IGP-as-a-Backup for Robust Software-Defined Networks

Olivier Tilmans Stefano Vissicchio

Université Catholique de Louvain
ICTEAM / INGI / IP Networking Lab
✉ name.surname@uclouvain.be

CNSM2014 – Nov. 20 2014

“Use the right tool for the right job”

Distributed network protocols and SDN

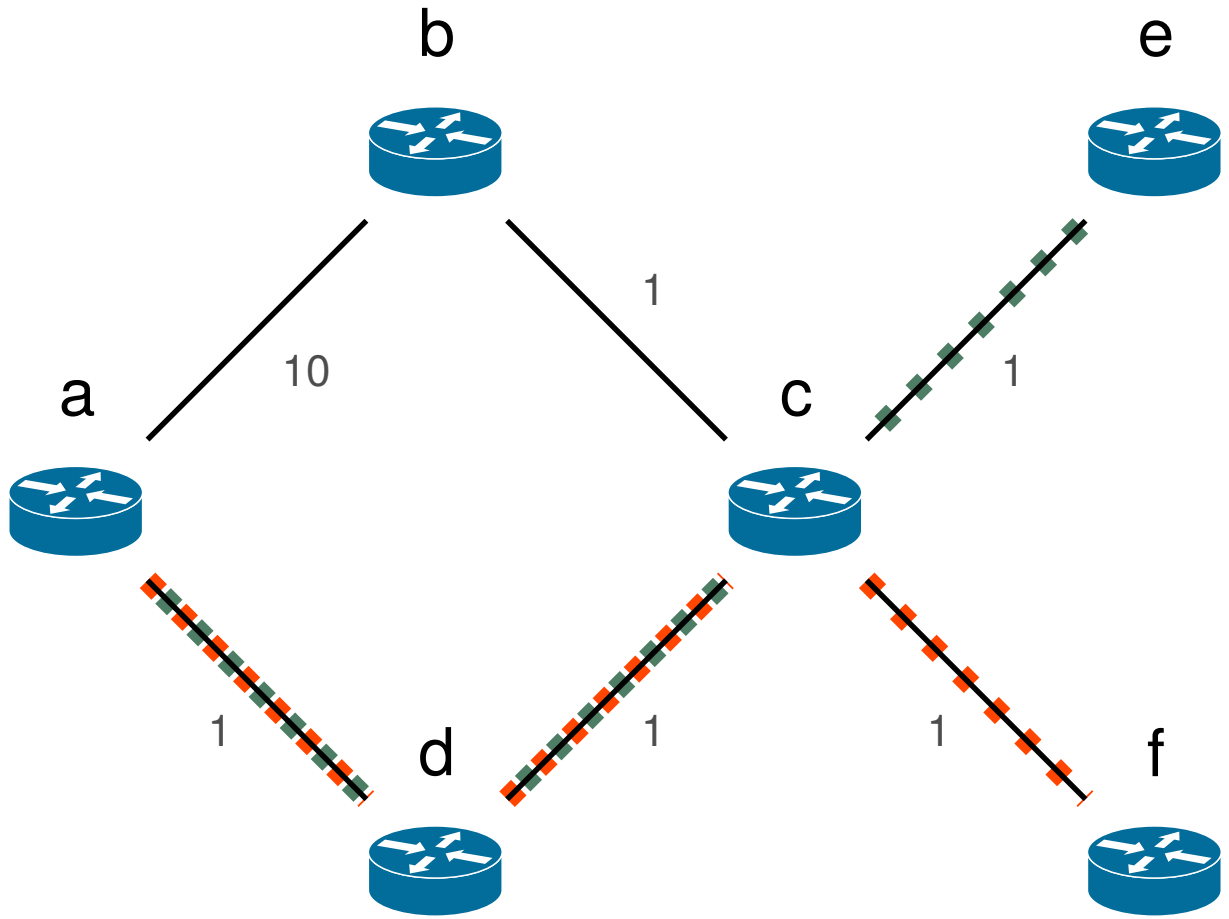
IGPs are distributed by nature

Flooding of reachability information

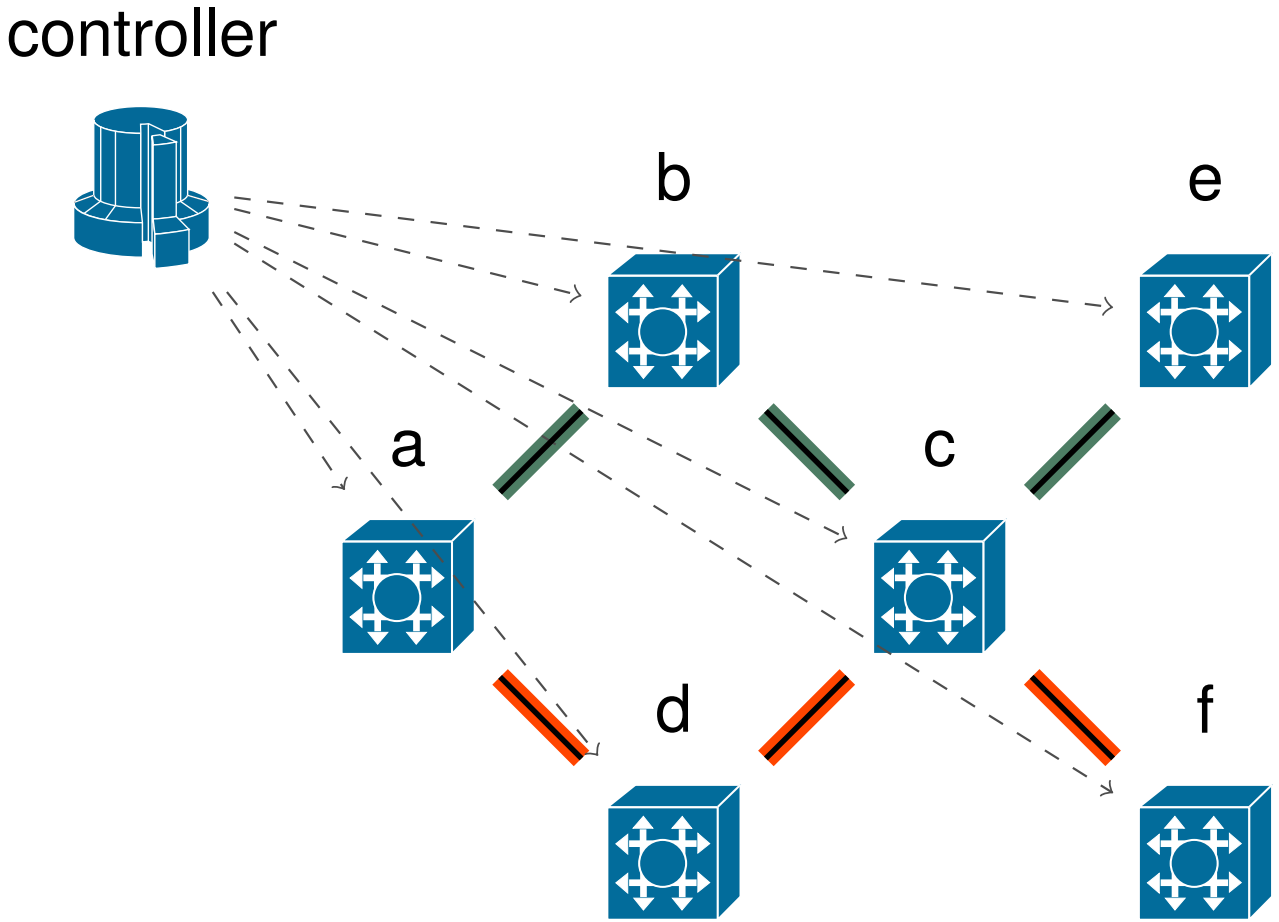
Each node infers the current map of the topology

Global shortest-path routing is achieved

Nodes only forward packets according to the overall shortest-path



SDN technologies are more expressive



Openflow enables for arbitrary behaviors

Match	**Action**
tcp, dst.port=2	output=3
in_port=2, ip_proto=89	drop
tcp, src.port=1234	rewrite:src.port=4567, output=1

Explicit control over the paths on a per-device basis

Handling failures

Because bad things **will** happen

Recovery in SDN is hard as switches are not autonomous

The controller is a new type of failures

Recovery in SDN is hard as switches are not autonomous

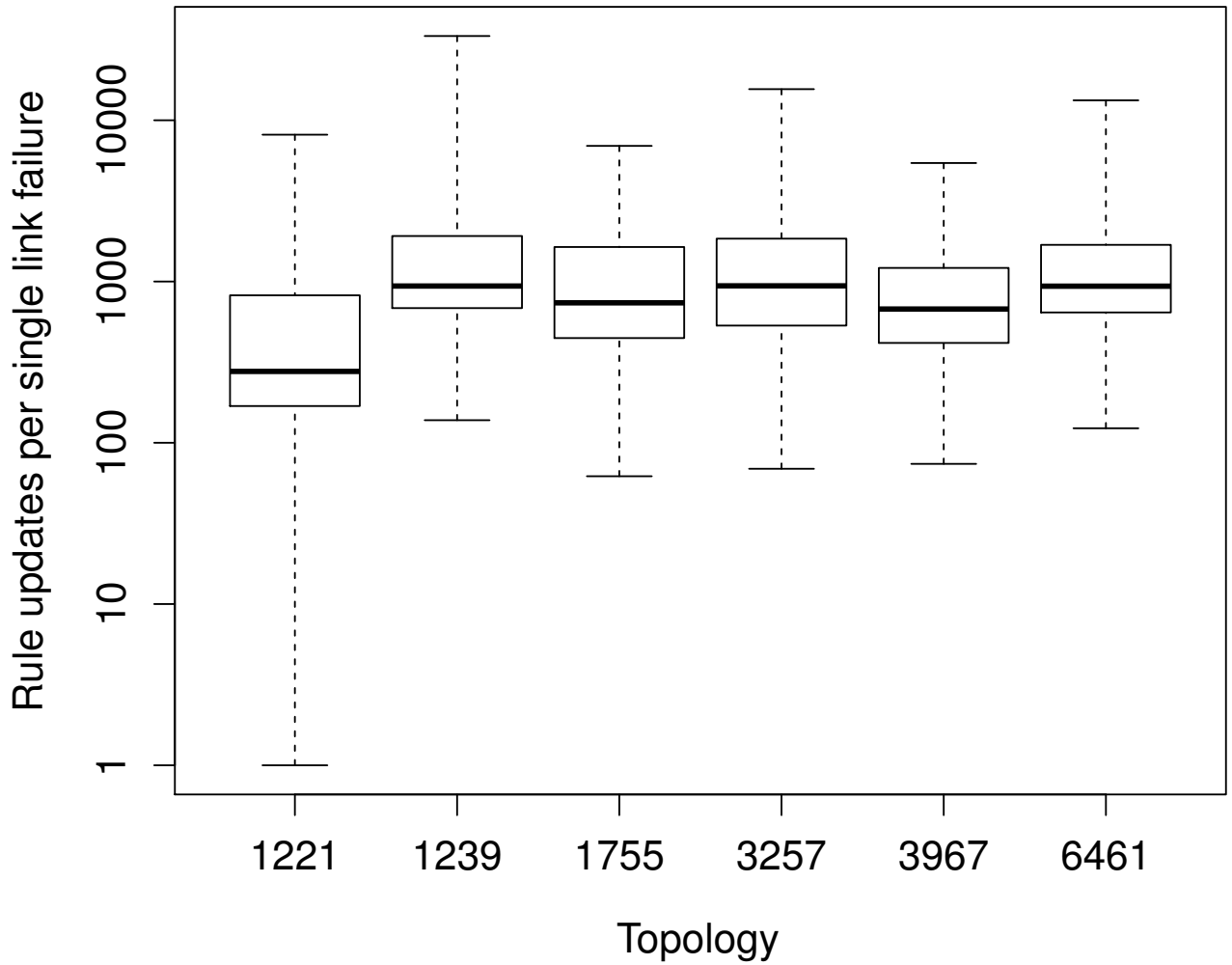
The controller is a new type of failures

