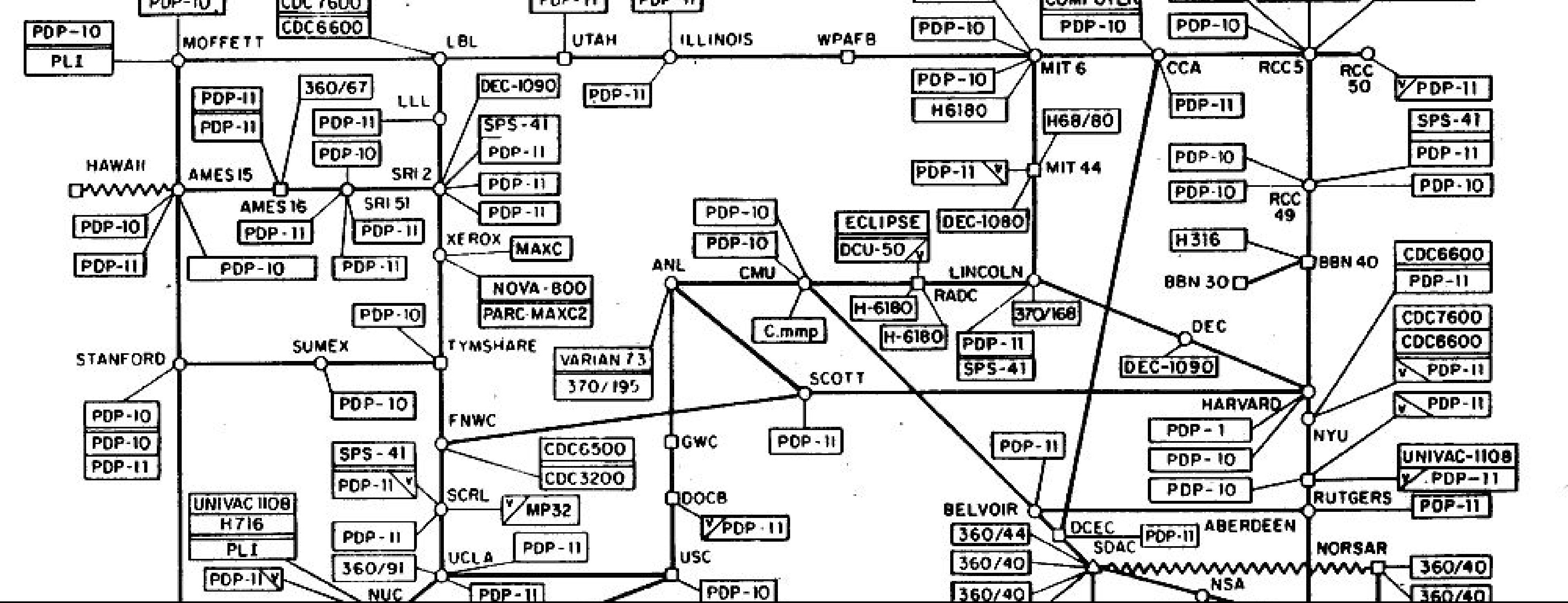


software defined networking

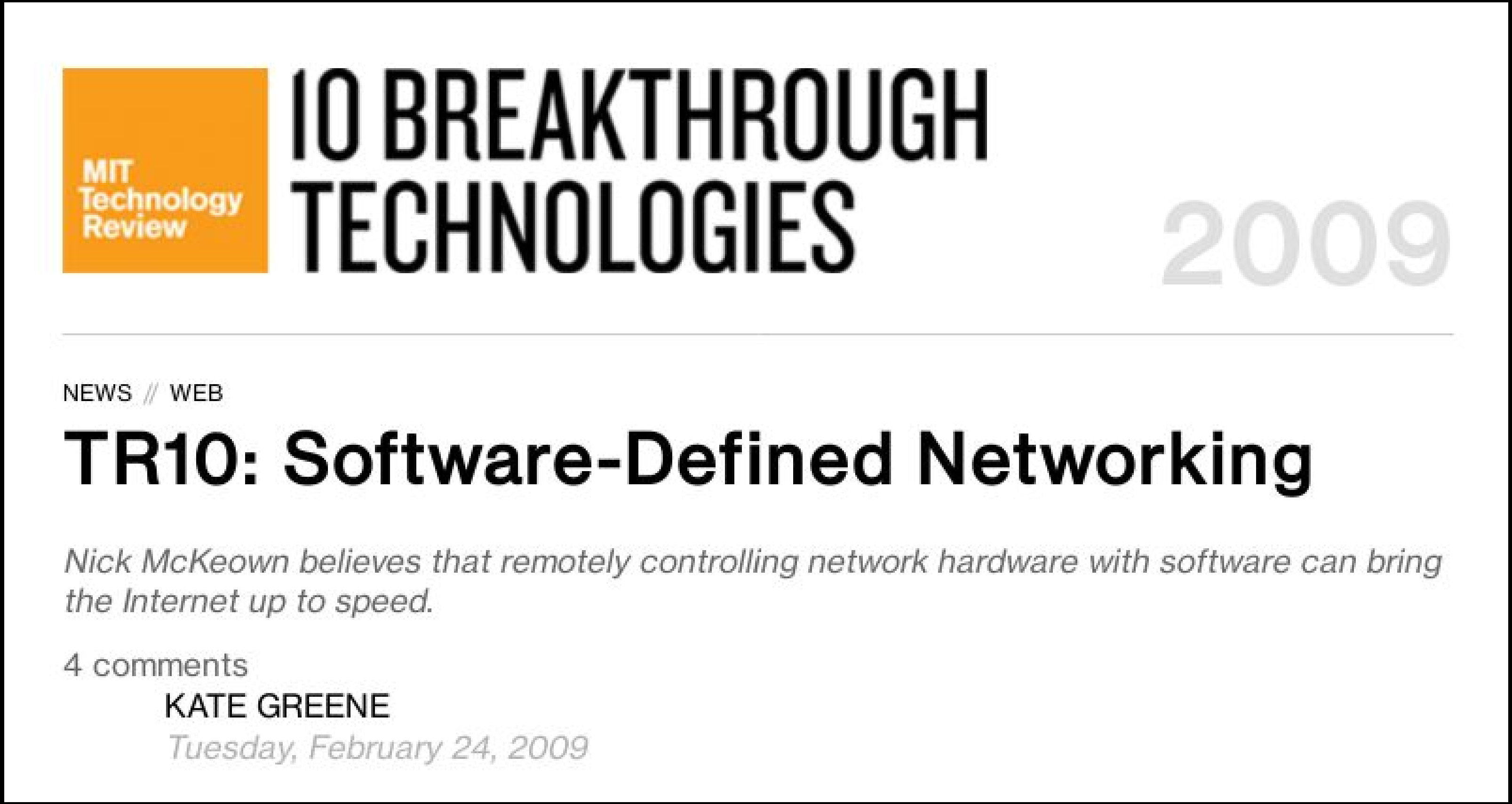
CS 6410 Fall 2017

Eric Campbell and Rolph Recto

software defined networking

software defined networking

(OpenFlow, originally)



MIT Technology Review

10 BREAKTHROUGH TECHNOLOGIES 2009

NEWS // WEB

TR10: Software-Defined Networking

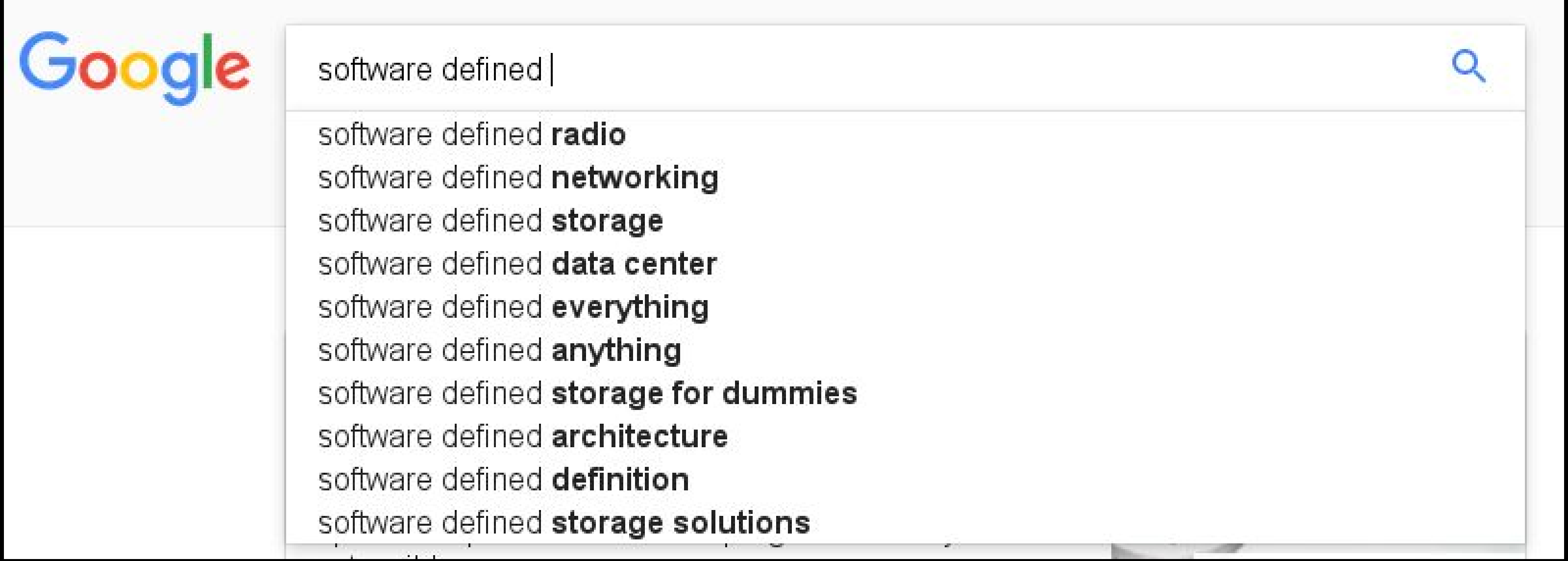
Nick McKeown believes that remotely controlling network hardware with software can bring the Internet up to speed.

4 comments

KATE GREENE

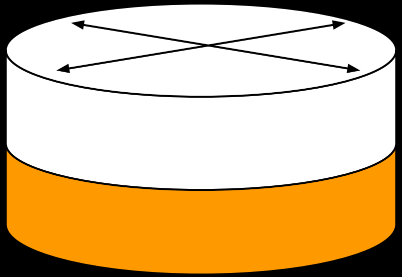
Tuesday, February 24, 2009

*“Stanford computer scientist Nick McKeown and colleagues developed a standard called OpenFlow that essentially opens up the Internet to researchers, allowing them to define data flows using software–a sort of ‘***software-defined networking***.’ Installing a small piece of OpenFlow firmware (software embedded in hardware) gives engineers access to flow tables, rules that tell switches and routers how to direct network traffic.”*



software defined

software defined **radio**
software defined **networking**
software defined **storage**
software defined **data center**
software defined **everything**
software defined **anything**
software defined **storage for dummies**
software defined **architecture**
software defined **definition**
software defined **storage solutions**



control plane

routing

isolation

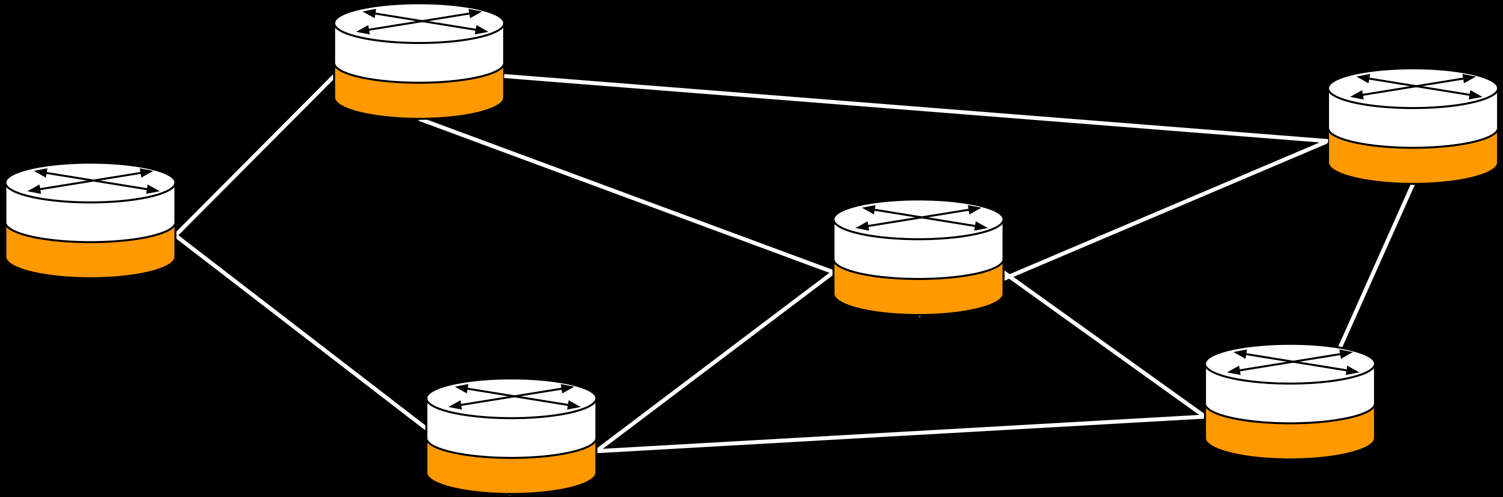
traffic engineering

data plane

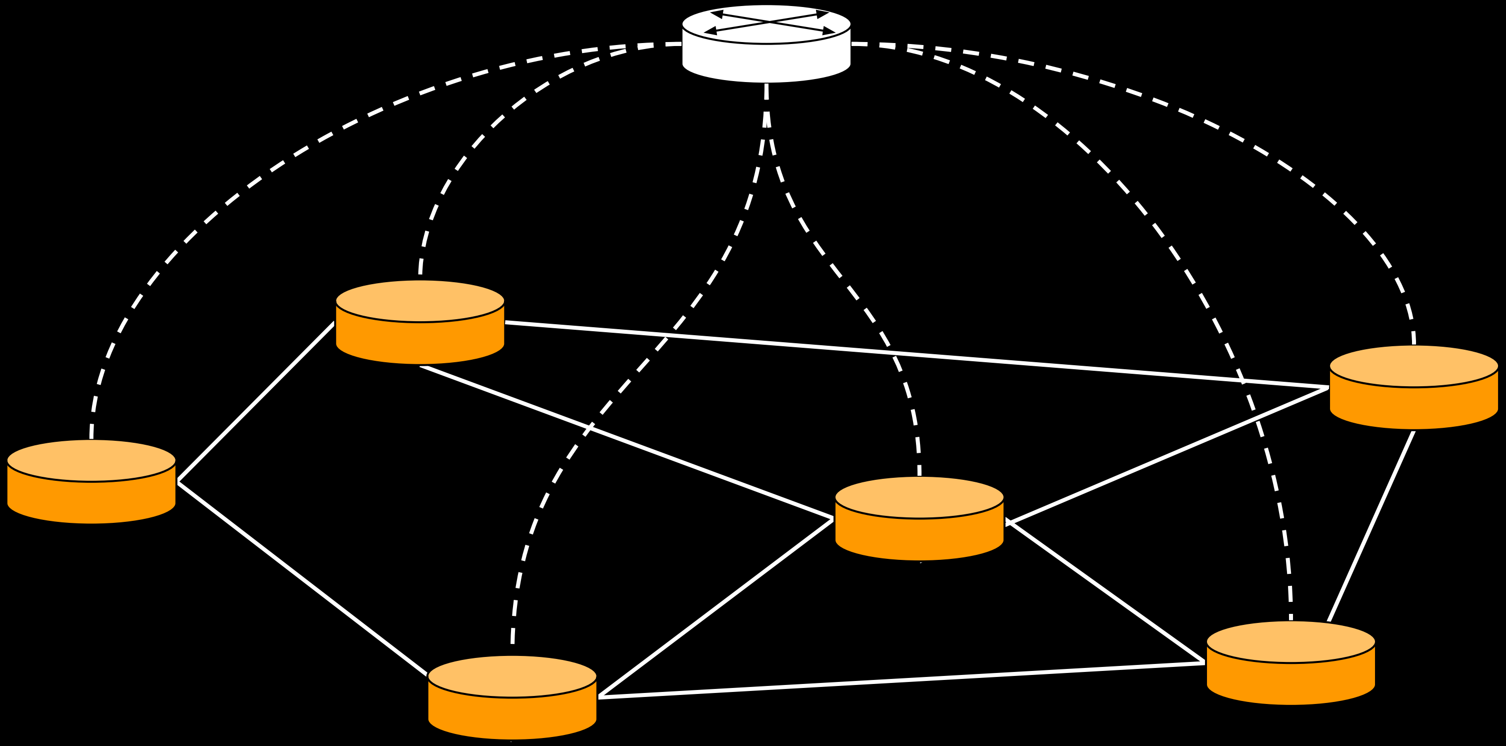
packet forwarding

packet scheduling

traditional networking



SDN



software defined networking

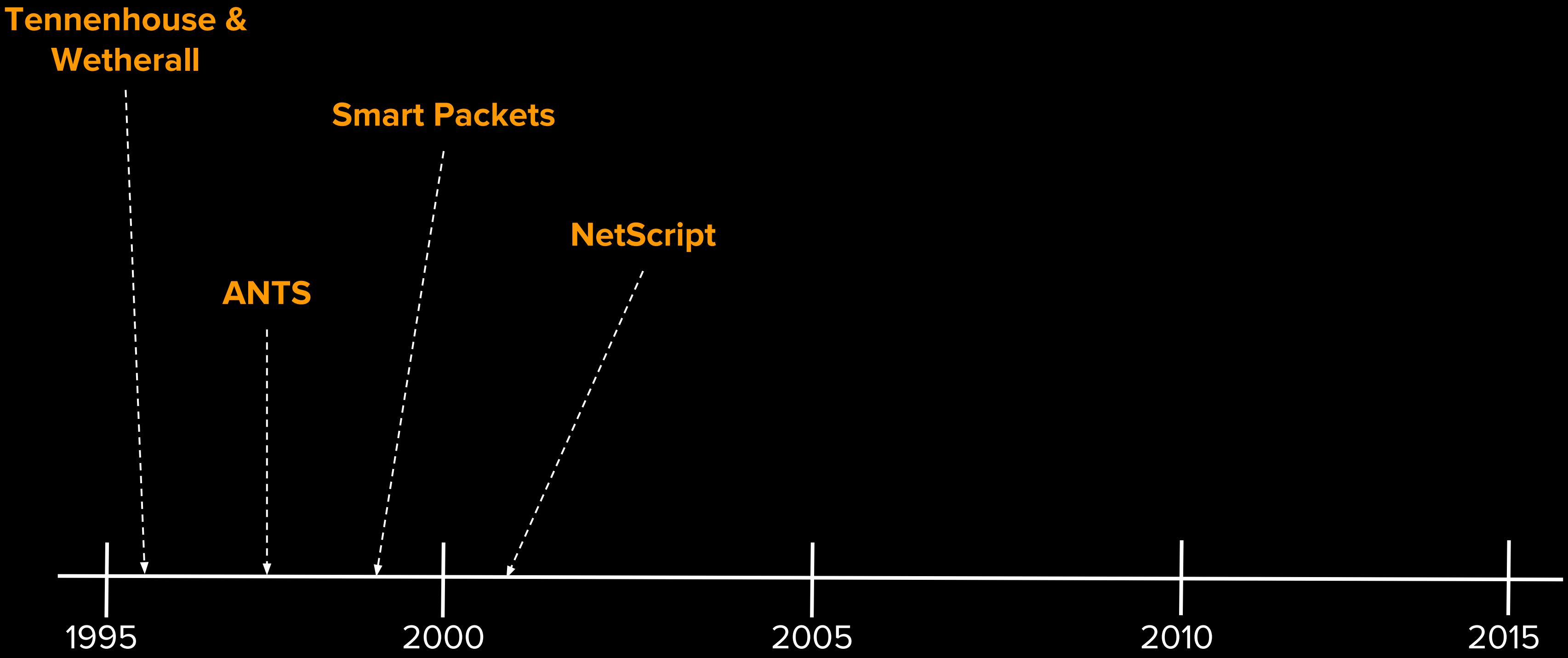
(programmable)

