Advanced Topics in Communication Networks

Internet Routing and Forwarding

Laurent Vanbever

nsg.ee.ethz.ch

15 Sep 2020

Lecture starts at **14:15**

In this lecture, you'll learn…

In this lecture, you'll learn

how to optimize the

- performance
- flexibility
- reliability

of large-scale network infrastructures

In this lecture, you'll learn

how to optimize the

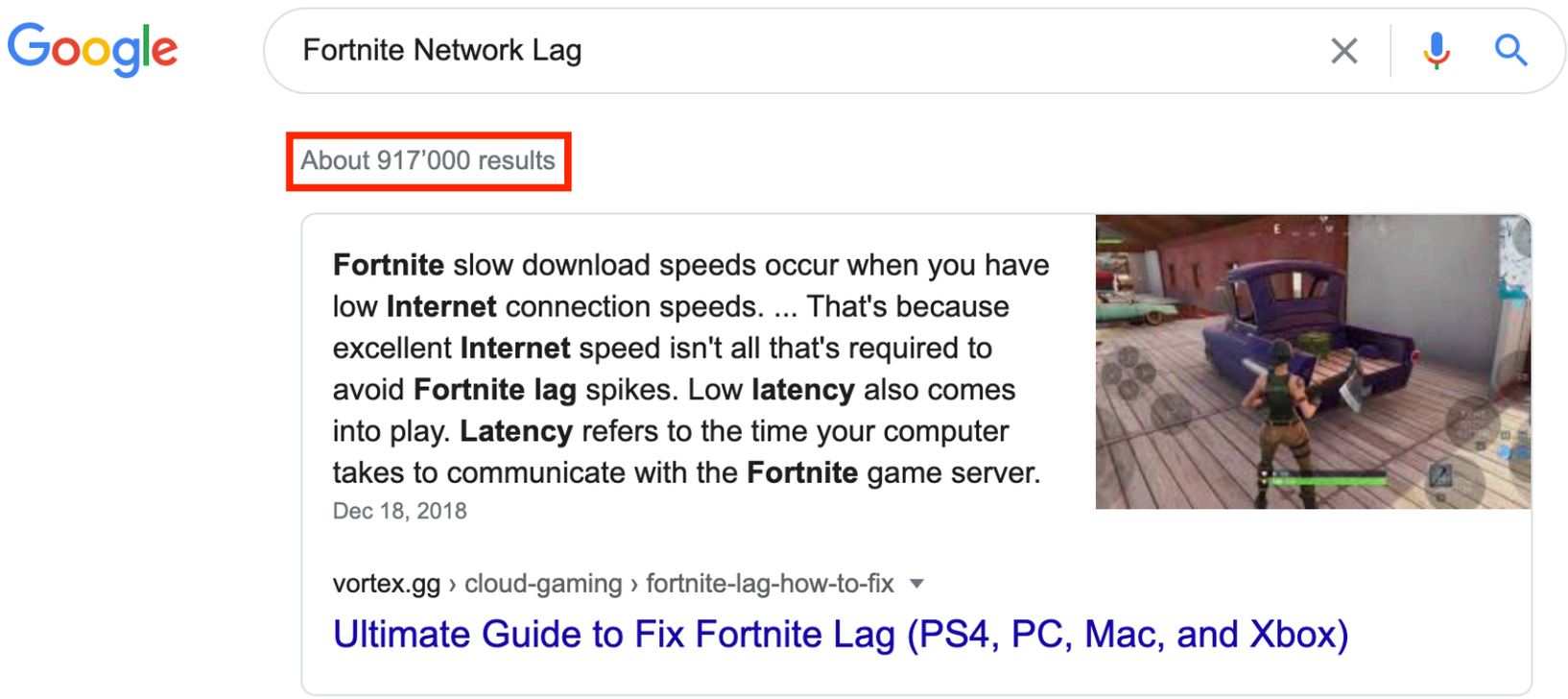
- performance
- flexibility
- reliability

of large-scale network infrastructures

Why should *you* care, **as a user?**

Fortnite, Epic Games

Many gamers care about low latency!



44% of UK Online Multiplayer Gamers Still Complain About Lag

Tuesday, June 23rd, 2020 (2:21 pm) - Score 4,169

Email | Link News 52 Comments

A new survey of 1,000 UK video gamers, which was commissioned by full fibre broadband ISP Hyperoptic, has found that 44% of adult respondents named "*the internet lagging*" as the most infuriating aspect of online gaming, while 27% also complained about other players' slow online speeds.

The survey revealed that respondents are now spending an average of 10.5 hours online gaming a week (up from 8 hours pre-lockdown) and more than half said the hobby has kept them entertained during this time (we assume the other half must have been either losing or playing a really boring game, since they aren't mentioned). Meanwhile 43% said they found playing online had helped them to stay connected with others.

How to reduce lag in PC games

By NICK GREENE Last updated 26 Mar 2020

Source: Pixab

Wh
yo
fru
Int
my

Actiontec PRODUCTS SERVICES RESOURCES LEARN SUPPORT

How to Reduce Latency or Lag in Gaming

s and Gateways G.hn

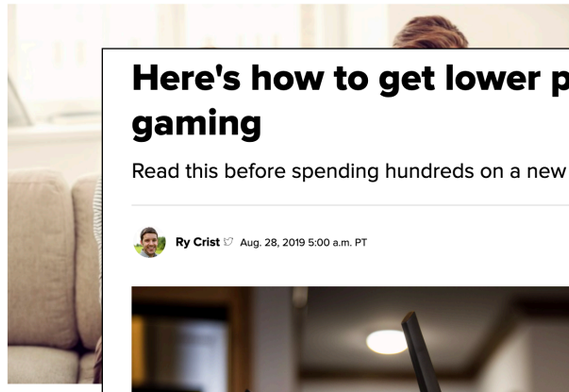
o gaming. Low latency means there is a
e chance for a connection loss or delay.
ean instant death. A wired connection is
iminates the possibility of lag.

g. Then we'll look at some opportunities
ork.

