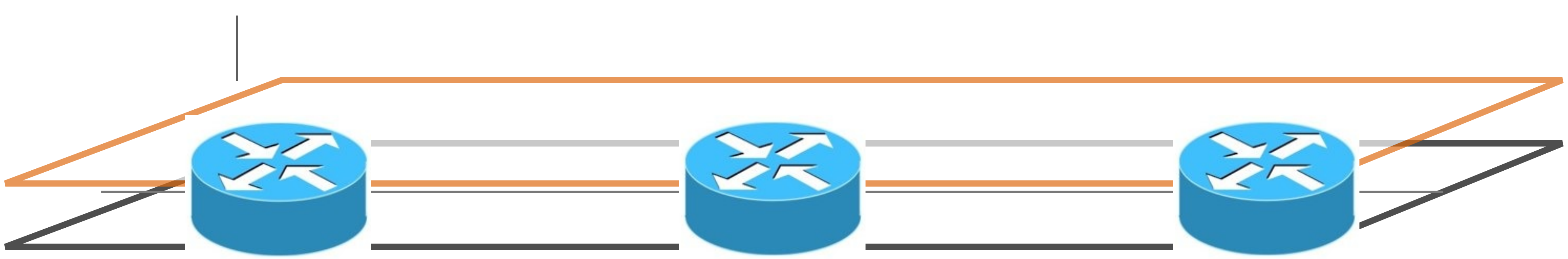
Distributed installation is controlled
by injecting carefully-computed information



We study which messages to inject
for controlling intra-domain routing protocols

**link-state IGP**

| input | function | output |
|---|---|---|
| weighted topology → | shortest-path computation → | forwarding paths |

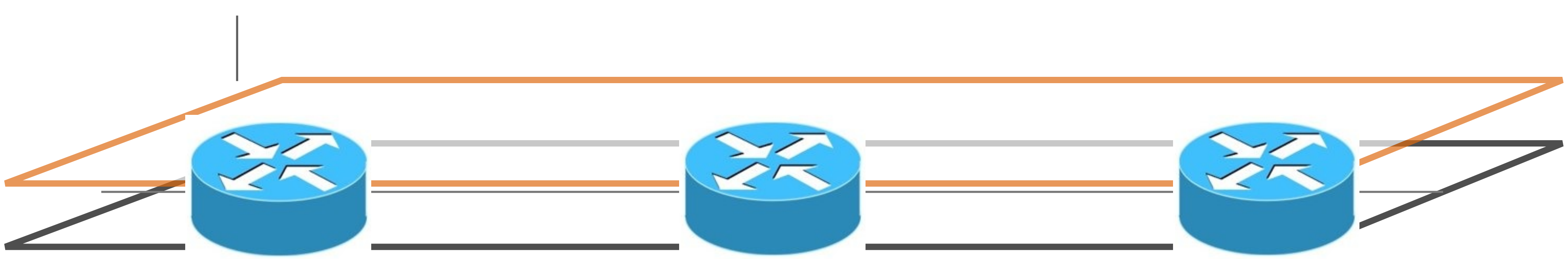


# The output of the controlled protocol is specified by operators' requirements

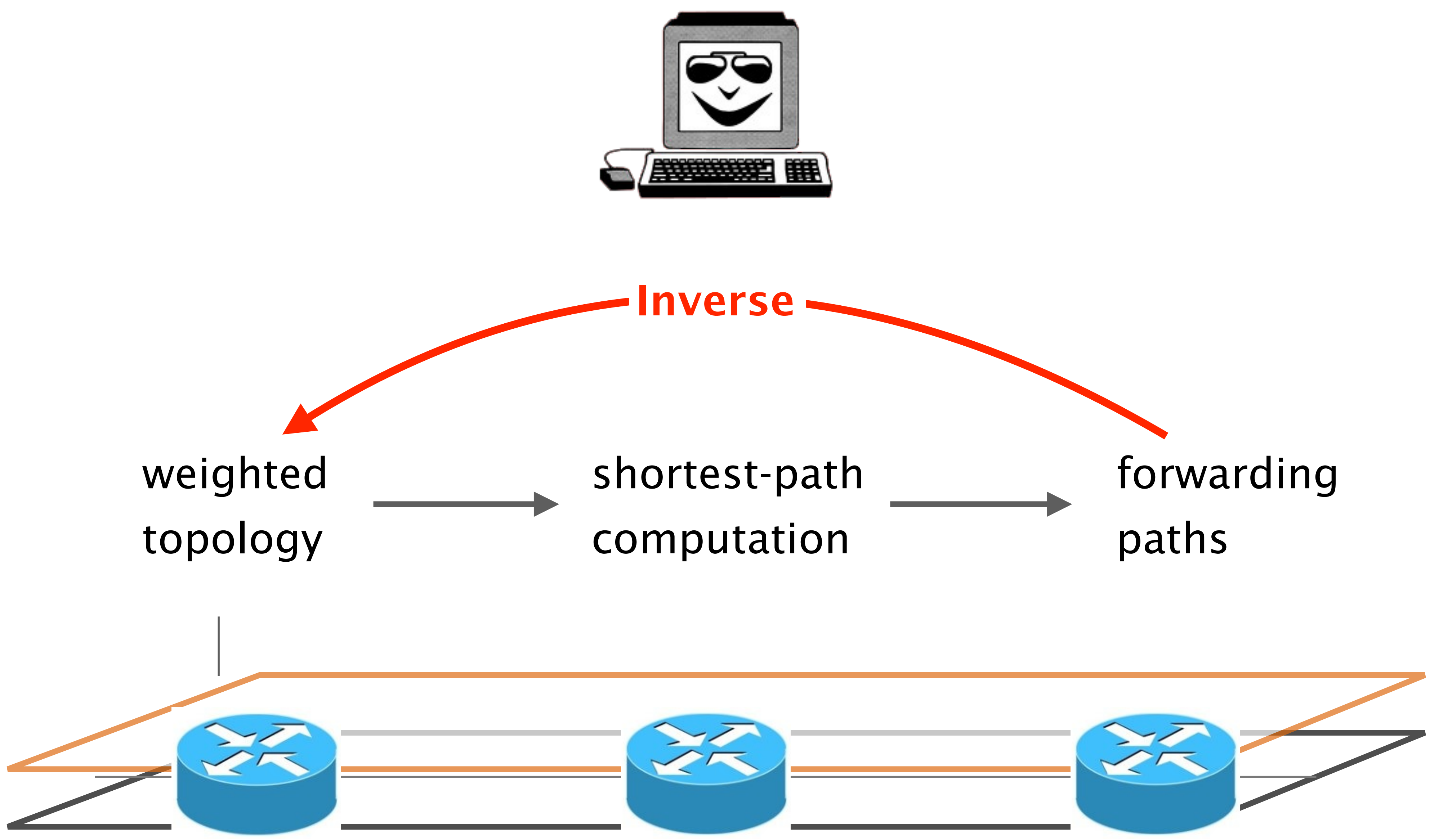
provided by operators
or controller optimizers
(e.g., DEFO)

link-state IGP

| input | function | output |
|---|---|---|
| weighted topology | → shortest-path computation | → forwarding paths |