Two families of recovery techniques:

Reactive approaches

Switches ask the controller *“What to do?”*

Performance of reactive approaches vary with the network size



Recovery in SDN is hard as switches are not autonomous

The controller is a new type of failures

Two families of recovery techniques:

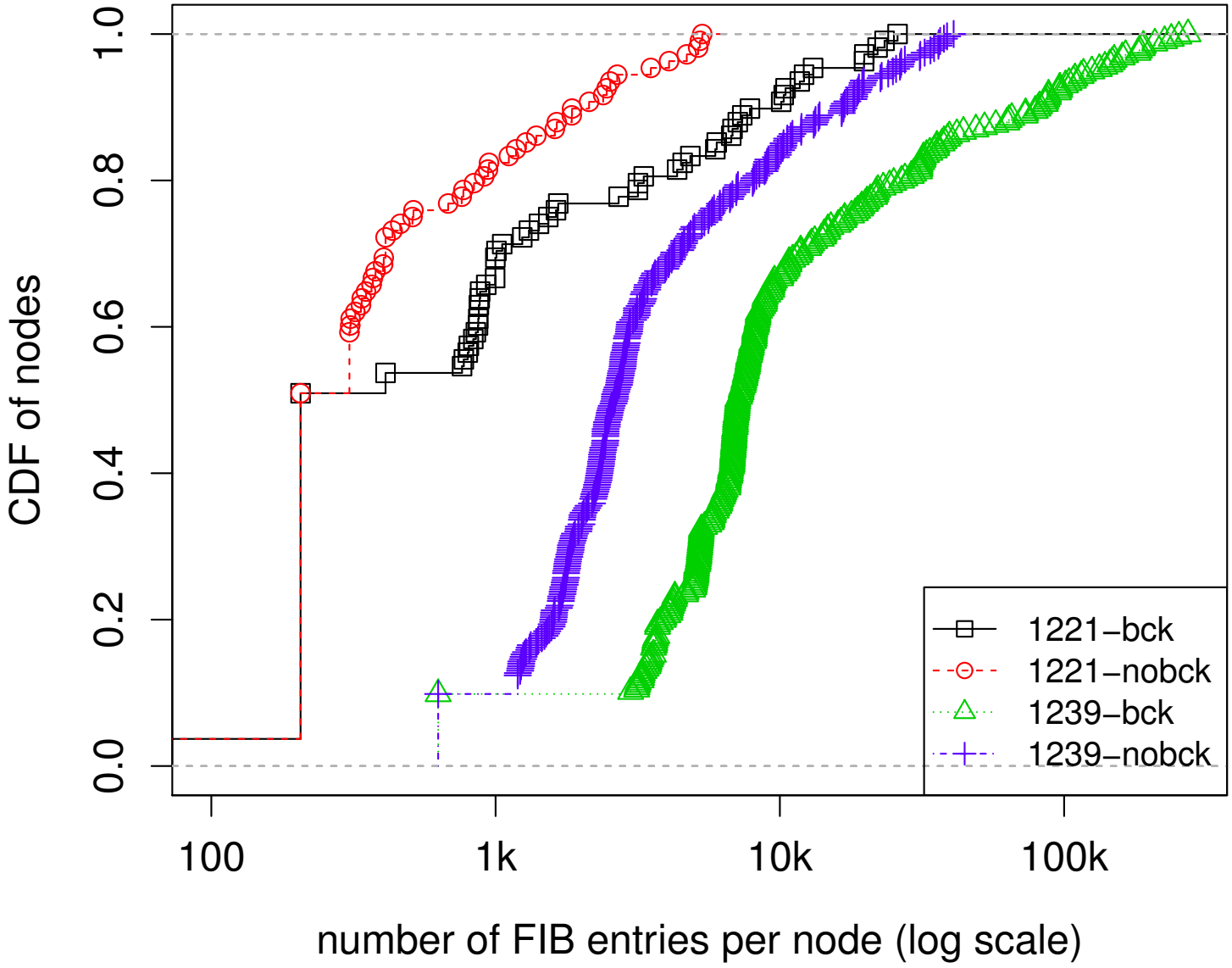
Reactive approaches

Switches ask the controller *"What to do?"*

Proactive approaches

Switches have backup rules
"If X happens do this
If Y happens do that
If Z happens do…"

Proactive approaches come at a price



IGPs are highly resilient by design

Scales well with large networks

Connectivity will be restored

Convergence can be very fast!

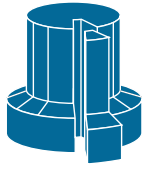
SDN are more expressive but IGPs are more resilient

Why picking only one?