7 Ways to Reduce Latency in Online Gaming

Sarah Pike
January 4, 2016

0
Comments



Here's how to get lower ping for online gaming

Read this before spending hundreds on a new gaming router.

Ry Crist Aug. 28, 2019 5:00 a.m. PT

3

OUTFOX About Us Forum Support Blog Download LOG IN START A FREE TRIAL

OUTSMART. OUTMANEUVER. OUTPERFORM.

THE ULTIMATE OPTIMIZED GAMING NETWORK

START A FREE TRIAL

MPROVES GAME CONNECTIONS

HASTE Features How Haste Works Supported Games Haste Pro Pricing GET HASTE NOW

RECLAIM YOUR CONNECTION

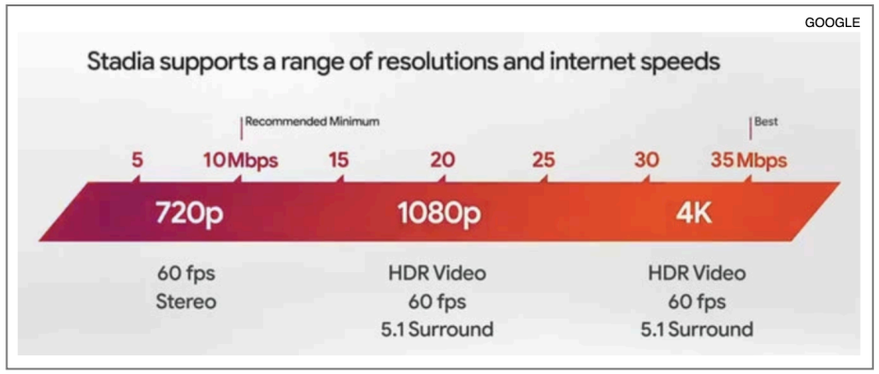
DEFEAT PING AND STABILIZE YOUR CONNECTION

GET HASTE NOW

No Credit Card Required for Haste Free

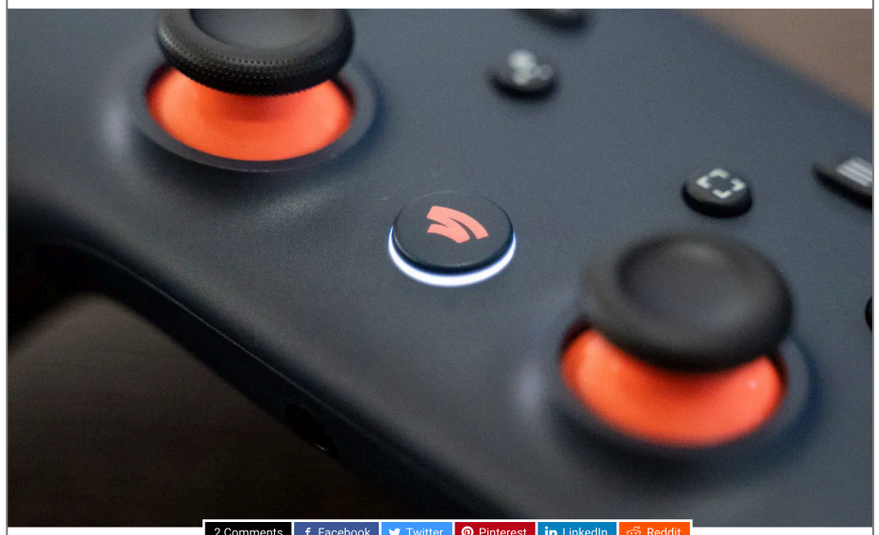
The problem of poor network performance will intensify as video games go streaming-based

Google Stadia



How to improve your internet connection for Google Stadia

Kyle Bradshaw - Nov. 19th 2019 12:30 pm PT @SkylledDev



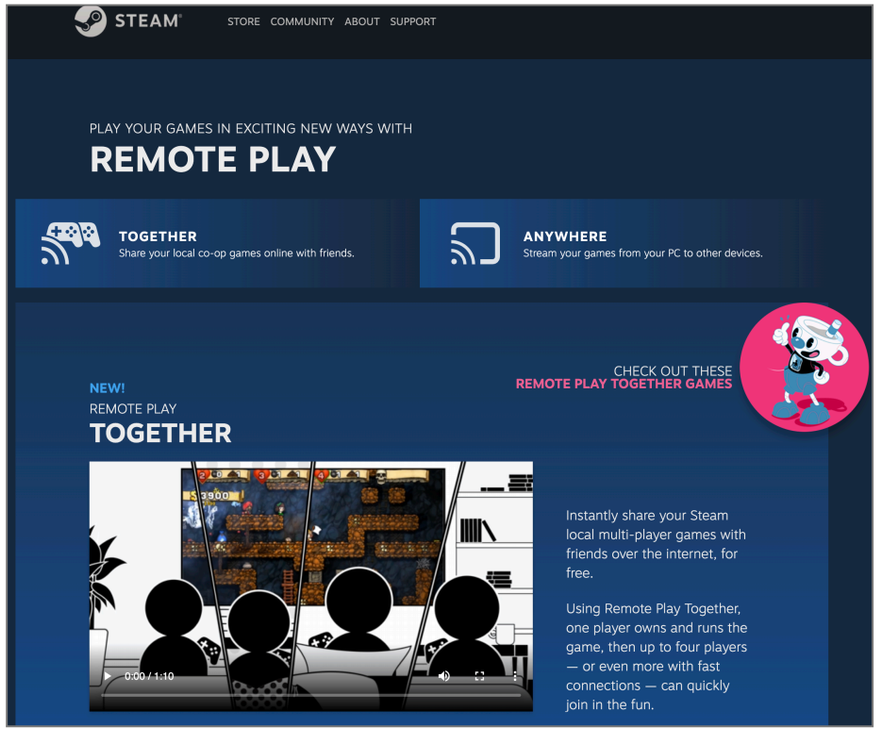
2 Comments | Facebook | Twitter | Pinterest | LinkedIn | Reddit

Google Stadia, as a game streaming service that can run at up to 4k and 60 frames per second, is very demanding on your internet connection. So we've collected a few ways you can improve your internet connection to Google Stadia to be both better and faster and have less lag.