# To control IGP output, the Fibbing controller inverts the shortest-path function



# Central Control over Distributed Routing

fibbing.net

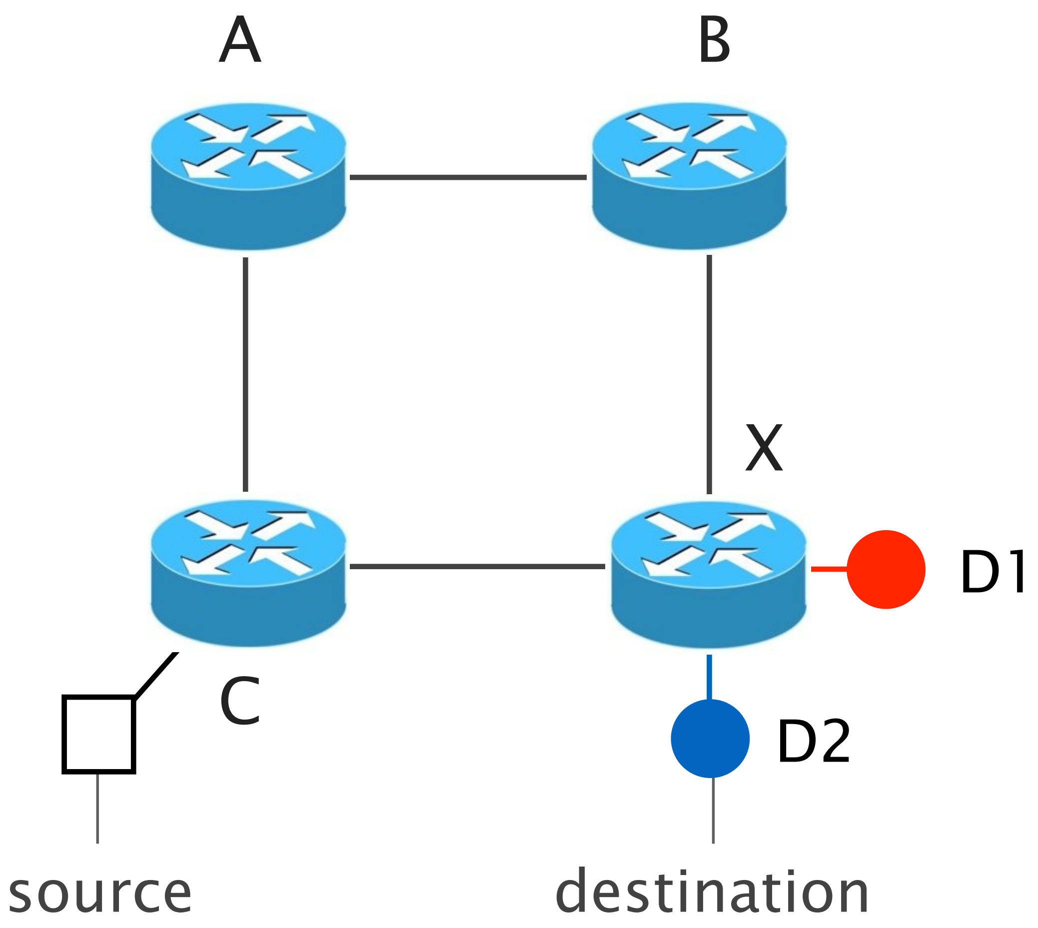

1 Manageability

2 Flexibility

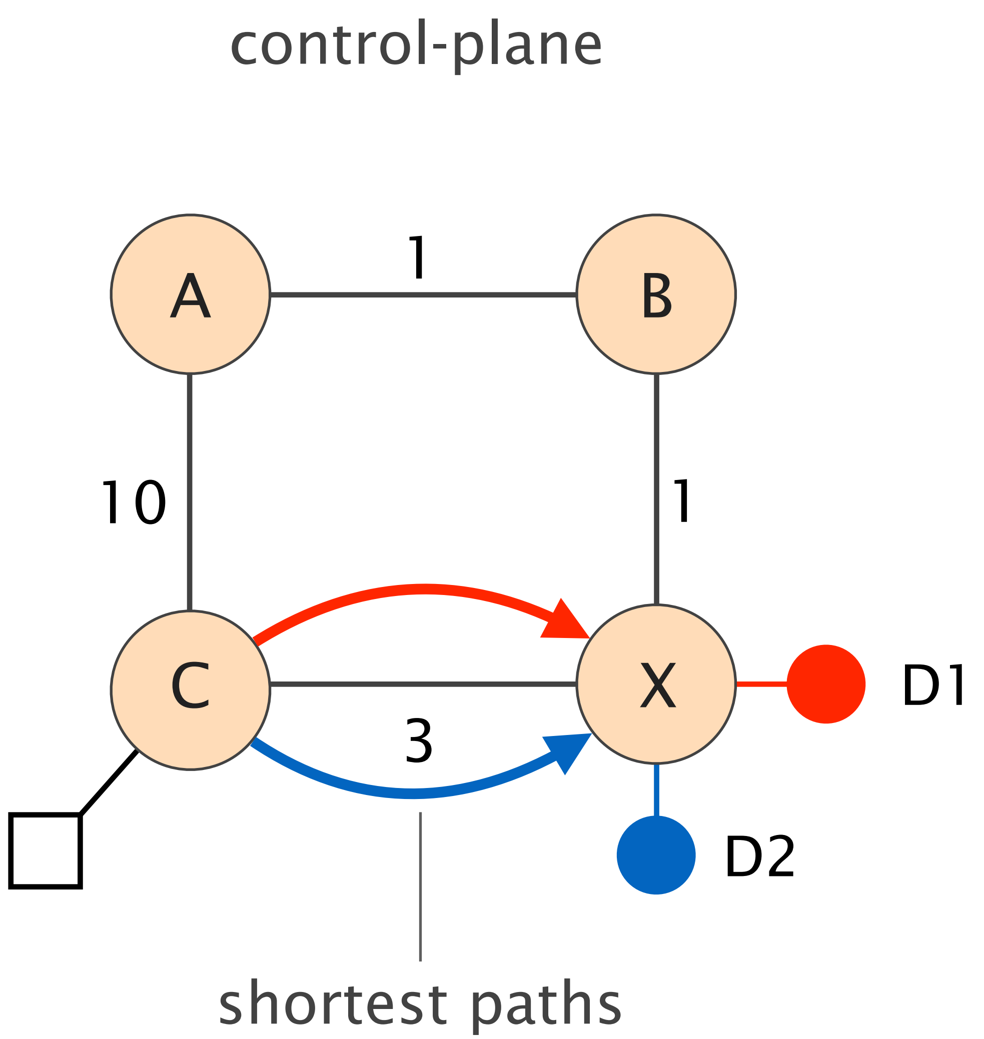
3 Scalability

4 Robustness

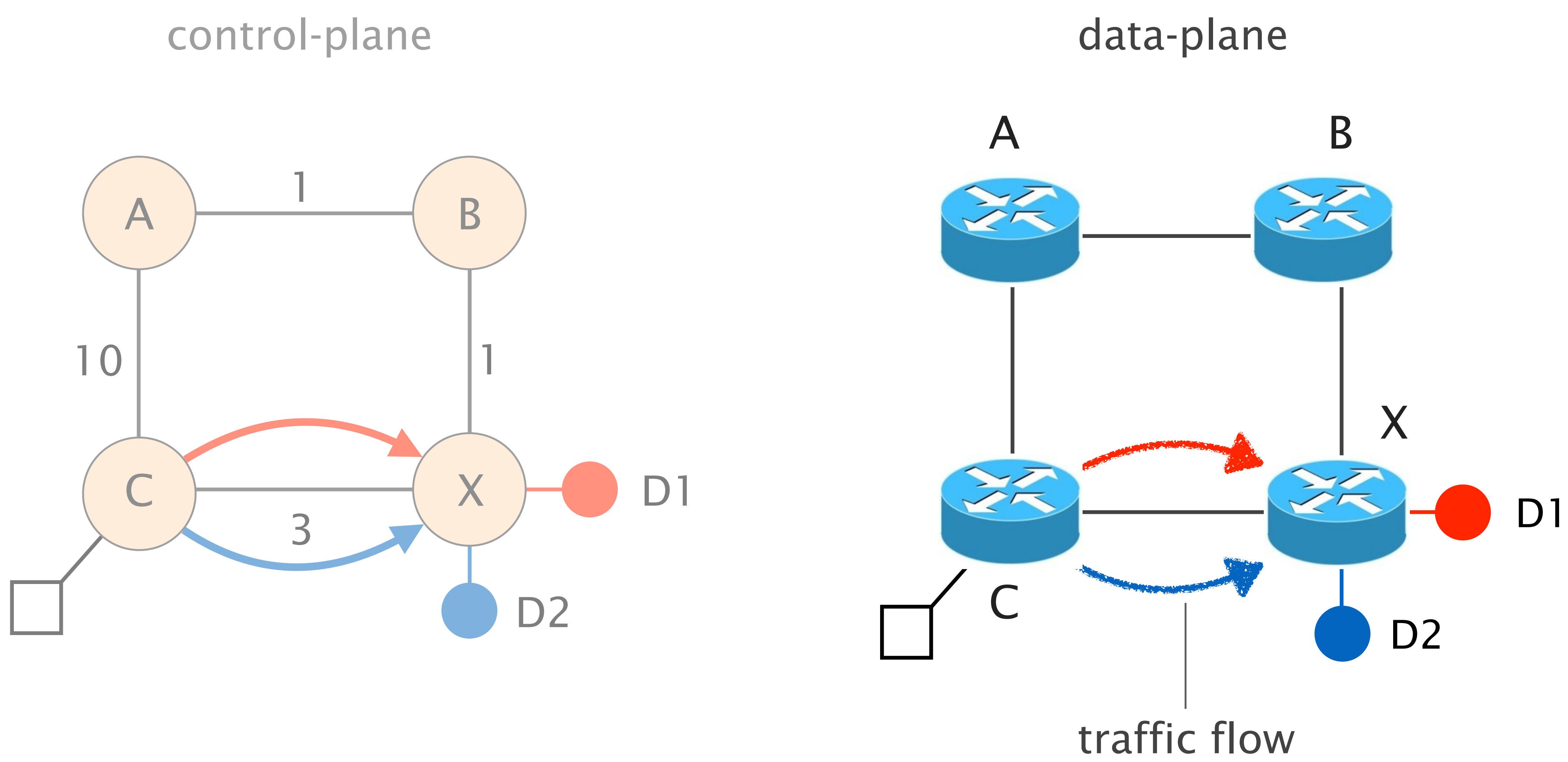
Consider this simple network
(implemented with Cisco routers)



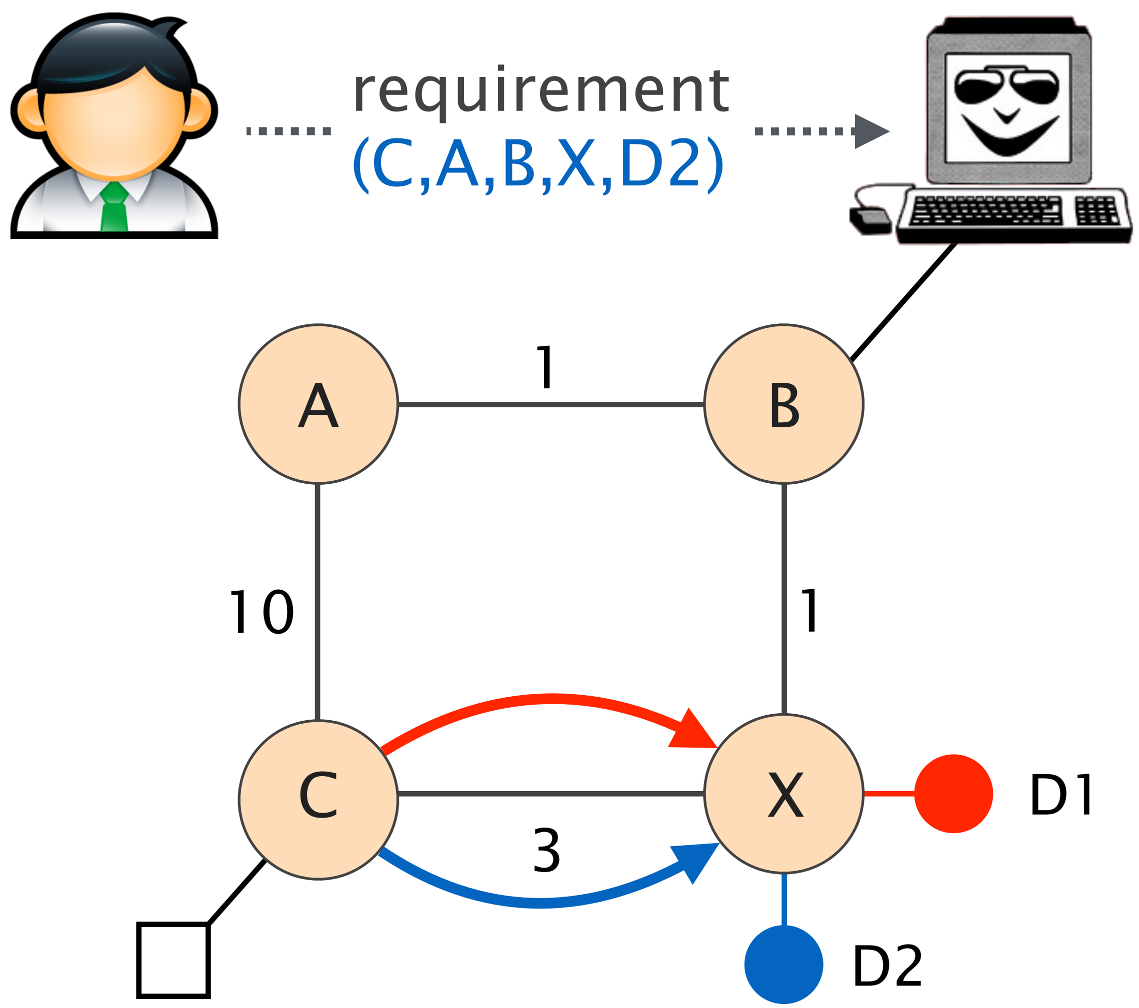
# An IGP control-plane computes shortest paths on a shared weighted topology



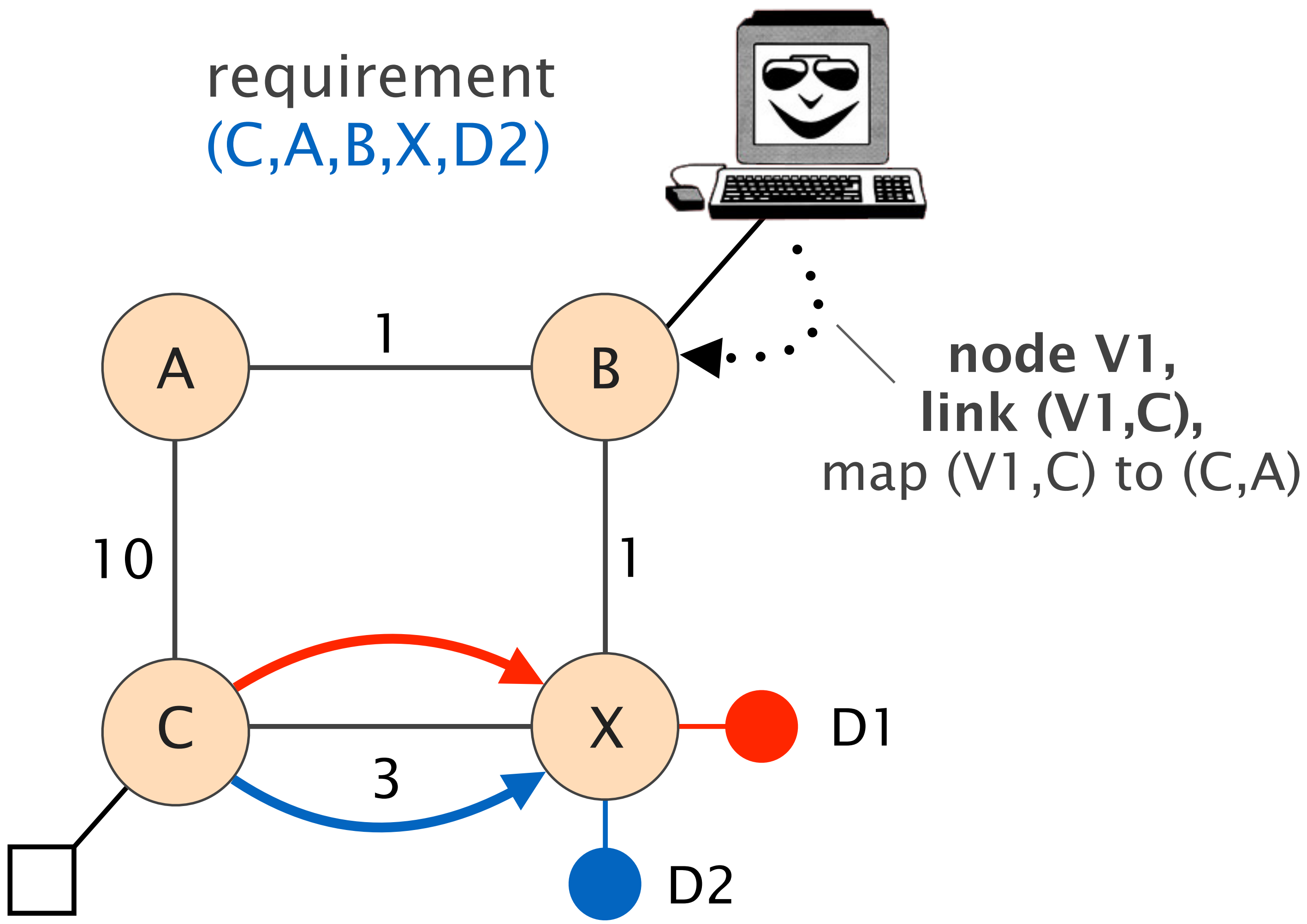
# IGP shortest paths are translated into forwarding paths on the data-plane



