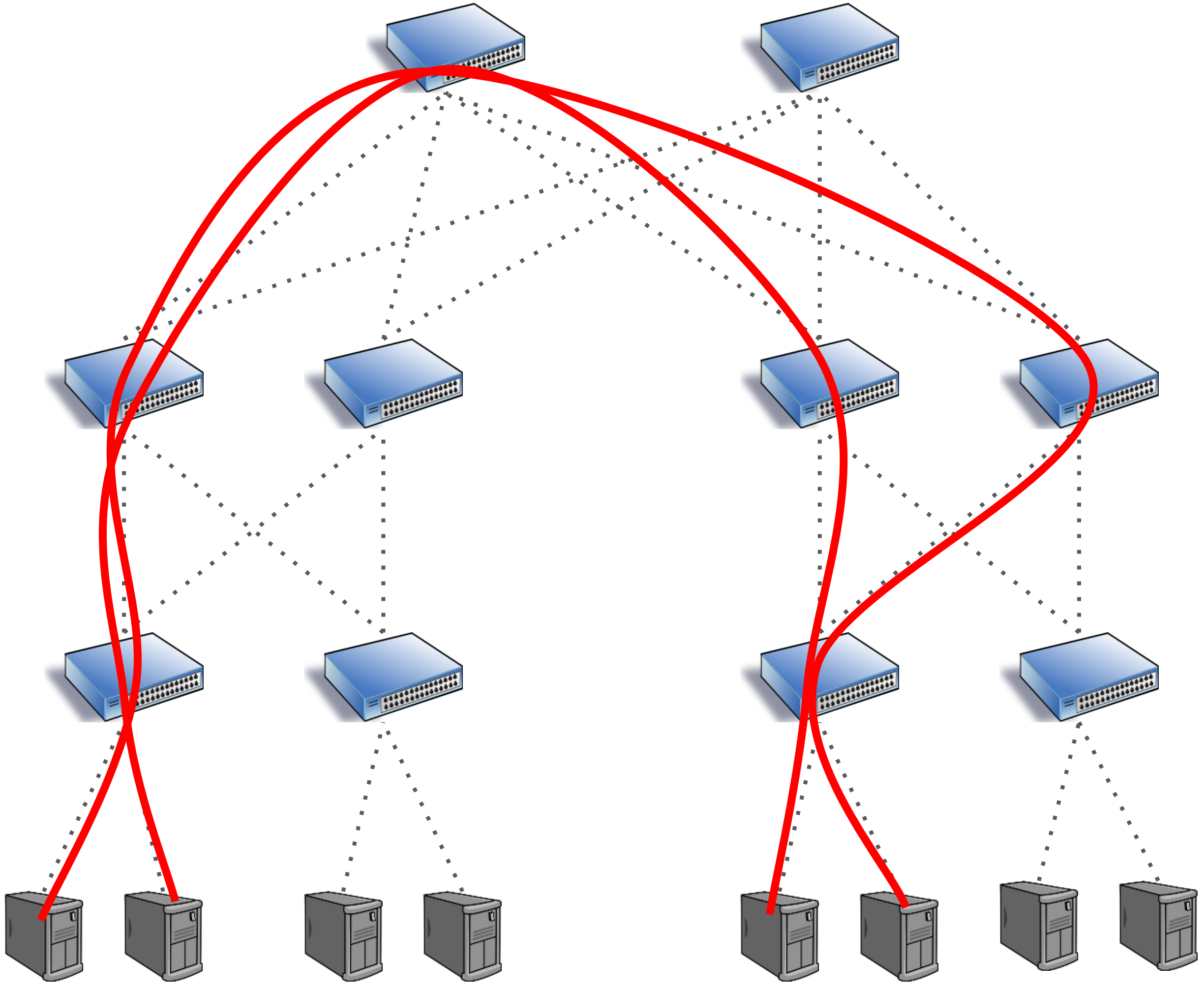
ESPRES: Transparent SDN Update Scheduling