Active Networks

Separating the Data and Control Planes

OpenFlow

SDN Today



Tennenhouse & Wetherall

ANTS

Smart Packets

NetScript

use pulls

technology pushes

network ossification

desire for unified middlebox interface

use pulls

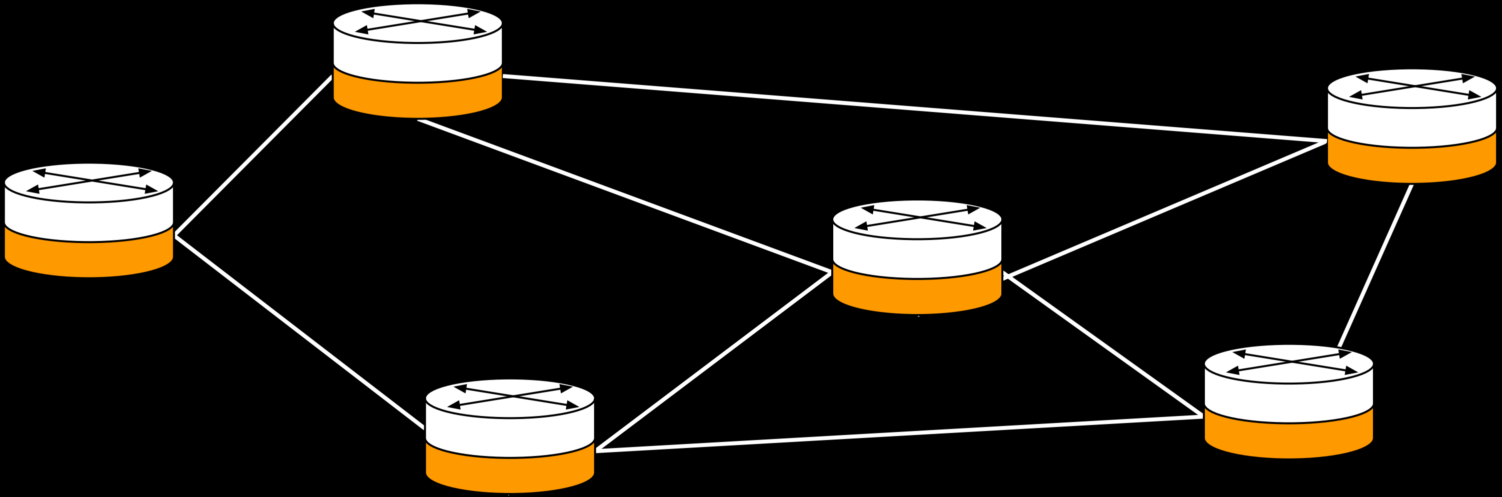
technology pushes

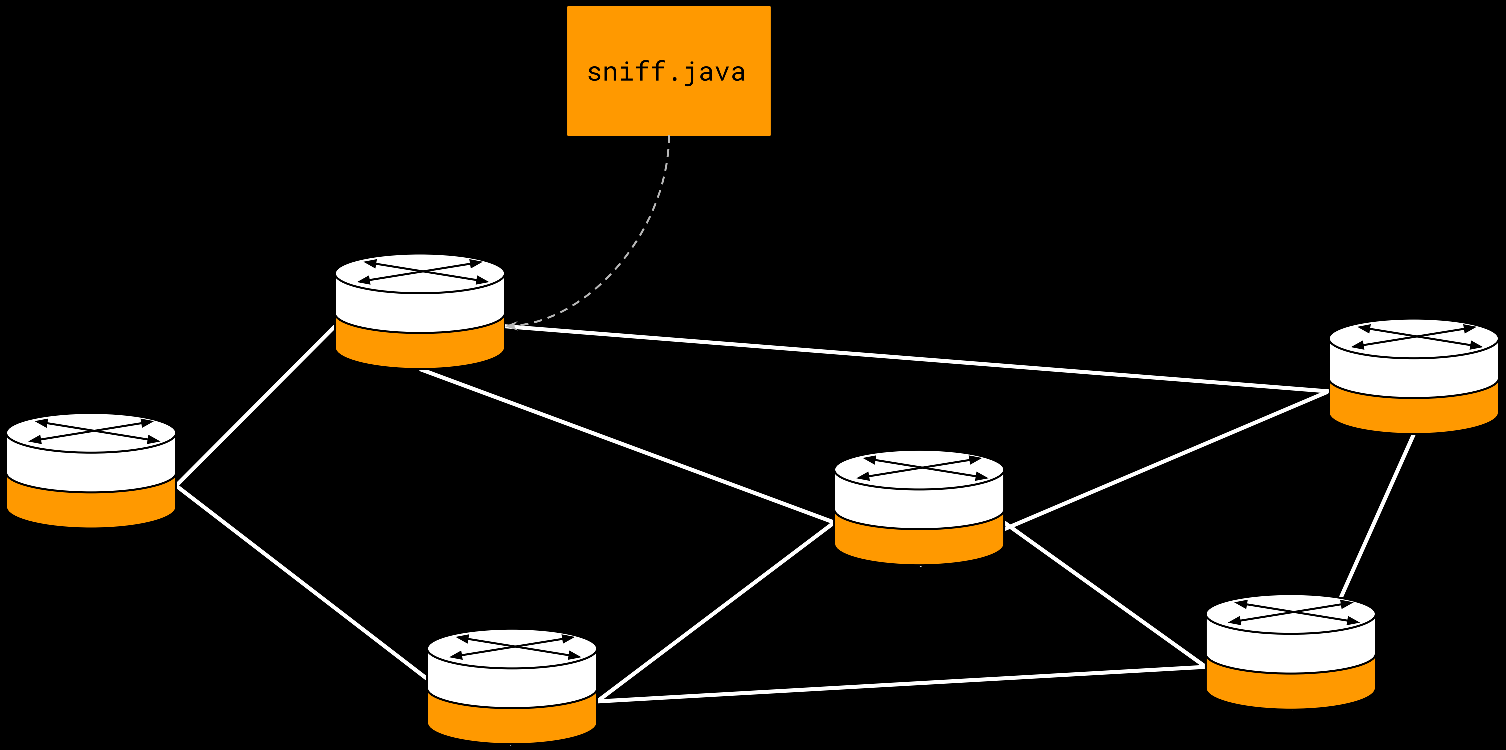
lower compute costs

advances in
programming languages

DARPA Active Networks

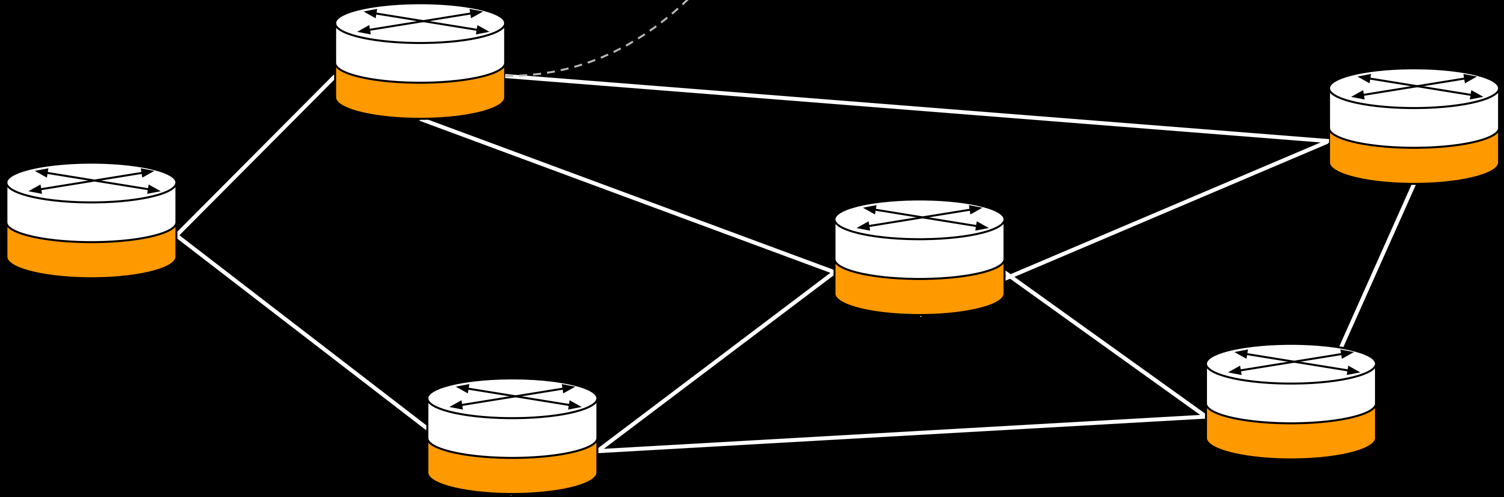
programmable switches



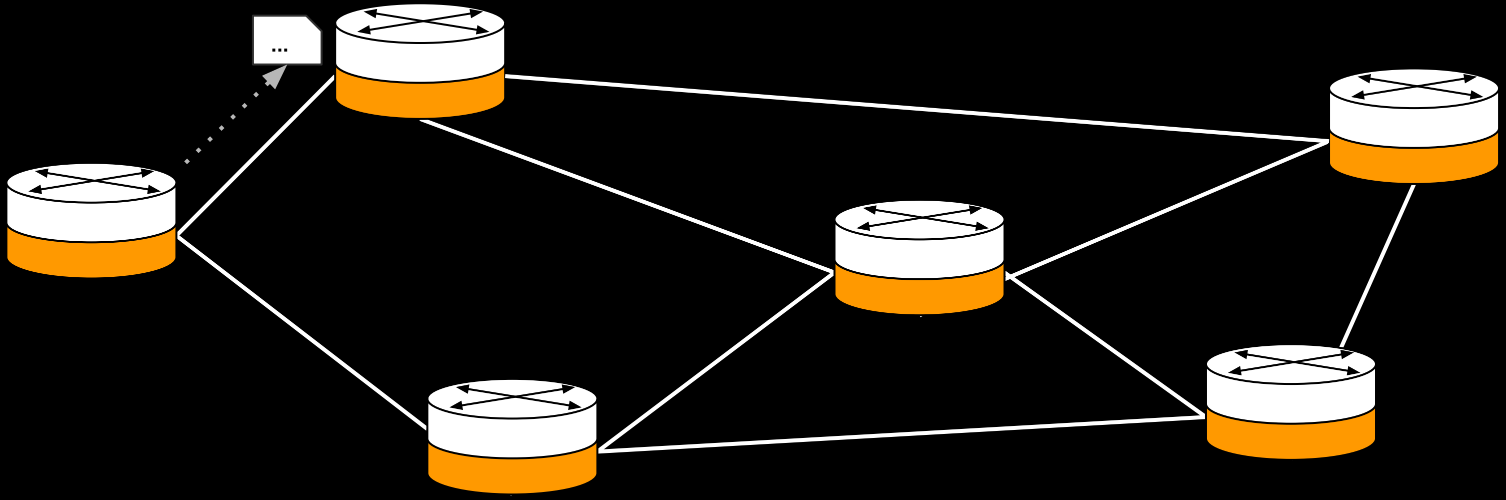

sniff.java

with `sniff.java` installed,
switch maintains table of
packet counts by source IP