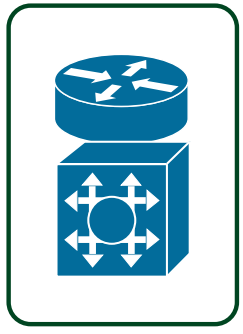
IBSDN Components

Operator

IBSDN Controller



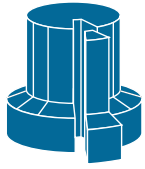
IBSDN Node



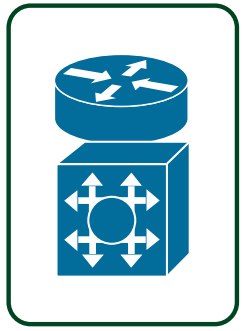
IBSDN Components

Operator

IBSDN Controller



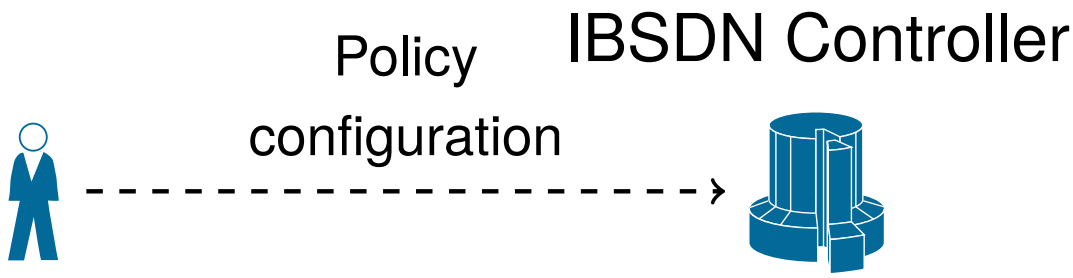
IBSDN Node



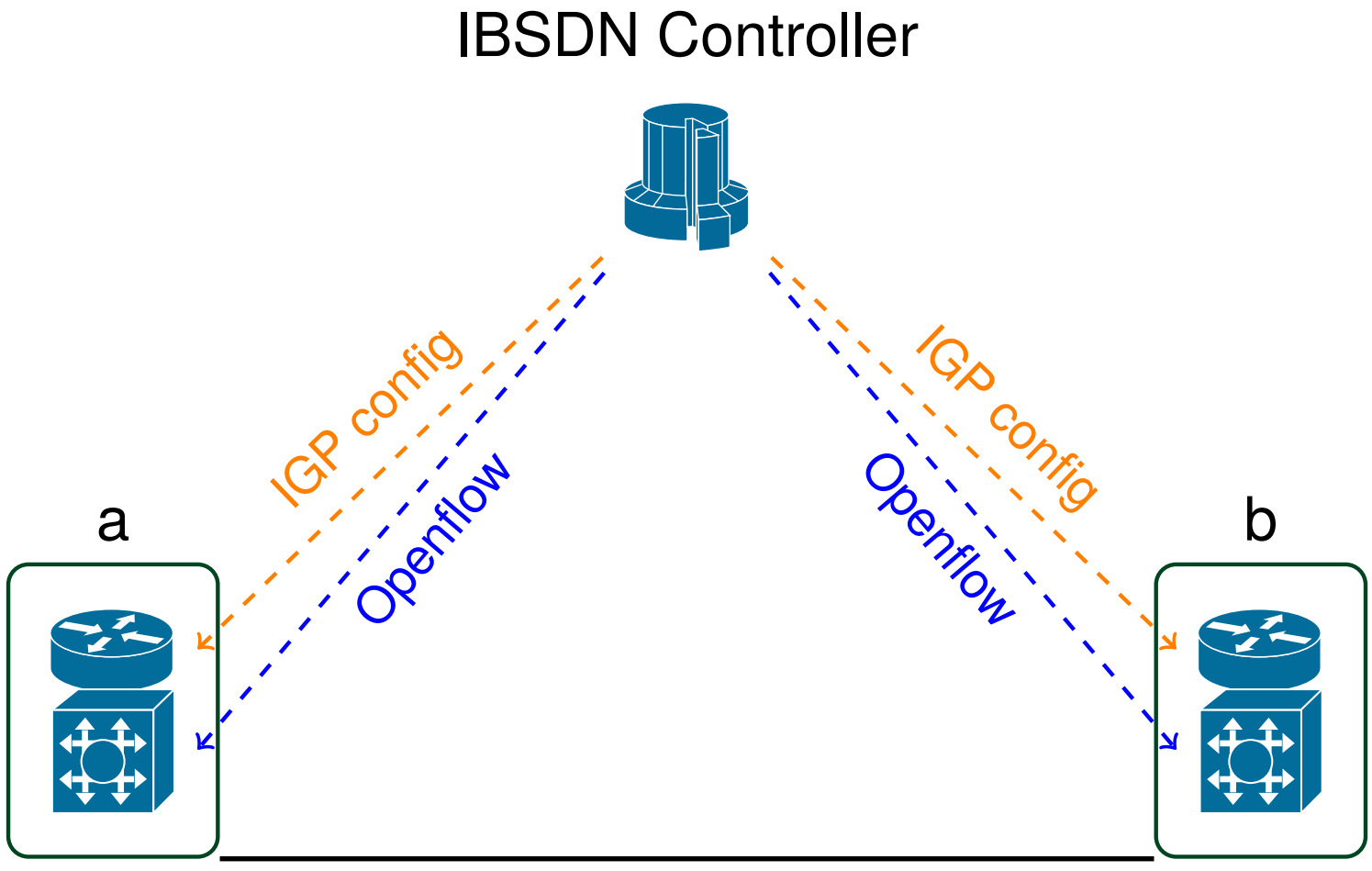
Local Agent

SDN switch

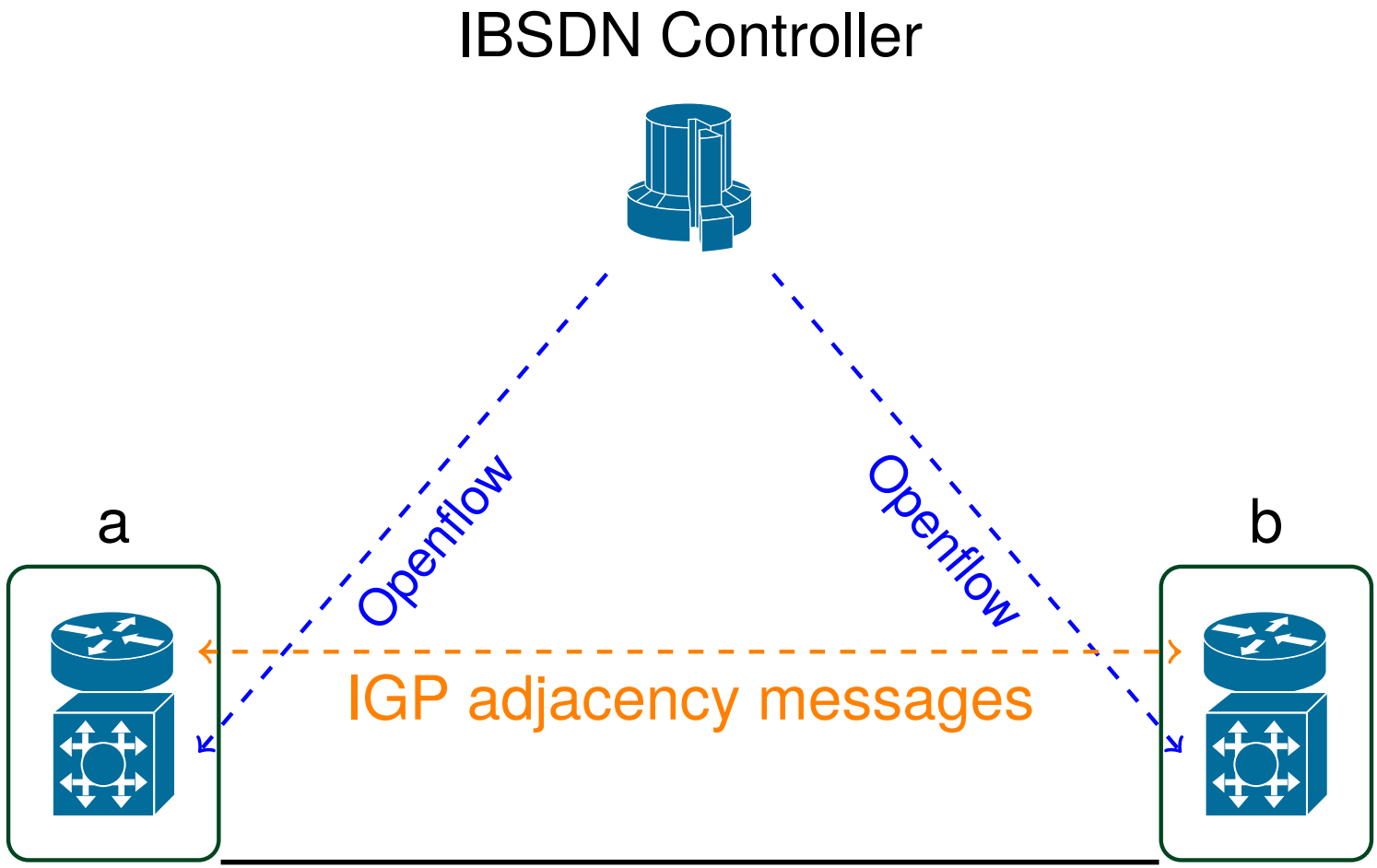
IBSDN is an Hybrid Architecture



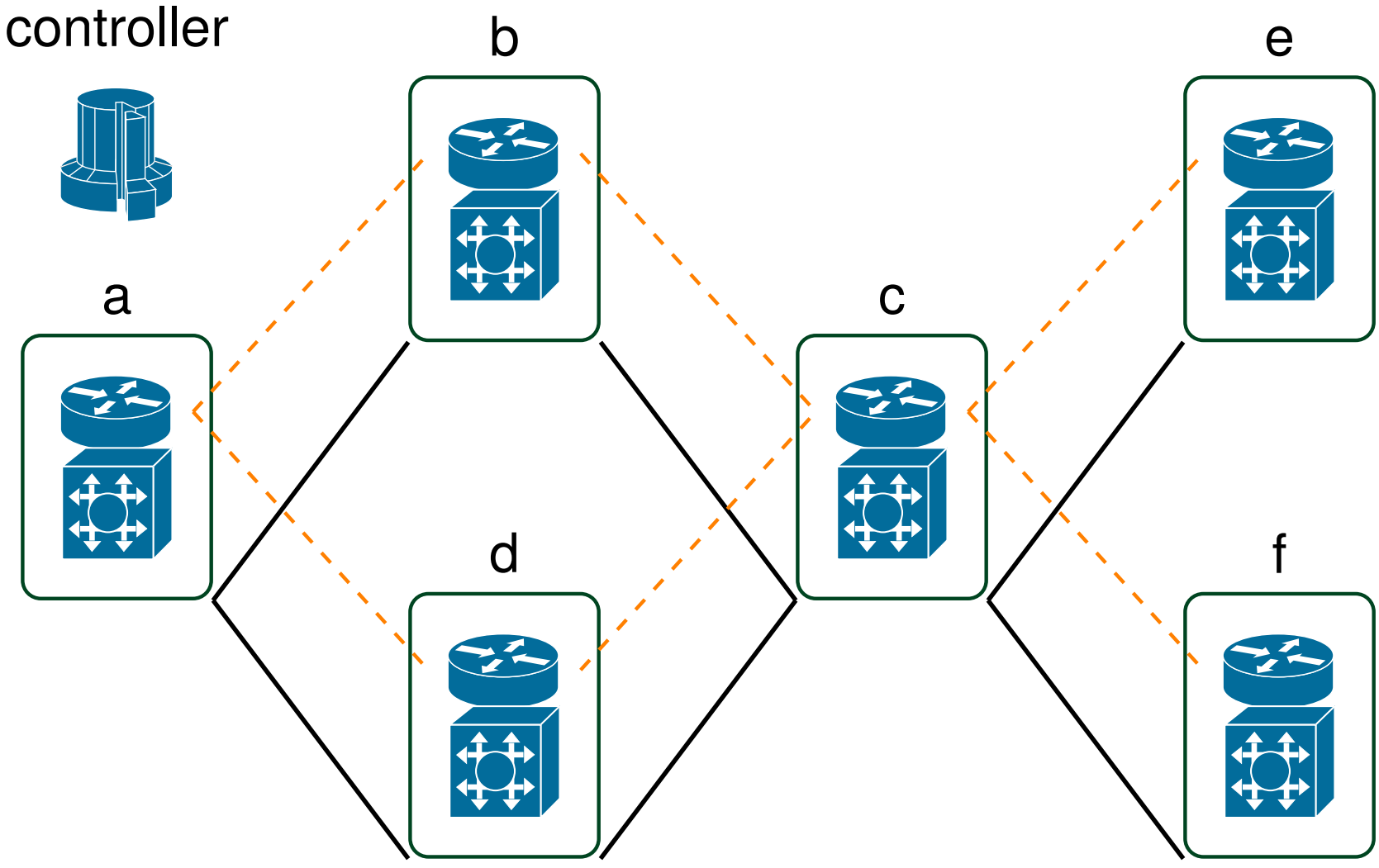