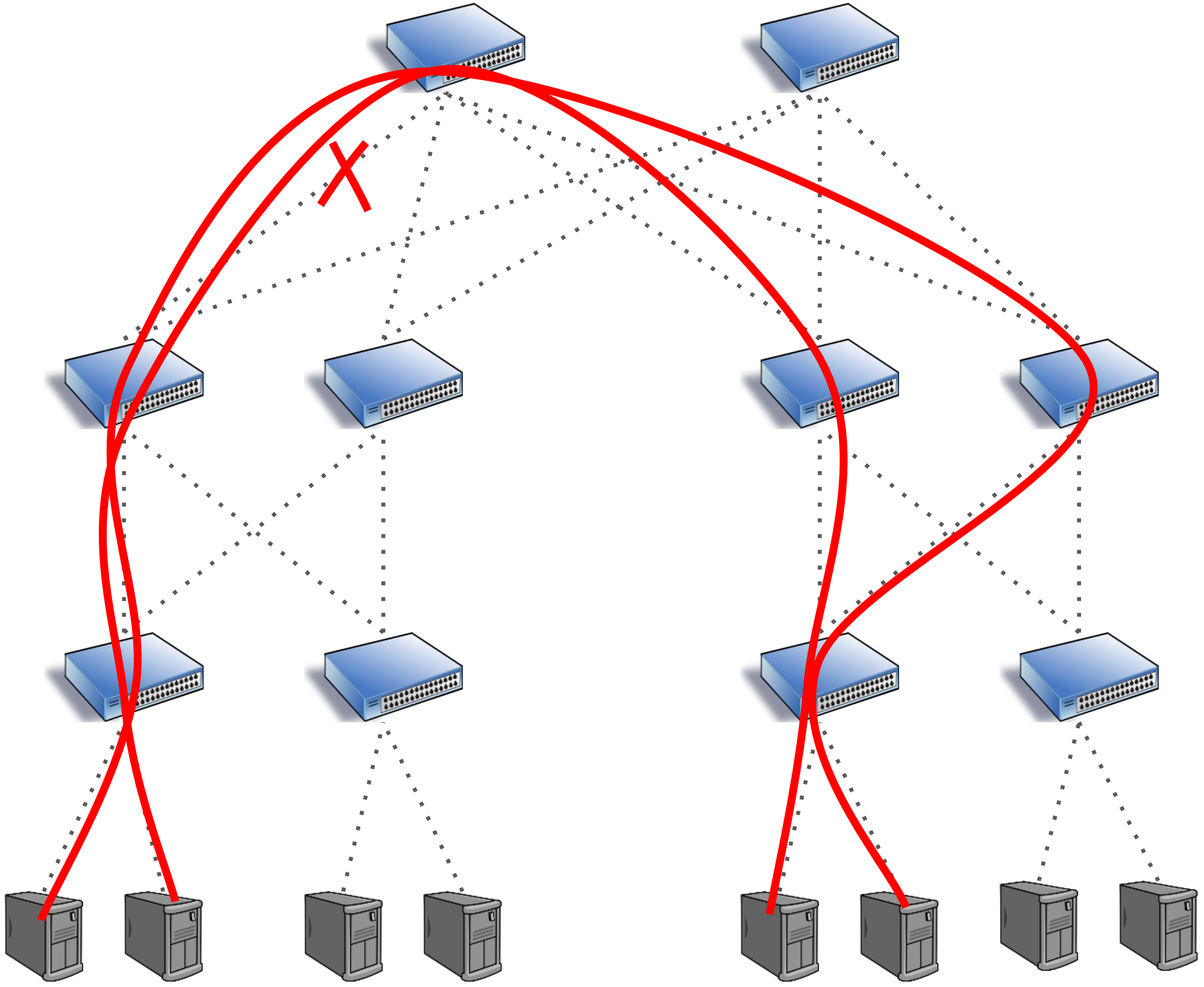
Peter Perešíni @ EPFL
Maciej Kuzniar @ EPFL
Marco Canini @ UCLouvain
Dejan Kostić @ KTH