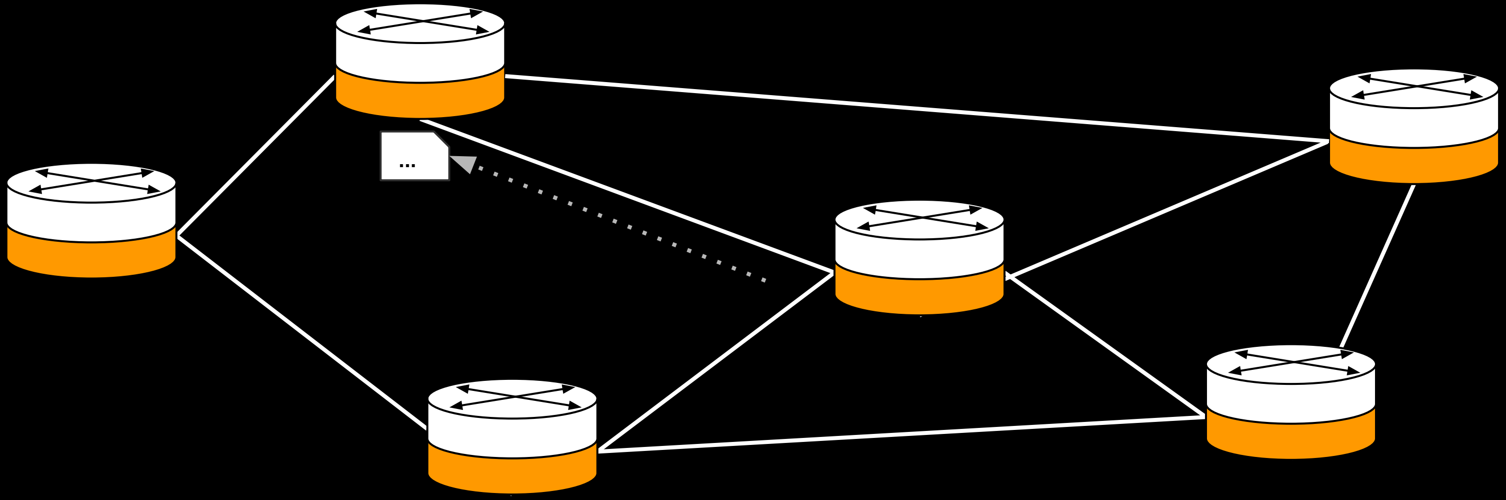
srcIP	count



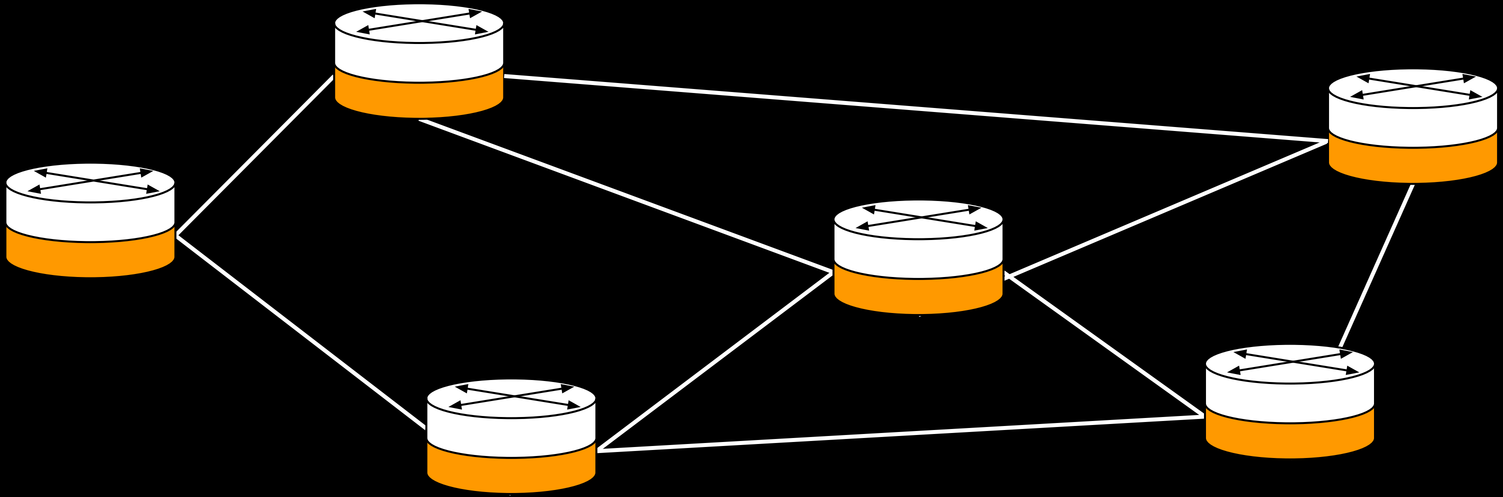
srcIP	count
10.0.0.1	1



srcIP	count
10.0.0.1	1
10.0.0.2	1



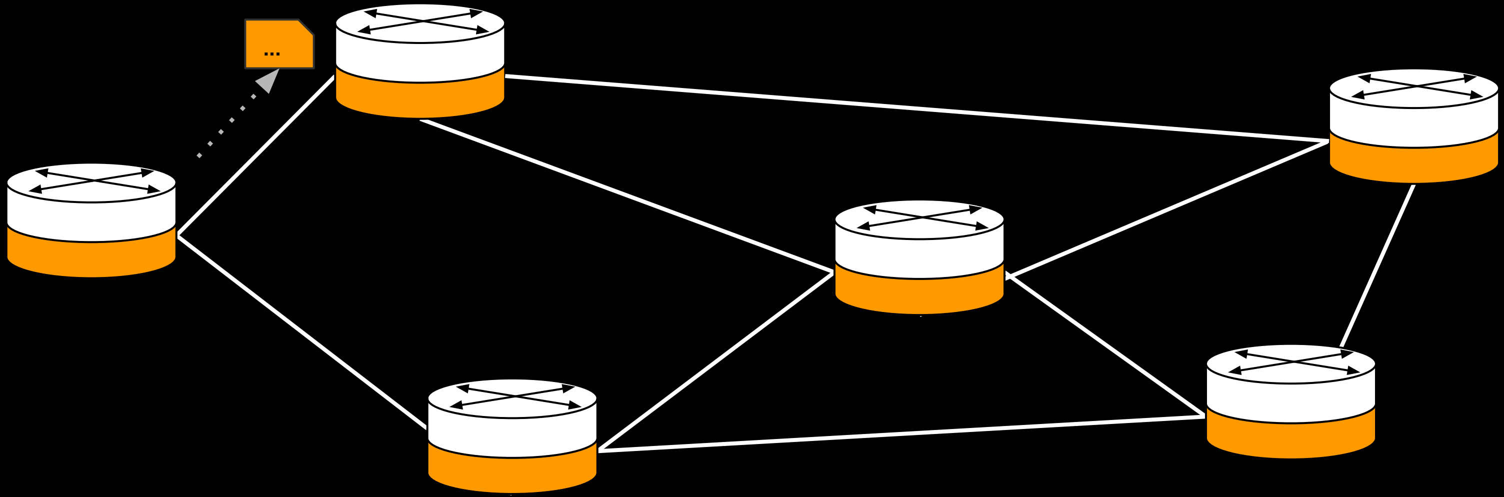
capsules

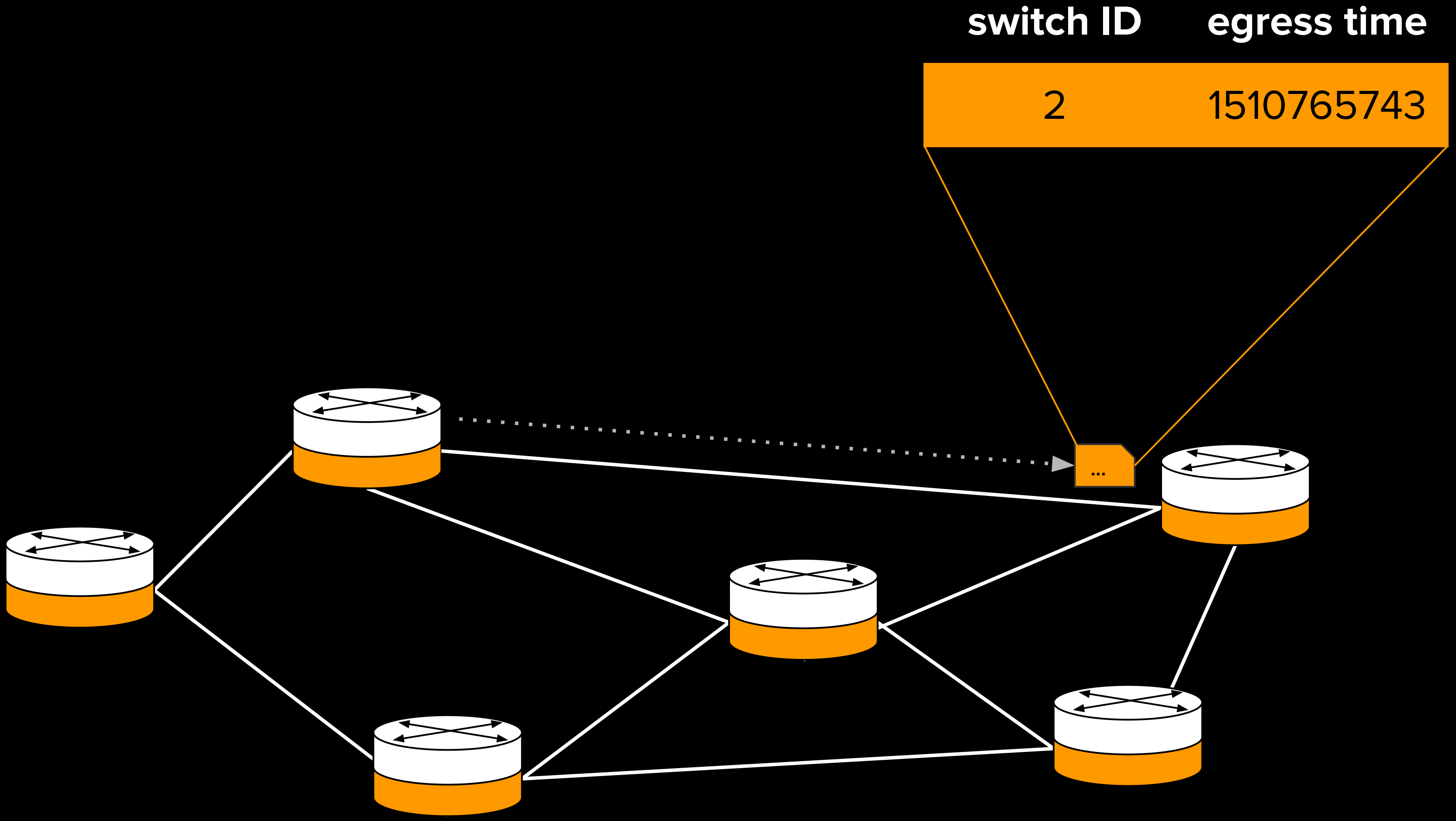


packet contains instructions to push switch info at every hop

Tiny Packet Programs
Jeyakumar et al, 2014

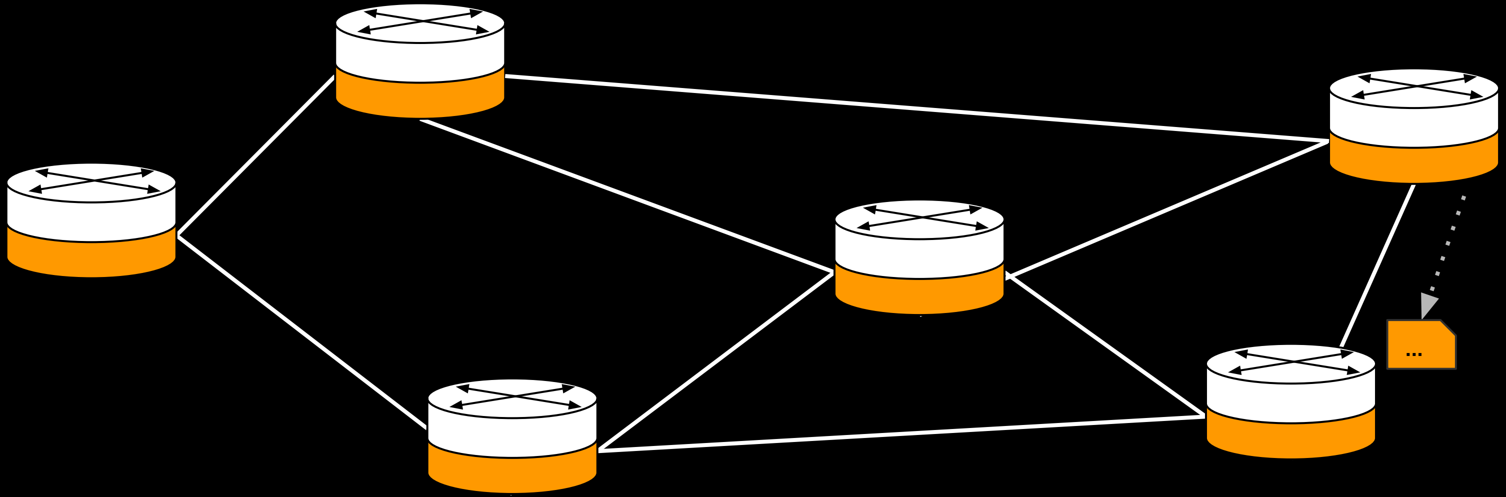
In-Band Network Telemetry
Kim et al, 2016





switch ID	egress time
2	1510765743

switch ID	egress time
3	1510765745
2	1510765743



whither active networks?

whither active networks?

performance and security concerns

no "killer app"

no practical deployment plan

> *"The misconception that packets would necessarily carry Java code written by end users made it possible to dismiss active network research as too far removed from real networks and inherently unsafe."*
>
> "The Road to SDN," Feamster et al 2014

whither active networks?

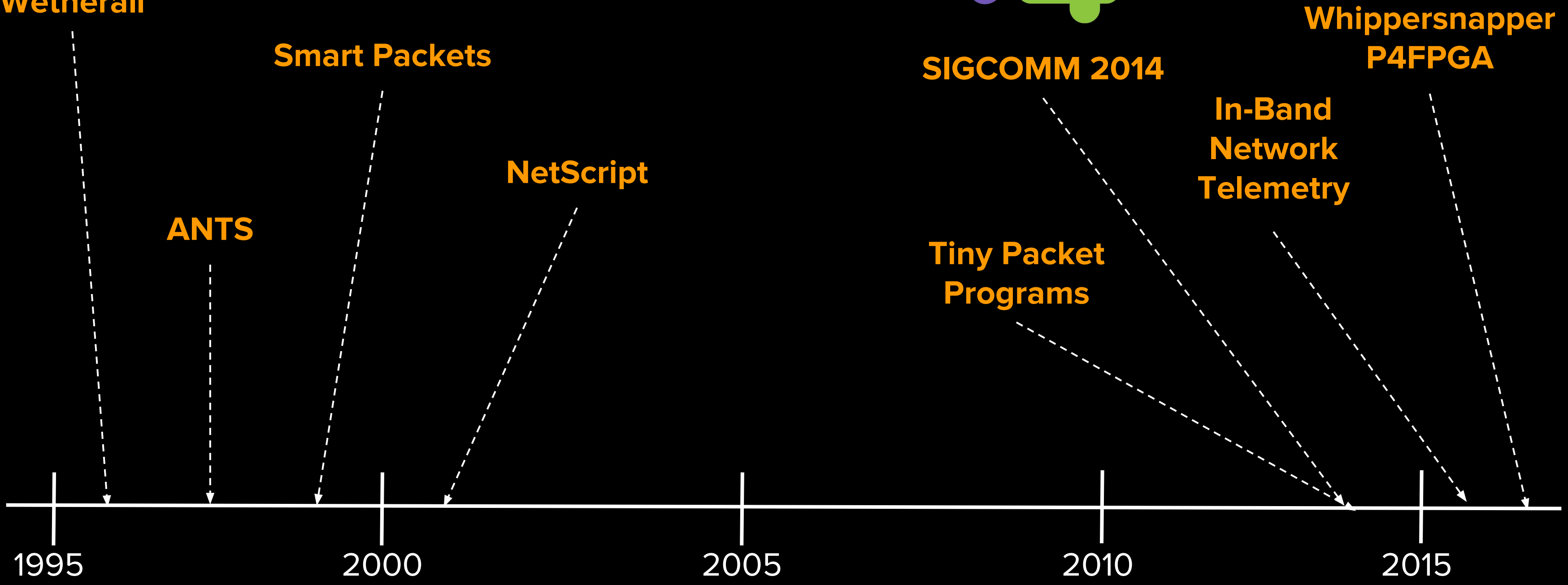
performance and security concerns

no “killer app”

no practical deployment plan

“The misconception that packets would necessarily carry Java code written by end users made it possible to dismiss active network research as too far removed from real networks and inherently unsafe.”

“The Road to SDN,” Feamster et al 2014



Tennenhouse & Wetherall

ANTS

Smart Packets

NetScript

Tiny Packet Programs

SIGCOMM 2014

In-Band Network Telemetry

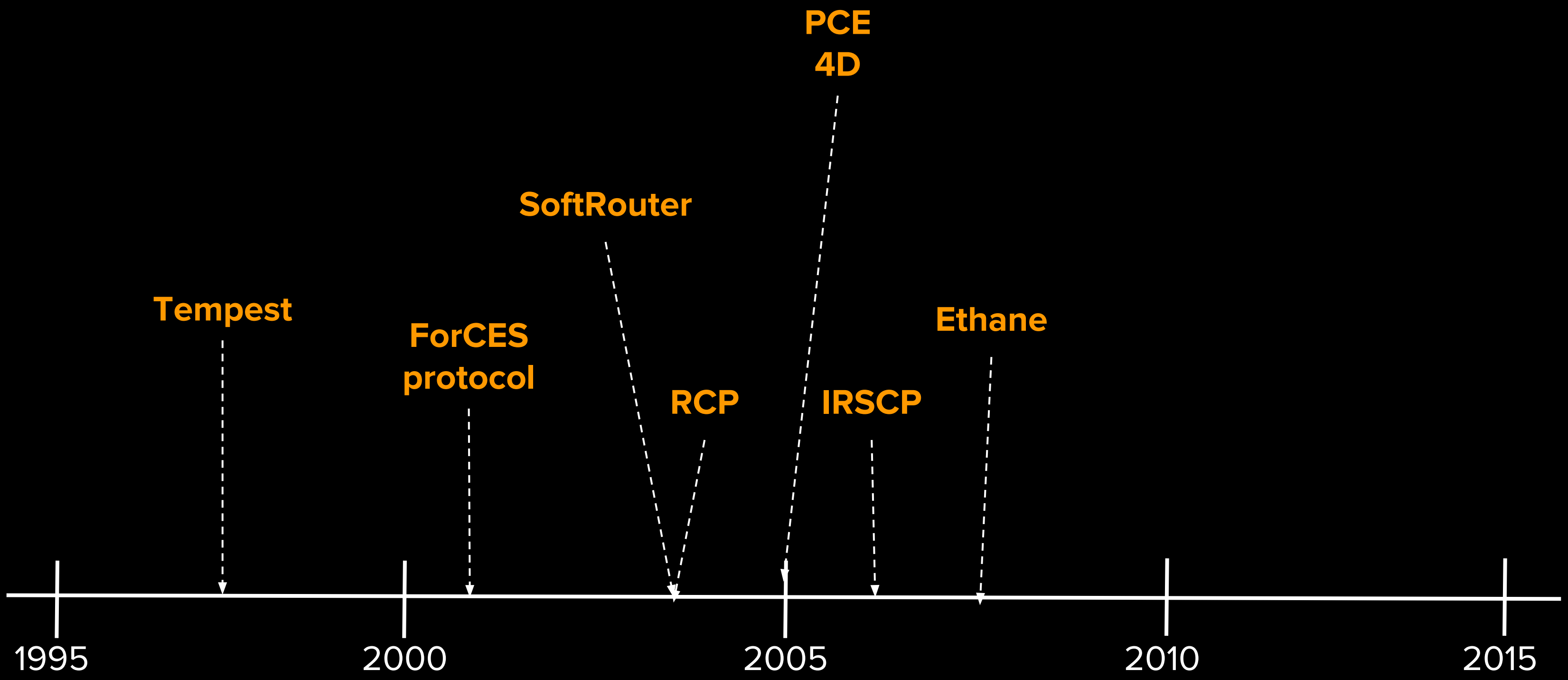
Whippersnapper P4FPGA

Active Networks

Separating the Data and Control Planes

OpenFlow

SDN Today



PCE
4D

SoftRouter

Tempest

ForCES protocol

RCP

IRSCP

Ethane

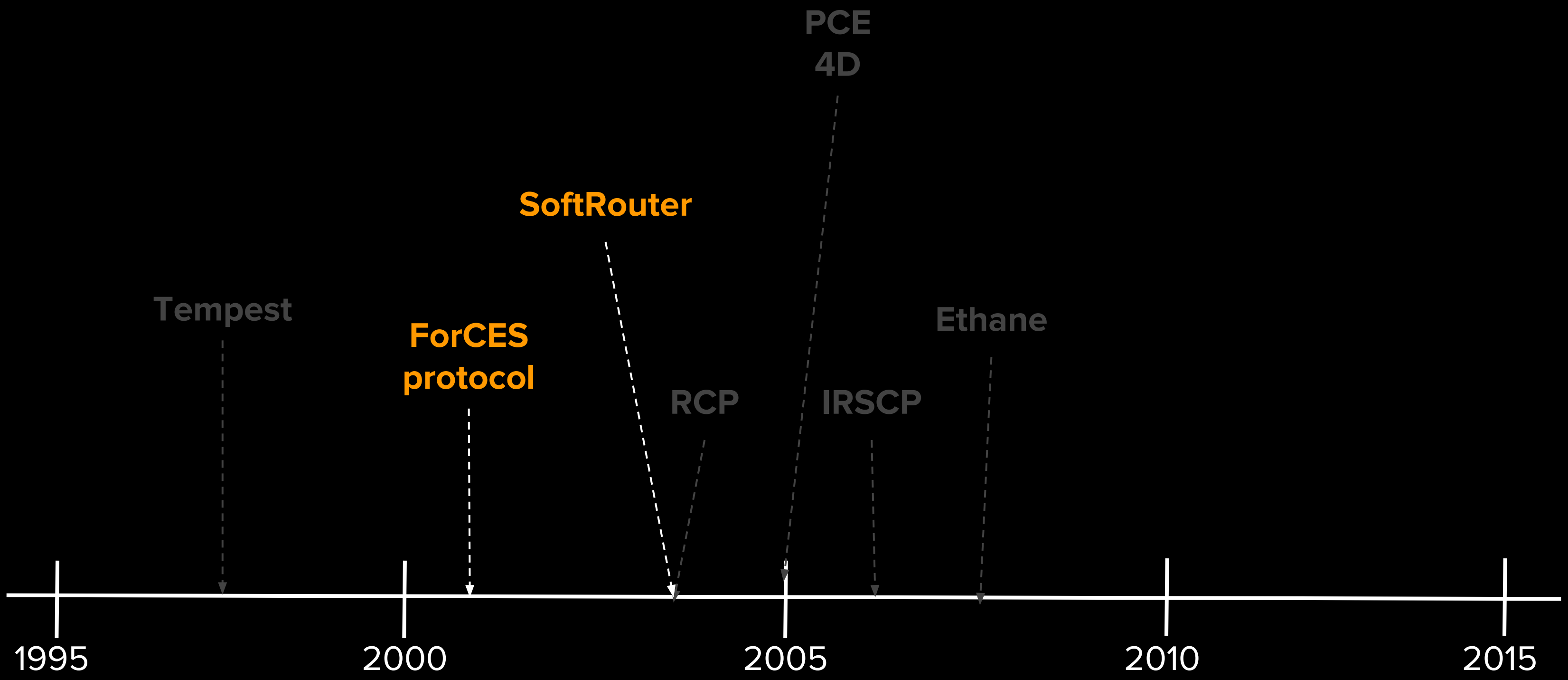
1995

2000

2005

2010

2015


PCE
4D
SoftRouter
Tempest
ForCES
protocol
Ethane
RCP
IRSCP
1995
2000
2005
2010
2015

use pulls

technology pushes

burgeoning network speeds

insufficient network reliability

specialized services (VPNs)

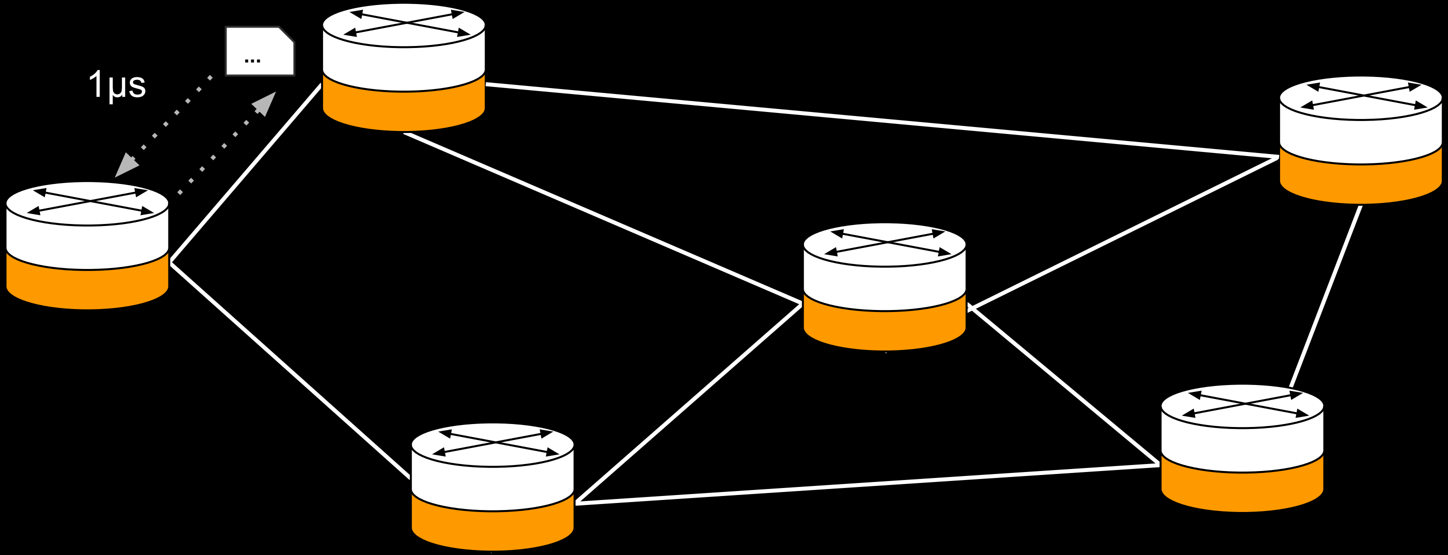
use pulls

technology pushes

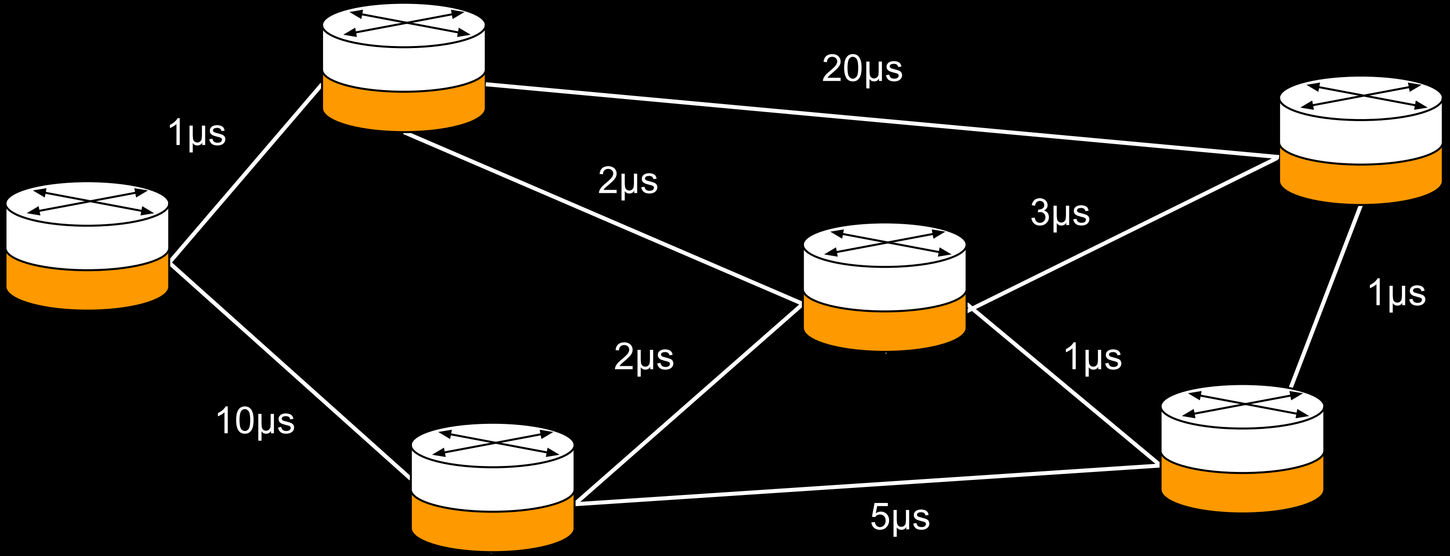
open interface between control and data planes