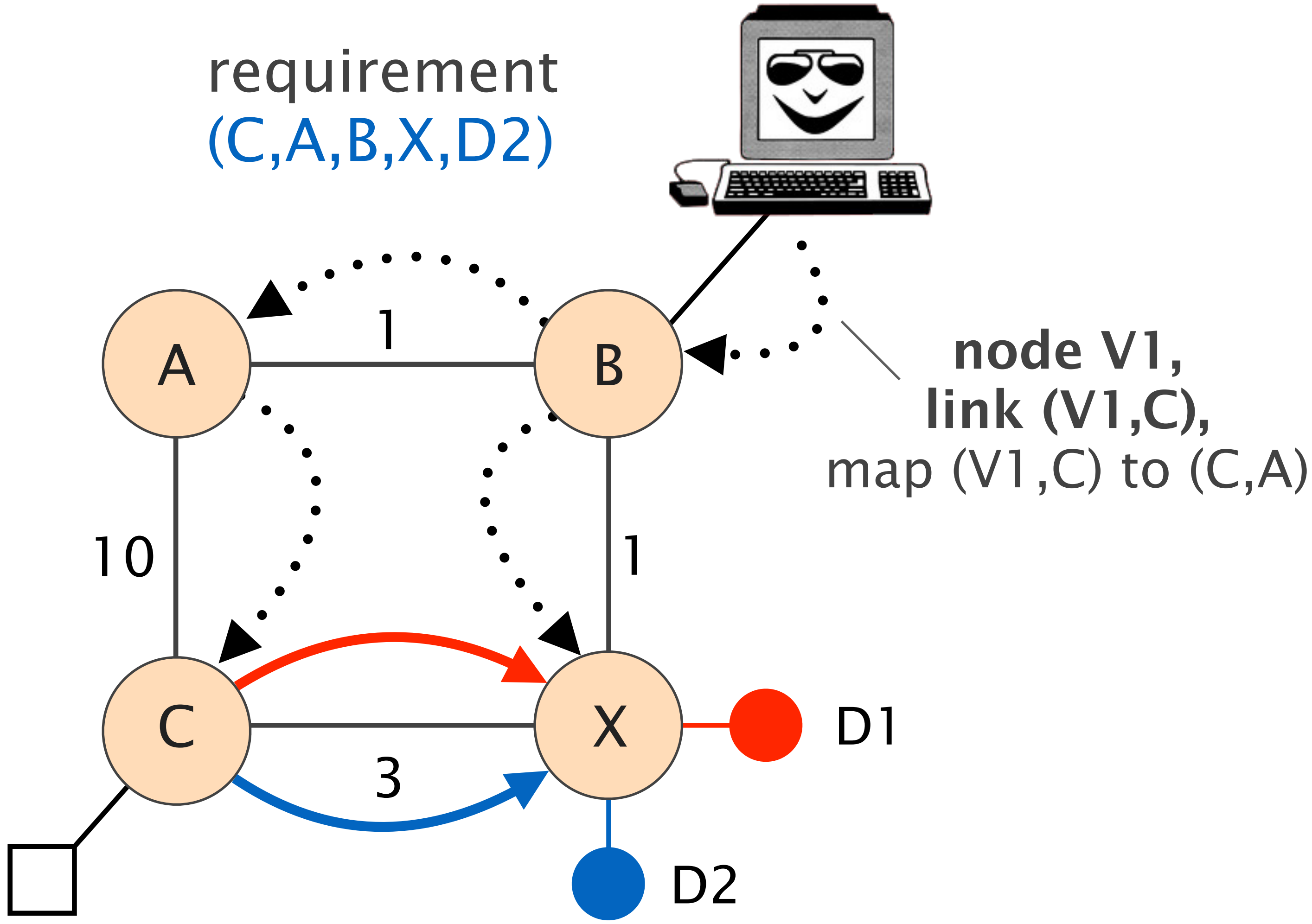
In Fibbing, operators can ask
the controller to modify forwarding paths



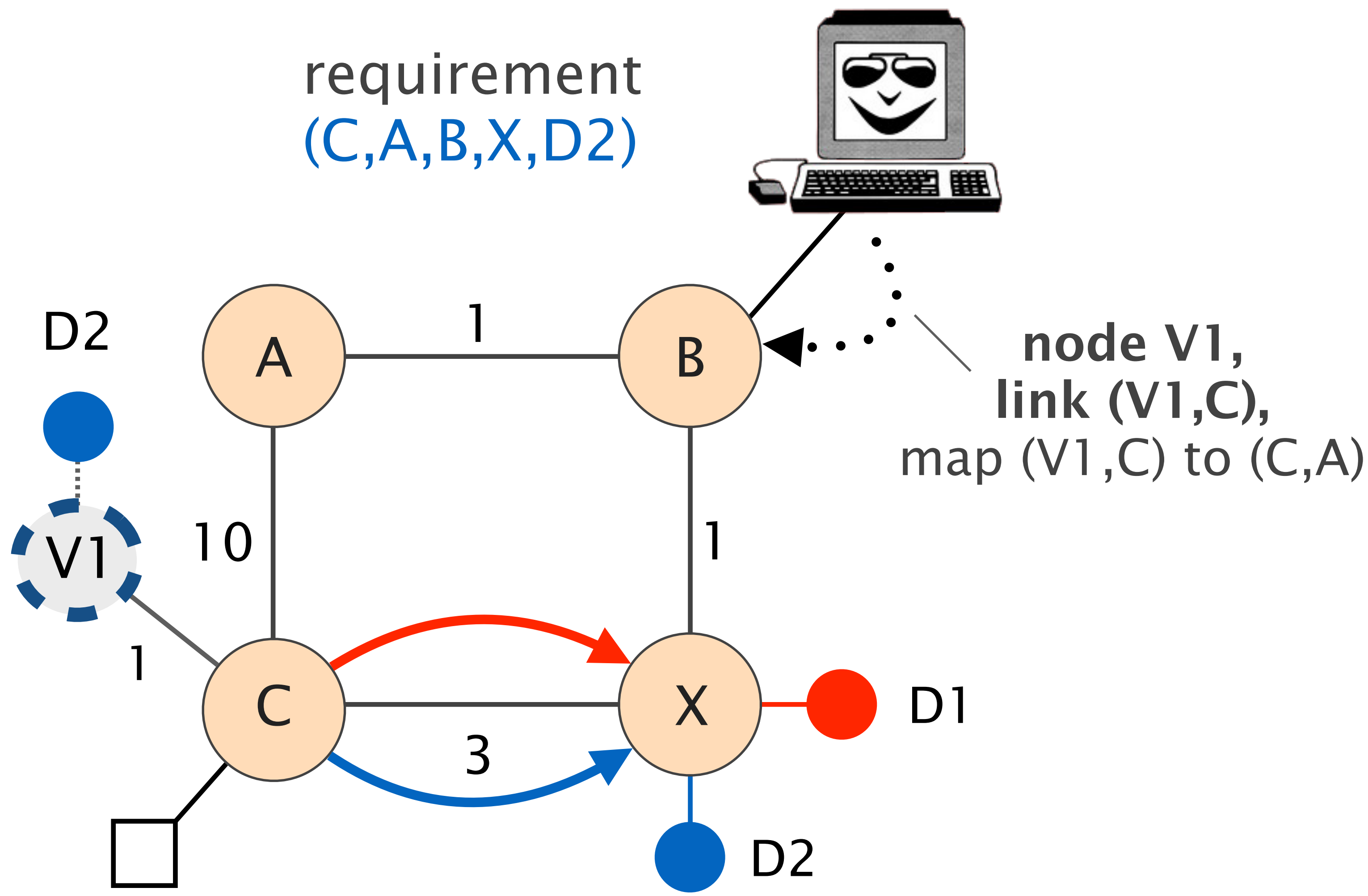
The Fibbing controller injects information on *fake nodes and links* to the IGP control-plane



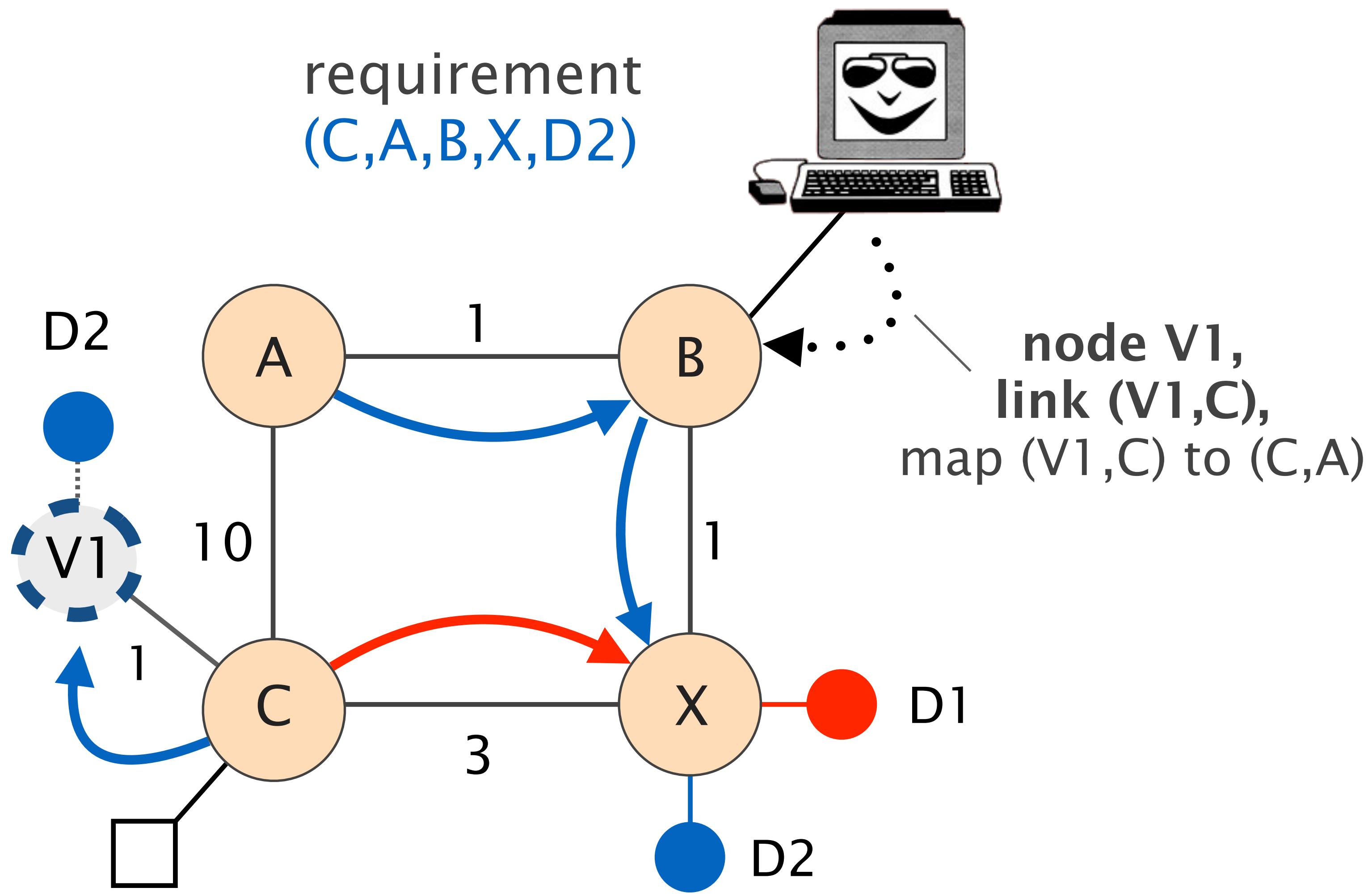
# Informations are flooded to all IGP routers in the network



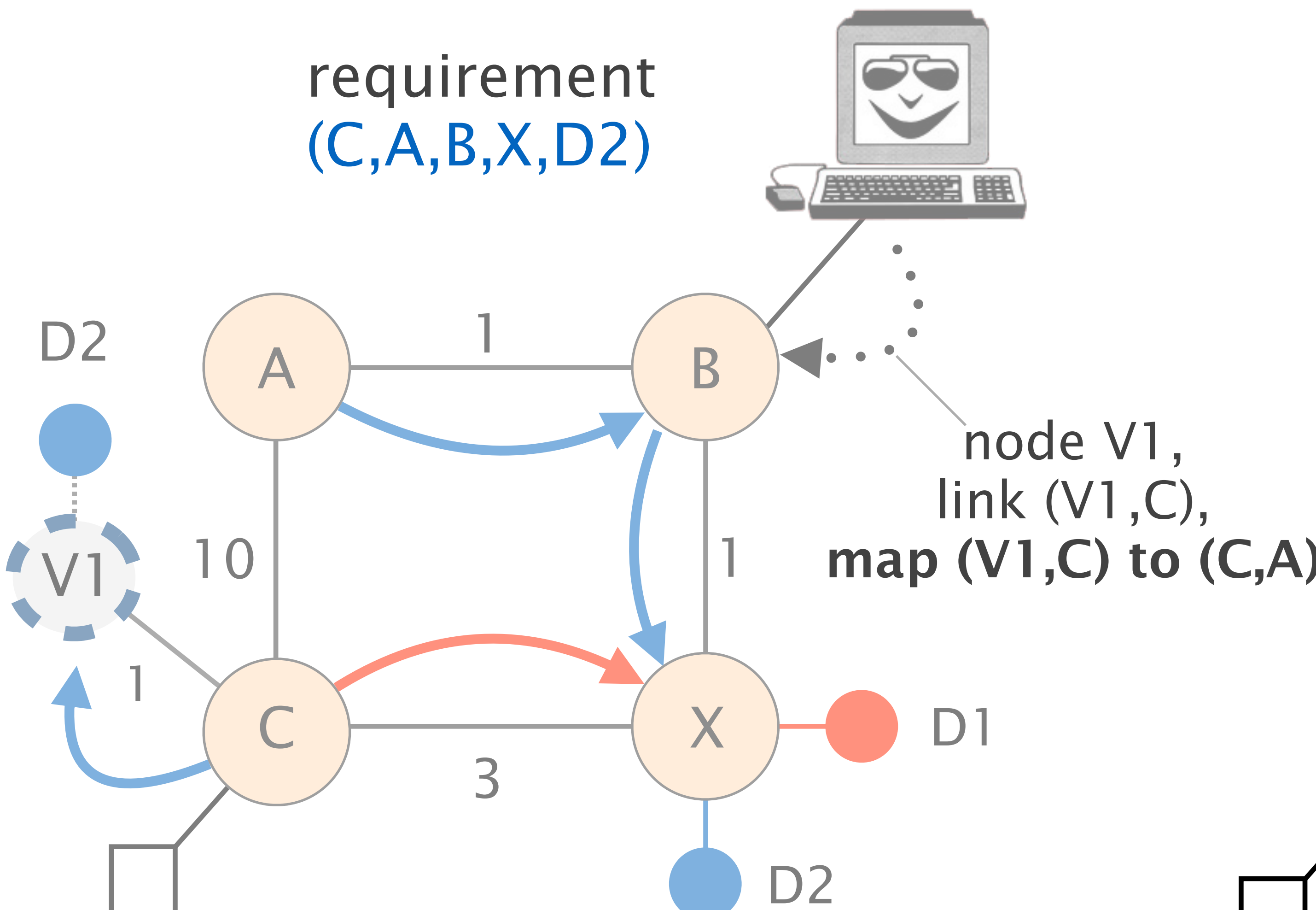
# Fibbing messages *augment* the topology seen by all IGP routers