Network events trigger big updates



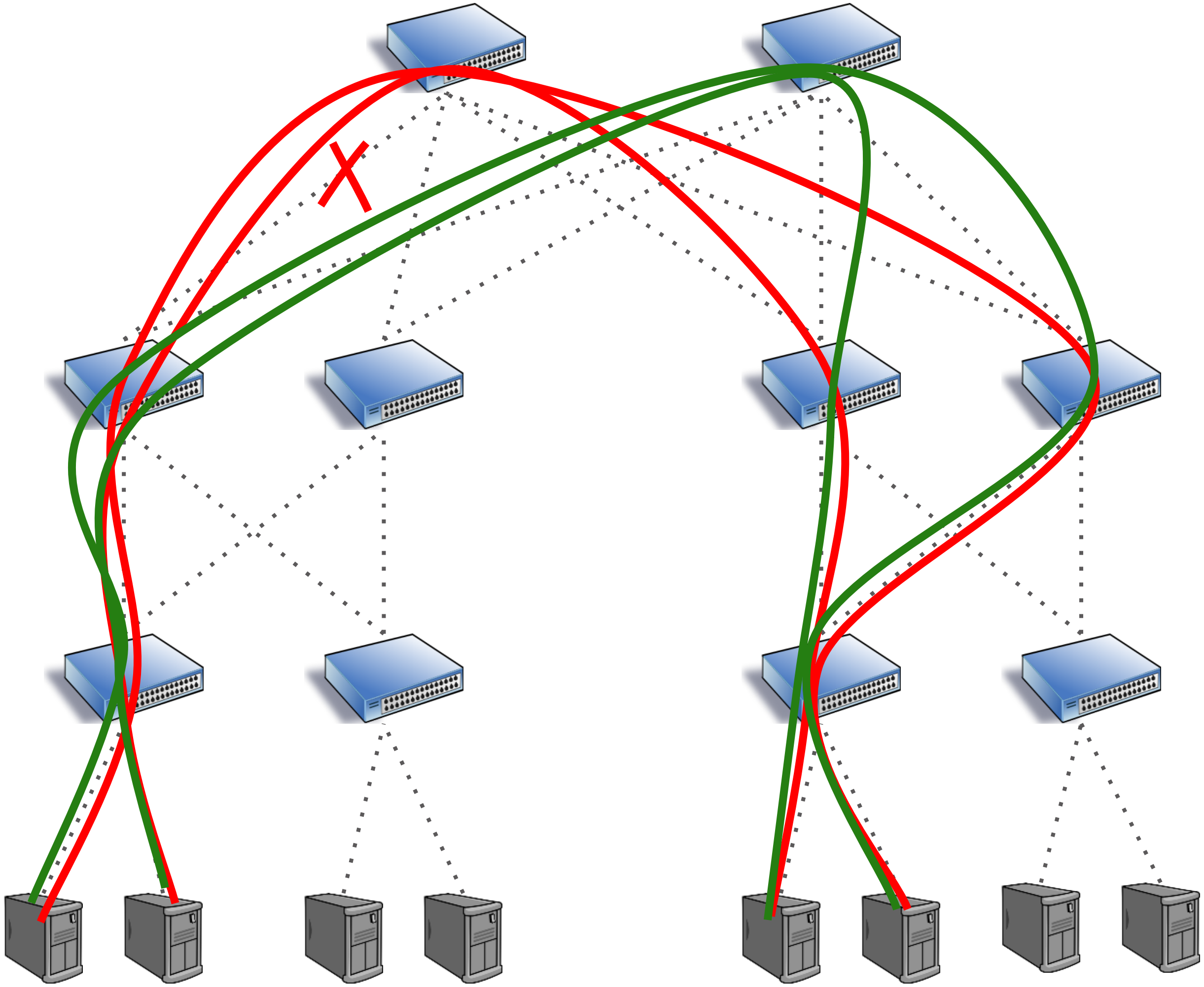
Topology changes

Network events trigger big updates



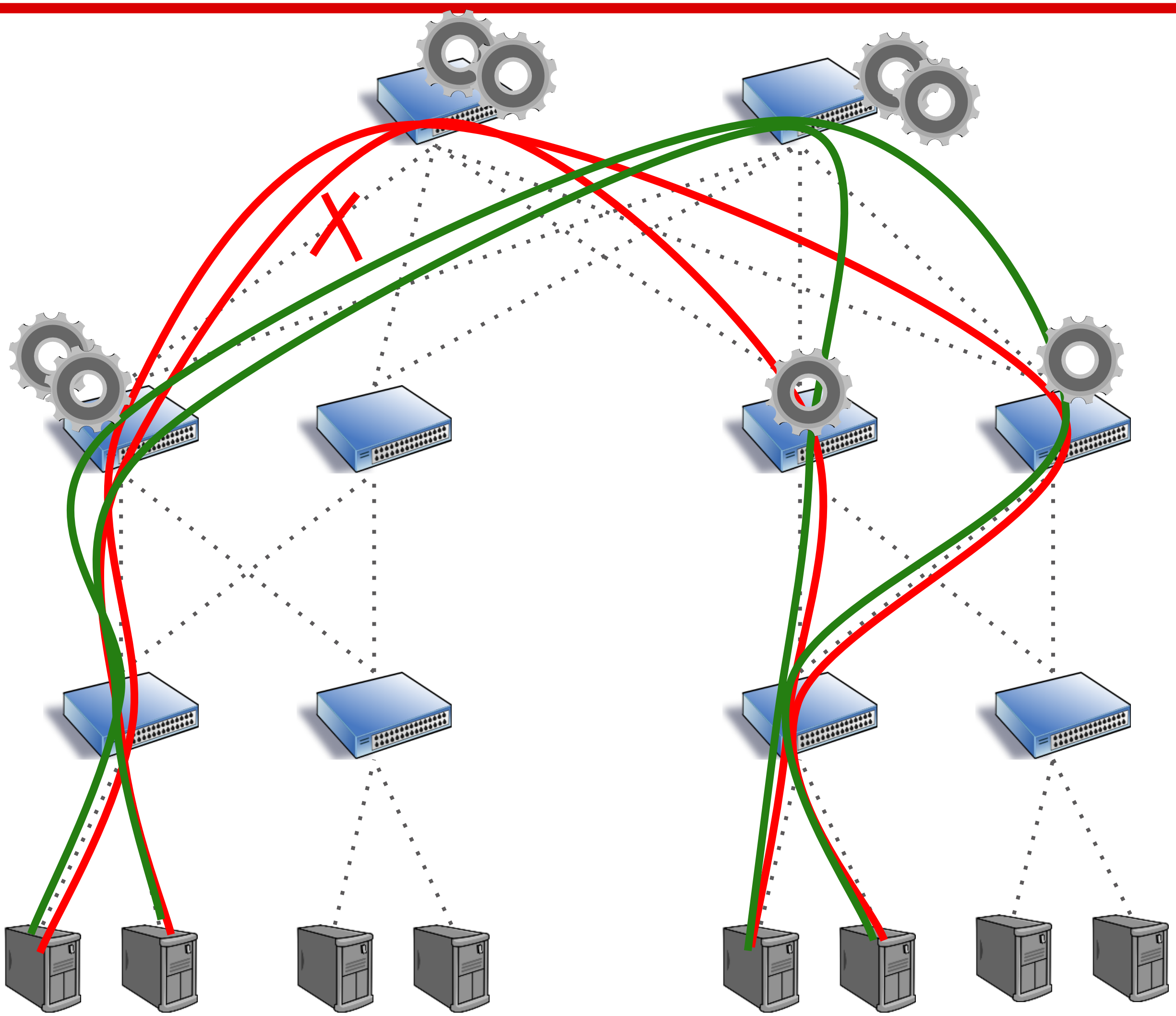
Topology changes

Network events trigger big updates



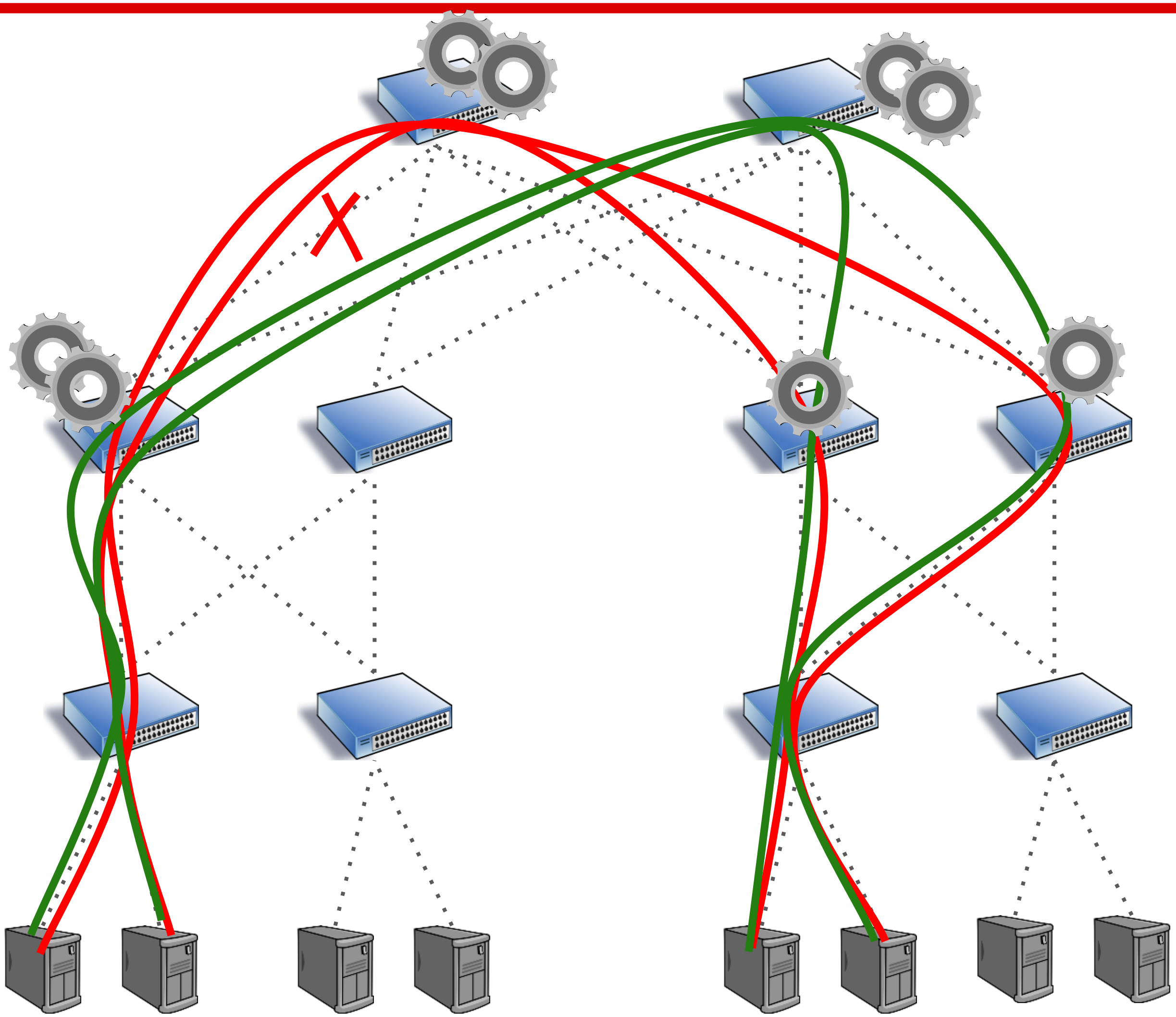