logically centralized control

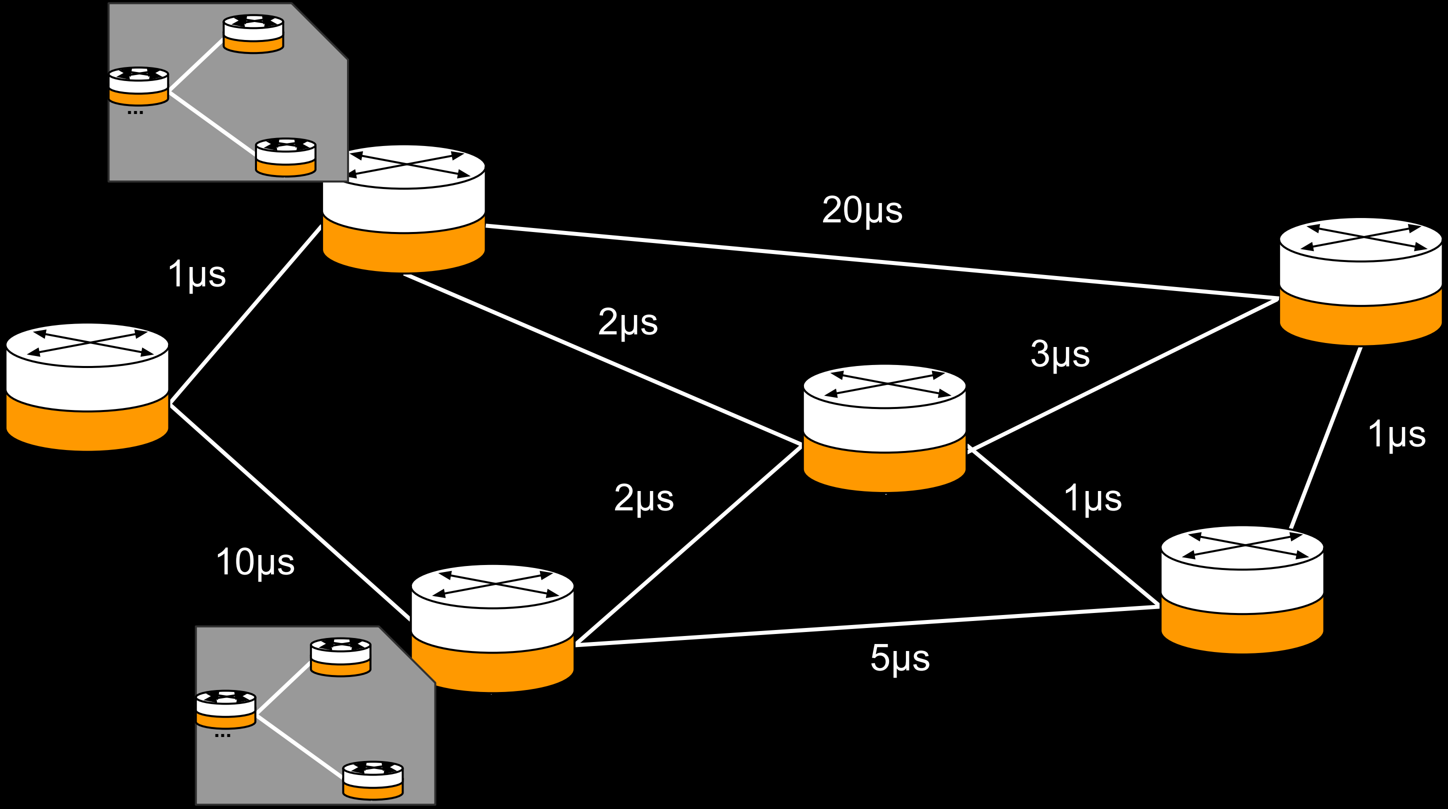
Open Shortest Path First (OSPF)



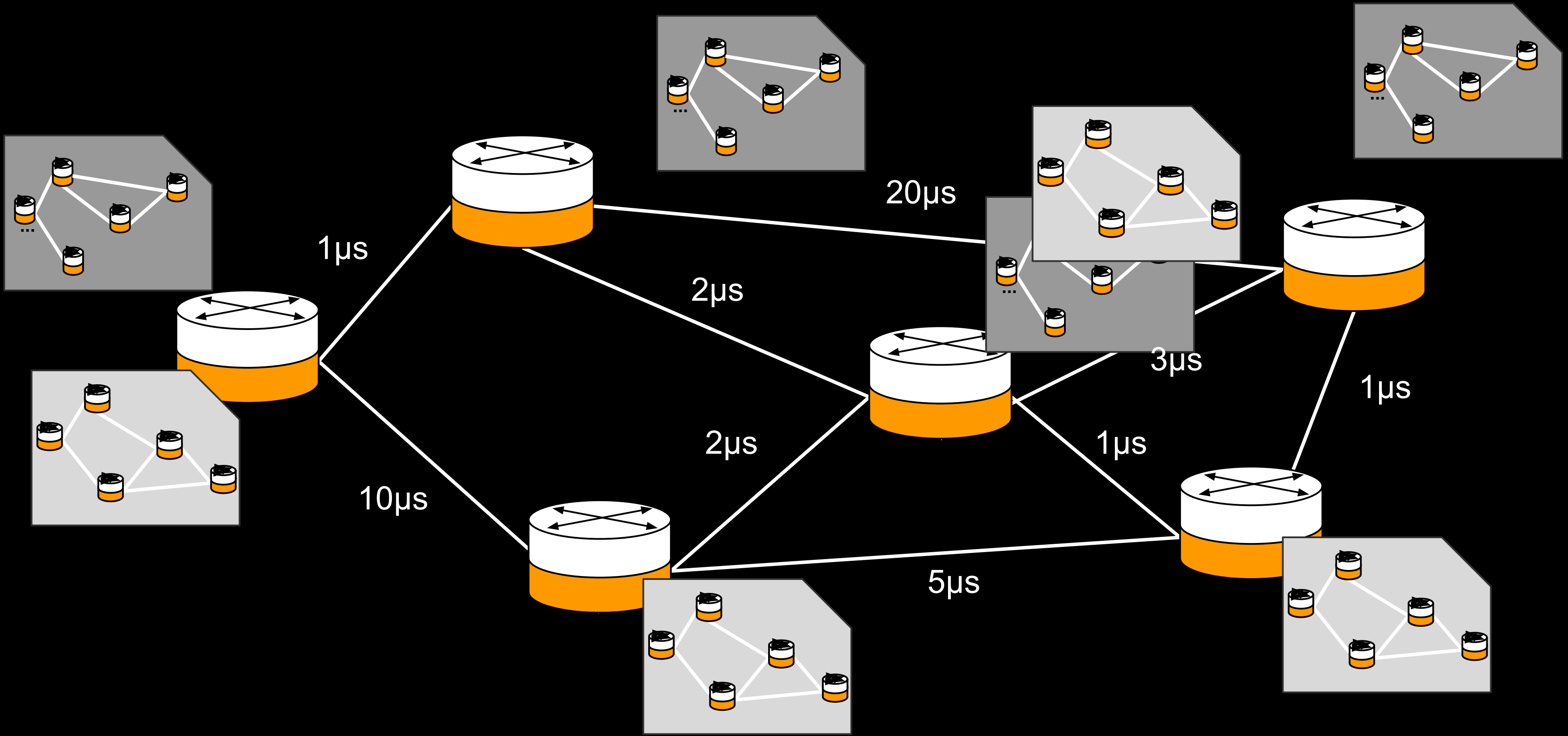
OSPF



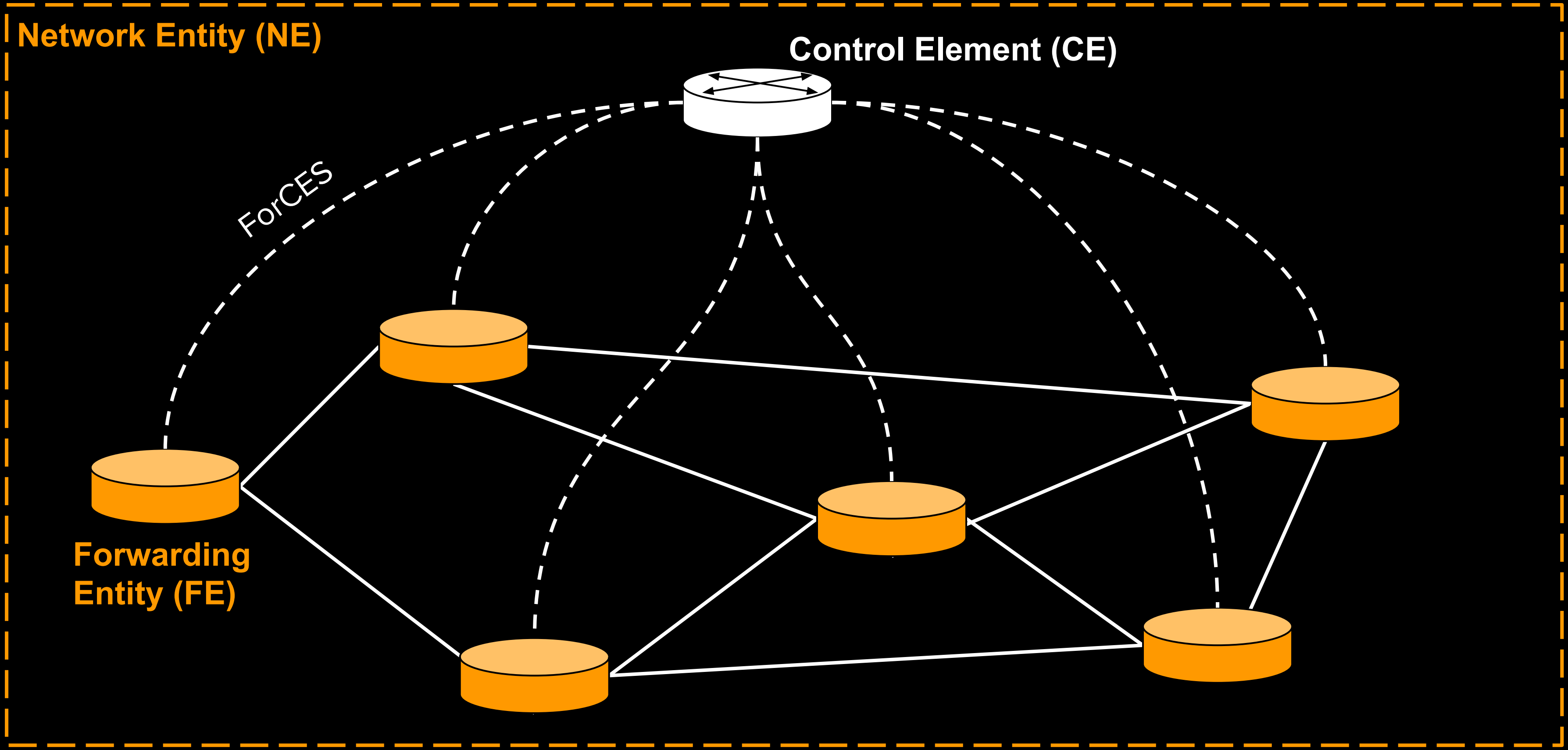
OSPF



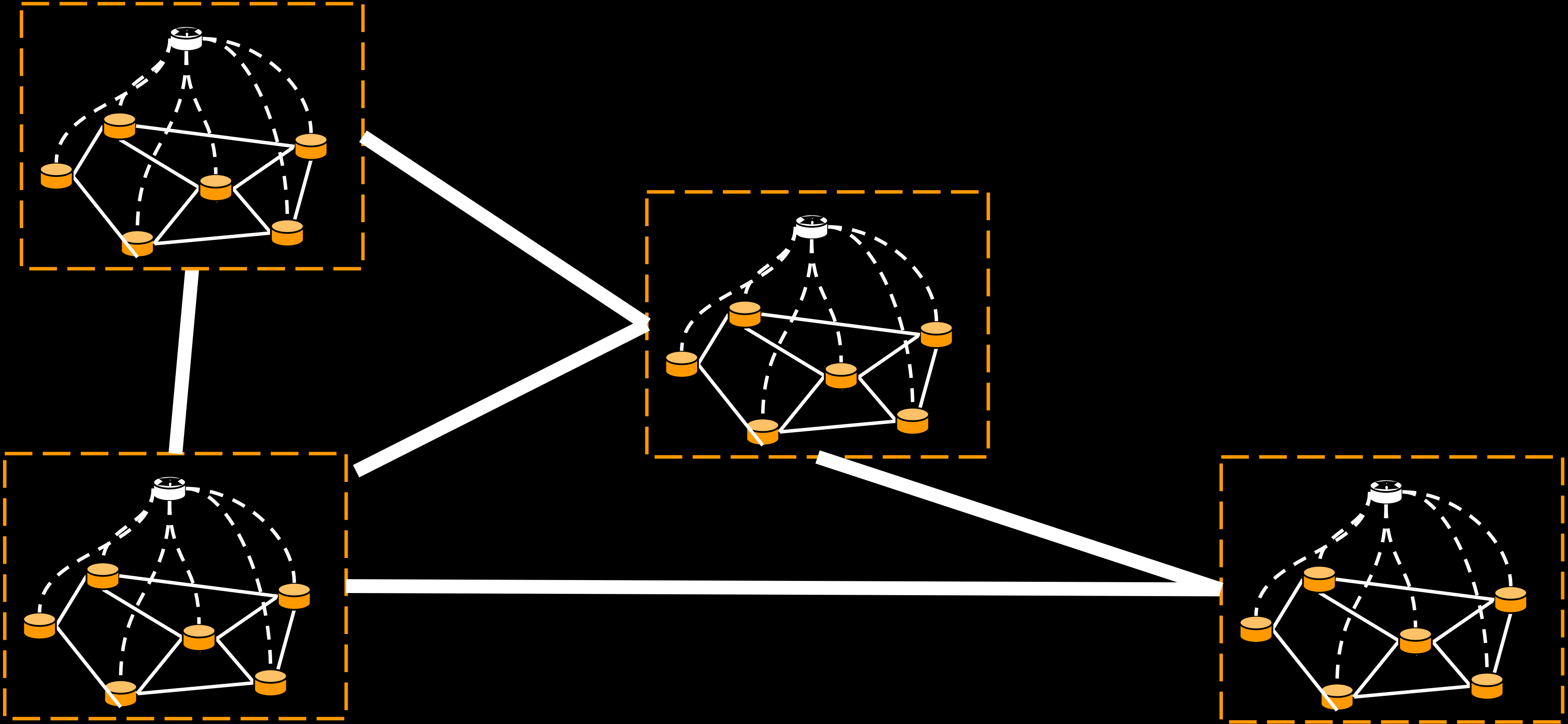
OSPF



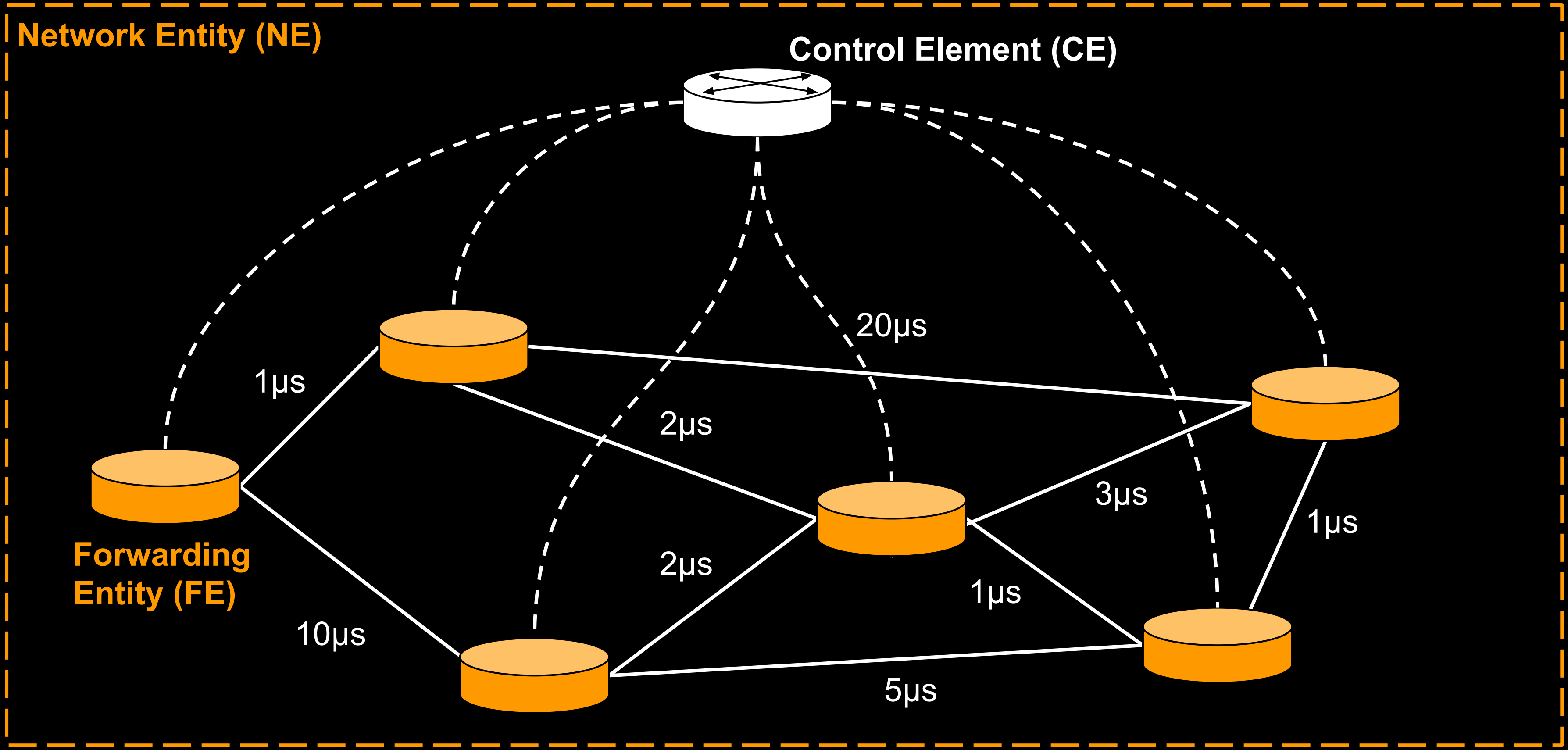
SoftRouter


Network Entity (NE)
Control Element (CE)
ForCES
Forwarding Entity (FE)