IBSDN is an Hybrid Architecture



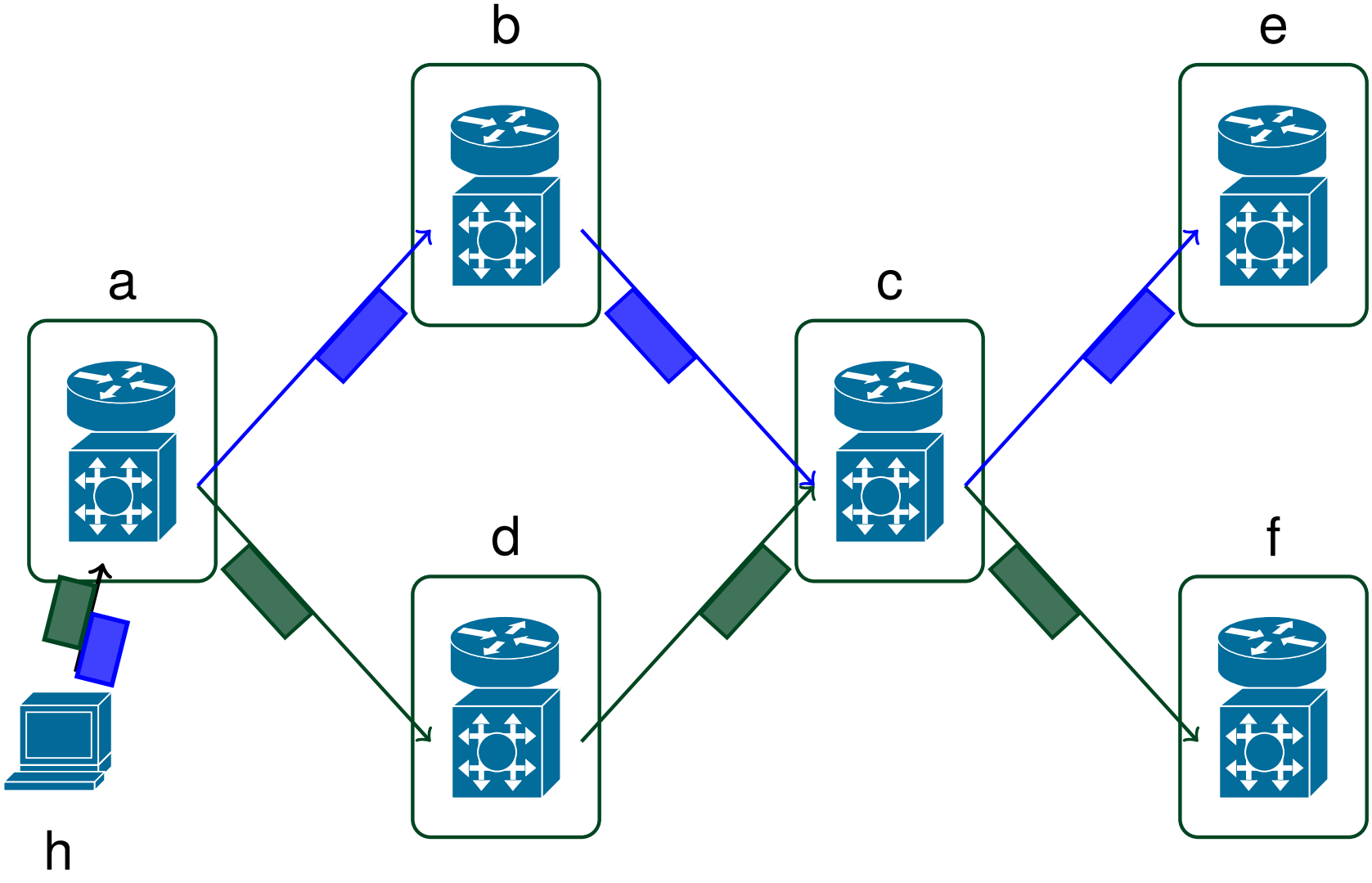
IBSDN is an Hybrid Architecture



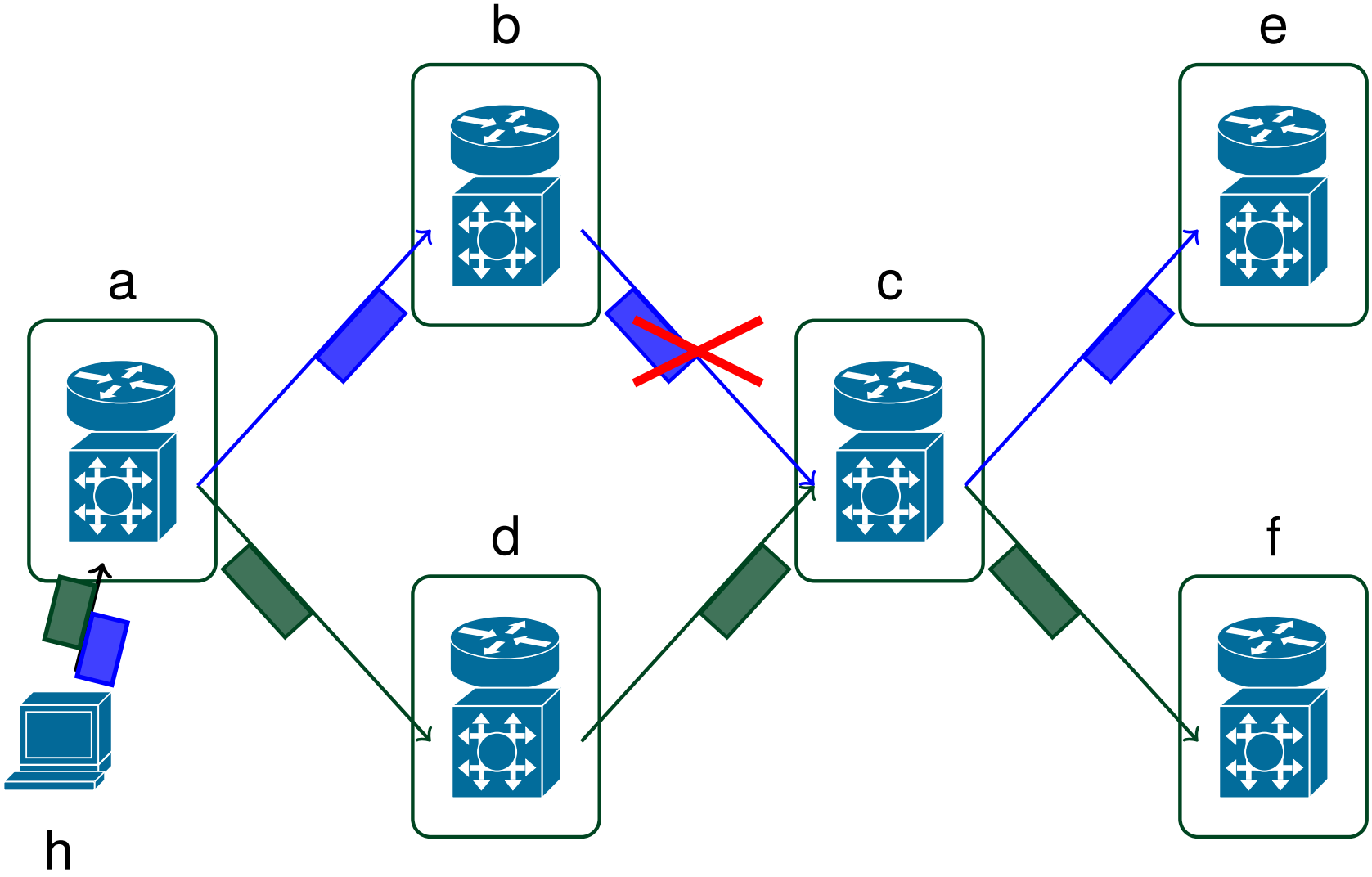
IBSDN offers the same expressiveness than Openflow during normal operation



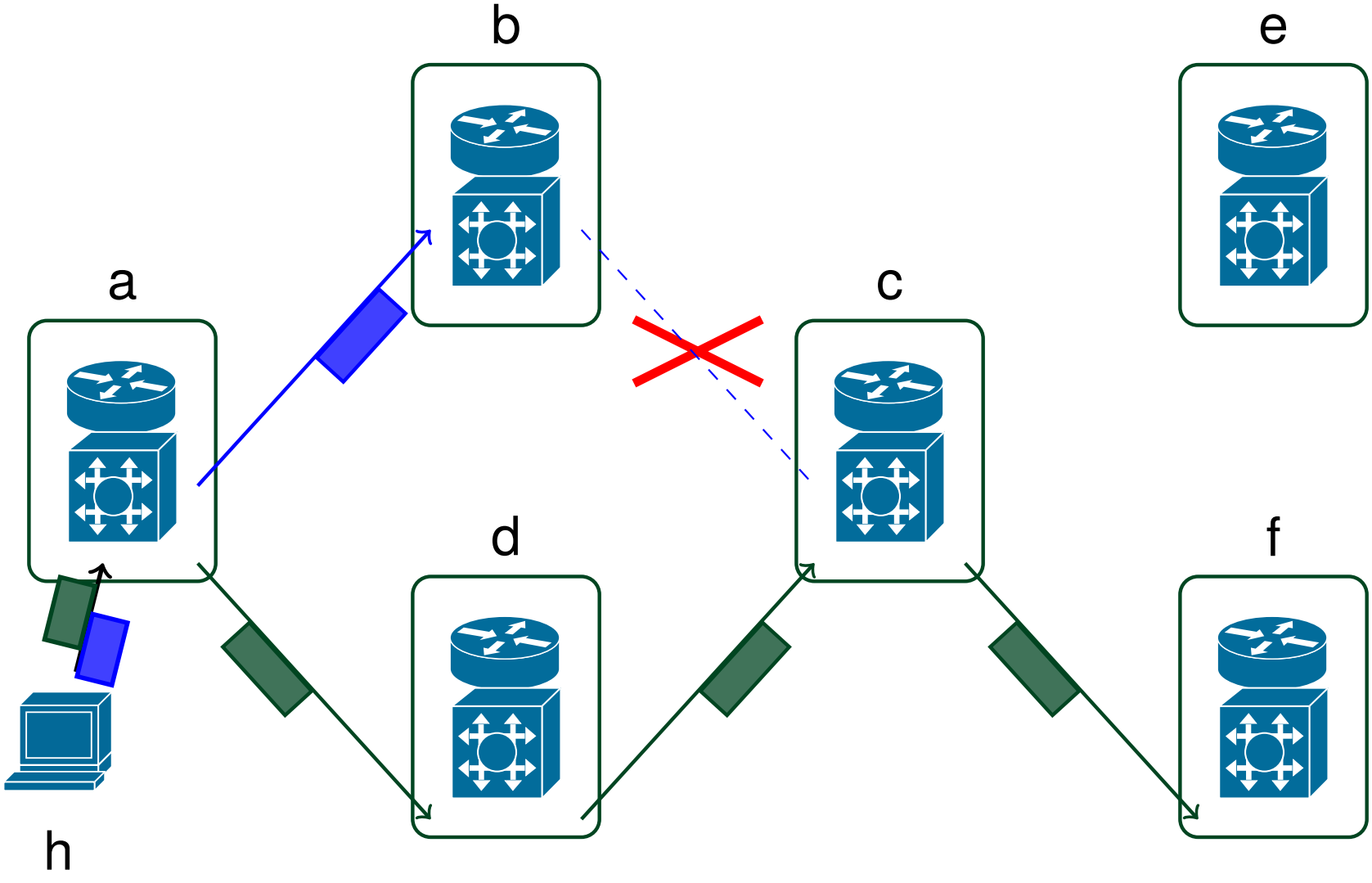
IBSDN offers the same expressiveness than Openflow during normal operation



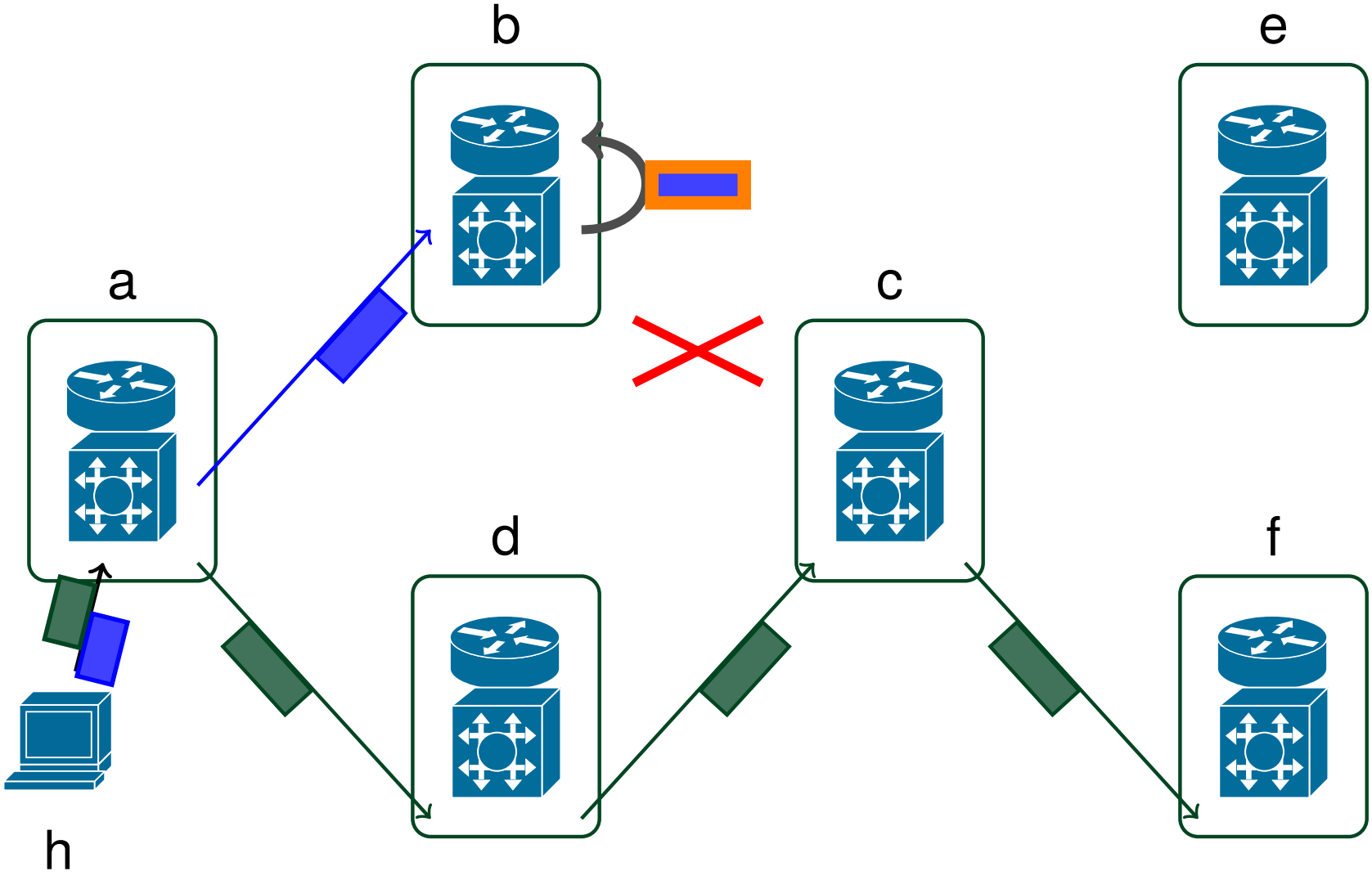
IBSDN offers the same expressiveness than Openflow during normal operation