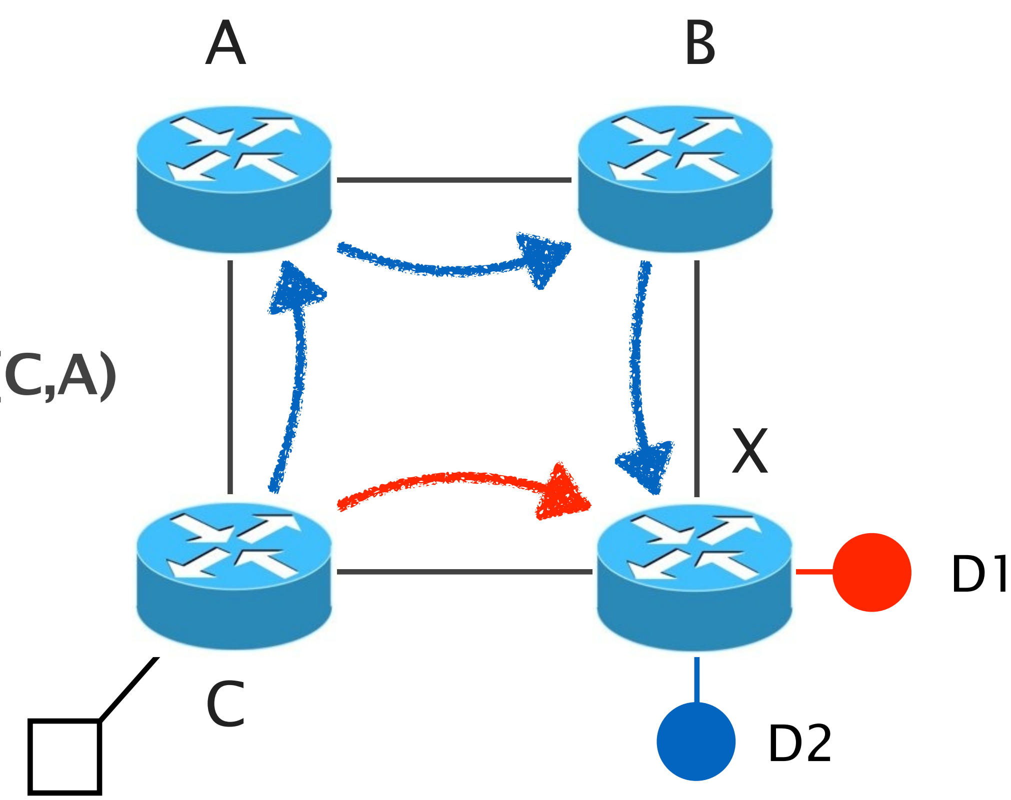


# Augmented topologies translate into new control-plane paths



# Augmented topologies translate into new *data-plane* paths



# Fibbing can program arbitrary per-destination paths

Theorem Any set of forwarding DAGs can be enforced by Fibbing

# Fibbing can program arbitrary per-destination paths

Theorem Any set of forwarding DAGs can be enforced by Fibbing

paths to the same destination do not create loops

# By achieving full per-destination control, Fibbing is highly flexible

Theorem Any set of forwarding DAGs can be enforced by Fibbing

- fine-grained traffic steering (middleboxing)
- per-destination load balancing (traffic engineering)
- backup paths provisioning (failure recovery)