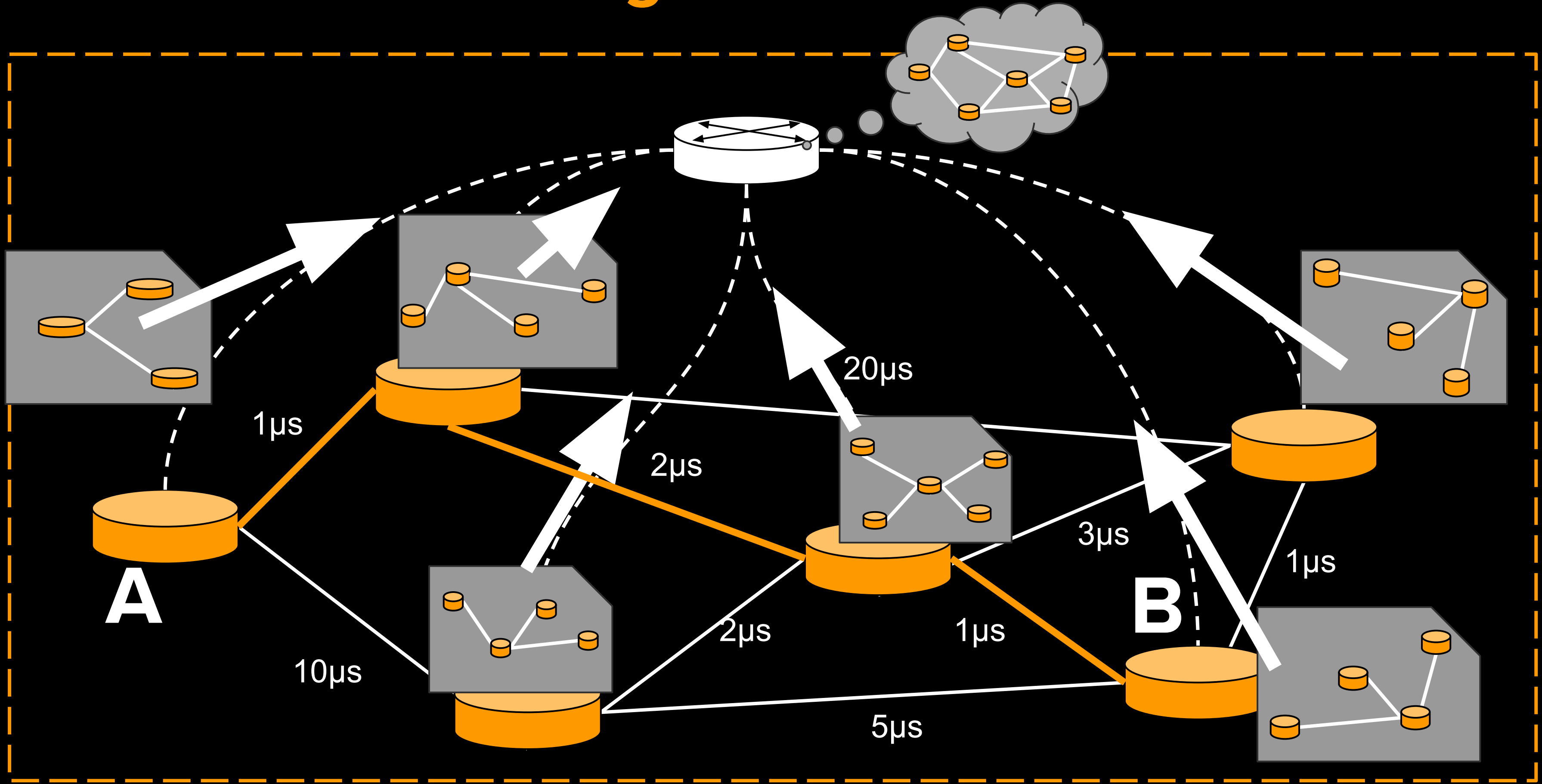
SoftRouter



SoftRouter


Network Entity (NE)
Control Element (CE)
20μs
1μs
2μs
3μs
1μs
2μs
1μs
10μs
5μs
Forwarding Entity (FE)

Shortest Path Routing



whither SoftRouter (and others)?

vendors didn’t adopt ForCES (and others)

not general enough

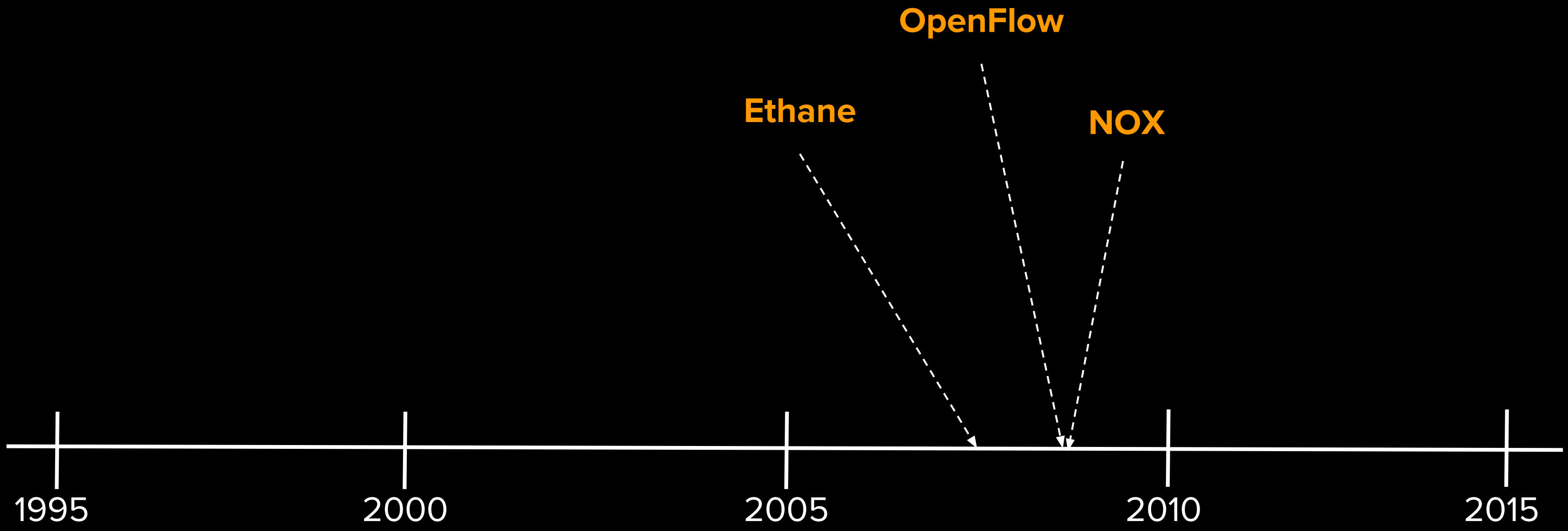
no practical deployment plan

Active Networks

Separating the Data and Control Planes

OpenFlow

SDN Today


OpenFlow
Ethane
NOX
1995
2000
2005
2010
2015

OpenFlow: enabling innovation in campus networks

Nick McKeown, Tom Anderson, Hari Balakrishnan,
Guru Parulkar, Larry Peterson,
Jennifer Rexford, Scott Shenker, Jonathan Turner

SIGCOMM 2008

Nick McKeown
Stanford

Jennifer Rexford
Princeton

Scott Shenker
Berkeley

Nick McKeown
Stanford

Jennifer Rexford
Princeton

Scott Shenker
Berkeley

Martín Casado
Stanford

networking research

use pulls

technology pushes

market factors

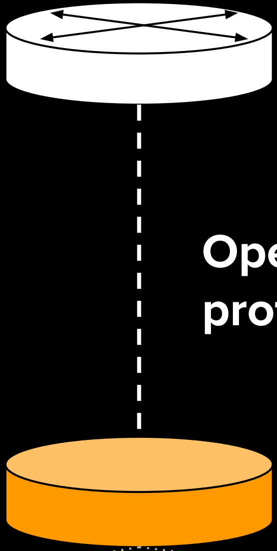
datacenter networks

use pulls

technology pushes

backwards compatible

general packet processing
(more fields to match on)