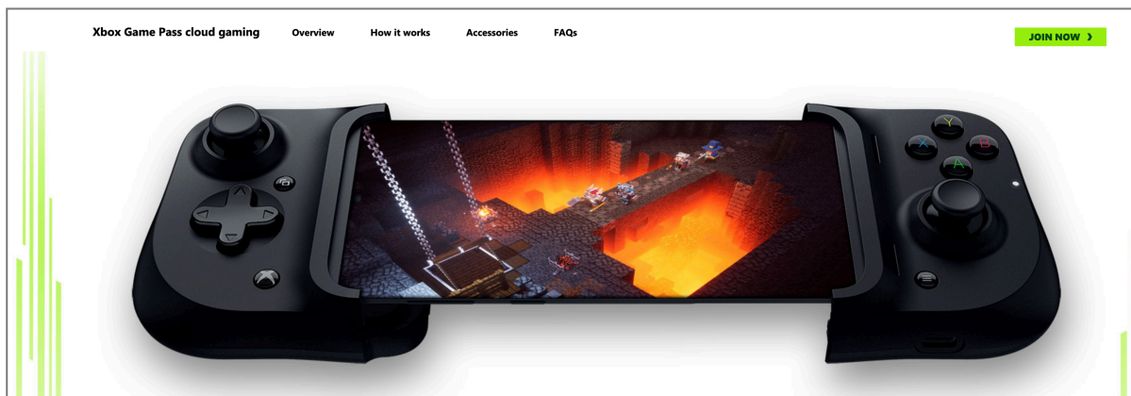
Before we can dive into the different methods you can use to improve your connection, let's take a second to discuss the two main attributes of your network that can affect Google Stadia or introduce lag.

The first is your download speed, or how long it takes to download files. Your download speed

Steam Remote Play



Xbox Game Pass Cloud Gaming



Audio/video-conferencing is also highly sensitive to network performance

and *many* others…

And, needless to say,
so is video streaming…

Where does this come from?

By default, IP networks only offer
a (possibly lousy) best effort service

By default, IP networks only offer
a (possibly lousy) best effort service

best effort service

IP routers do their best to deliver packets to their destination without making any promises

best effort service

IP routers do their best to deliver packets to their destination without making any promises

For many applications and users
doing "its best" is just not good enough anymore

In this lecture, you'll learn

how to optimize the

- performance
- flexibility
- reliability

of large-scale network infrastructures

Why should *you* care, as a **network operator?**

IP networks carry always more critical services

IP networks carry always more critical services

Since the end of 2017,

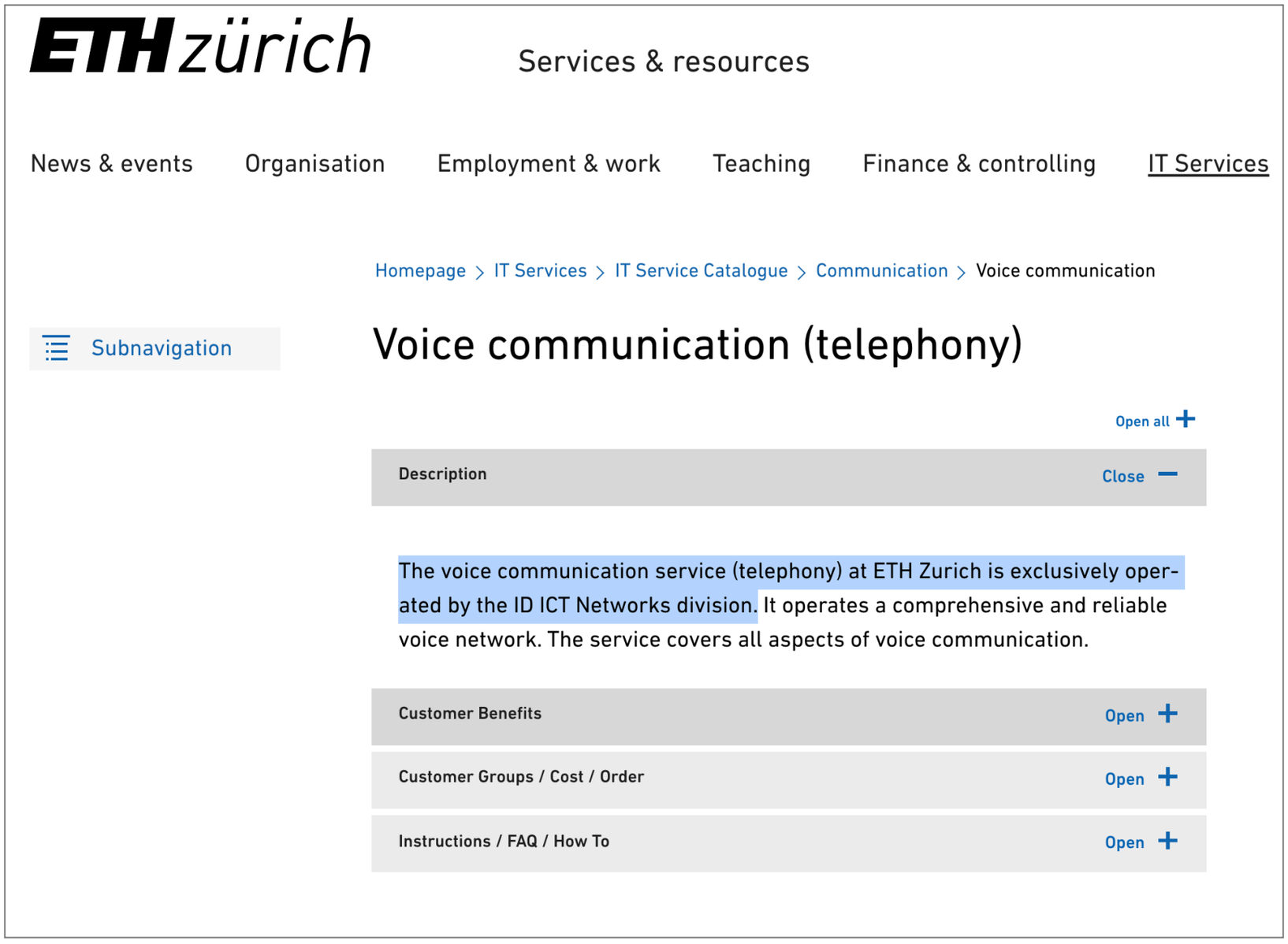
All fixed-network services

(telephony, TV, and Internet)