# Central Control over Distributed Routing

## fibbing.net

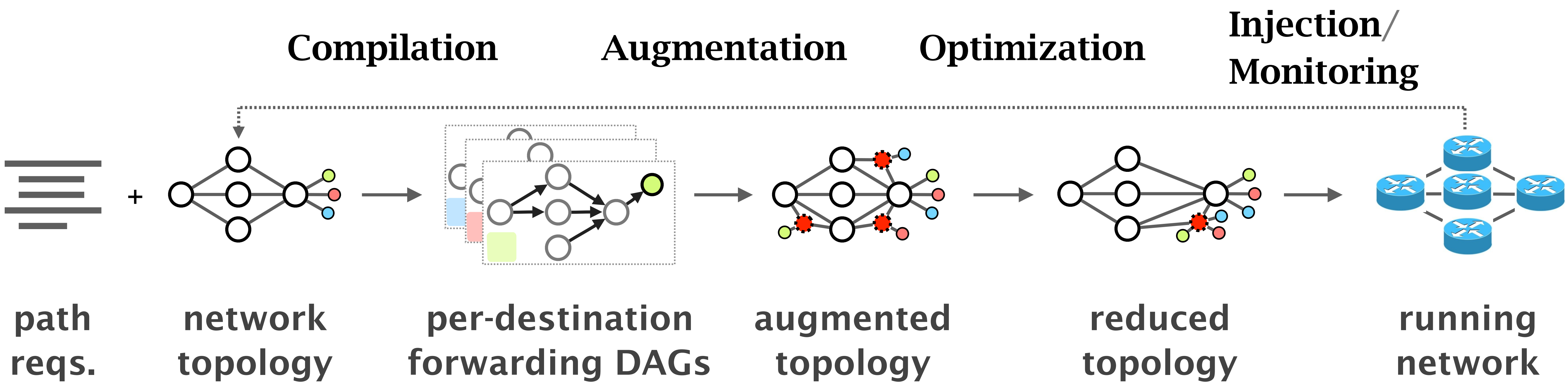

1 Manageability

2 Flexibility

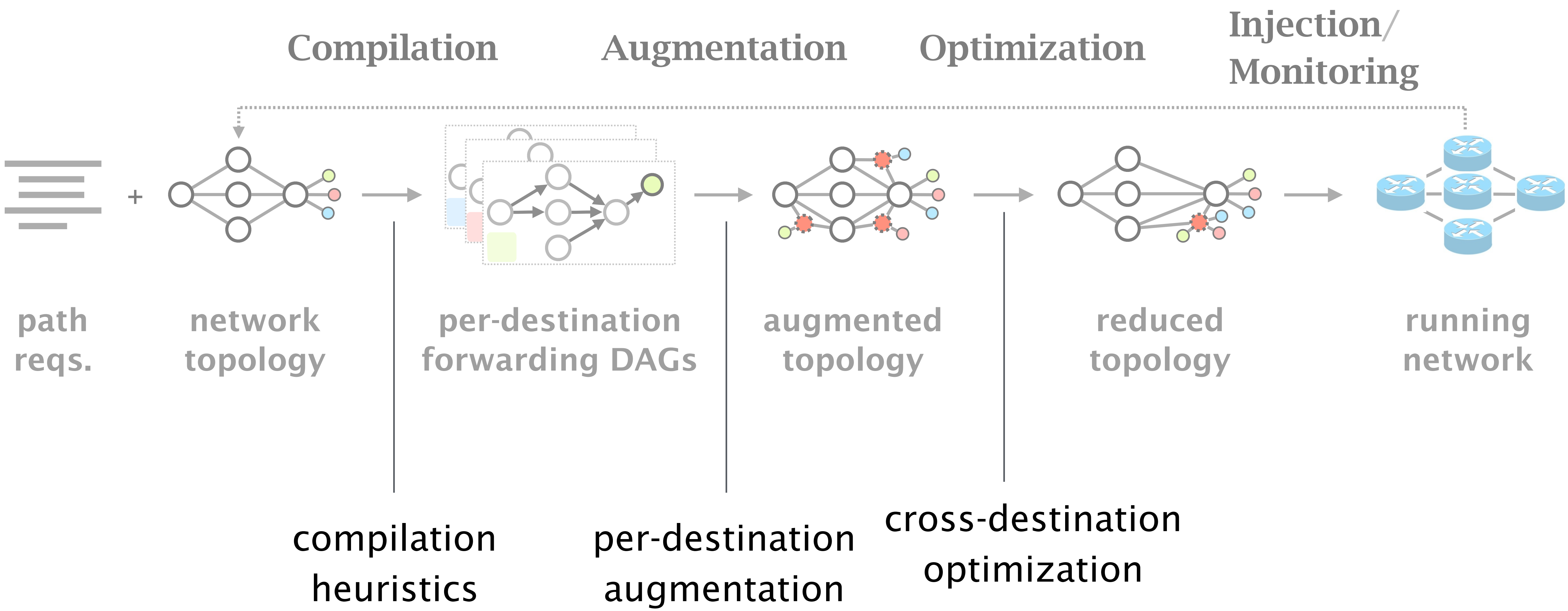
3 Scalability

4 Robustness

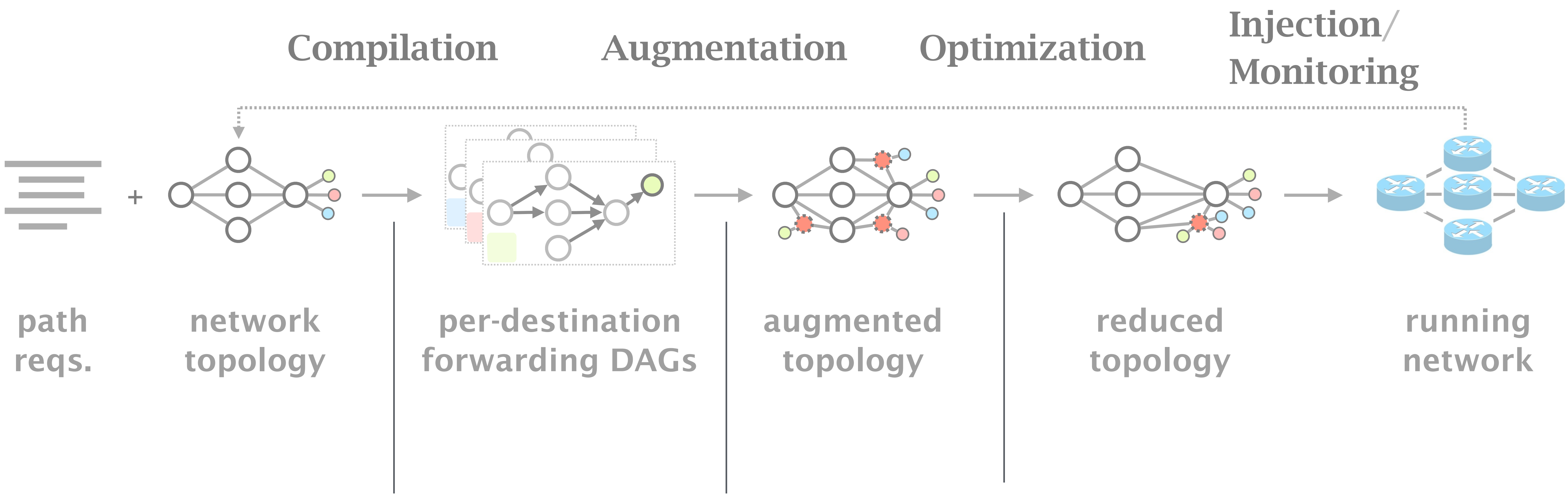
# We implemented a Fibbing controller



We also propose algorithms to compute augmented topologies of limited size



For our Fibbing controller, we propose algorithms to be run in sequence



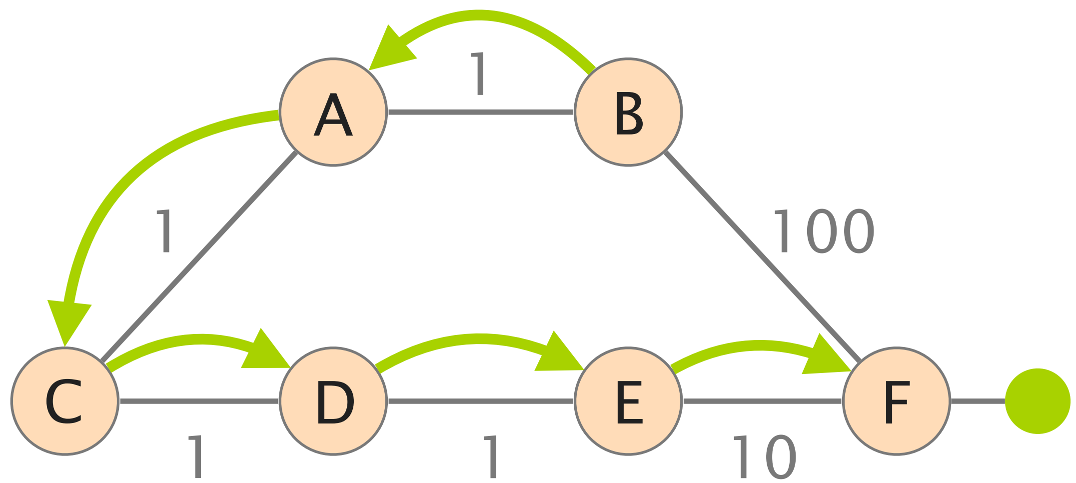

compilation heuristics

per-destination augmentation

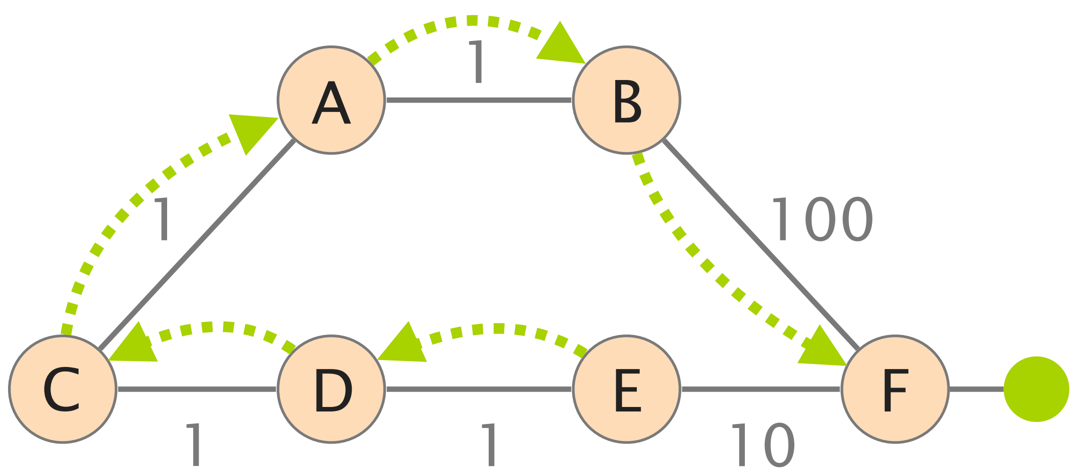
1. simple
2. merger

cross-destination optimization

Consider the following example,
with a drastic forwarding path change

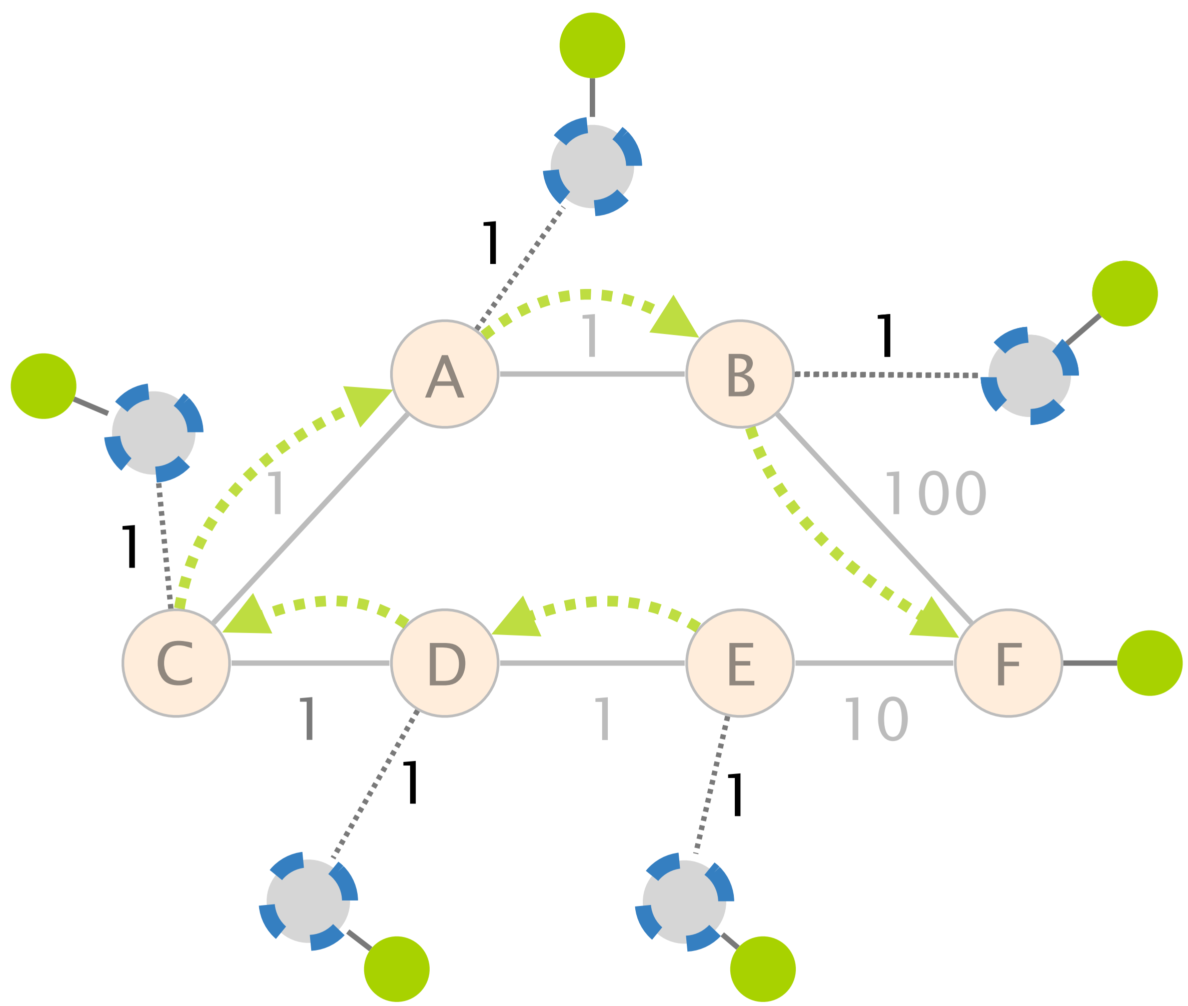



original shortest-path
“down and to the right”

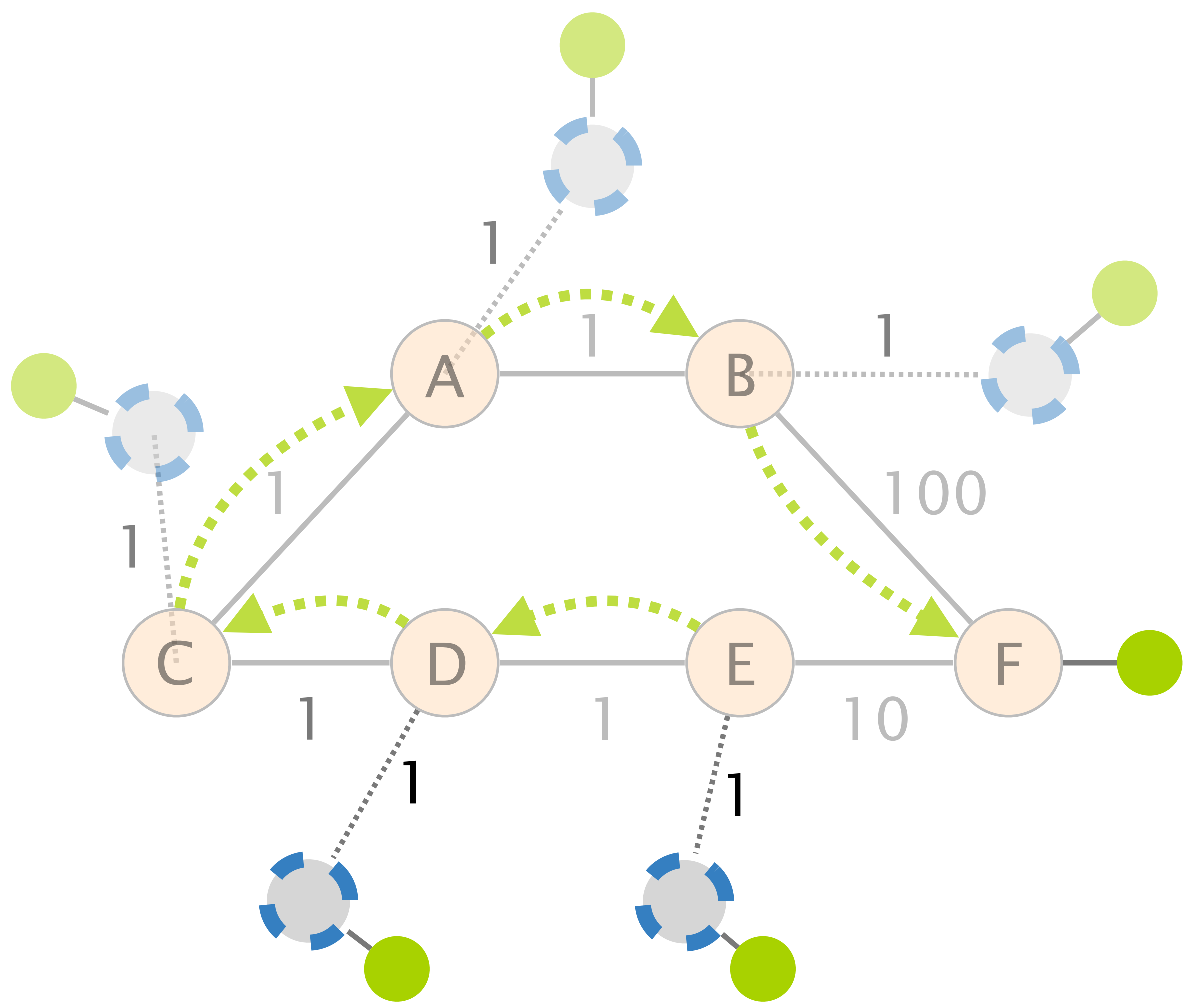



desired shortest-path
“up and to the right”

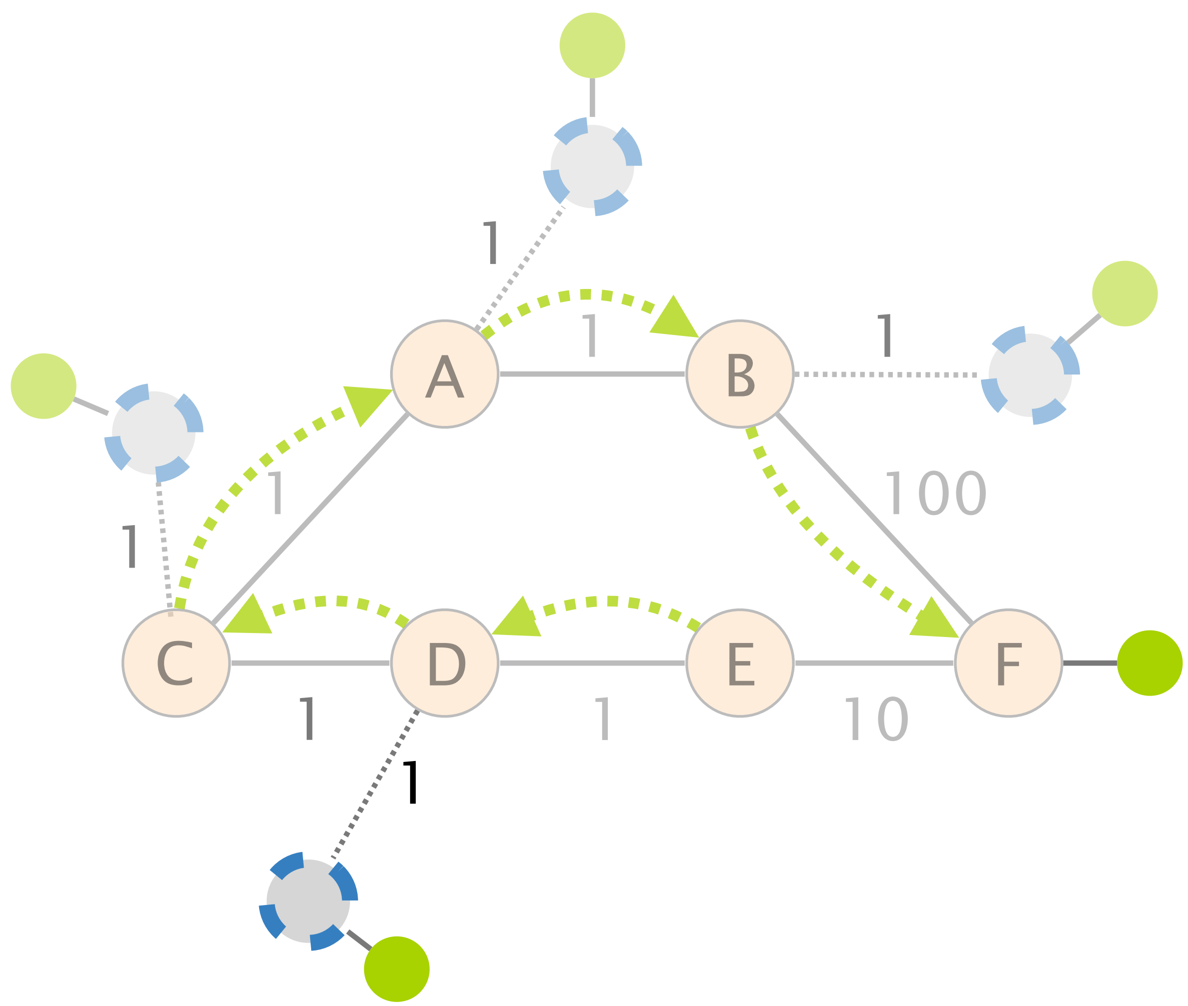
# Simple adds one fake node for every router that has to change next-hop