Topology changes

Network events trigger big updates



Topology changes

Network events trigger big updates

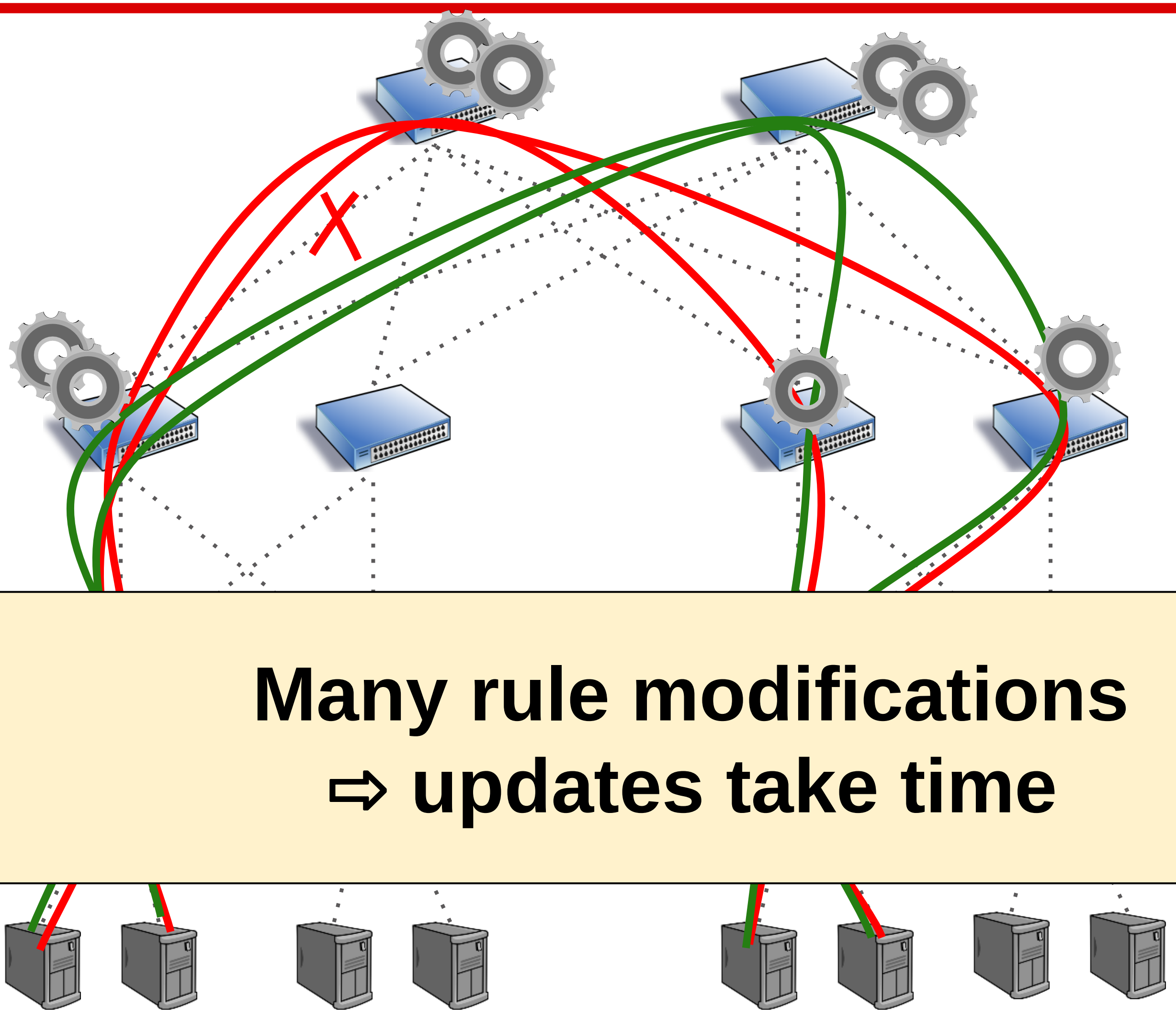


Topology changes

Traffic Engineering

Policy changes

Network events trigger big updates



Topology changes

Traffic Engineering

Policy changes

Many rule modifications
⇨ updates take time

What if we reorder installations...

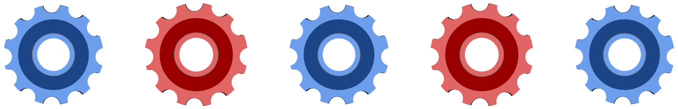
Update touching two (independent) flows

- blue - 3 changes on a switch
- red - 2 changes on a switch

What if we reorder installations...