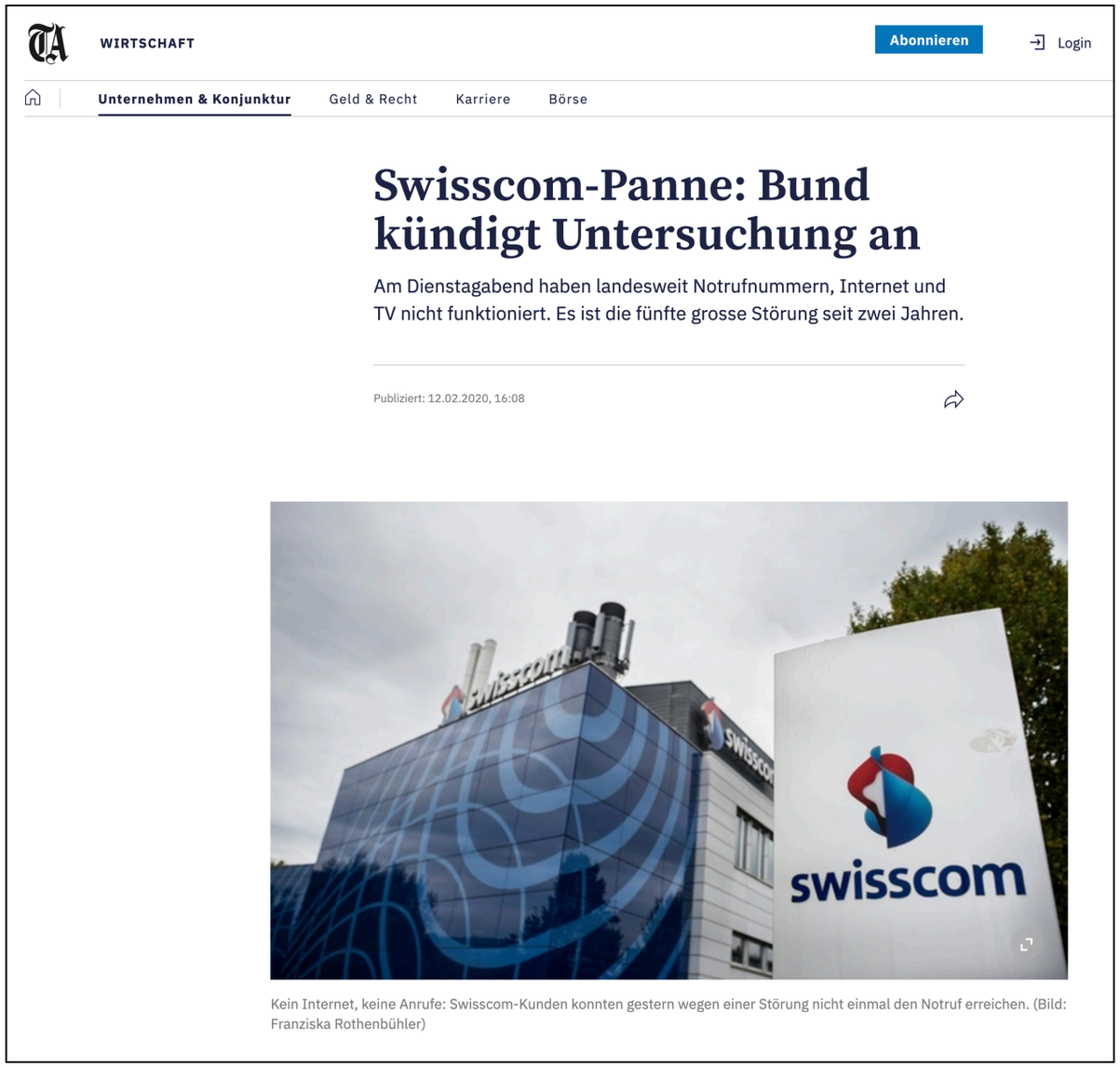
run on IP technology

Source: https://www.swisscom.ch/en/business/enterprise/themen/work-smart/all-ip.html

Same applies at ETH Zürich when you pick up the phone



With great powers come great responsibilities



WIRTSCHAFT

Abonnieren

Login

Unternehmen & Konjunktur | Geld & Recht | Karriere | Börse

Swisscom-Panne: Bund kündigt Untersuchung an

Am Dienstagabend haben landesweit Notrufnummern, Internet und TV nicht funktioniert. Es ist die fünfte grosse Störung seit zwei Jahren.

Publiziert: 12.02.2020, 16:08

Kein Internet, keine Anrufe: Swisscom-Kunden konnten gestern wegen einer Störung nicht einmal den Notruf erreichen. (Bild: Franziska Rothenbühler)

February 2020

Emergency numbers

(117, 112, 144, 118)

were not reachable for >1.5h

due to a major outage

in Swisscom

https://www.tagesanzeiger.ch/wirtschaft/unternehmen-und-konjunktur/bei-swisscom-laeuft-im-moment-gar-nichts/story/20660403