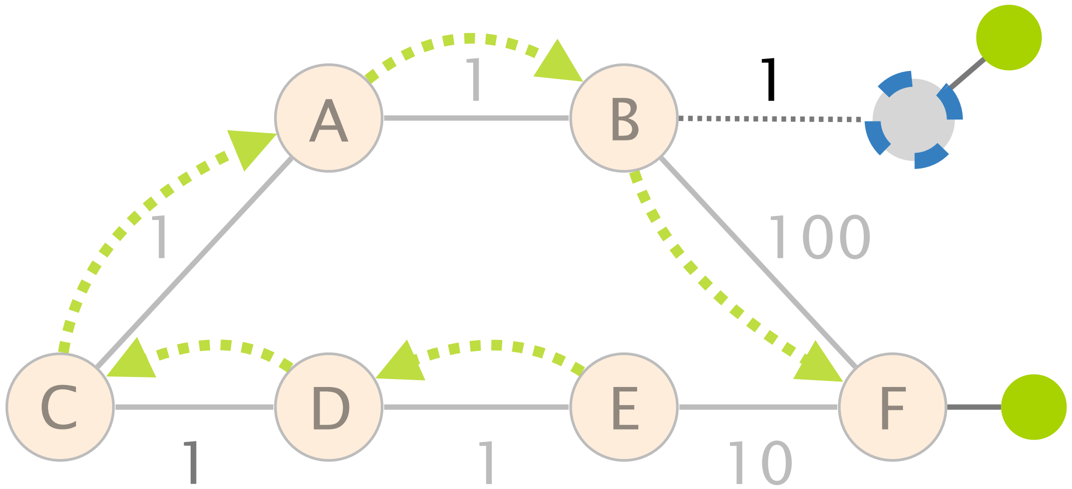
# Merger iteratively merges fake nodes (starting from Simple's output)



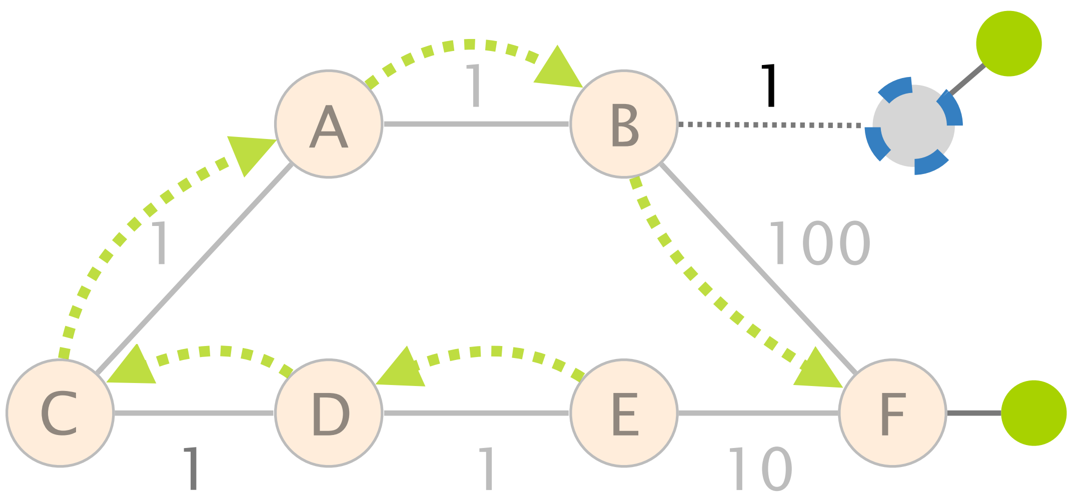
# Merger iteratively merges fake nodes (starting from Simple's output)



This way, Merger programs multiple next-hop changes with a single fake node



This way, Merger programs multiple next-hop changes with a single fake node




Previous SDN solutions (e.g., RCP) cannot do the same

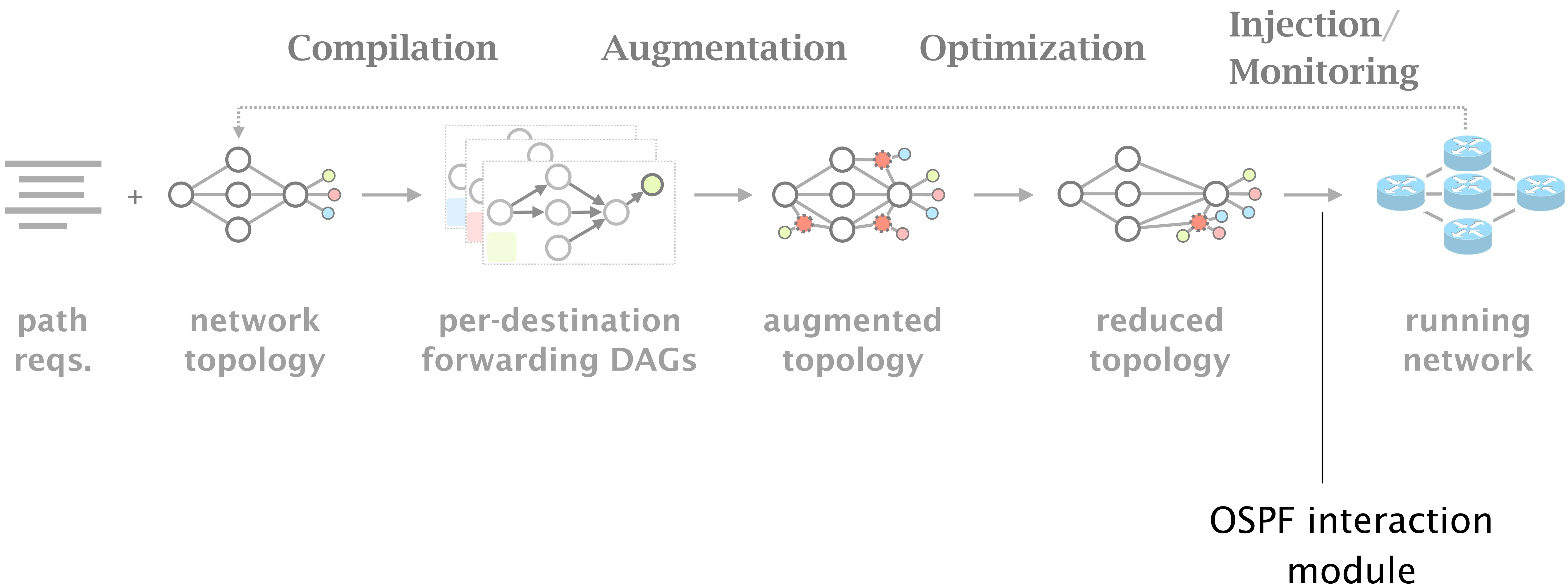
# Simple and Merger achieve different trade-offs in terms of time and optimization efficiency

We ran experiments on Rocketfuel topologies, with at least 25% of nodes changing next-hops

- Simple runs in milliseconds
  Merger takes 0.1 seconds

- Merger reduces fake nodes by up to 50%
  and up to 90% with cross-destination optimization

# We implemented the machinery to listen to OSPF and augment the topology



Experiments on real routers show that Fibbing has very limited impact on routers

| # fake nodes | router memory (MB) | |
|---|---|---|
| 1 000 | 0.7 | |
| 5 000 | 6.8 | |
| 10 000 | 14.5 | |
| 50 000 | 76.0 | |
| 100 000 | 153 | DRAM is cheap |

>> # real routers

# Experiments on real routers show that Fibbing has very limited impact on routers

| # fake nodes | router memory (MB) | |
|---|---|---|
| 1 000 | 0.7 | |
| 5 000 | 6.8 | |
| 10 000 | 14.5 | |
| 50 000 | 76.0 | |
| 100 000 | 153 | DRAM is cheap |

CPU utilization always under 4%

# Experiments on real routers show that Fibbing does not impact IGP convergence

Upon link failure, we registered *no difference* in the (sub-second) IGP convergence with

- no fake nodes
- up to 100,000 fake nodes and destinations

Experiments on real routers show that Fibbing achieves fast forwarding changes

| # fake nodes | installation time (seconds) | |
|---|---|---|
| 1 000 | 0.9 | |
| 5 000 | 4.5 | |
| 10 000 | 8.9 | |
| 50 000 | 44.7 | |
| 100 000 | 89.50 | 894.50 μs/entry |

# Central Control over Distributed Routing

fibbing.net


1 Manageability

2 Flexibility

3 Scalability

4 Robustness

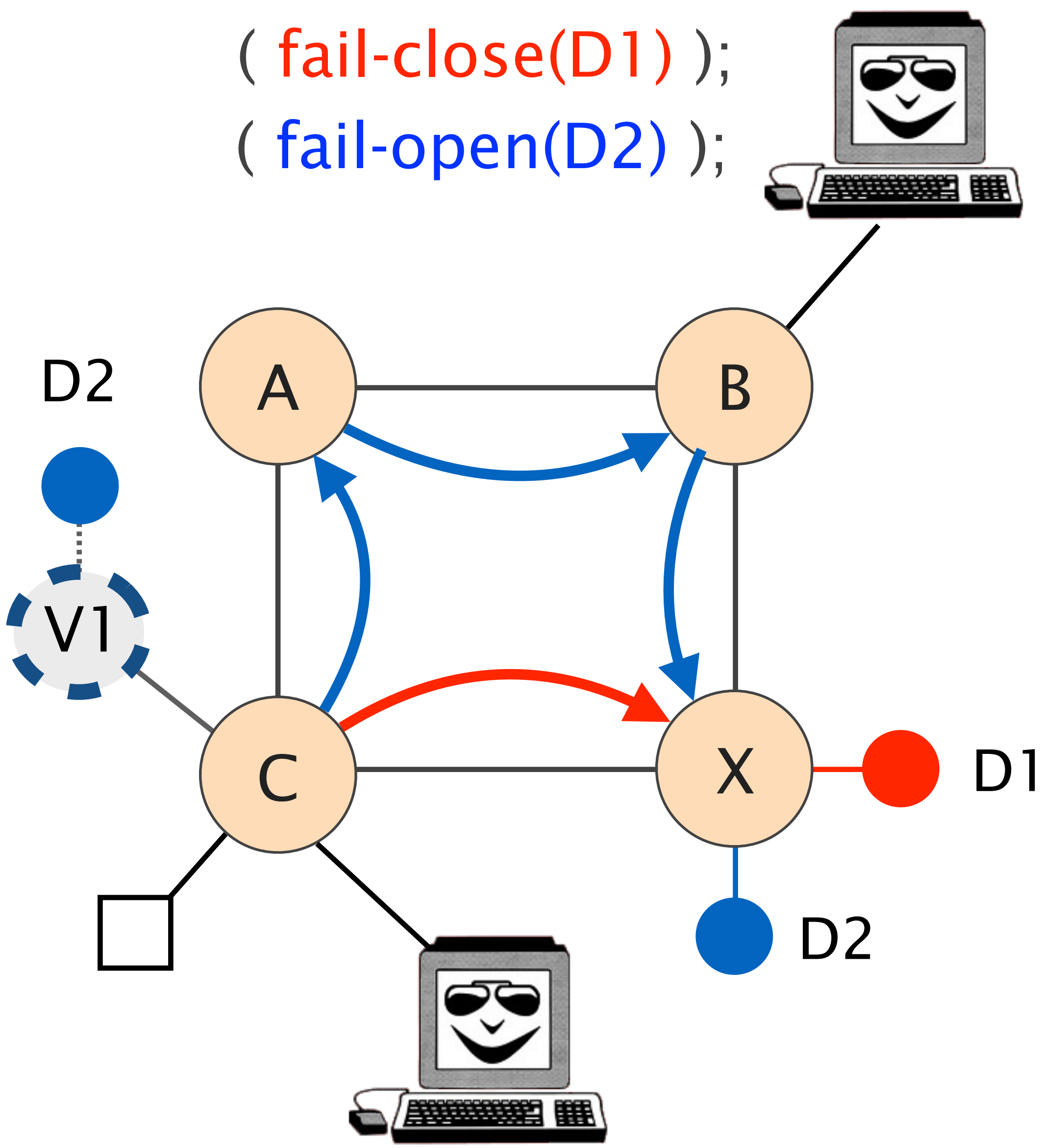
# Fibbing improves robustness by relying on the underlying IGP

- no controller action needed in some cases
  IGPs re-converge quickly [Filsfils07]