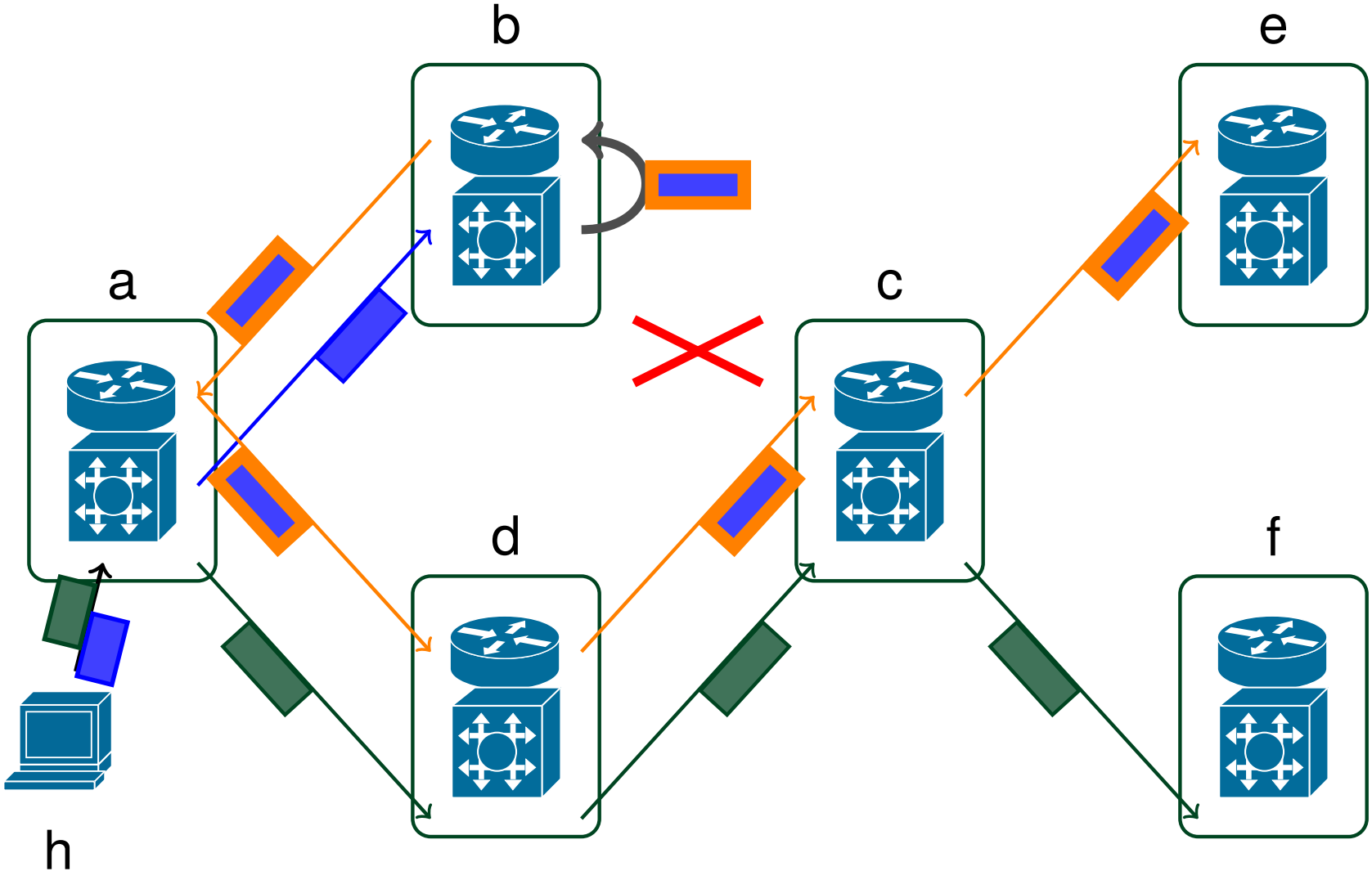
IBSDN reacts to failures by using the underlying IGP as failover



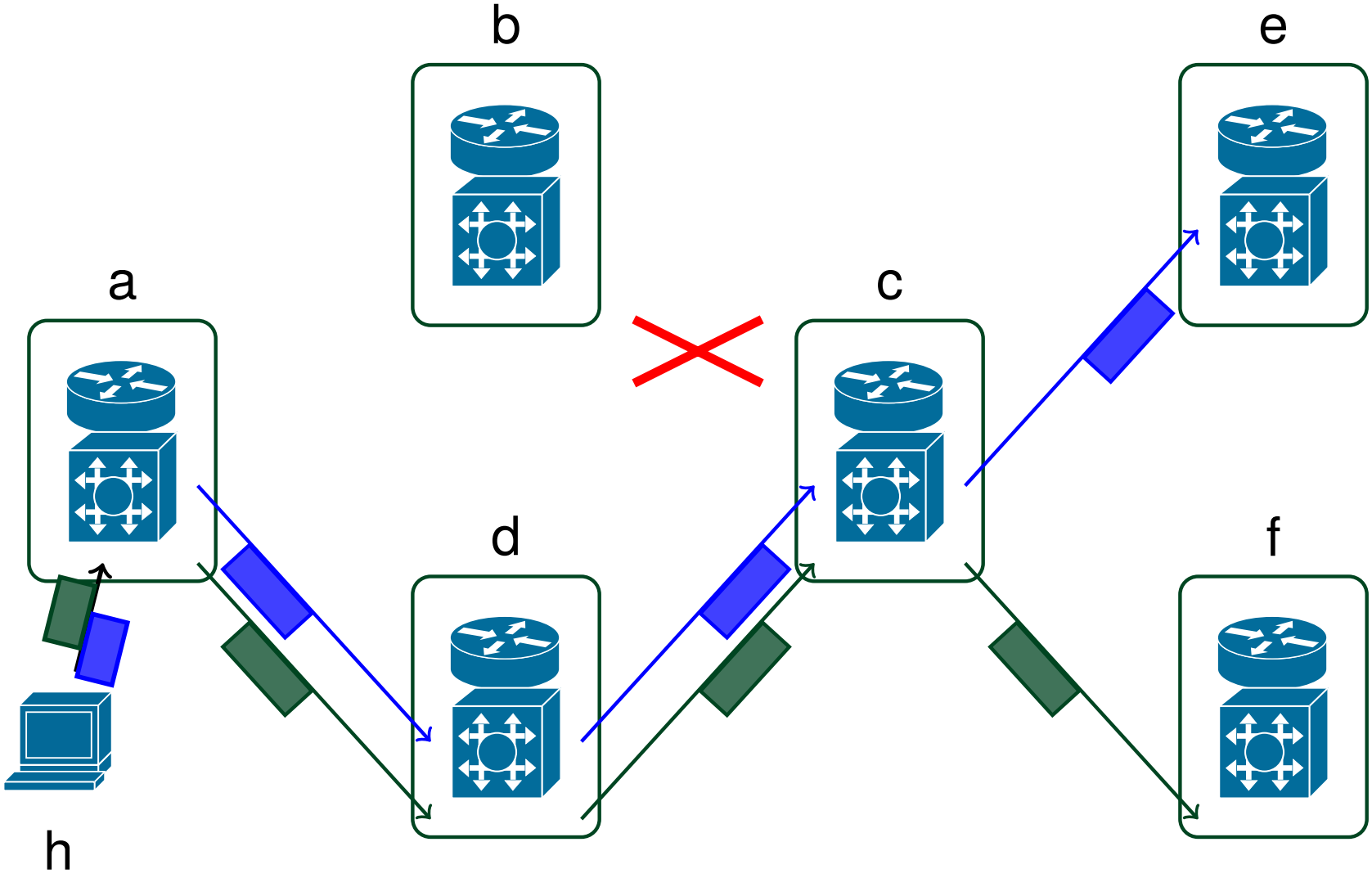
IBSDN reacts to failures by using the underlying IGP as failover



IBSDN reacts to failures by using the underlying IGP as failover



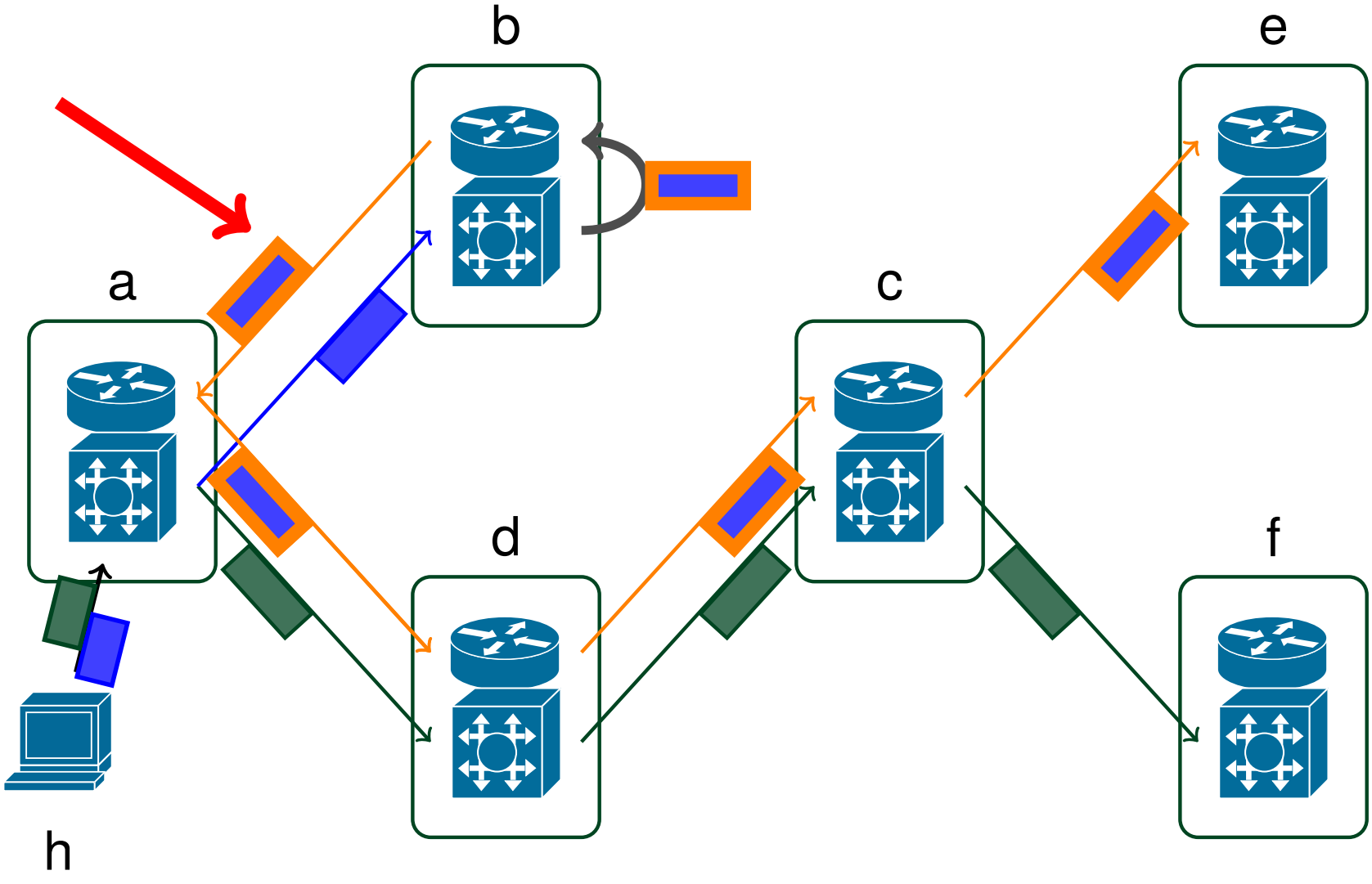
Until the controller computes and installs the new set of optimal SDN rules