IP networks carry always more traffic

critical services

IP networks carry always more traffic

always faster user speeds

x

always more users

x

always more content consumed

IP networks carry always more traffic

always faster user speeds

x

always more users

x

always more content consumed

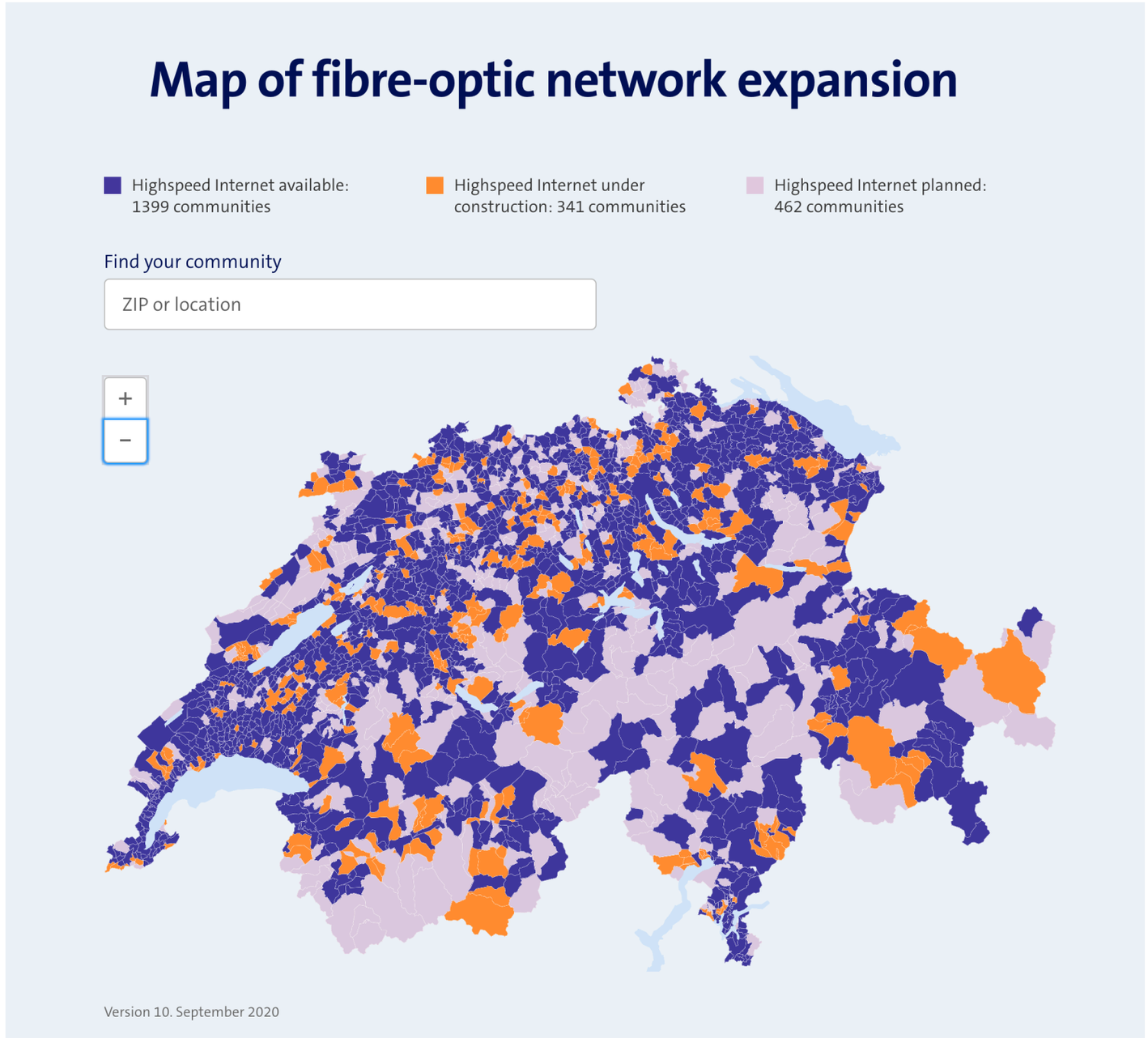
IP networks carry always more traffic

Expected growth in throughput by 2023

broadband	>x2	110 Mbps
cellular	>x3	44
Wi-Fi	>x3	92

Source: Cisco Annual Internet Report (2018–2023)

Up to 60% of Swiss homes and business will have access to Fiber connectivity by 2025