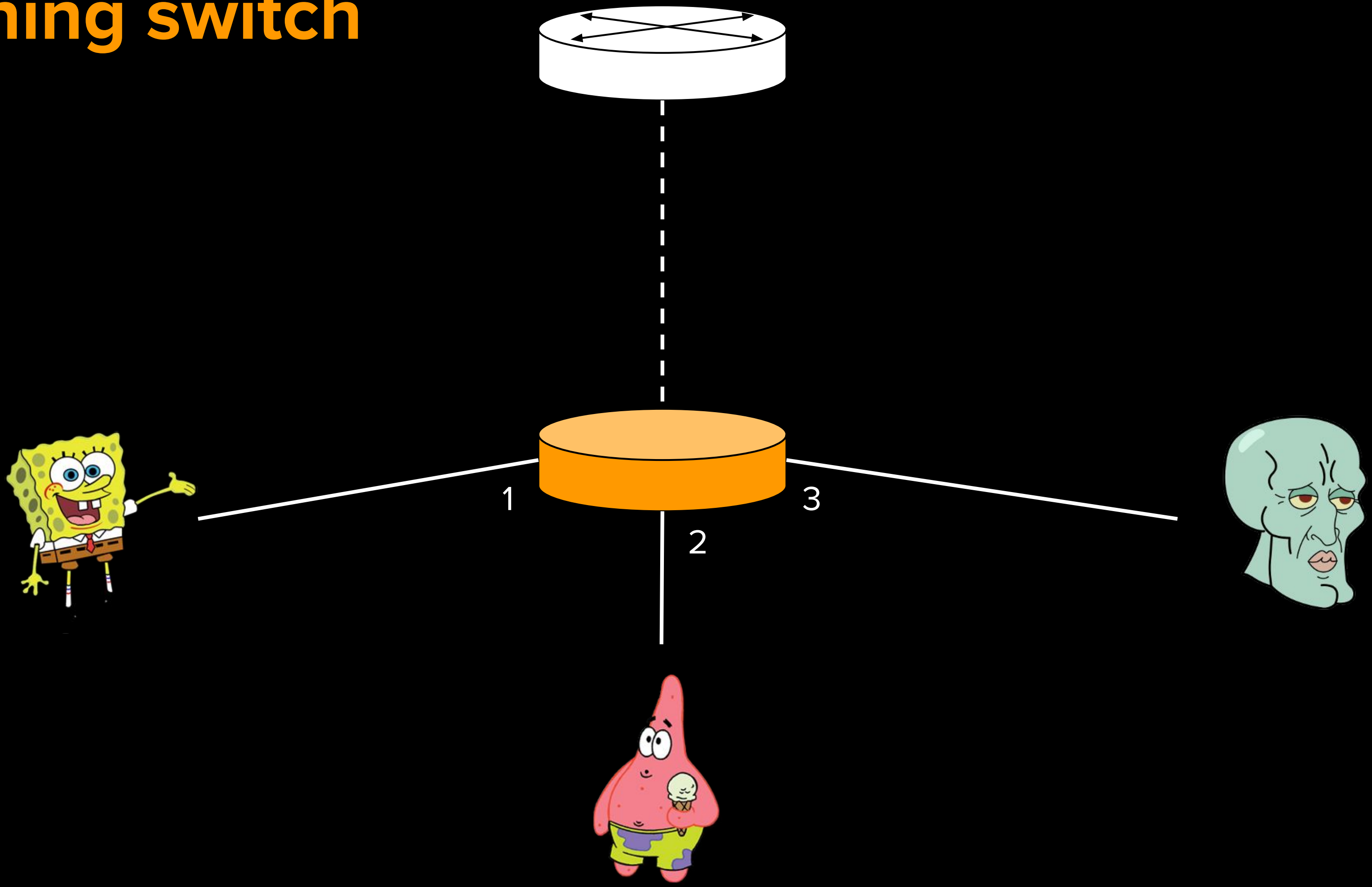


OpenFlow
protocol

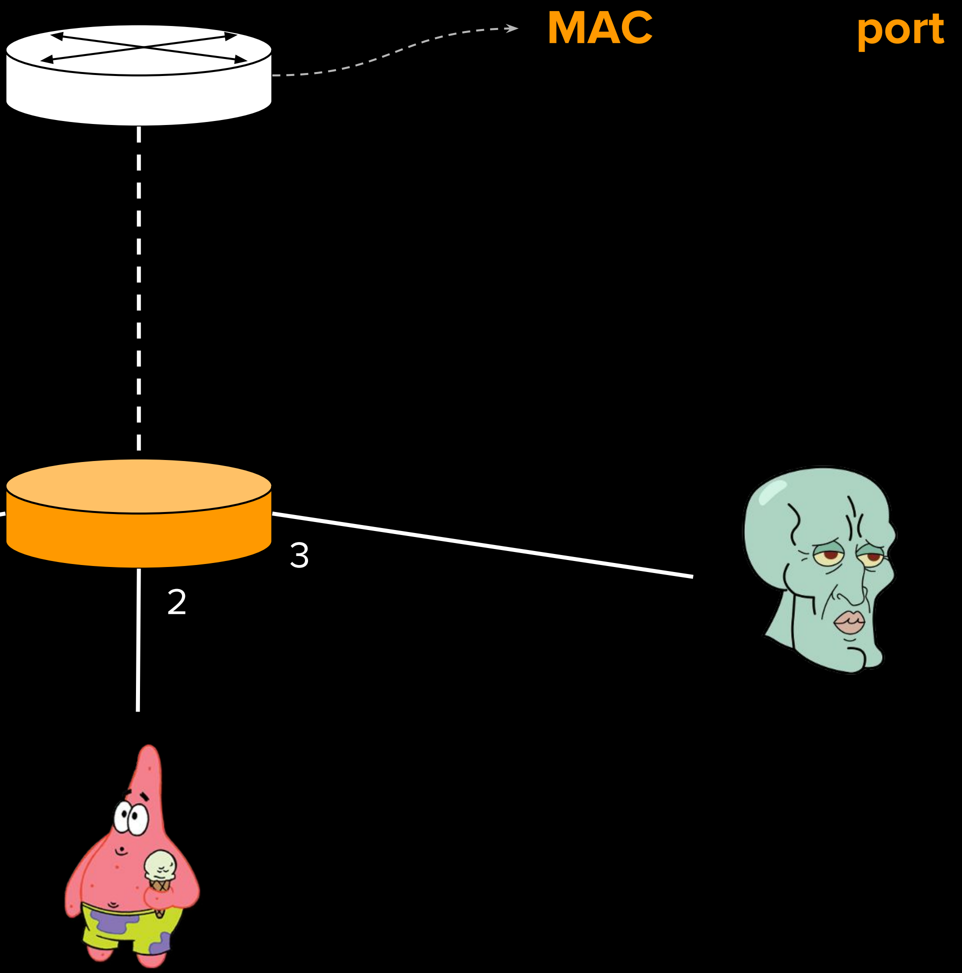
flow
table

fields	counter	action
srcIp=10.0.0.*, ipProto=TCP	10	pt = 2
dstPort=80	0	drop

learning switch

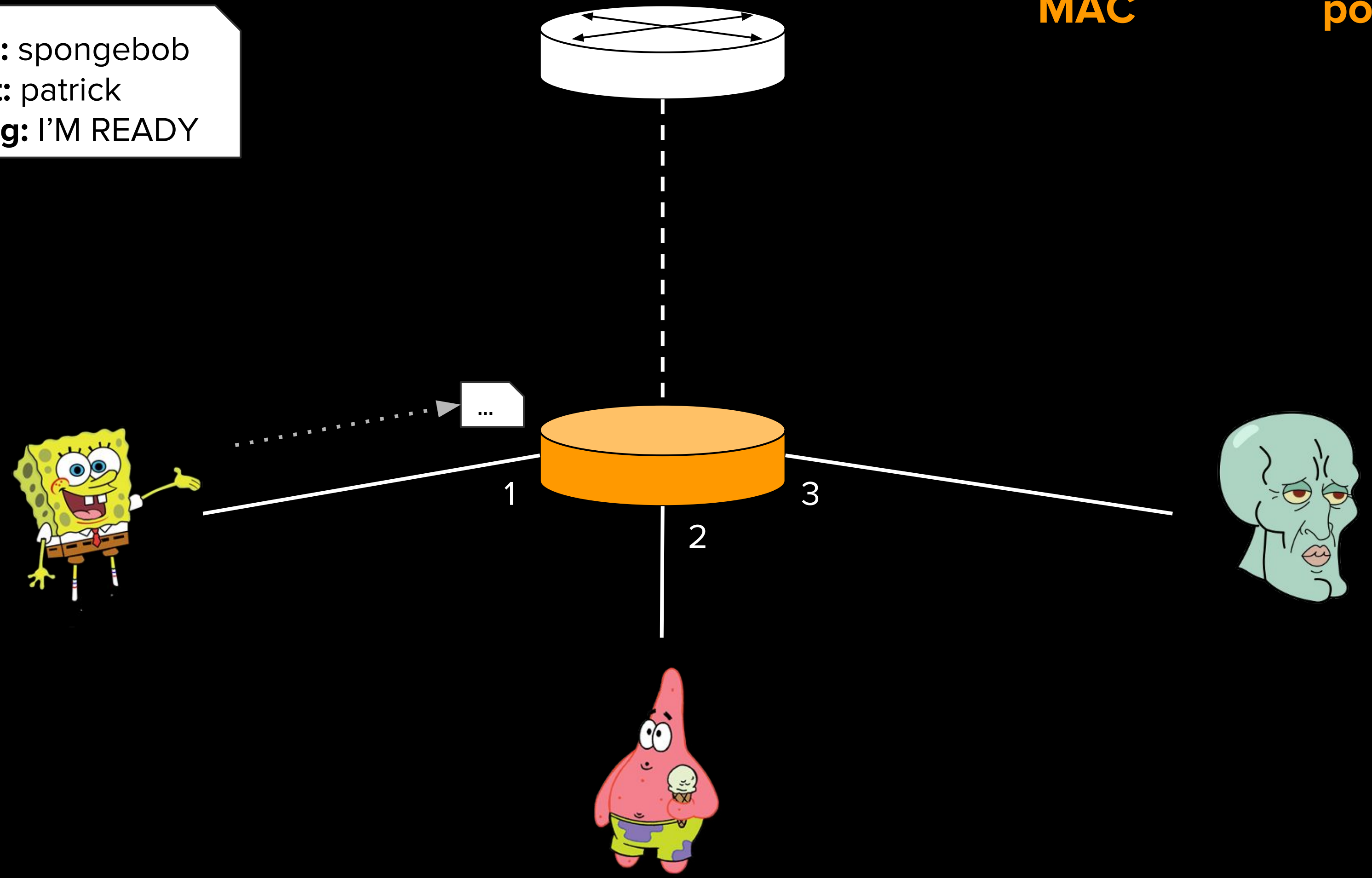


controller maintains hash table of MAC to port number mappings

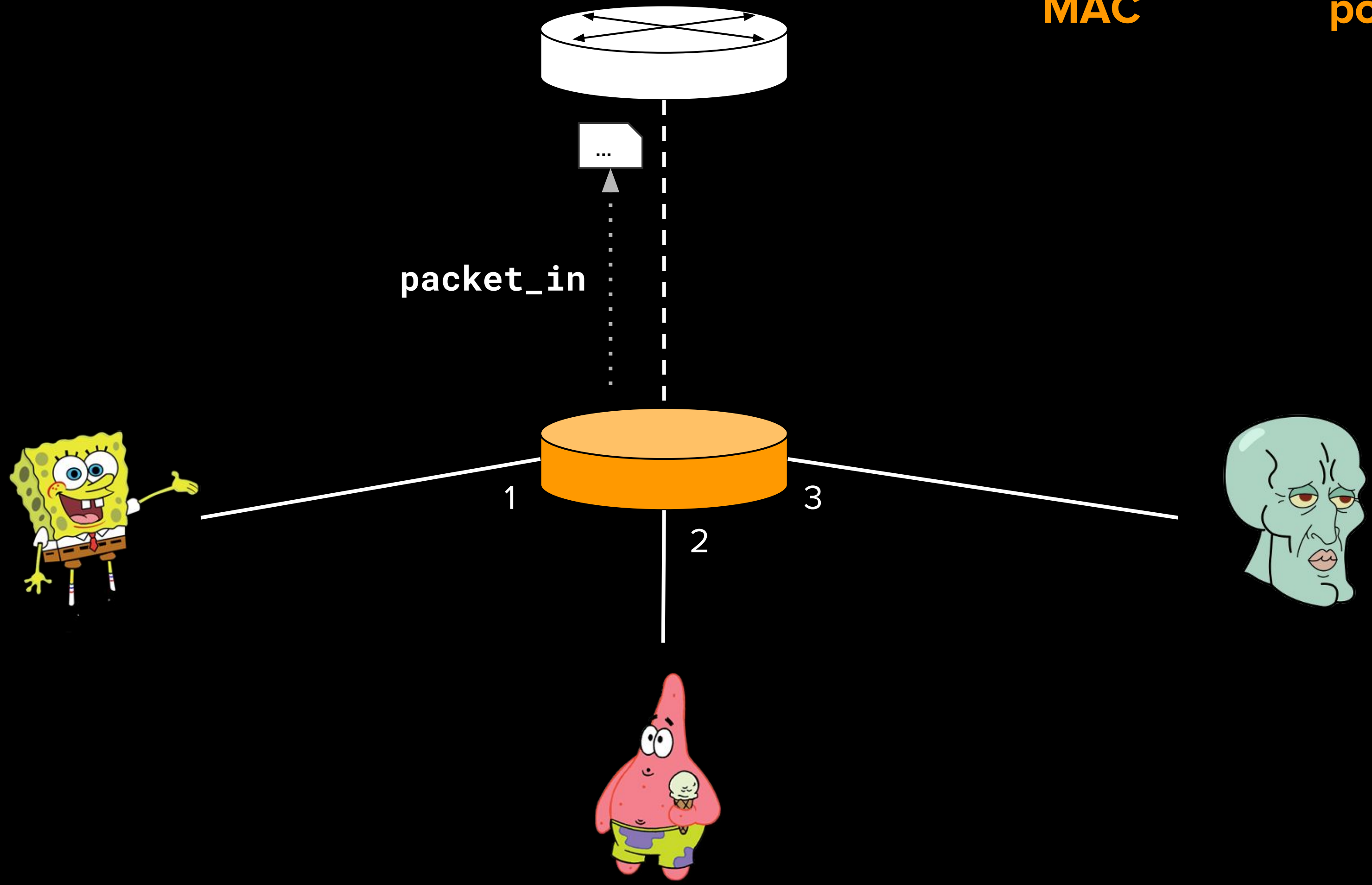


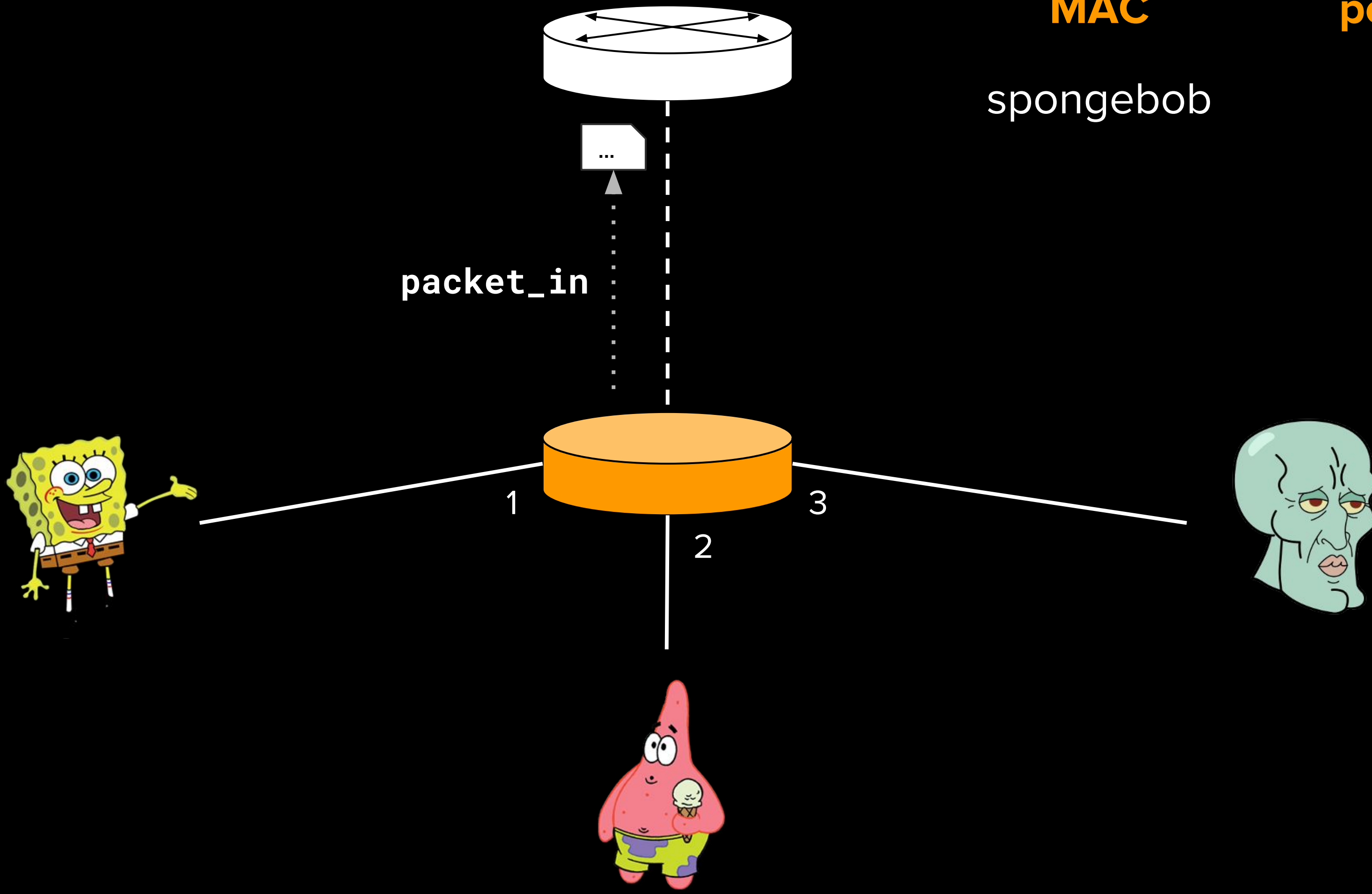
src: spongebob
dst: patrick
msg: I’M READY