Update touching two (independent) flows

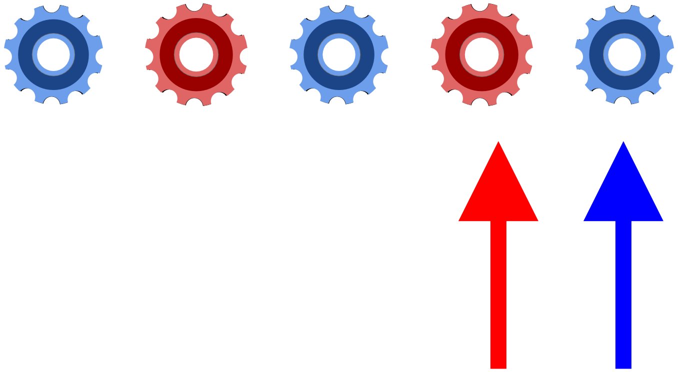
- blue - 3 changes on a switch
- red - 2 changes on a switch



What if we reorder installations...

Update touching two (independent) flows

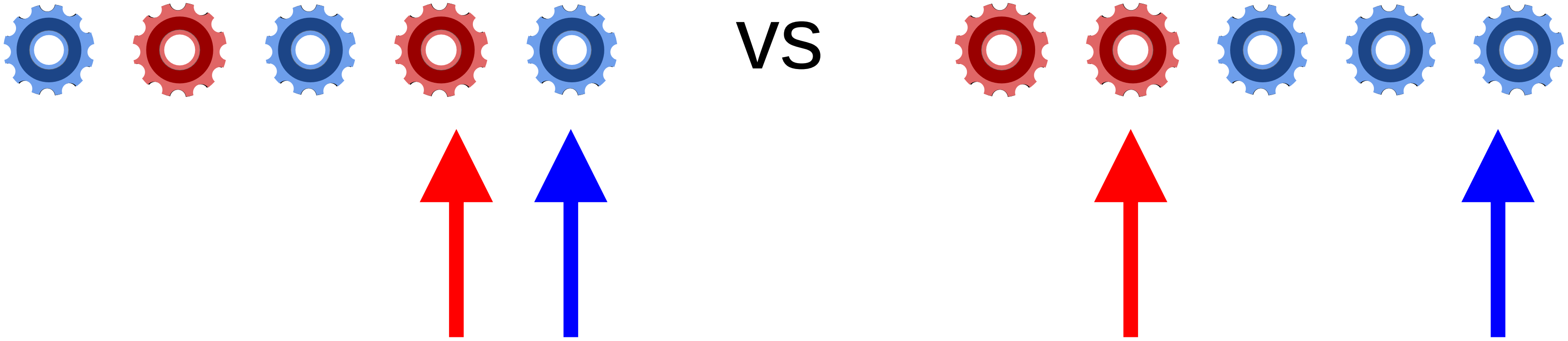
- blue - 3 changes on a switch
- red - 2 changes on a switch



What if we reorder installations...

Update touching two (independent) flows

- blue - 3 changes on a switch
- red - 2 changes on a switch



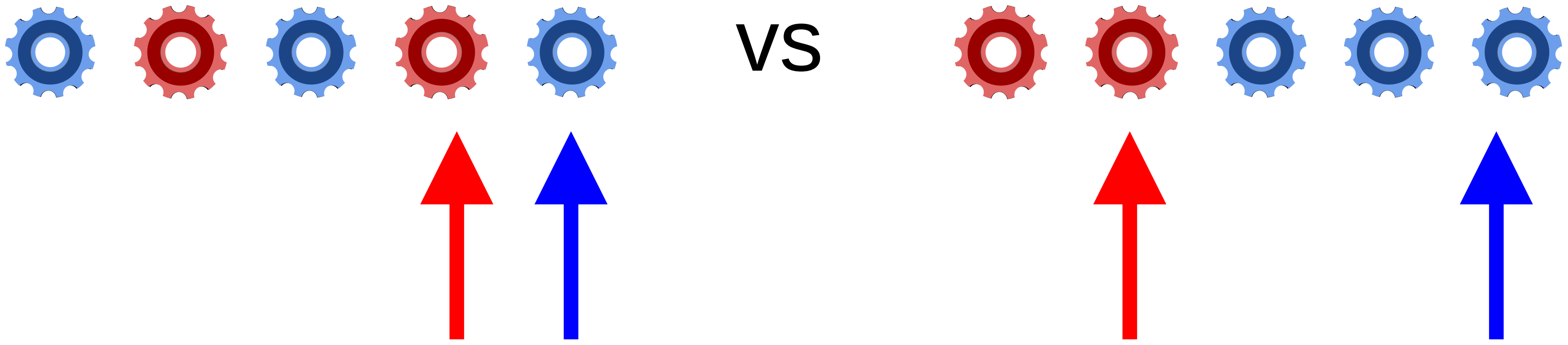
What if we reorder installations...