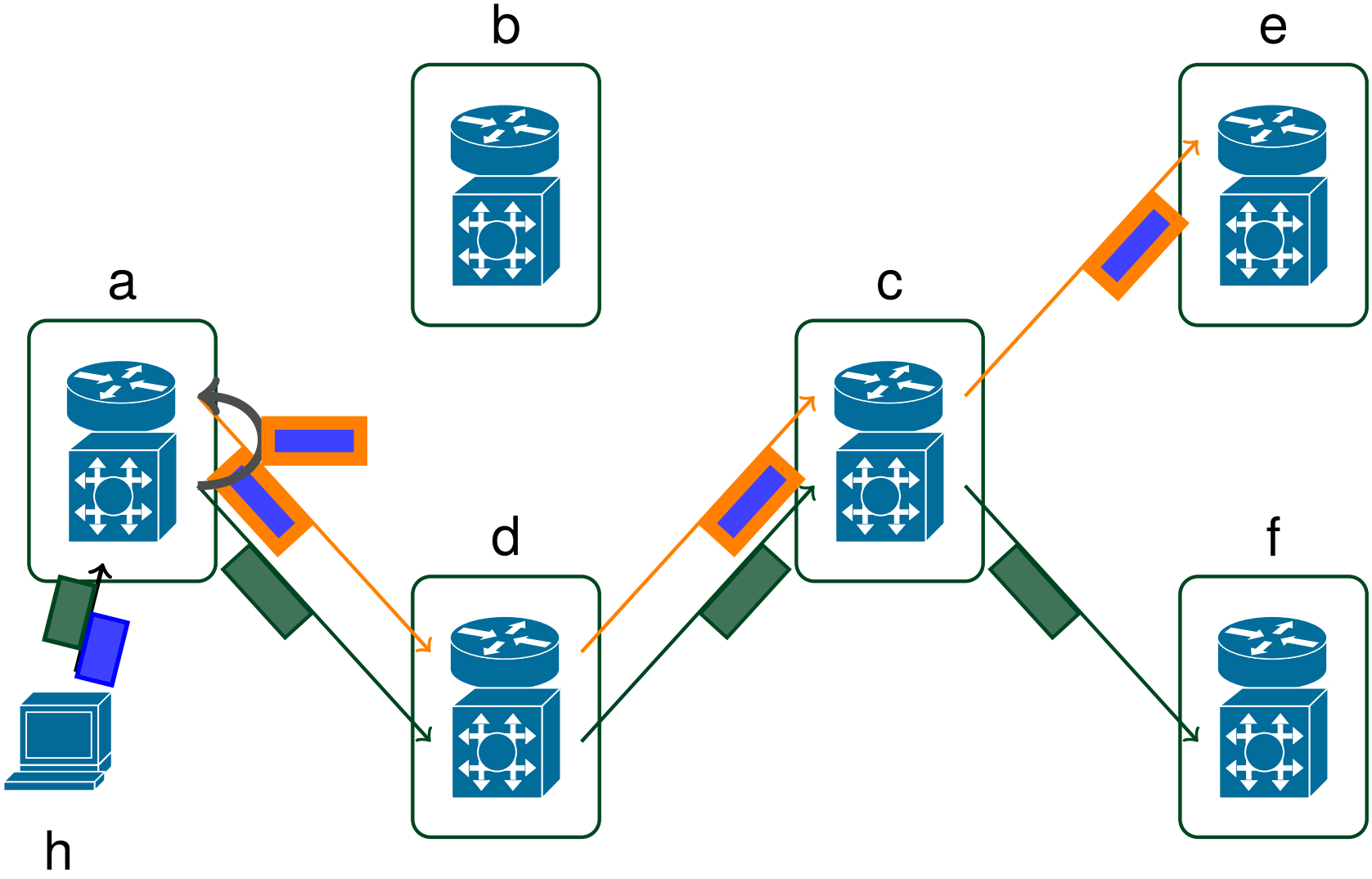
Enabling a performant IBSDN

Packet returns stretch the post-failure path

Packet returns stretch the post-failure path



We built a packet return removal procedure



Enabling a performant IBSDN

Packet returns stretch the post-failure path

Packet return removal procedure

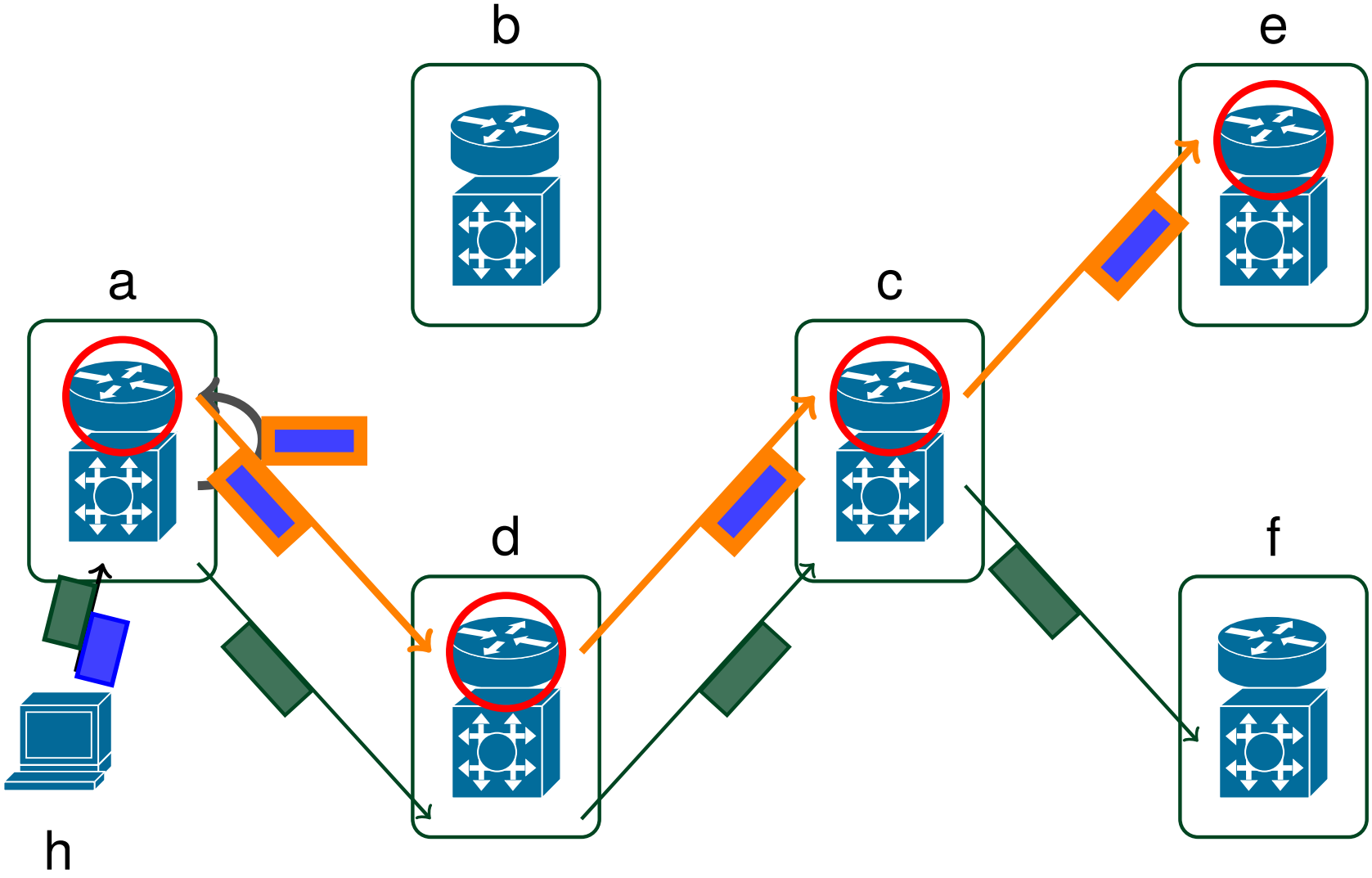
Enabling a performant IBSDN

Packet returns stretch the post-failure path

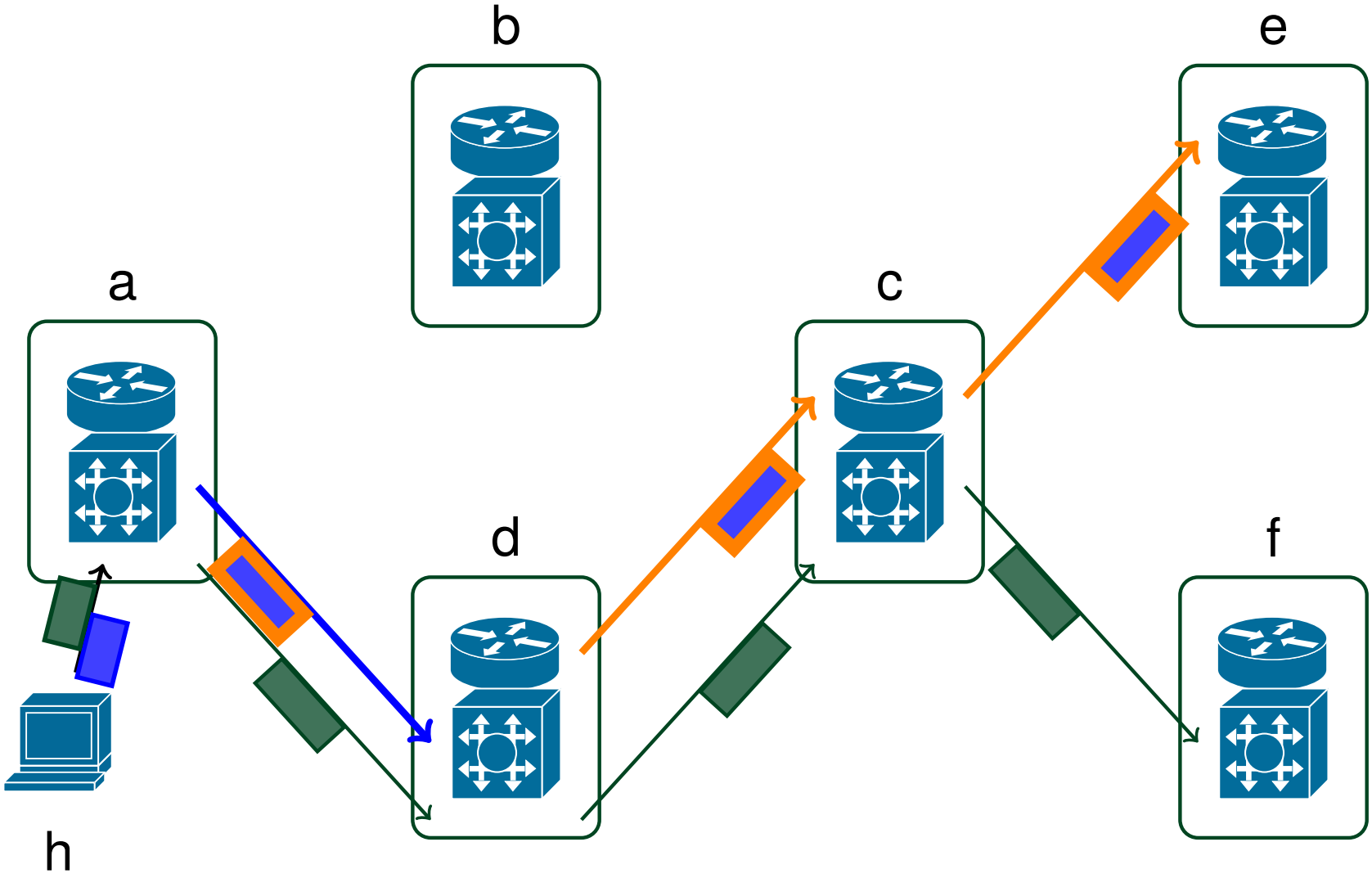
Packet return removal procedure

Forwarding packets through local agent is inefficient

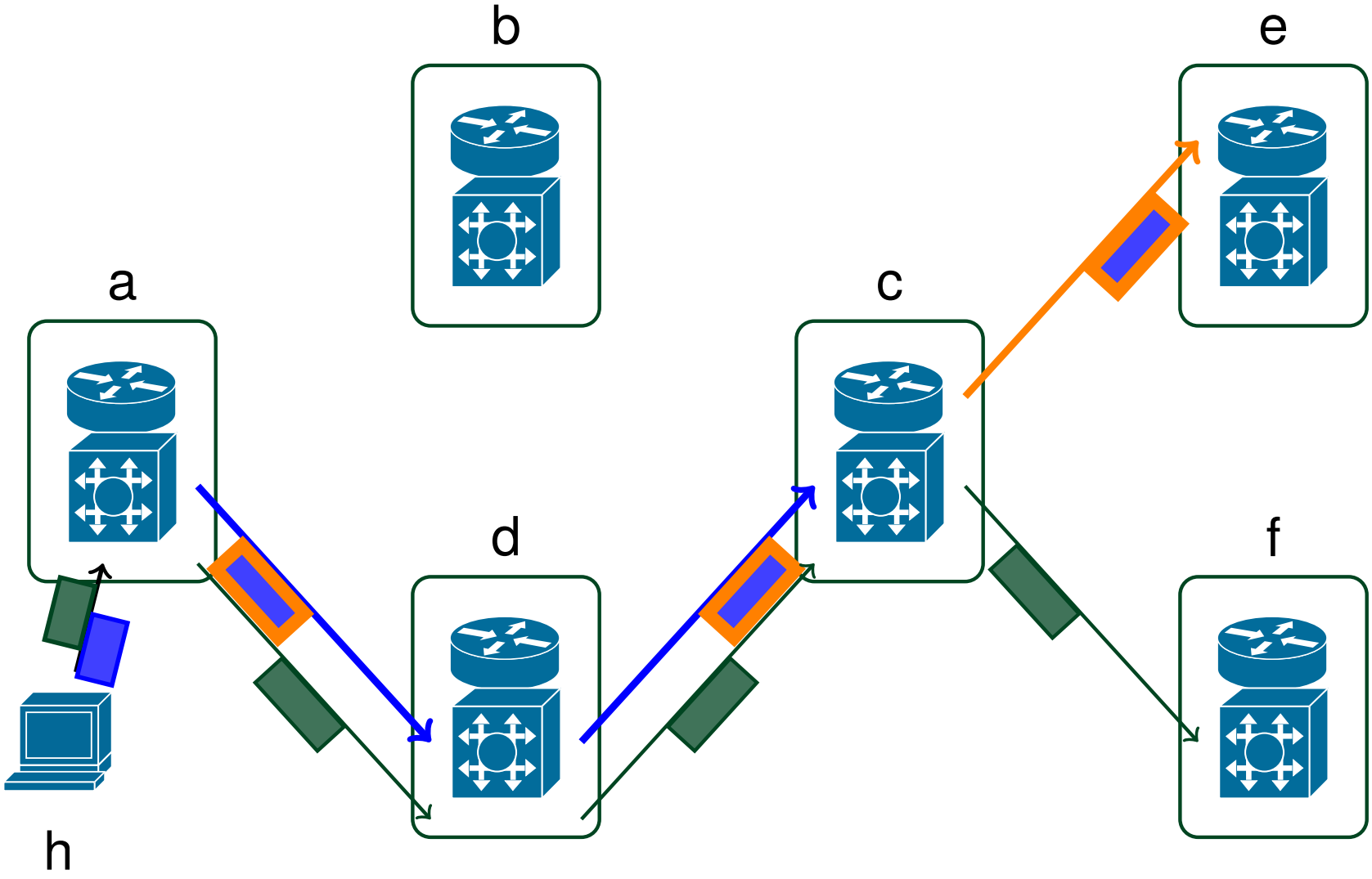
Forwarding through software local agents is slow



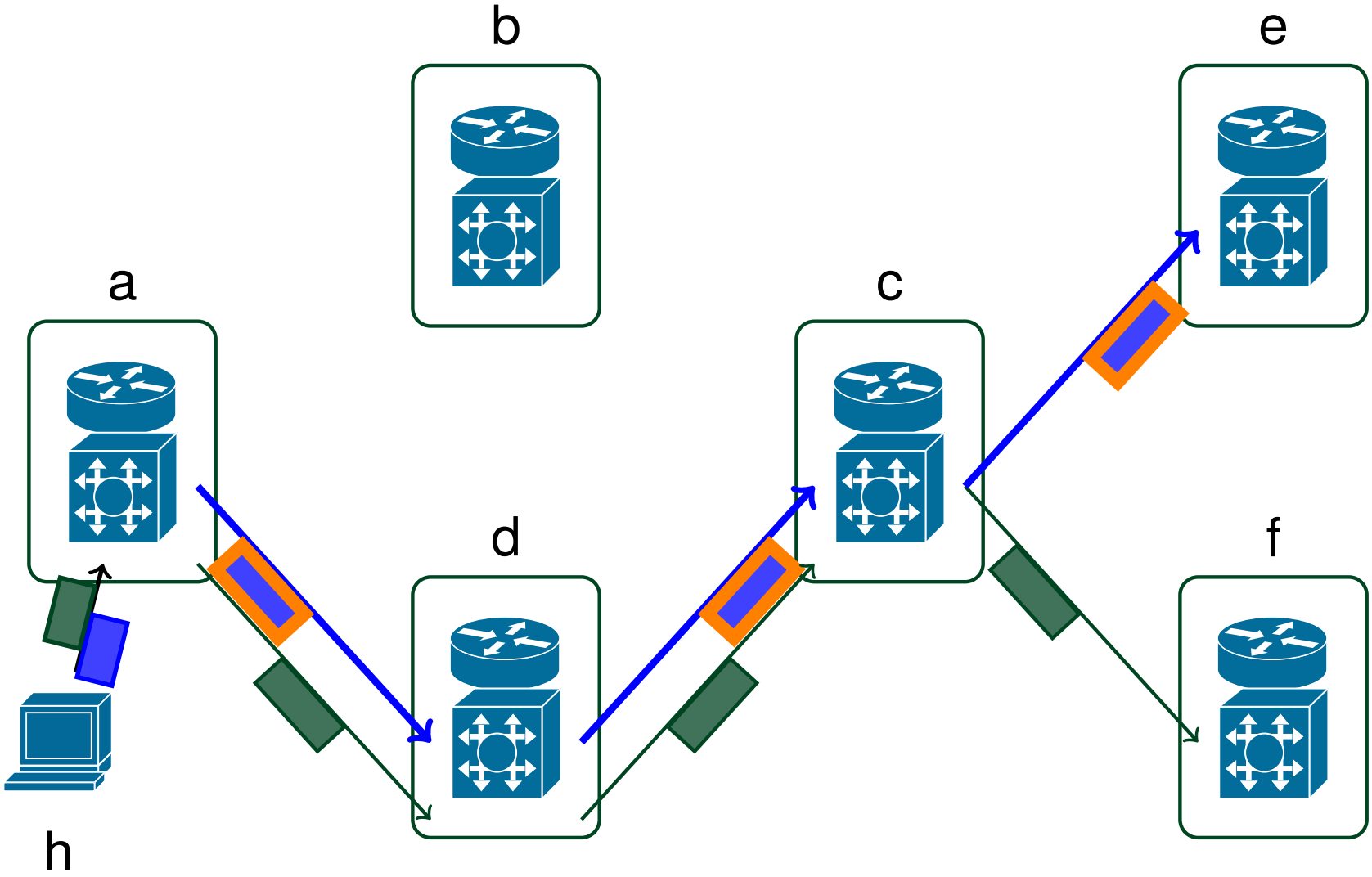
Removing slow local Forwarding



Removing slow local Forwarding