MAC	port

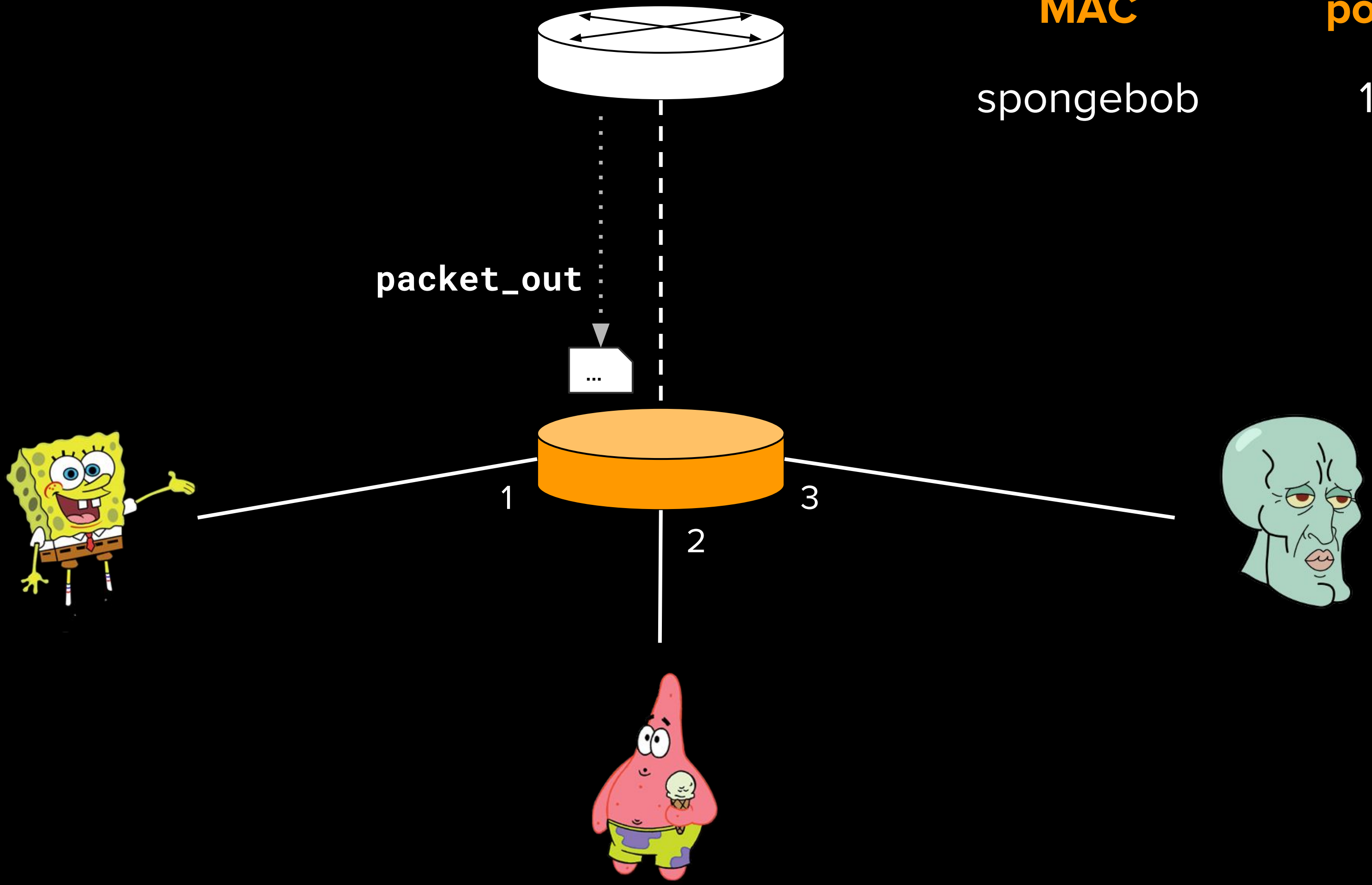


MAC	port

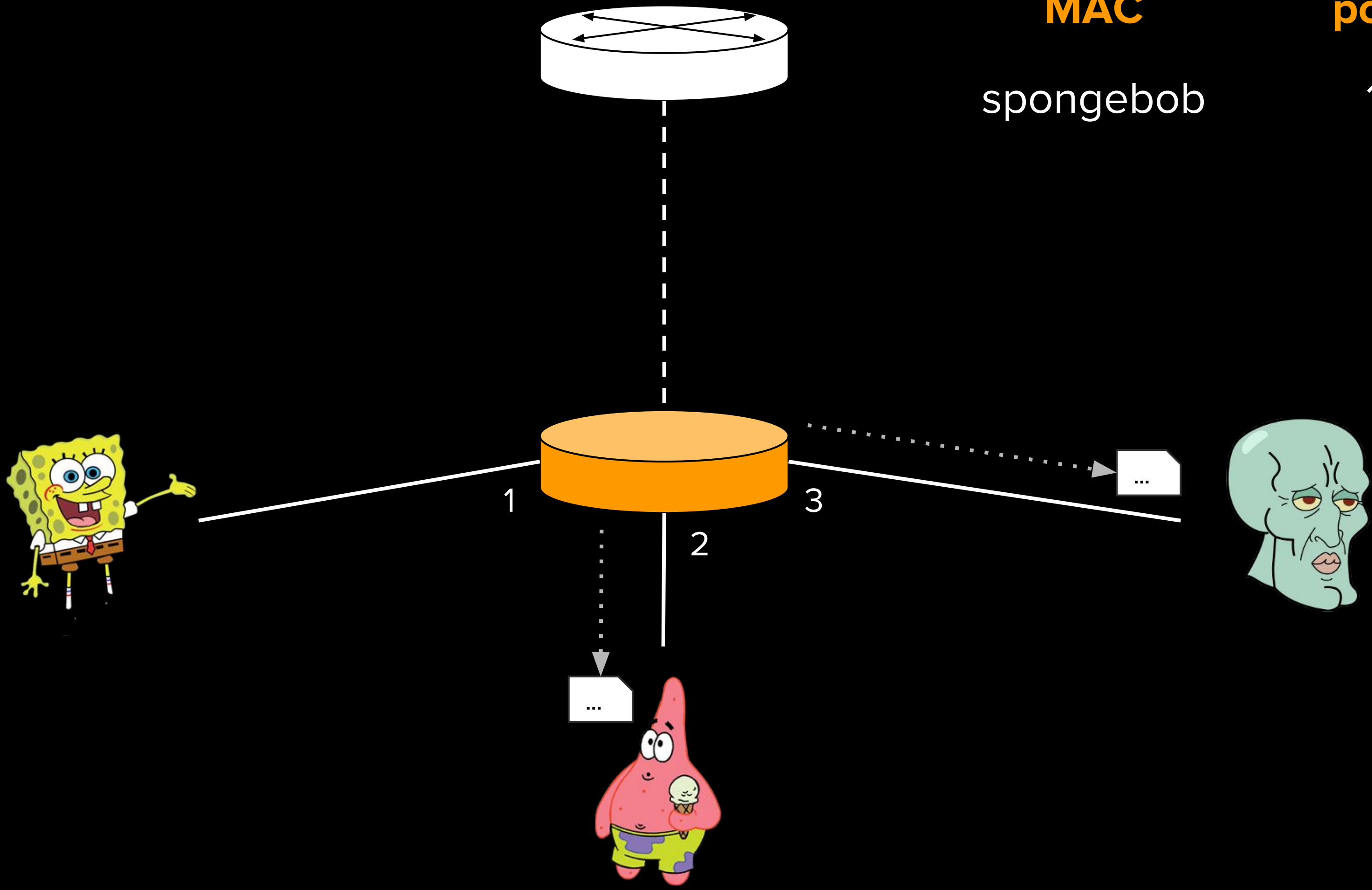




MAC	port
spongebob	1

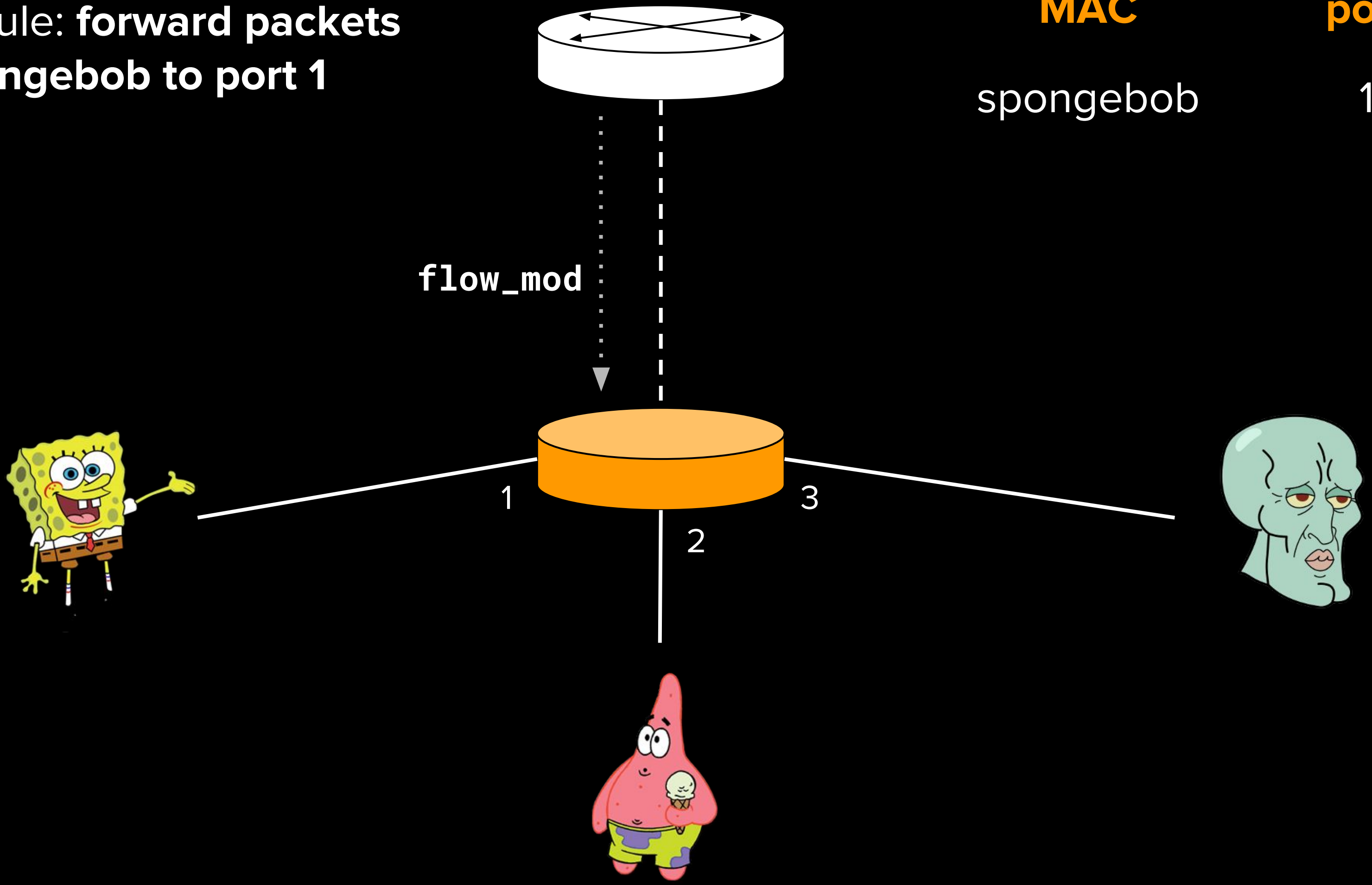


MAC	port
spongebob	1

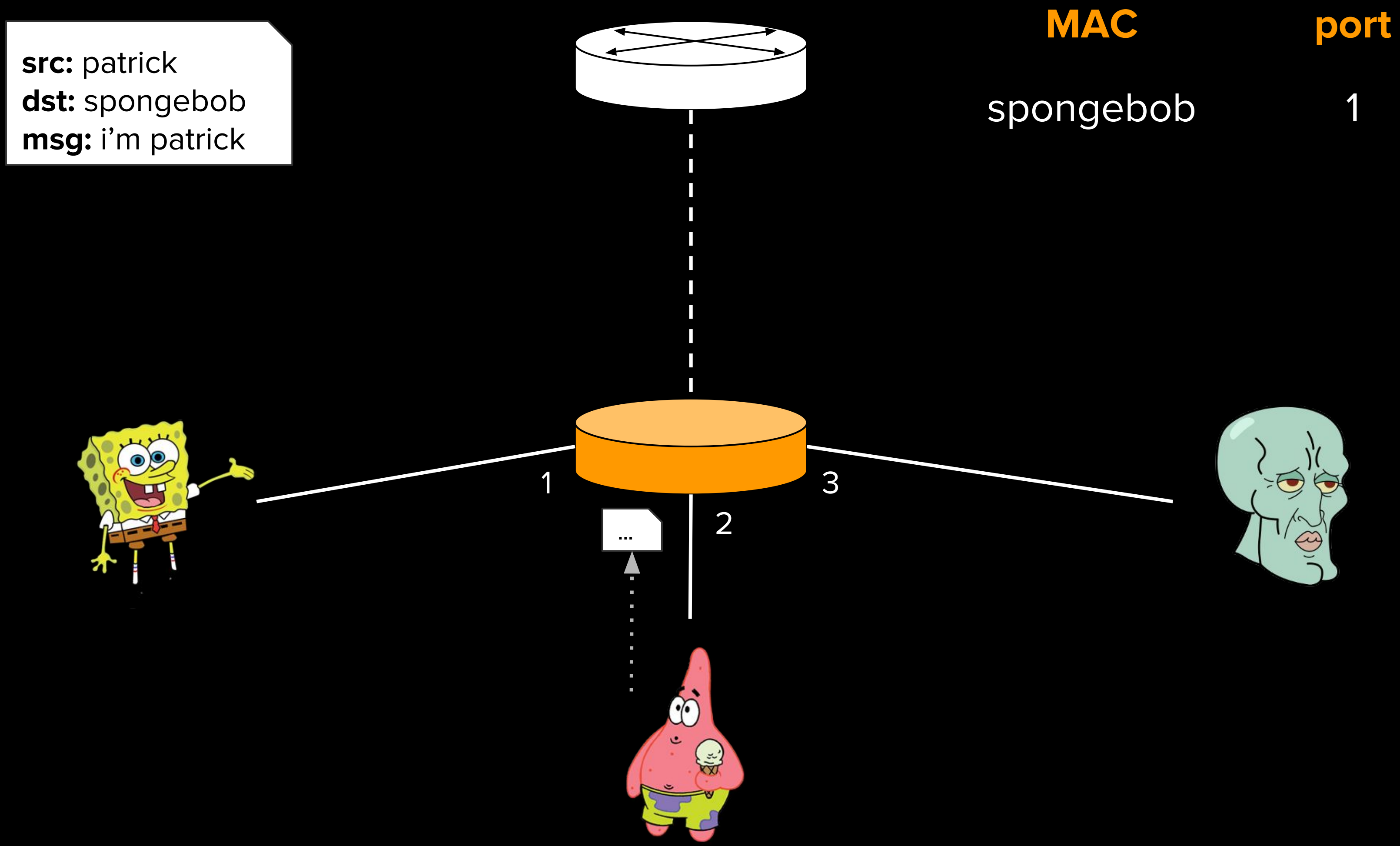


MAC	port
spongebob	1

install rule: **forward packets for spongebob to port 1**



MAC	**port**
spongebob	1

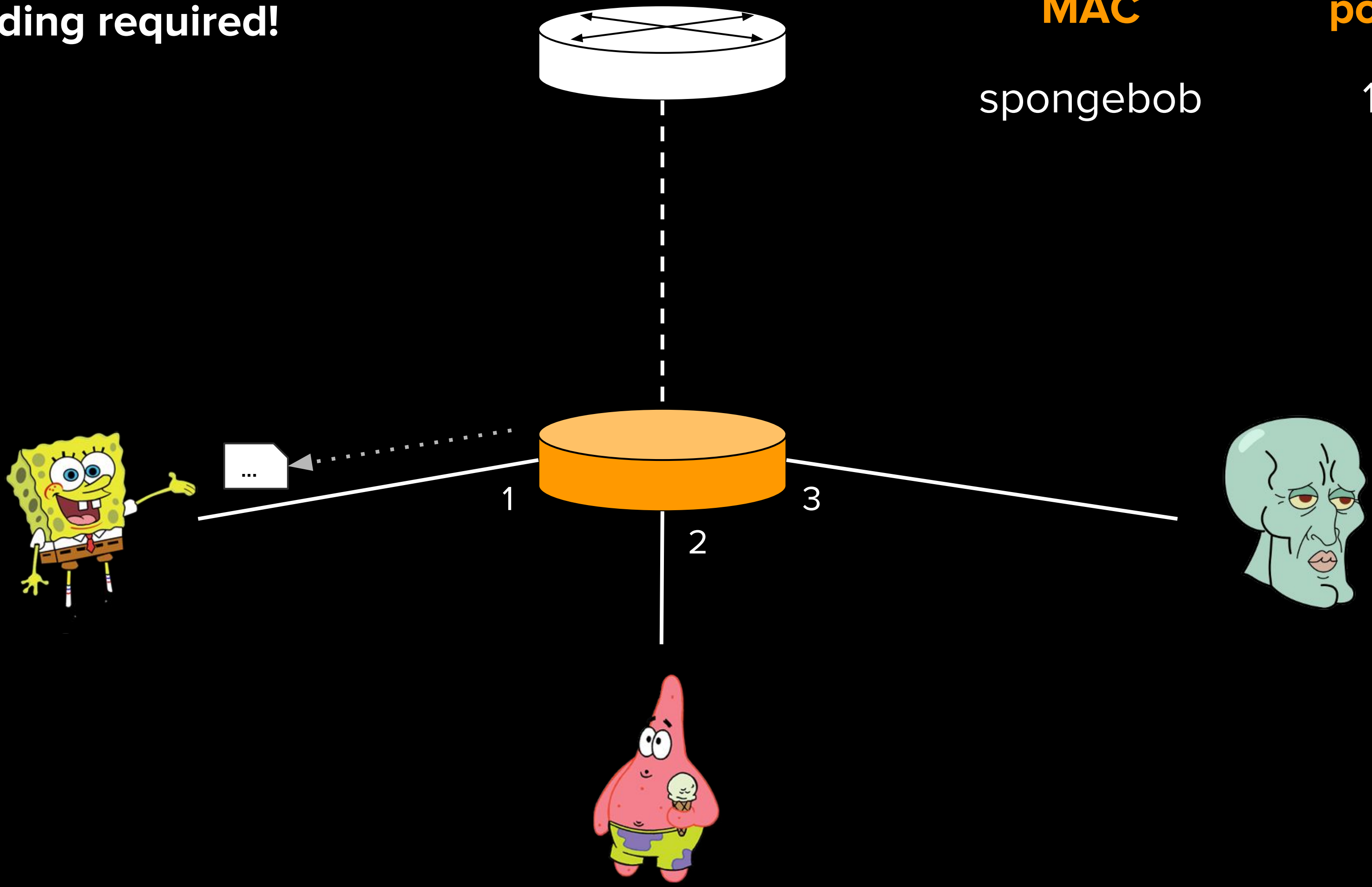


src: patrick
dst: spongebob
msg: i’m patrick

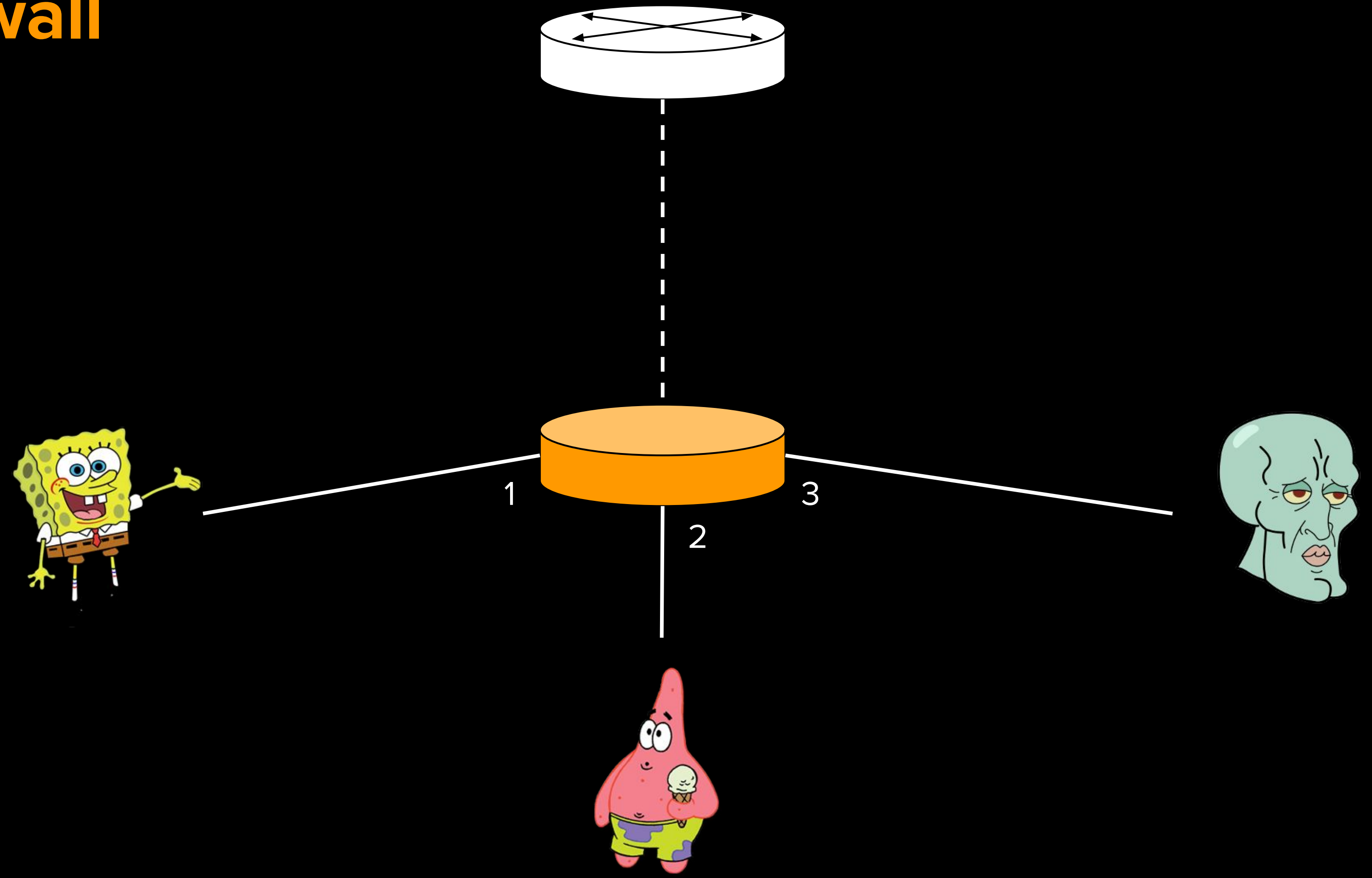
MAC	port
spongebob	1

no flooding required!

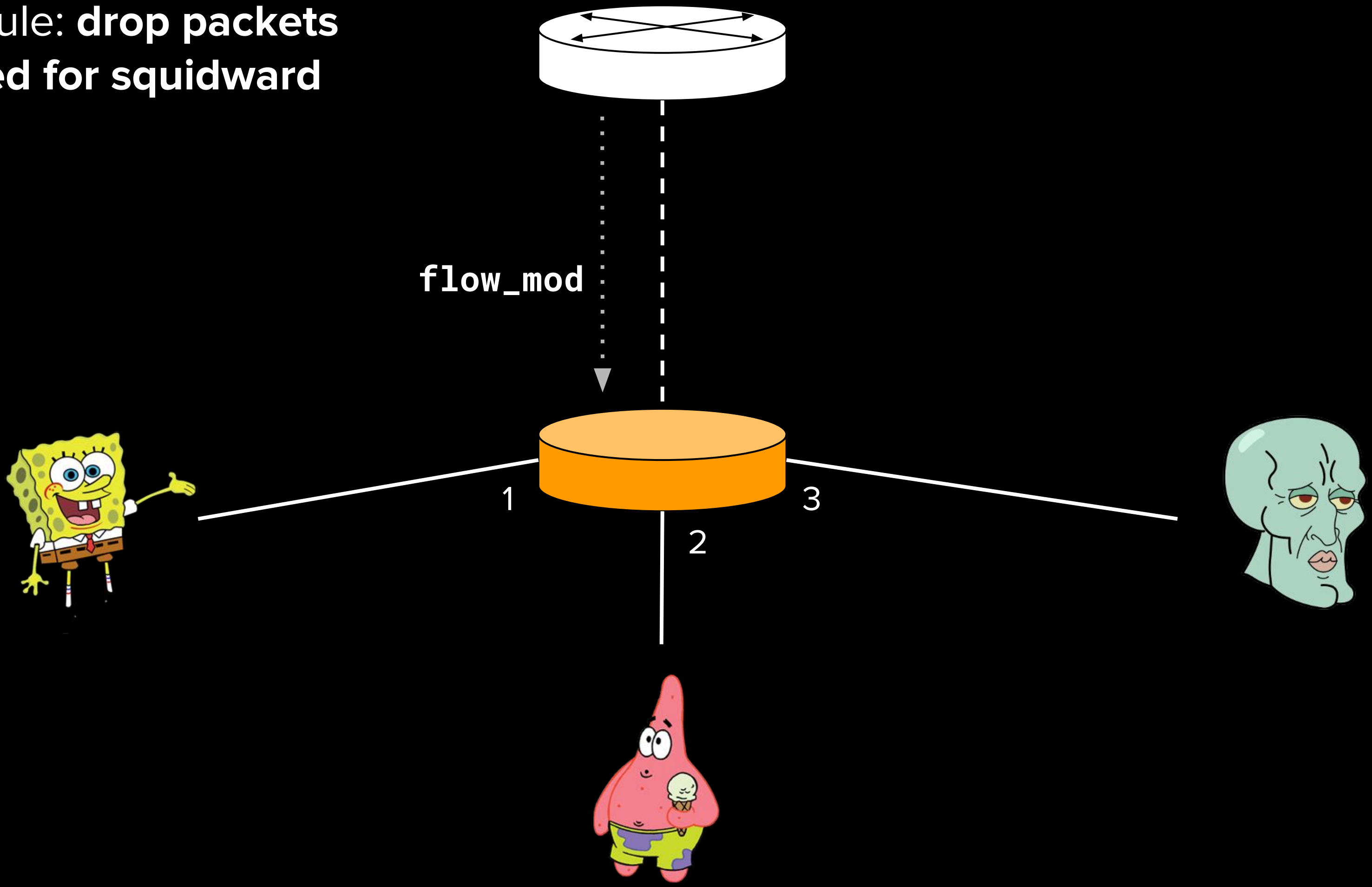
MAC	port
spongebob	1

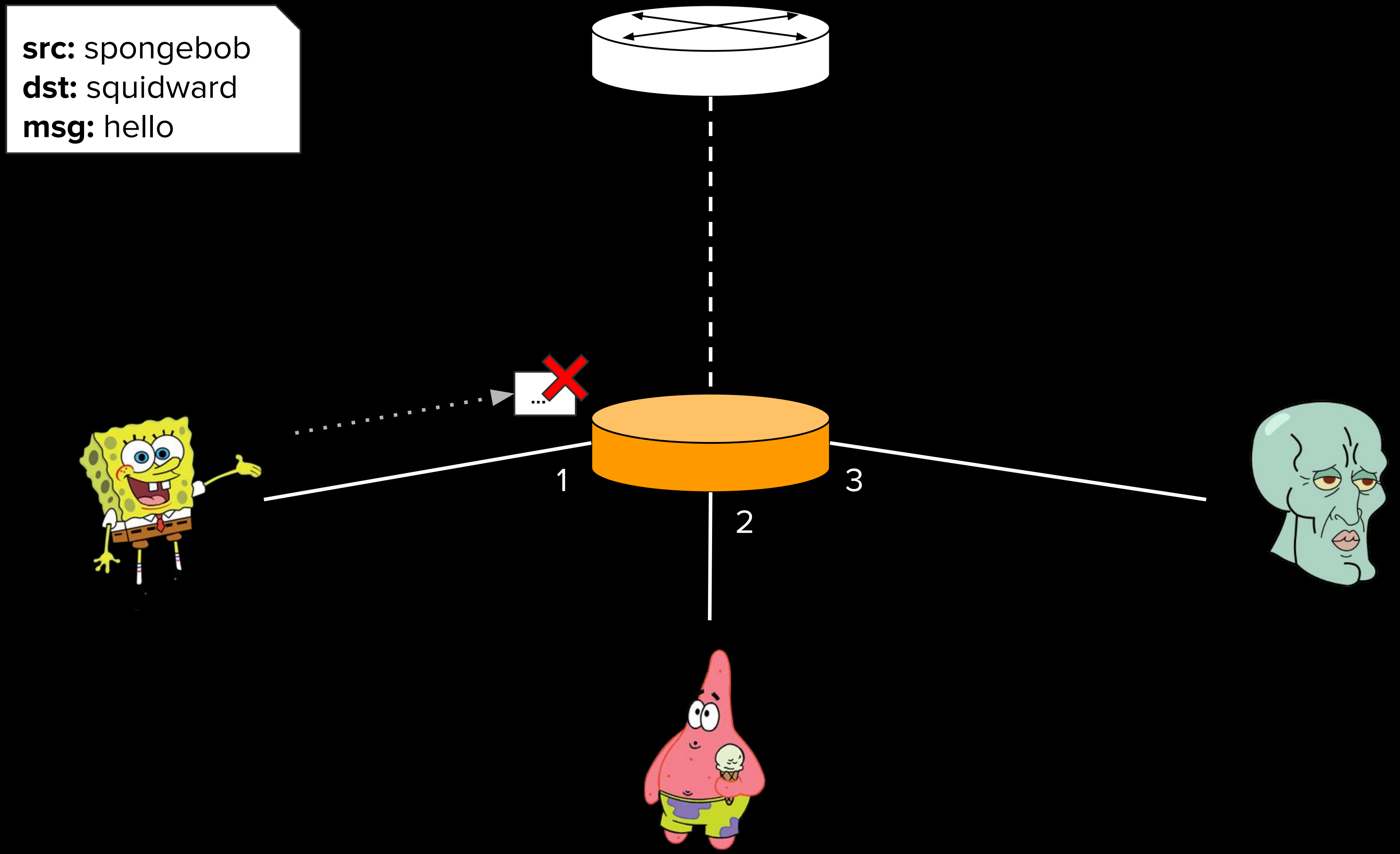


firewall



install rule: **drop packets destined for squidward**





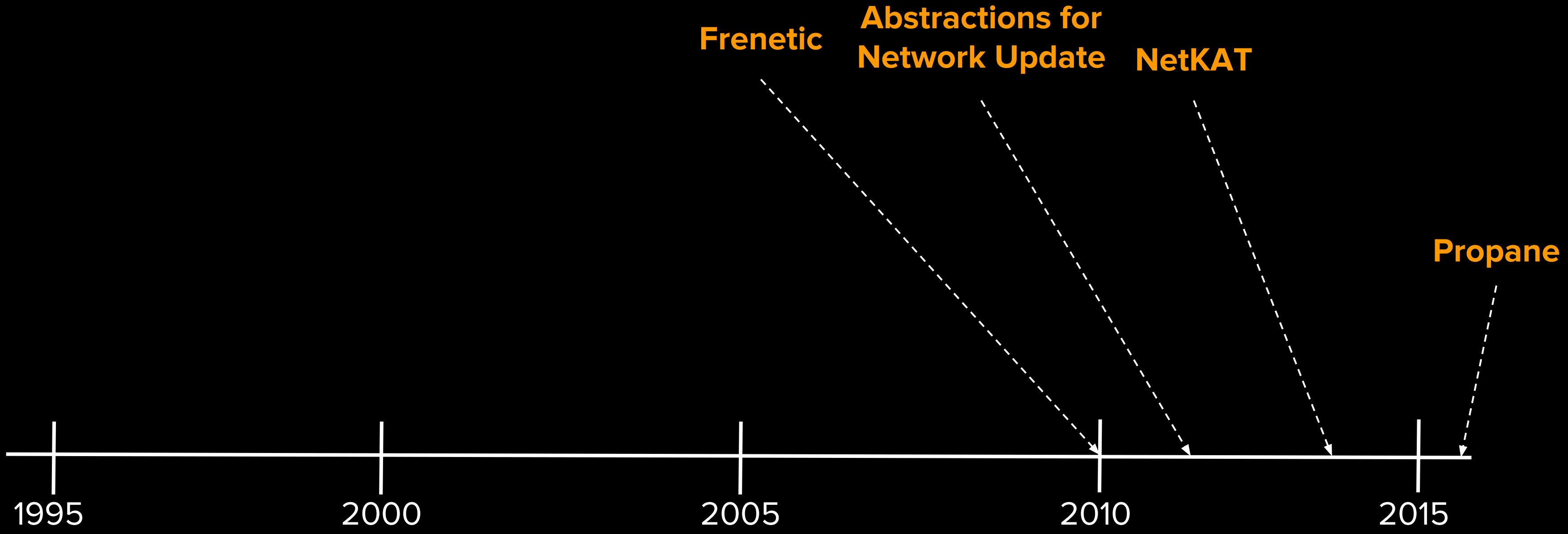
src: spongebob
dst: squidward
msg: hello