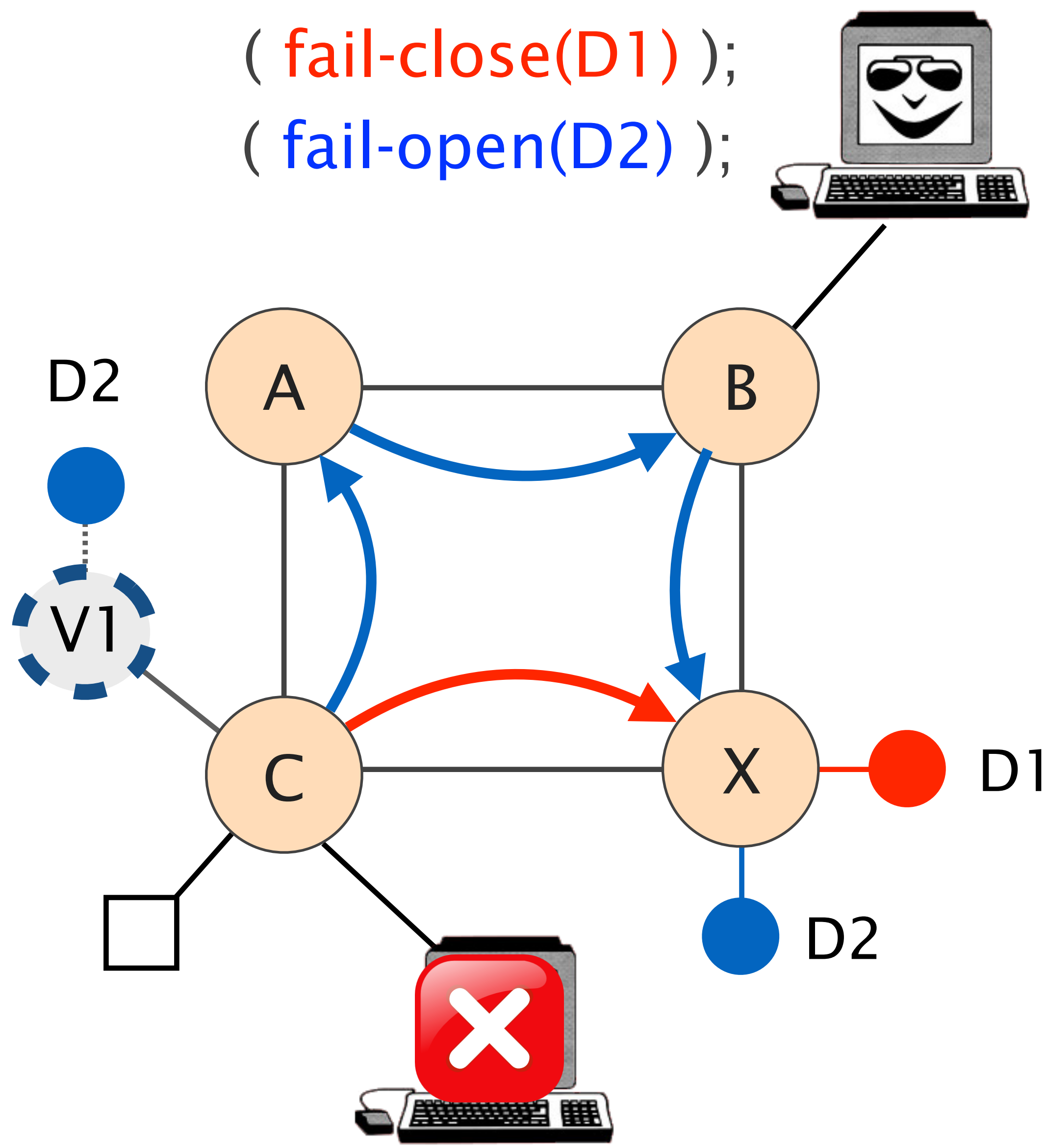
- IGP provides fast failure detection and control-plane sync
  thanks to its shared topology

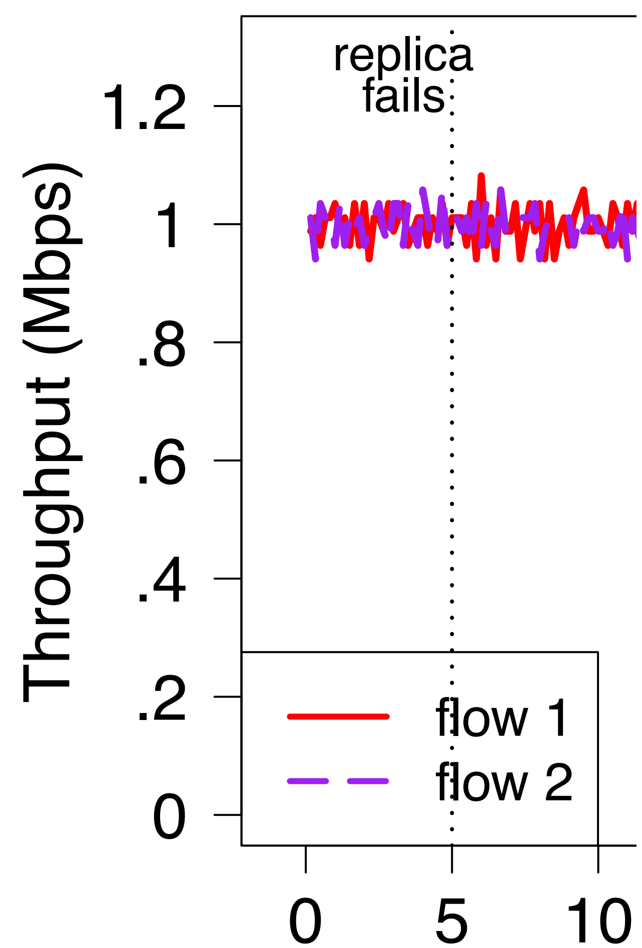
- Fibbing supports fail-open and fail-close semantics
  see paper

We ran a failure recovery case study,
with distributed Fibbing controller

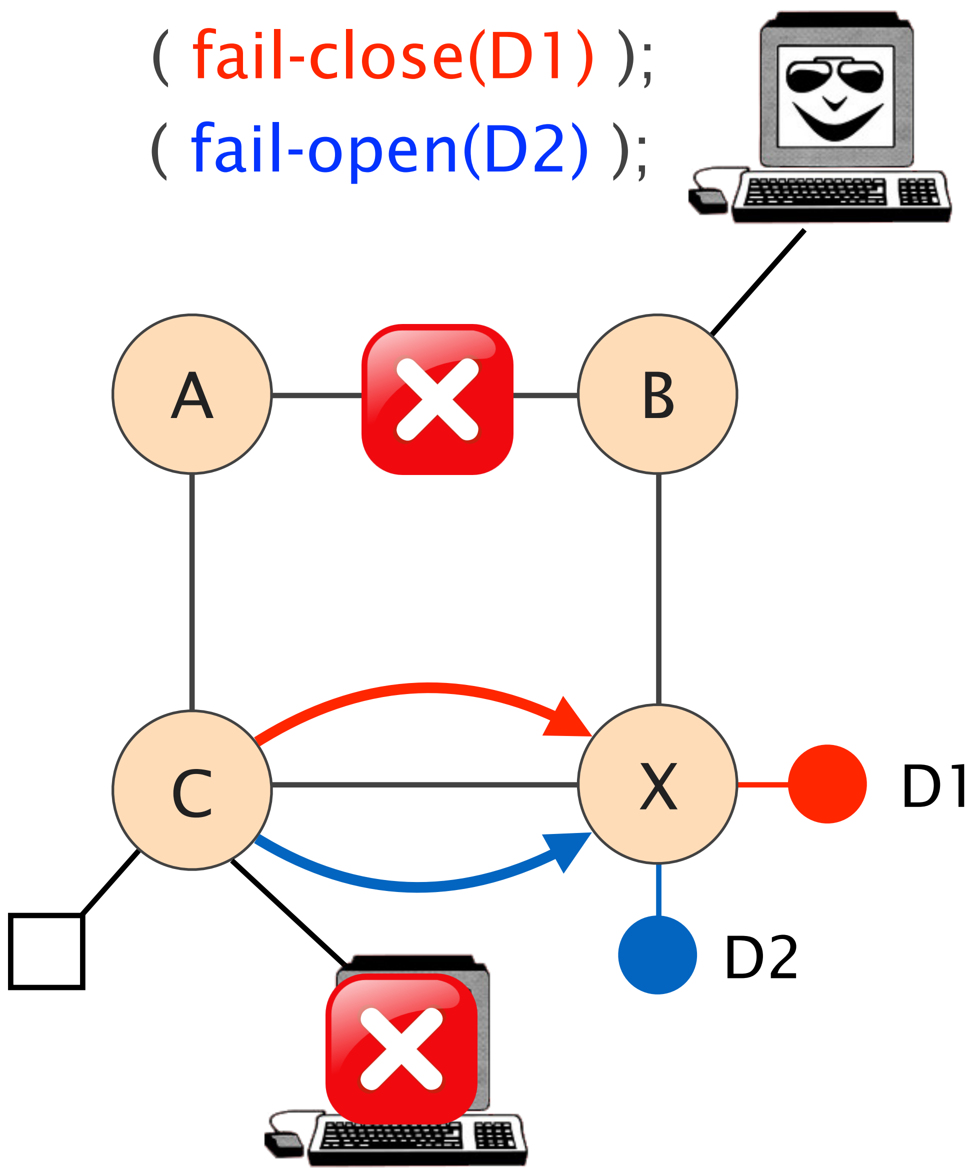


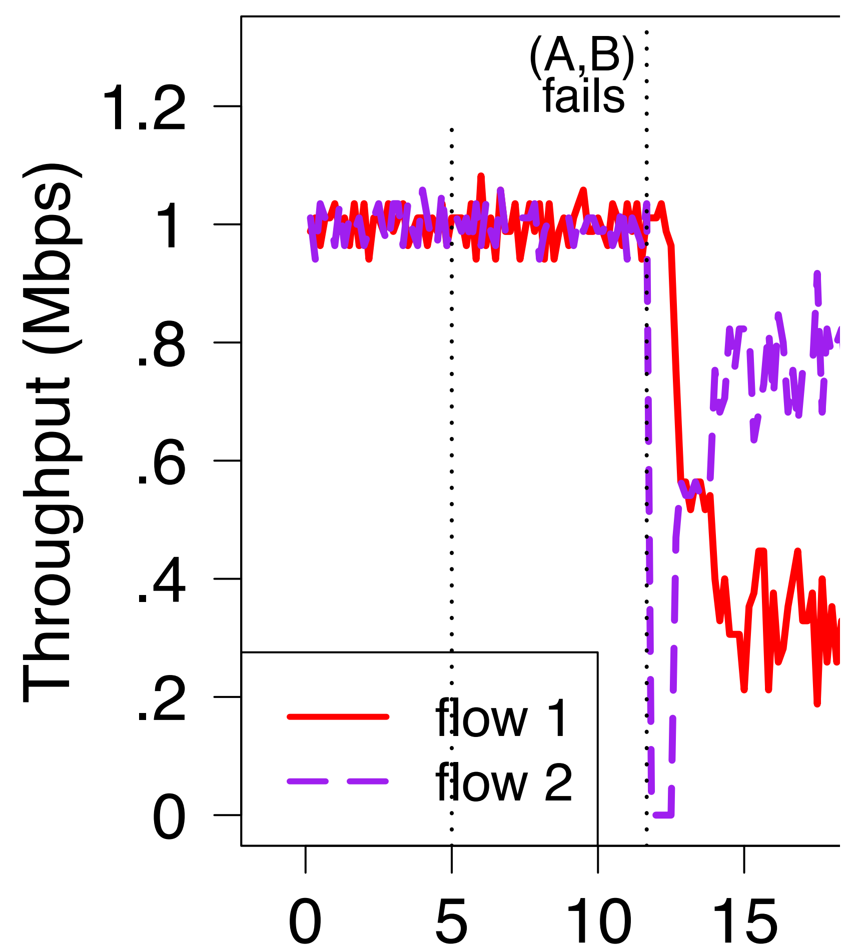
# Fibbing survives replica failures with no impact on forwarding