Removing slow local Forwarding



Enabling a performant IBSDN

Packet returns stretch the backup path

Forwarding packets through local agent is inefficient

Generalized packet return removal procedure

IBSDN has strong guarantees

Safety

Th.1 Connectivity is preserved for any combination of failures if there is no network partition, without any action from the controller

IBSDN has strong guarantees

Safety

Th.1 Connectivity is preserved for any combination of failures if there is no network partition, without any action from the controller

Efficiency

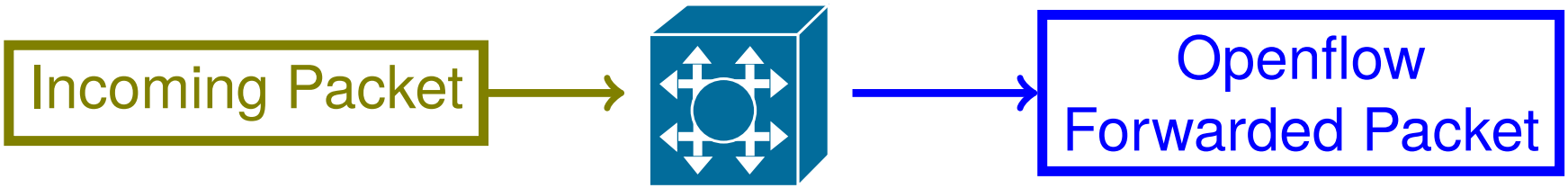
Th.2 Packet returns are removed in linear time

Th.3 Slow forwarding is removed in linear time

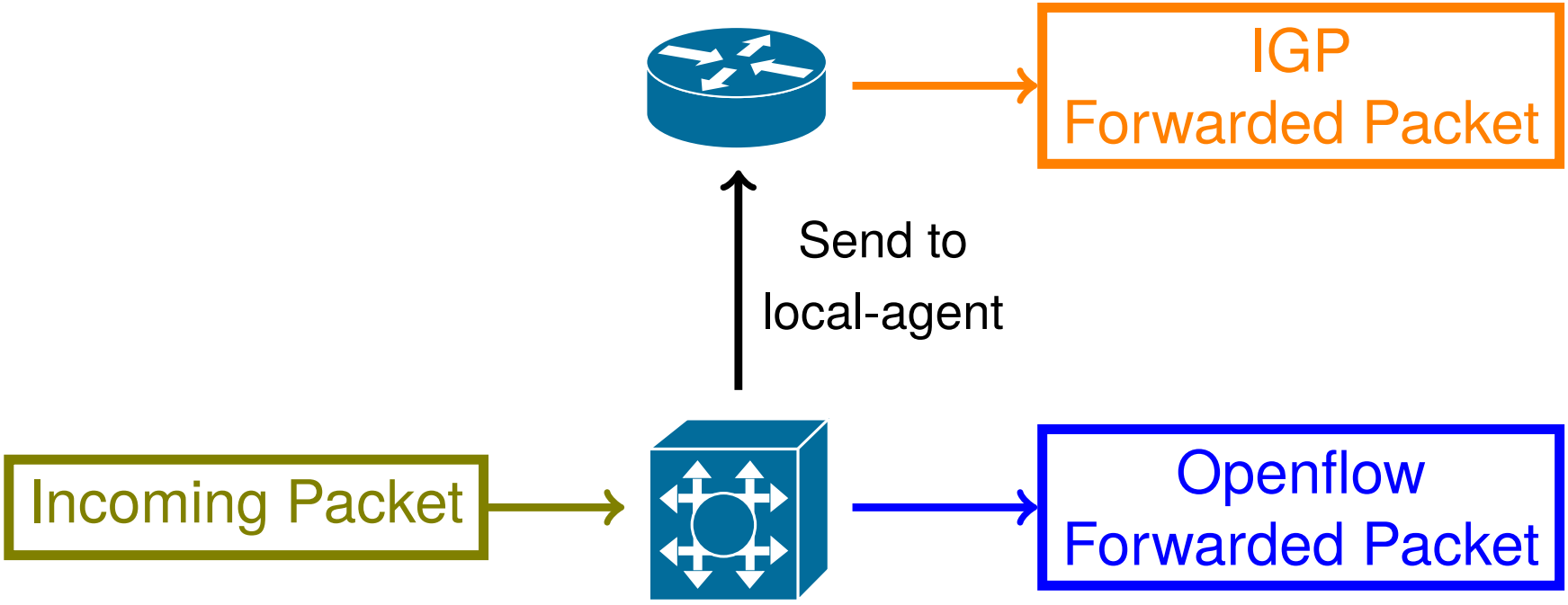
Implementation and Evaluation

Does it work in practice?