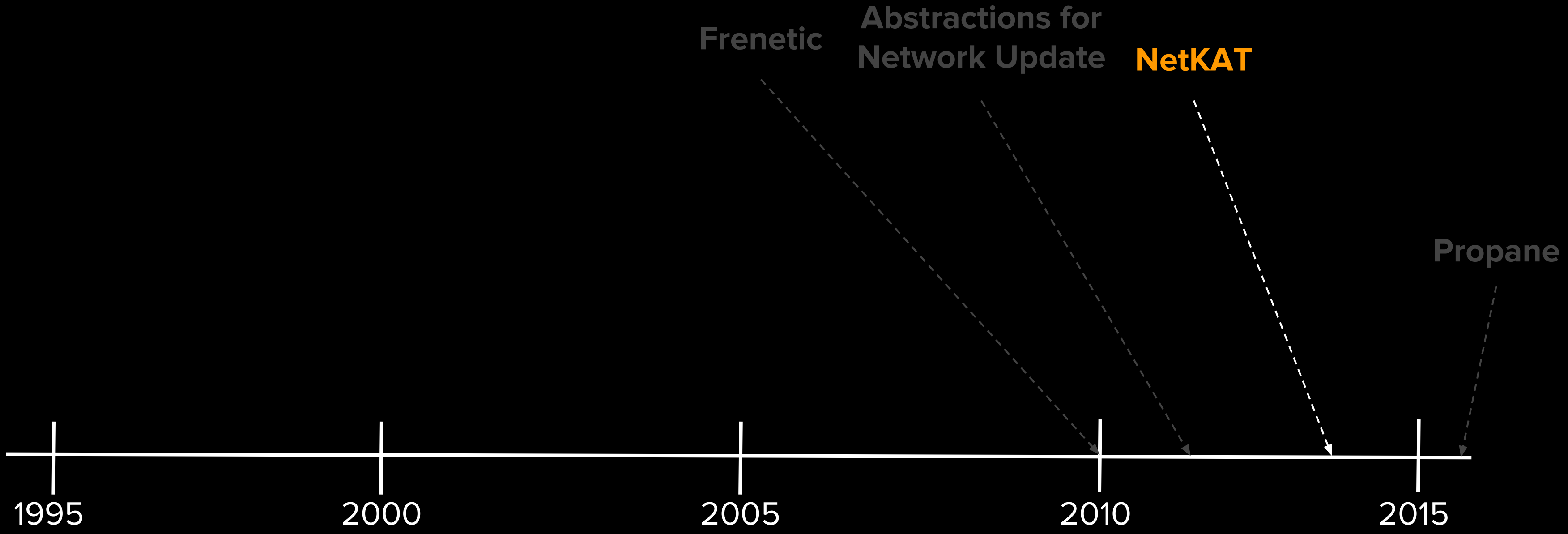
Active Networks

Separating the Data and Control Planes

OpenFlow/SDN

SDN Today


Frenetic
Abstractions for Network Update
NetKAT
Propane
1995
2000
2005
2010
2015


Frenetic
Abstractions for Network Update
NetKAT
Propane
1995
2000
2005
2010
2015

NetKAT

Forwarding Policy (F)

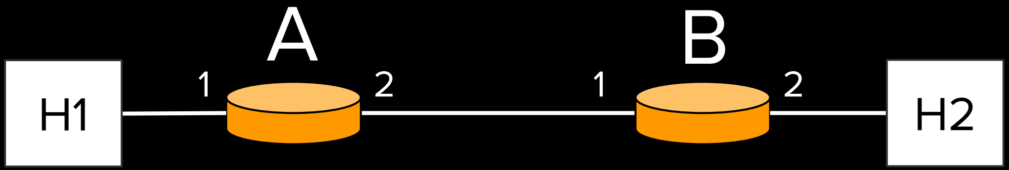
```
if sw==A then
     (if pt==1 then pt=2 elif pt==2 then pt=1)
elif sw==B then
     (if pt==1 then p=2 elif pt=2 then pt=1)
else drop
```

Network Behavior

```
run(F;T)
```

Topology (T)

```
if sw==A and pt==2 then (sw=B; pt=1)
elif sw==B and pt==1 then (sw=A; pt=2)
else drop
```



NetKAT

Forwarding Policy (F)

```
if sw==A then
    (if pt==1 then pt=2 elif pt==2 then pt=1)
elif sw==B then
    (if pt==1 then p=2 elif pt=2 then pt=1)
else drop
```

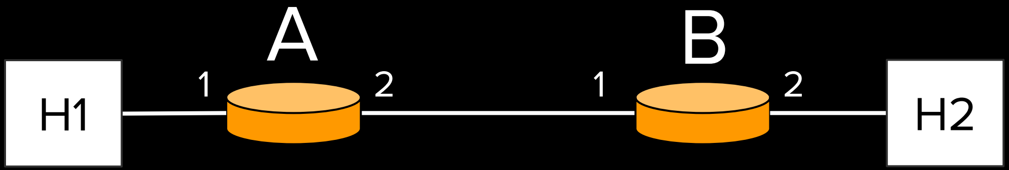
Topology (T)

```
if sw==A and pt==2 then (sw=B; pt=1)
elif sw==B and pt==1 then (sw=A; pt=2)
else drop
```

Intuition: These are functions from packets to sets of packets

Network Behavior

`run(F;T)`



Want to show:
SSH packets sent from **H1** get to **H2**

Invariant true when
P1 is equivalent to **P2**.

P1

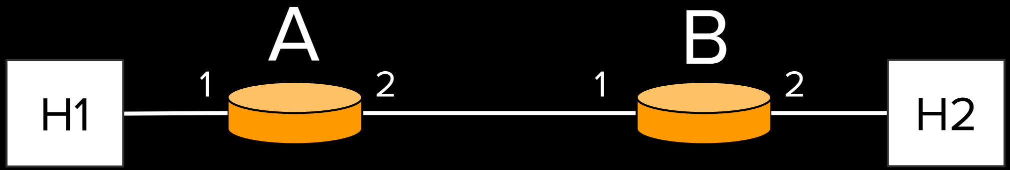
```
if typ==SSH and @h1
then run(F;T);
eventually(@h2)
```

P2

```
if typ==SSH and @h1
then run(F;T)
```

let `@h1` be `sw==A and pt==1`

let `@h2` be `sw==B and pt==2`



Want to show:
SSH packets sent from **H1** get to **H2**

Invariant true when
P1 is equivalent to **P2**.

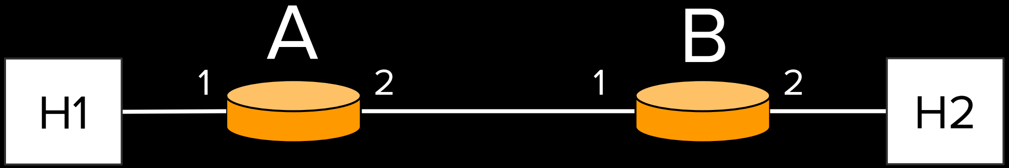
P1

```
if typ==SSH and @h1
then run(F;T);
eventually(@h2)
```

P2

```
if typ==SSH and @h1
then run(F;T)
```

SSH packets from **H1**



Want to show:
SSH packets sent from **H1** get to **H2**

Invariant true when
P1 is equivalent to **P2**.

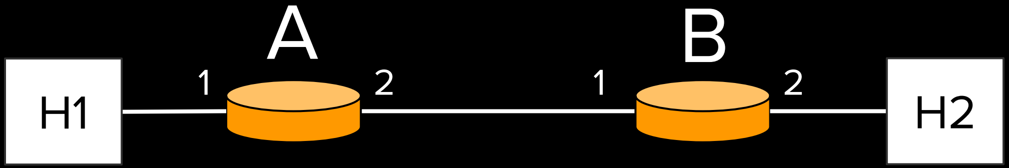
P1

```
if typ==SSH and @h1
then run(F;T);
eventually(@h2)
```

P2

```
if typ==SSH and @h1
then run(F;T)
```

“run” the network



Want to show:
SSH packets sent from **H1** get to **H2**

Invariant true when
P1 is equivalent to **P2**.

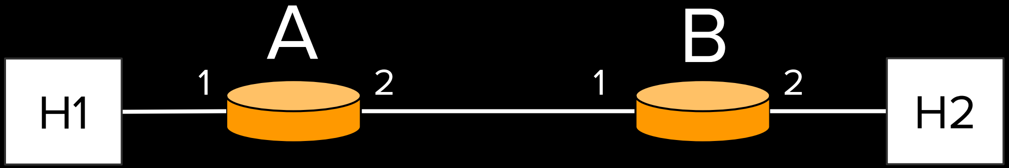
P1

```
if typ==SSH and @h1
then run(F;T);
eventually(@h2)
```

packets received by **H2**

P2

```
if typ==SSH and @h1
then run(F;T)
```

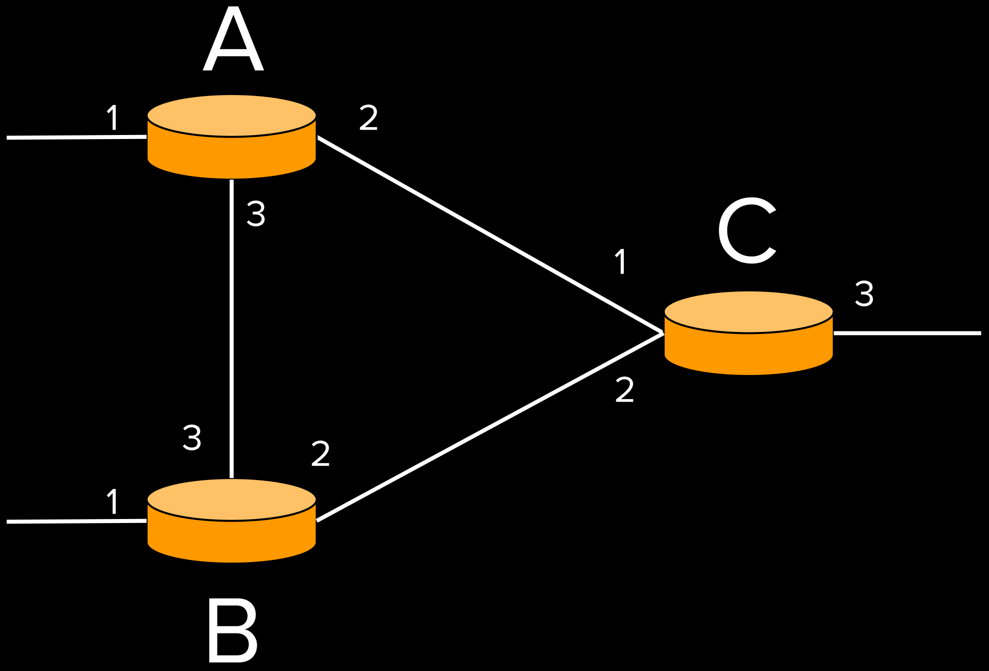


NetKAT

```
if sw == A then:
     if port == 1 then port := 2
     elif port == 2 then port := 3
     else drop

elif sw == B then:
     if port==1 drop
     else port := 1

elif sw == C
     if   port == 1 then port :=3
     elif port == 3 then port :=2
     elif port == 2 drop;
```

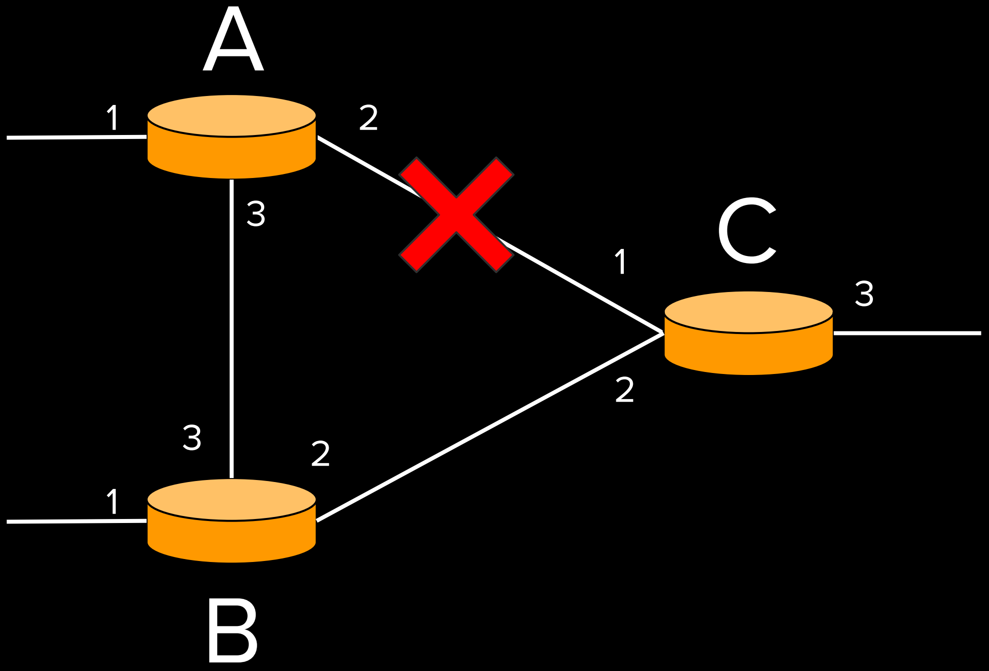


NetKAT

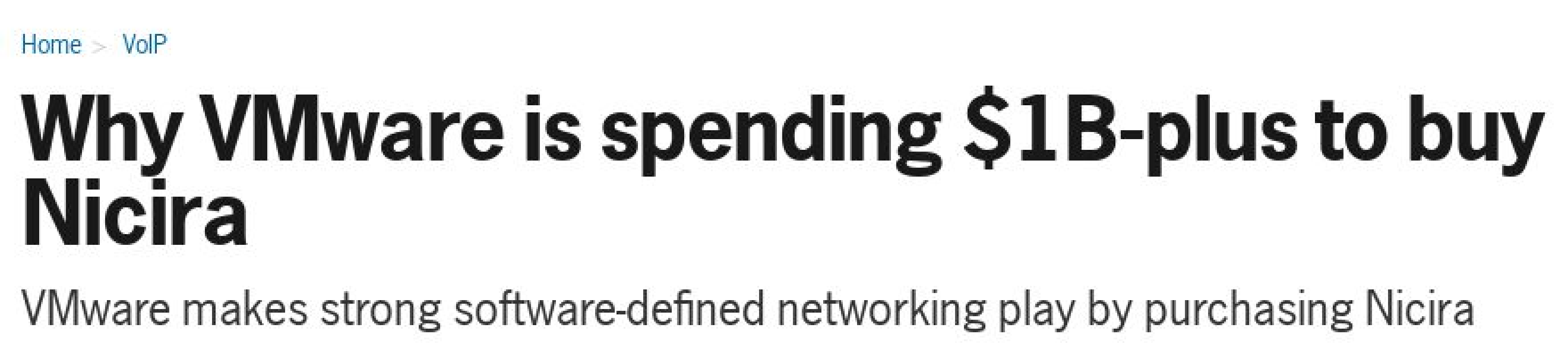
```
if sw == A then:
     if port == 2 then alert_ctrl
     else port := 3

if sw == B then:
     if port == 1 then drop
     elif port == 2 then port := 3
     elif port == 3 then port := 2

if sw == C
     if port == 1 then alert_ctrl
      elif port == 2 then port := 1
     elif port == 3 then port := 2
```



whither SDN?

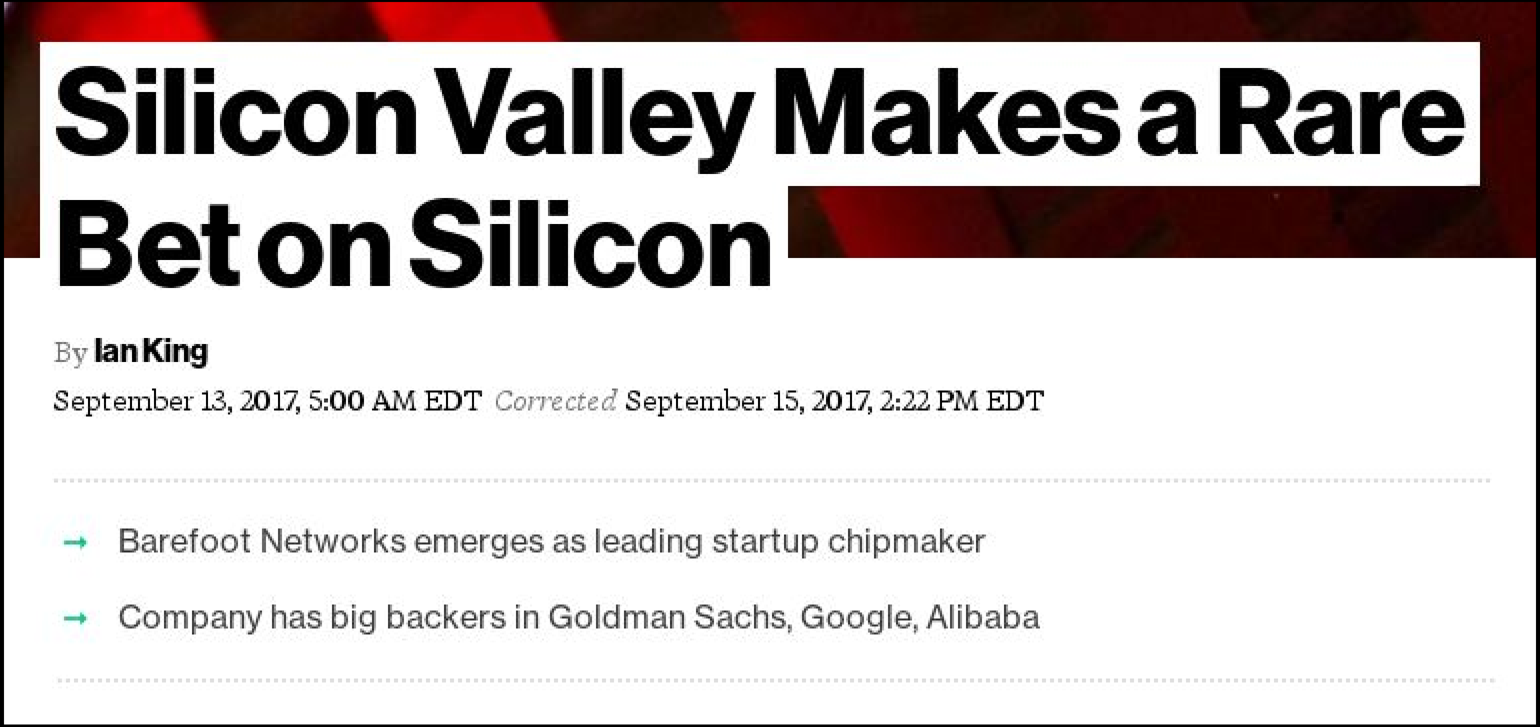


Home > VoIP

Why VMware is spending $1B-plus to buy Nicira

VMware makes strong software-defined networking play by purchasing Nicira

whither SDN?



Silicon Valley Makes a Rare Bet on Silicon

By Ian King

September 13, 2017, 5:00 AM EDT *Corrected* September 15, 2017, 2:22 PM EDT

- Barefoot Networks emerges as leading startup chipmaker
- Company has big backers in Goldman Sachs, Google, Alibaba

questions?

thanks!