Source: Swisscom

IP networks carry always more traffic

always faster user speeds

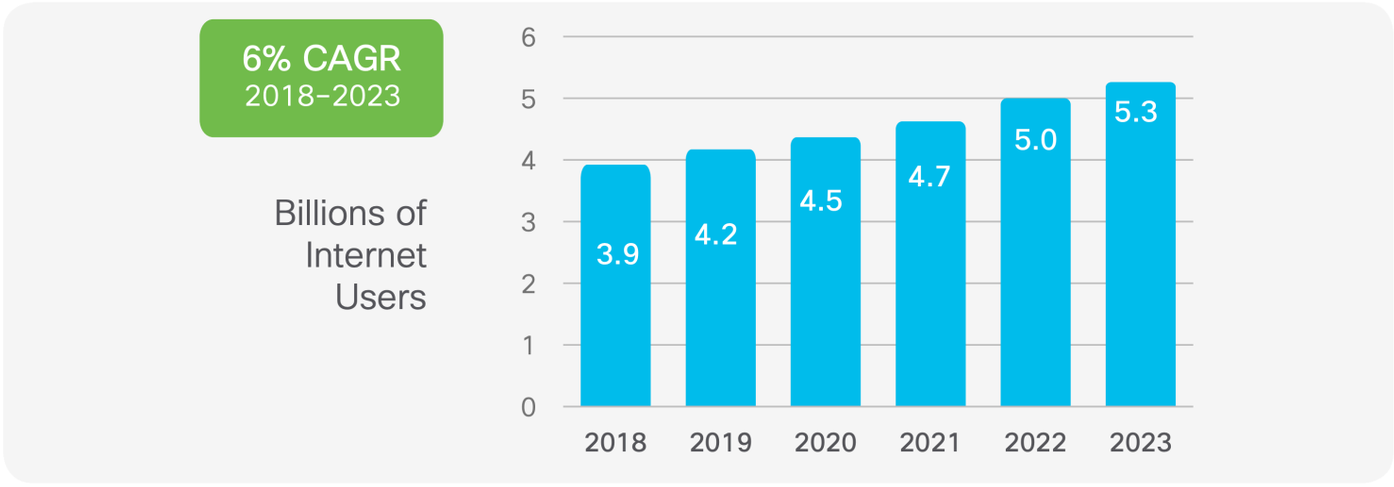
x

always more users

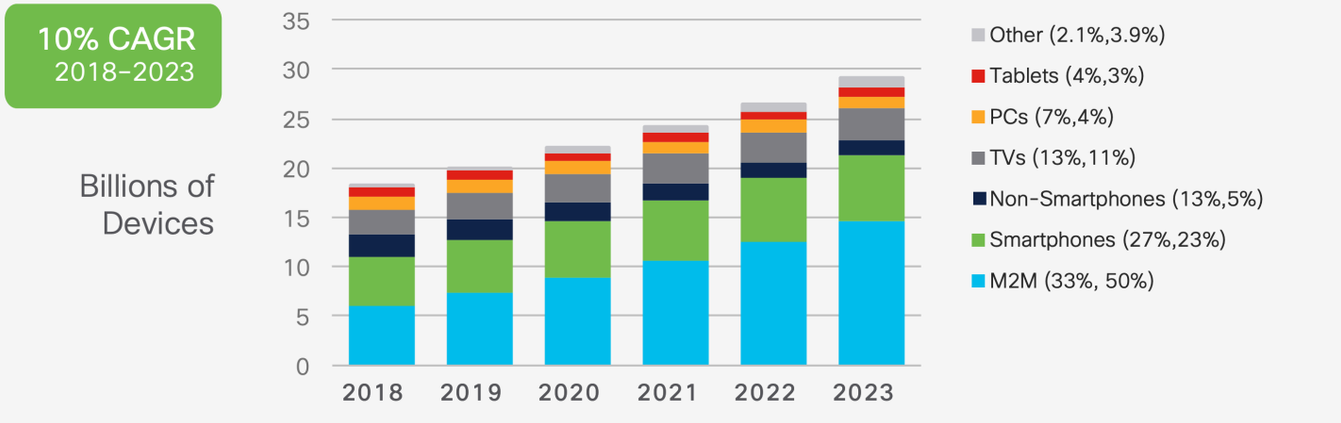
x

always more content consumed

Internet growth is still going on strong



Source: Cisco Annual Internet Report, 2018–2023



Source: Cisco Annual Internet Report, 2018–2023

IP networks carry always more traffic

always faster user speeds

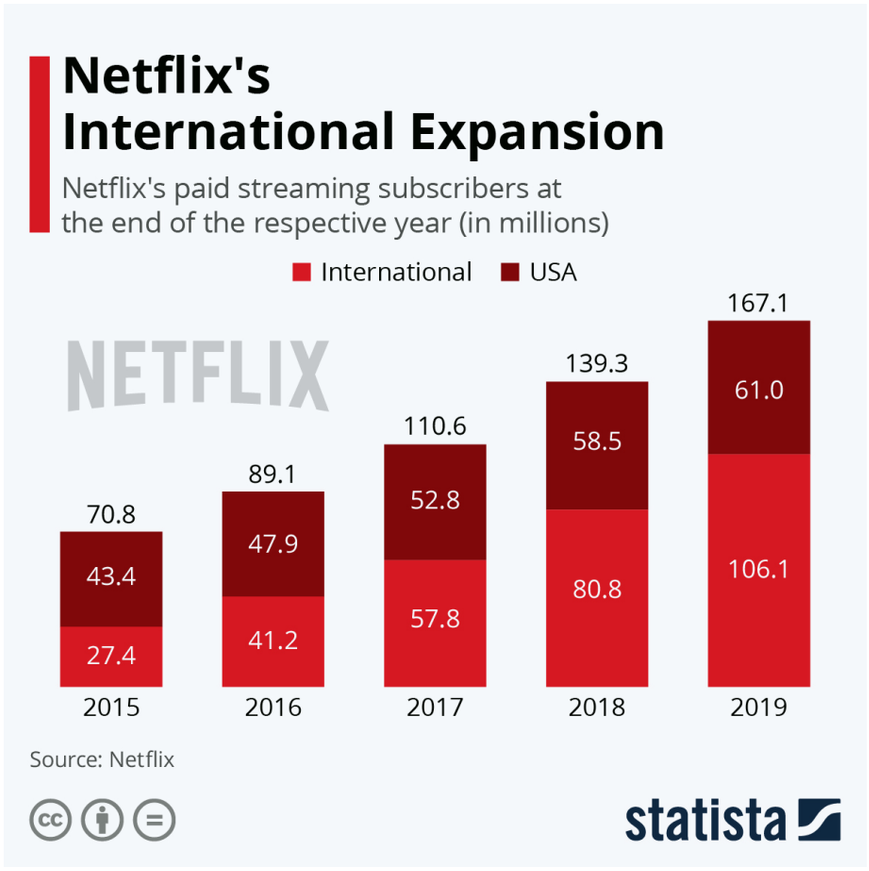
x

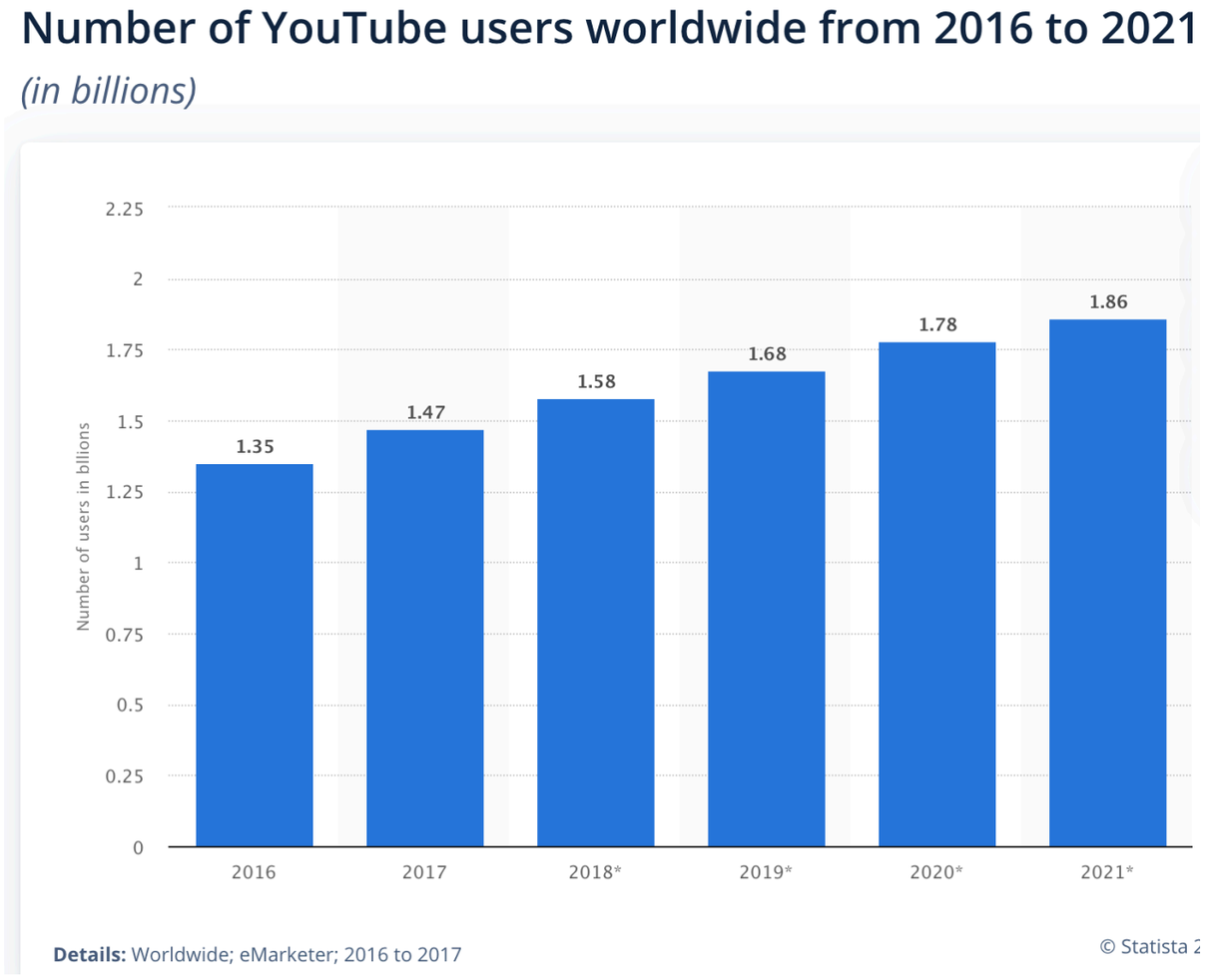
always more users

x

always more content consumed

Most of the Internet traffic is video already and we keep watching more of more of them





Ever-increasing network traffic stresses network infrastructures

One recent example

somehow arguable though

Netflix Lowers Video Quality to Aid Broadband ISP Congestion

Thursday, March 19th, 2020 (8:53 pm) - Score 5,555

Email | Link News 43 Comments

In a somewhat unusual development Netflix, the global internet video streaming giant, has responded to some concerns about broadband congestion in other European countries by agreeing with the EU to lower the quality (bitrate) of its videos for the next 30 days (i.e. easing the strain on fixed line ISPs and mobile operators).

The move will result in the company reducing video bitrates across all of their streams, although they have yet to say if those paying extra for the highest quality are going to be compensated. Netflix have also yet to confirm if this will apply to the United Kingdom, where the majority of fixed line providers have yet to really struggle with managing increased demand (we've covered this here).

https://www.ispreview.co.uk/index.php/2020/03/netflix-lowers-video-quality-to-aid-broadband-isp-congestion.html