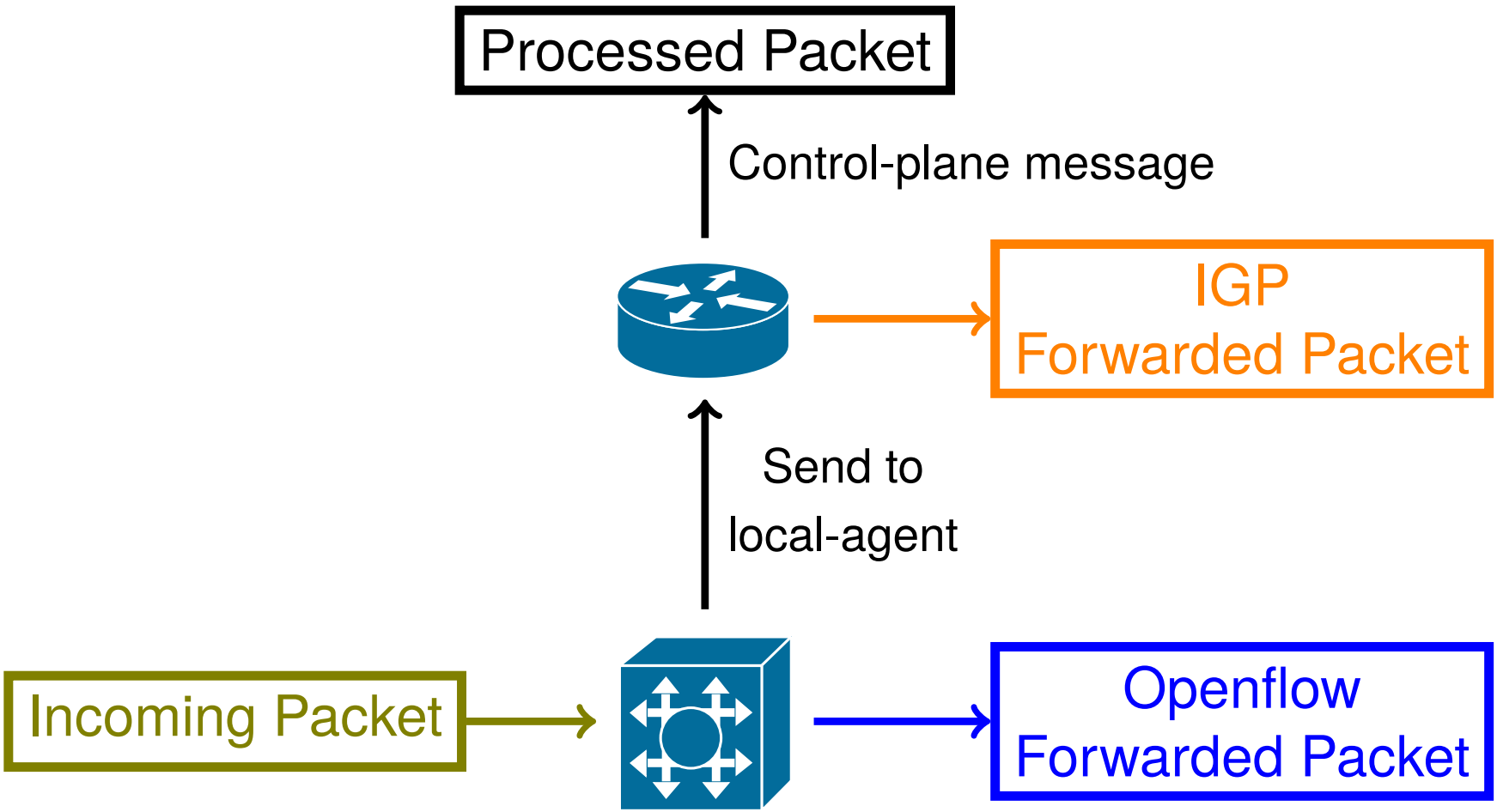
An IBSDN node is a coordinated stack of forwarding decisions



An IBSDN node is a coordinated stack of forwarding decisions



An IBSDN node is a coordinated stack of forwarding decisions



Can be implemented today!

Vanilla Openflow 1.1+

Uses the Logical OFPP_NORMAL Openflow port and Fast-failover groups

Implemented on Linux machines with modified Open vSwitch

Experiments in a virtual testbed confirmed IBSDN safety

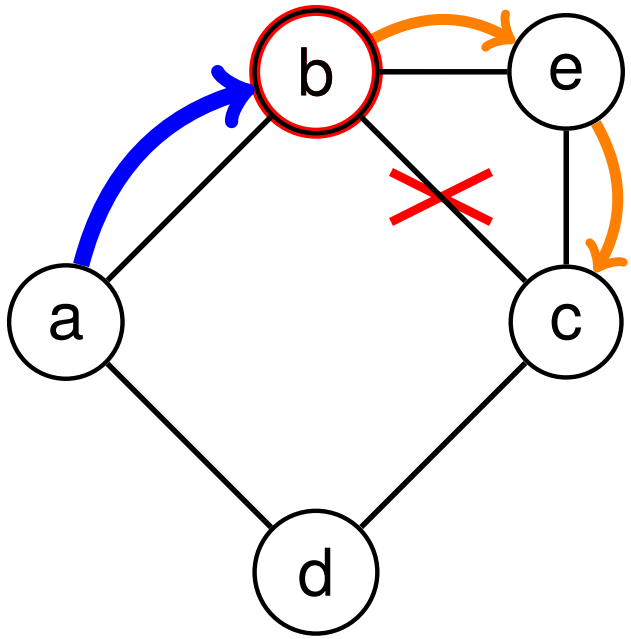
No packet losses if the failure does not trigger IGP convergence

In the worst case, IGP convergence is fast [1]

[1] *Francois, Pierre, et al. "Achieving sub-second IGP convergence in large IP networks." ACM SIGCOMM Computer Communication Review 35.3 (2005): 35-44.*

Evaluating the path stretch

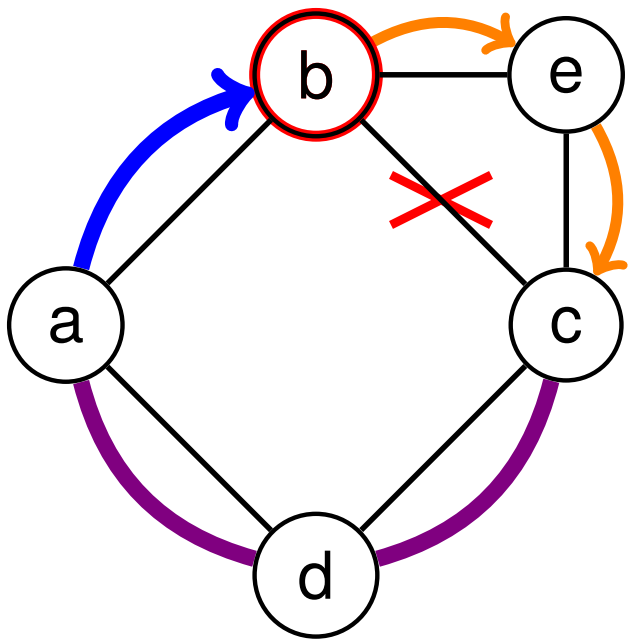
$$|\text{IBSDN path}| := |\text{Path until node adjacent to failure}| + |\text{IGP path from there to the destination}|$$



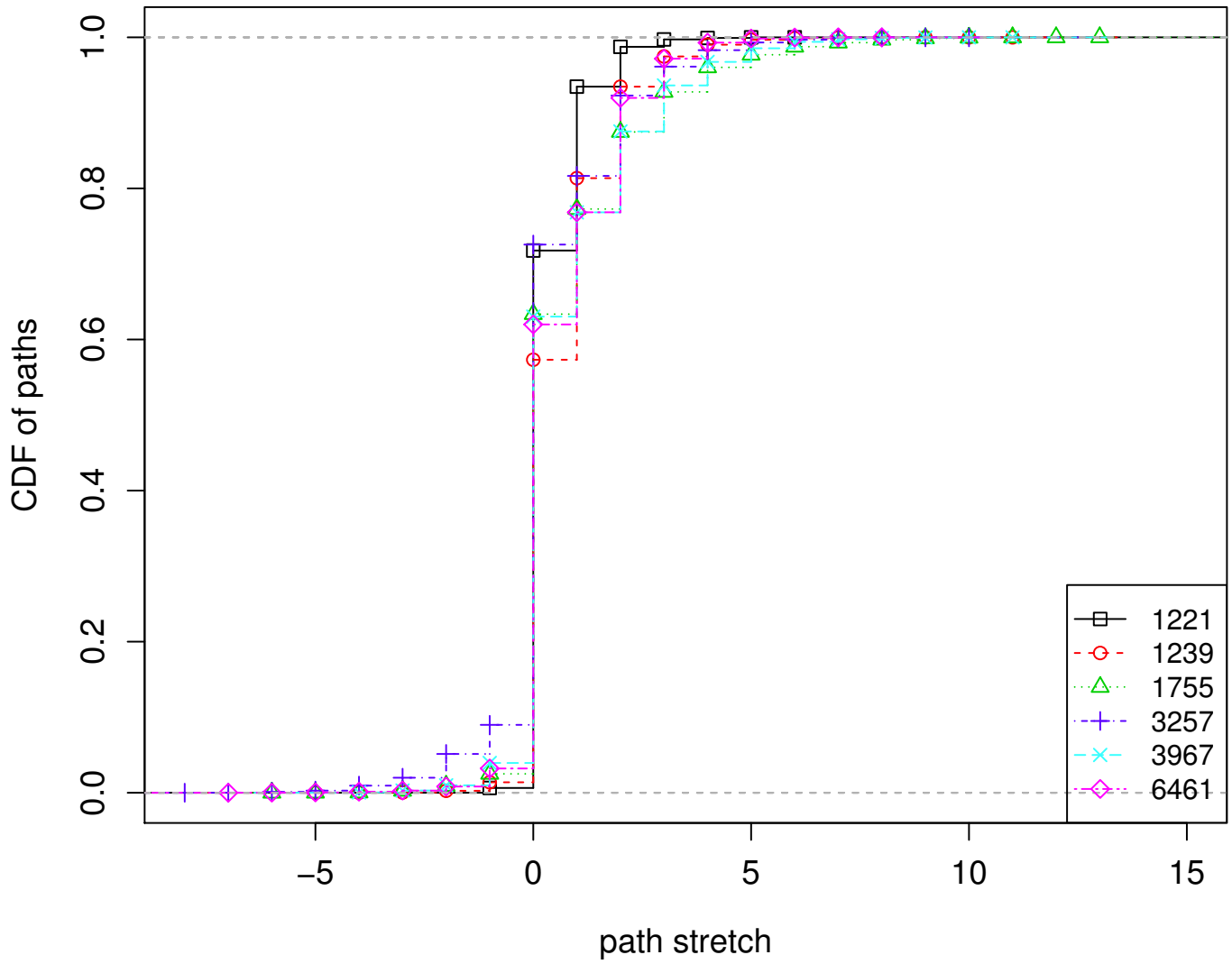
Evaluating the path stretch

$$
\begin{aligned}
|\text{IBSDN path}| &:= |\text{Path until node adjacent to failure}| \\
&+ |\text{IGP path from there to the destination}|
\end{aligned}
$$

$$
|\text{Path stretch}| := |\text{IBSDN backup path}| - |\text{IGP path}|
$$



The packet return removal procedure effectively removes most of the path stretches



IBSDN is not only about failure recovery

IBSDN is not only about failure recovery

- Incremental deployment of SDN functions
- Communication with an inband controller
- …

IGP-as-a-Backup for Robust Software-Defined Networks

IBSDN flanks a SDN with an IGP

Implements separation of concern in network management

Benefits from both control-planes