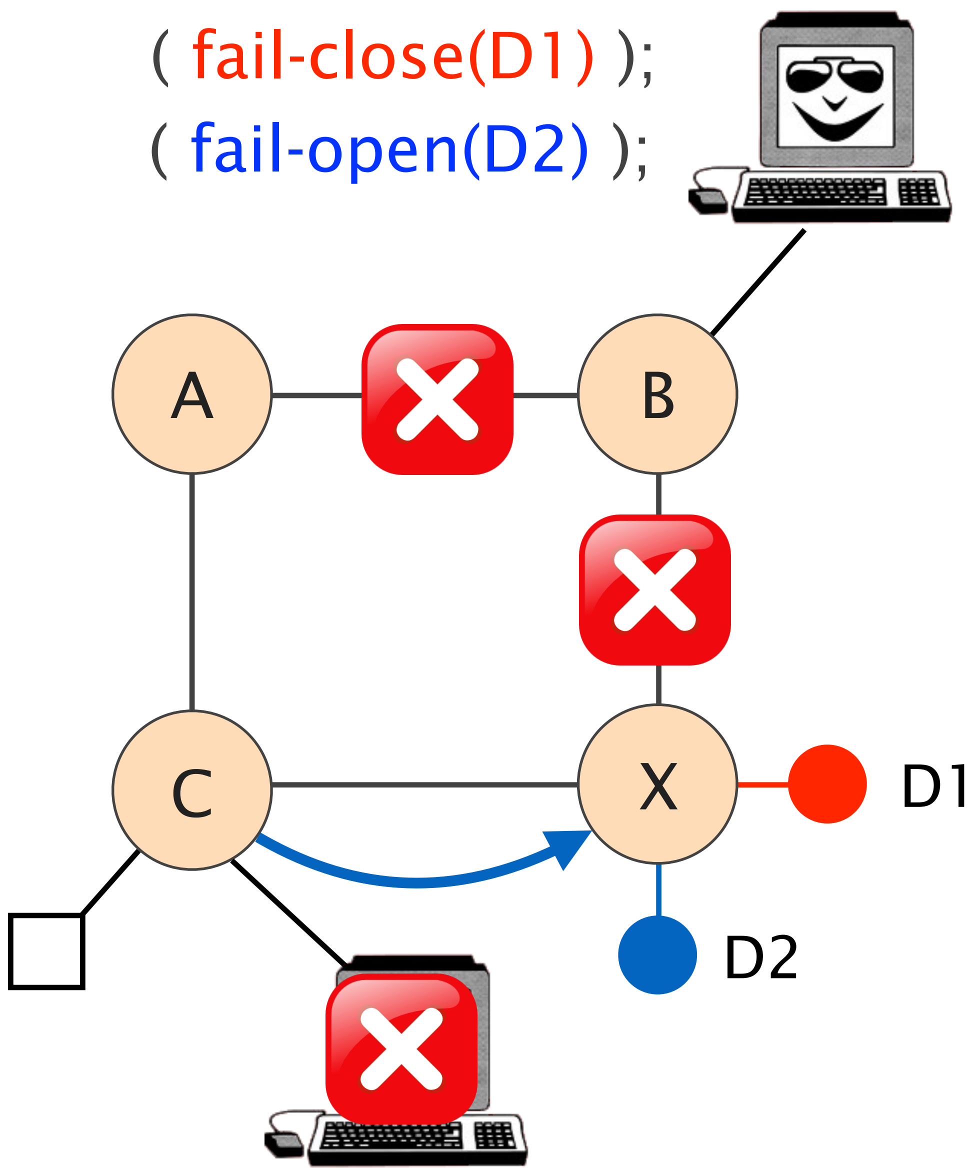



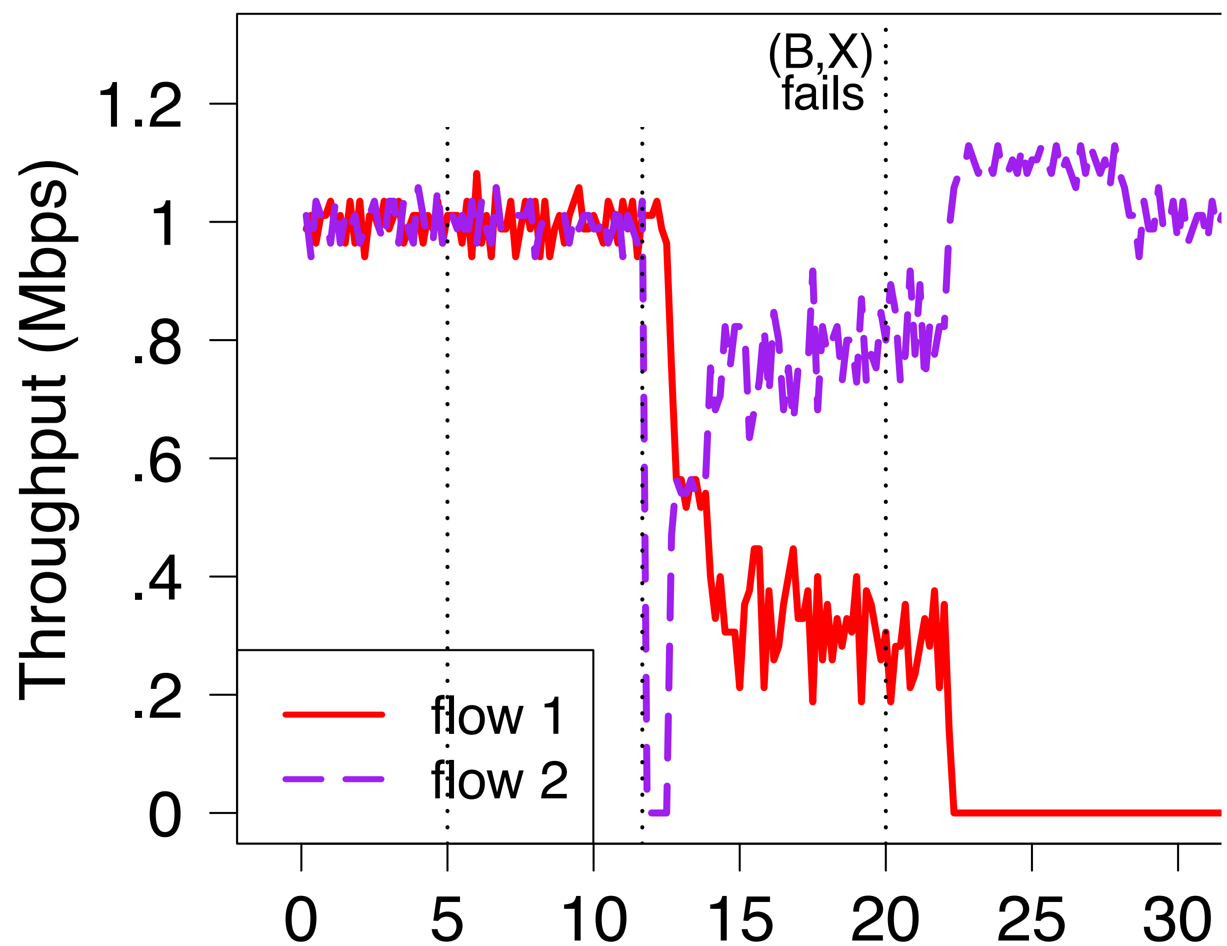
# Fibbing reacts to network failures quickly re-optimizing forwarding



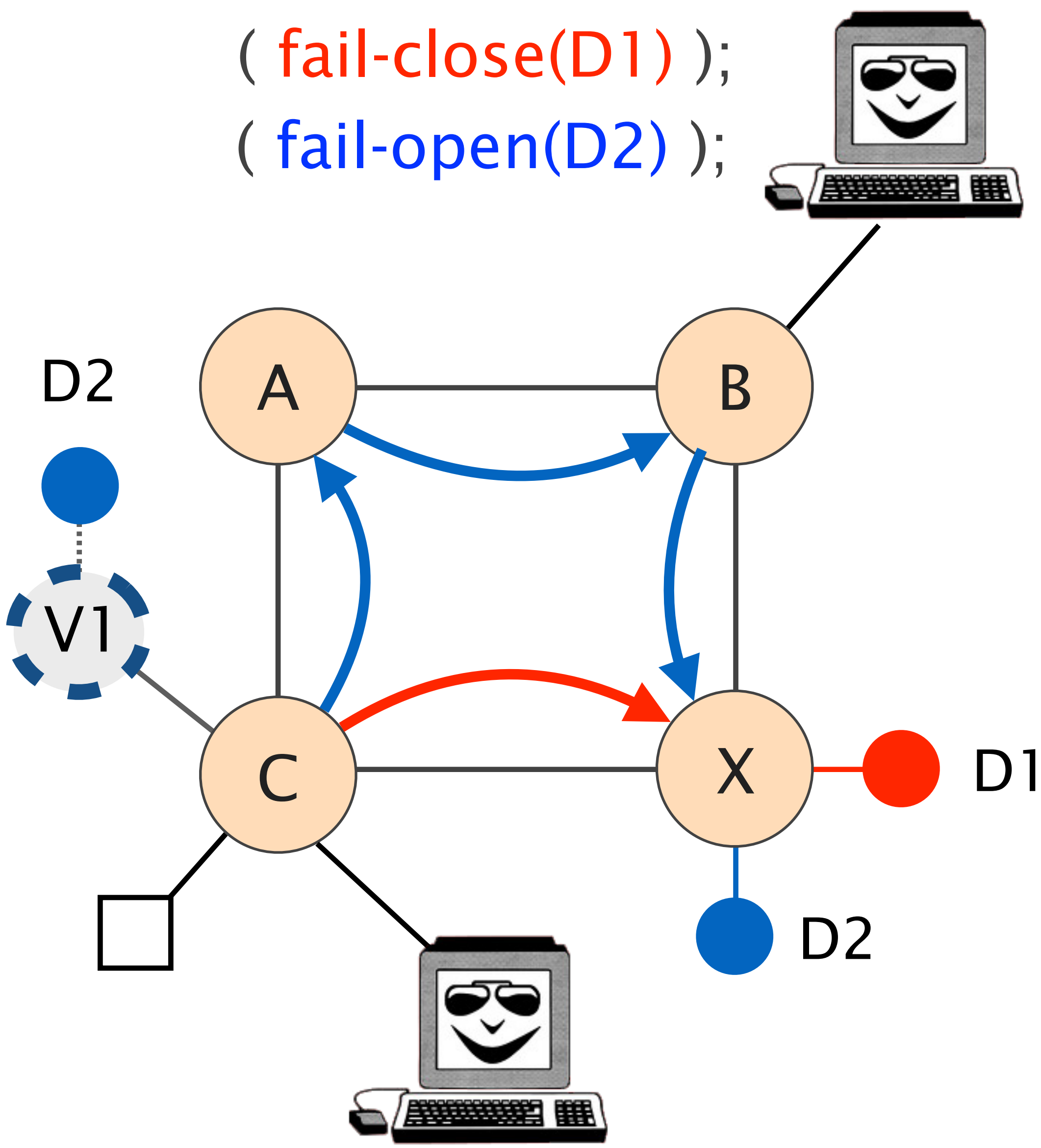



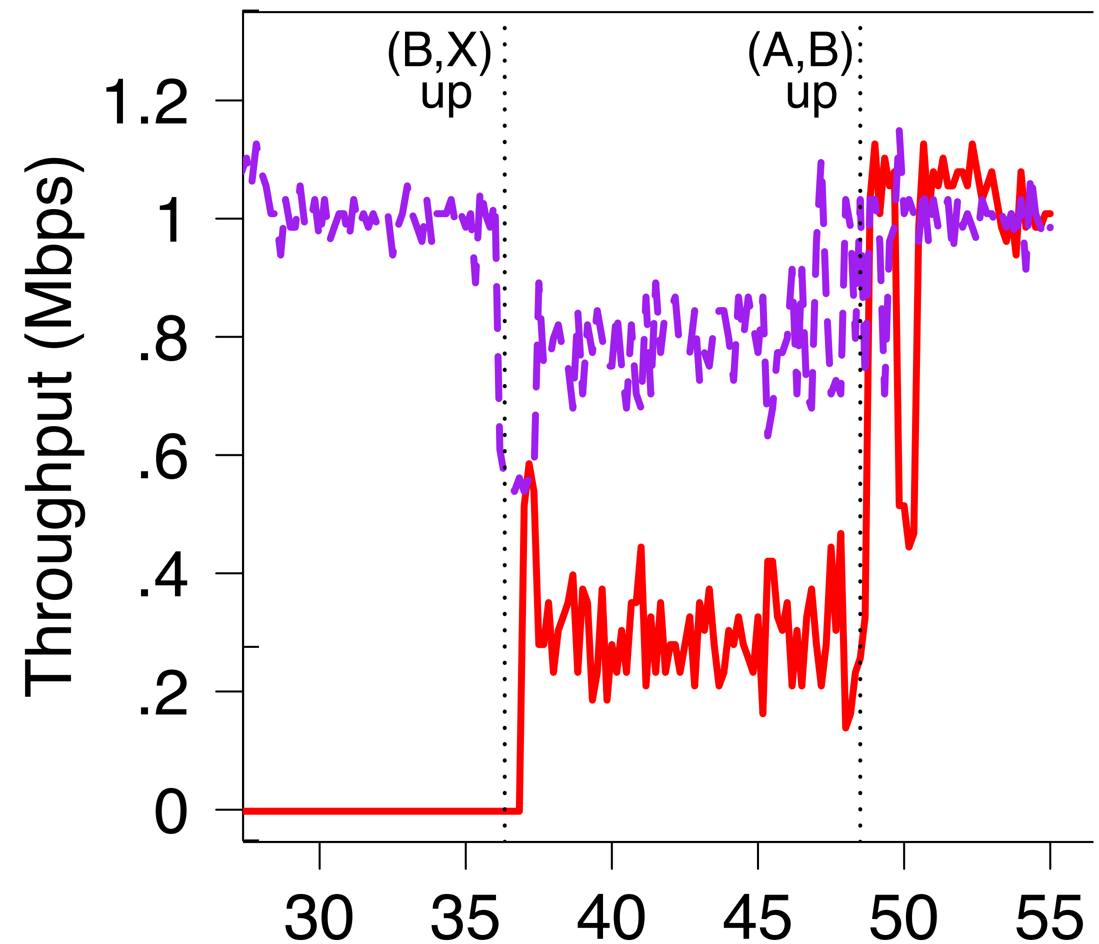
Fibbing reacts to partitions,
respecting fail-close and fail-open semantics



# Fibbing recovers correctly
# (as soon as failures are fixed)



# Fibbing shows the *benefits* of central control over distributed protocols

- Realizes SDN management model
  network-wide automated control

- Simplify controllers and improves robustness
  heavy work is still done by routers

- Works today, on existing networks
  avoids SDN deployment hurdles

# Central Control over Distributed Routing

fibbing.net


*Tell me lies, tell me sweet little lies*

— Fleetwood Mac

Stefano Vissicchio

stefano.vissicchio@uclouvain.be