Update touching two (independent) flows

total time same

but

different **ordering** of rule installation **matters**



Some challenges

Scheduling sounds easy but

- multiple switches affected
  - and switch-speeds are different over time
- rule dependencies
  - e.g. update ingress only after core installed
- control channel is FIFO
  - no reordering

ESPRES overview

(See paper for more details)

- Keep backlog of rule installations in ESPRES
    - enables re-ordering on control channel
- Schedule rules to be installed next
    - react on-the-fly to current switch conditions
    - needs to be fast
    - support flexible goals

Possible scheduling goals

Already illustrated:

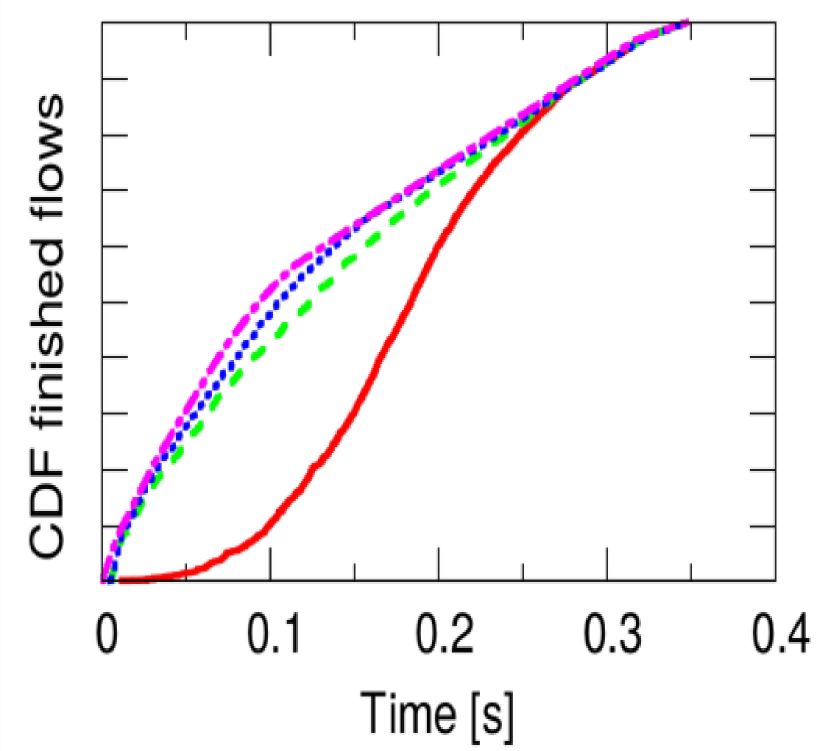
- install majority of flows sooner

See paper:

- decrease mid-update rule overhead
- decrease transient instabilities

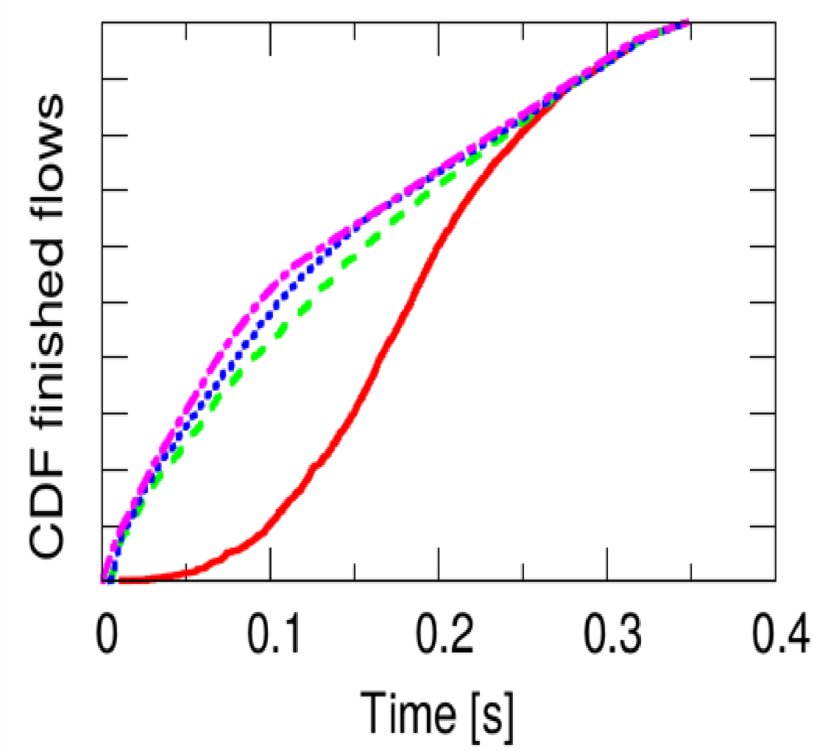
A taste of results

1000 new flows; 18 switches



A taste of results

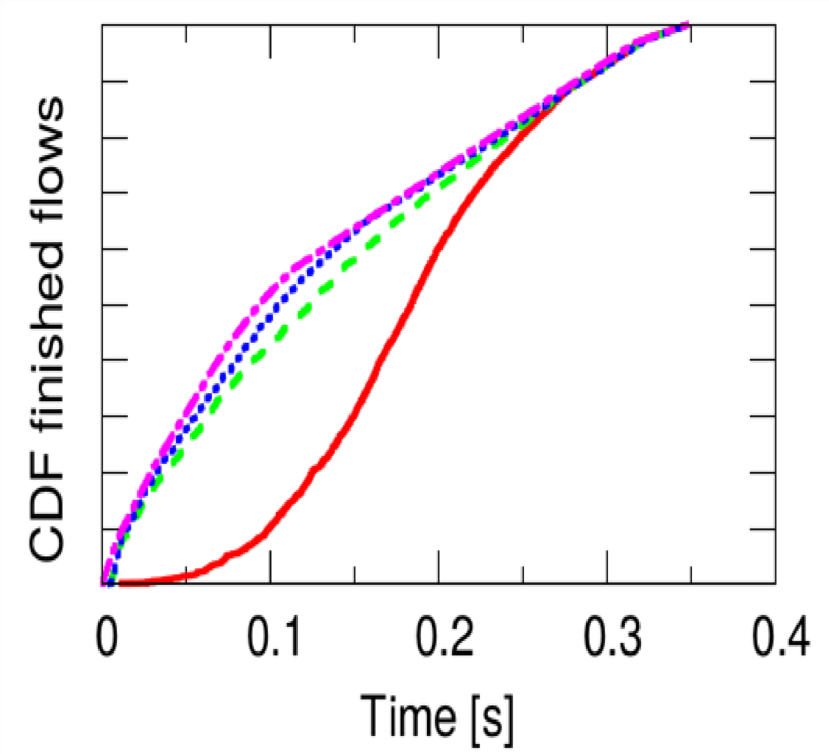
1000 new flows; 18 switches



- ➢ No scheduling (send rule when dependencies are met)

A taste of results

1000 new flows; 18 switches



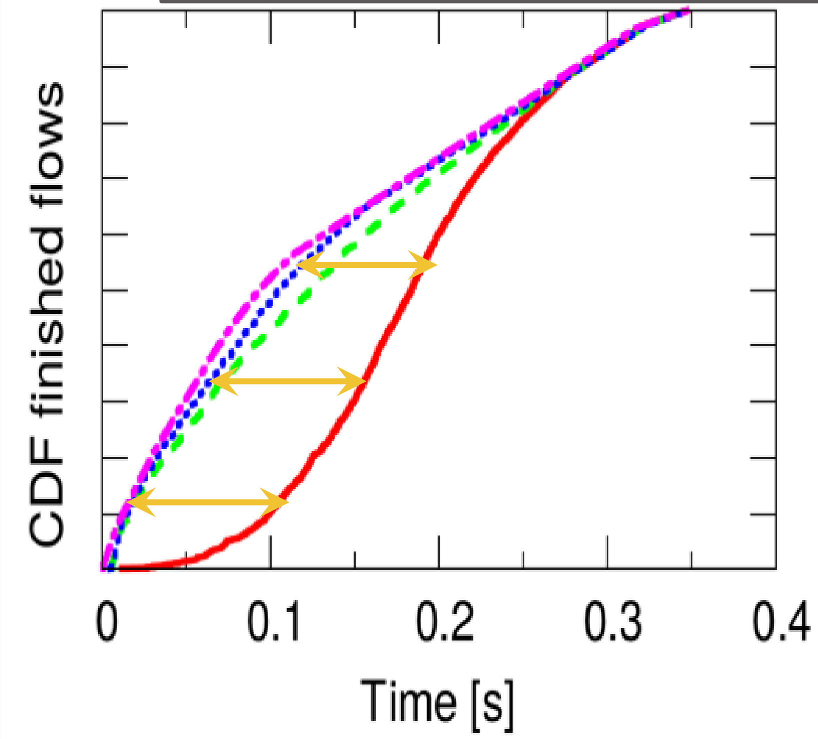
- No scheduling (send rule when dependencies are met)

- Espres A
- Espres B
- Optimal (Offline)

A taste of results

1000 n

We reduce completion time by ≥40% for half of flows



- ➢ No scheduling (send rule when dependencies are met)
- ➢ Espres A
- ➢ Espres B
- ➢ Optimal (Offline)

Summary

- Updates touch many flows
- Changing rule installation order helps achieve different goals

ESPRES

- ➢ Maintain backlog of rule installations at the controller
- ➢ Schedule their order on-the-fly