In this lecture, you'll learn

how to optimize the

- performance
- flexibility
- reliability

of large-scale network infrastructures

Techniques

Performance

Flexibility

Reliability

Traffic Engineering

Load Balancing

Quality of Service

Multicast

Virtual Private Networks

Fast Convergence

Techniques

Performance

Traffic Engineering

Load Balancing

Quality of Service

Multicast

Flexibility

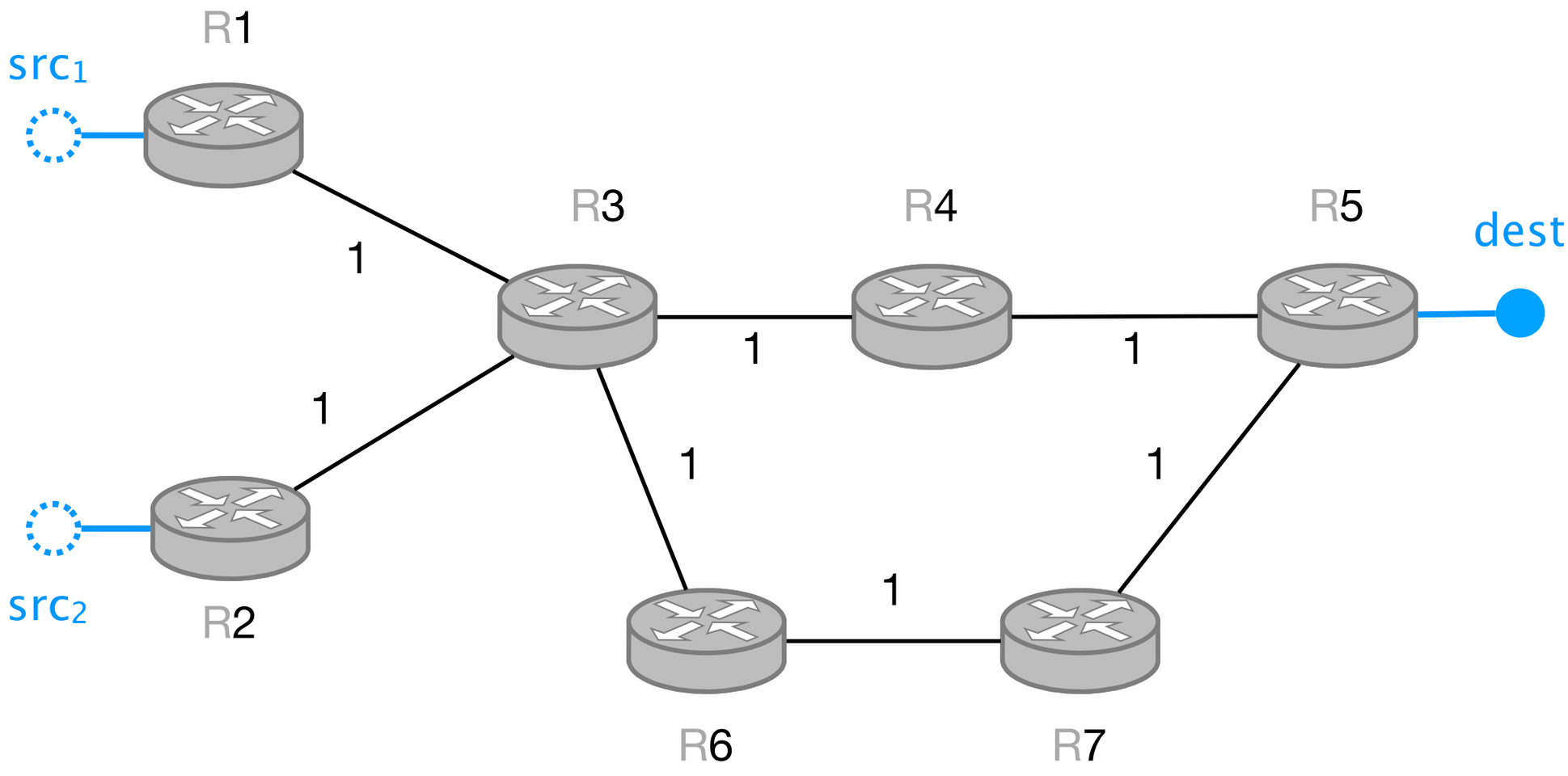
Virtual Private Networks

Reliability

Fast Convergence

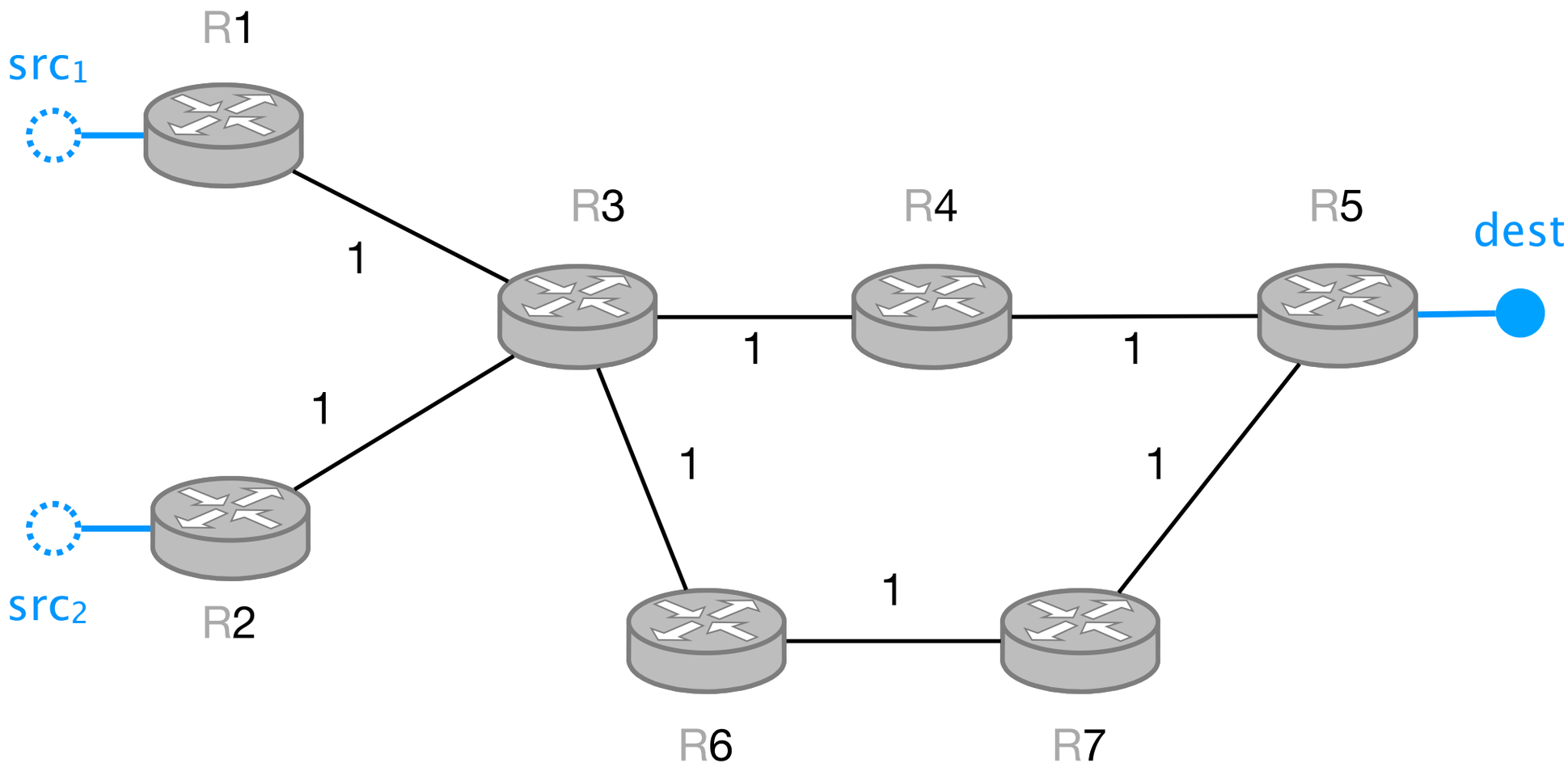
Consider this 10-Gbps network running vanilla OSPF

src_1 and src_2 send traffic to dest, 1 TCP flow each

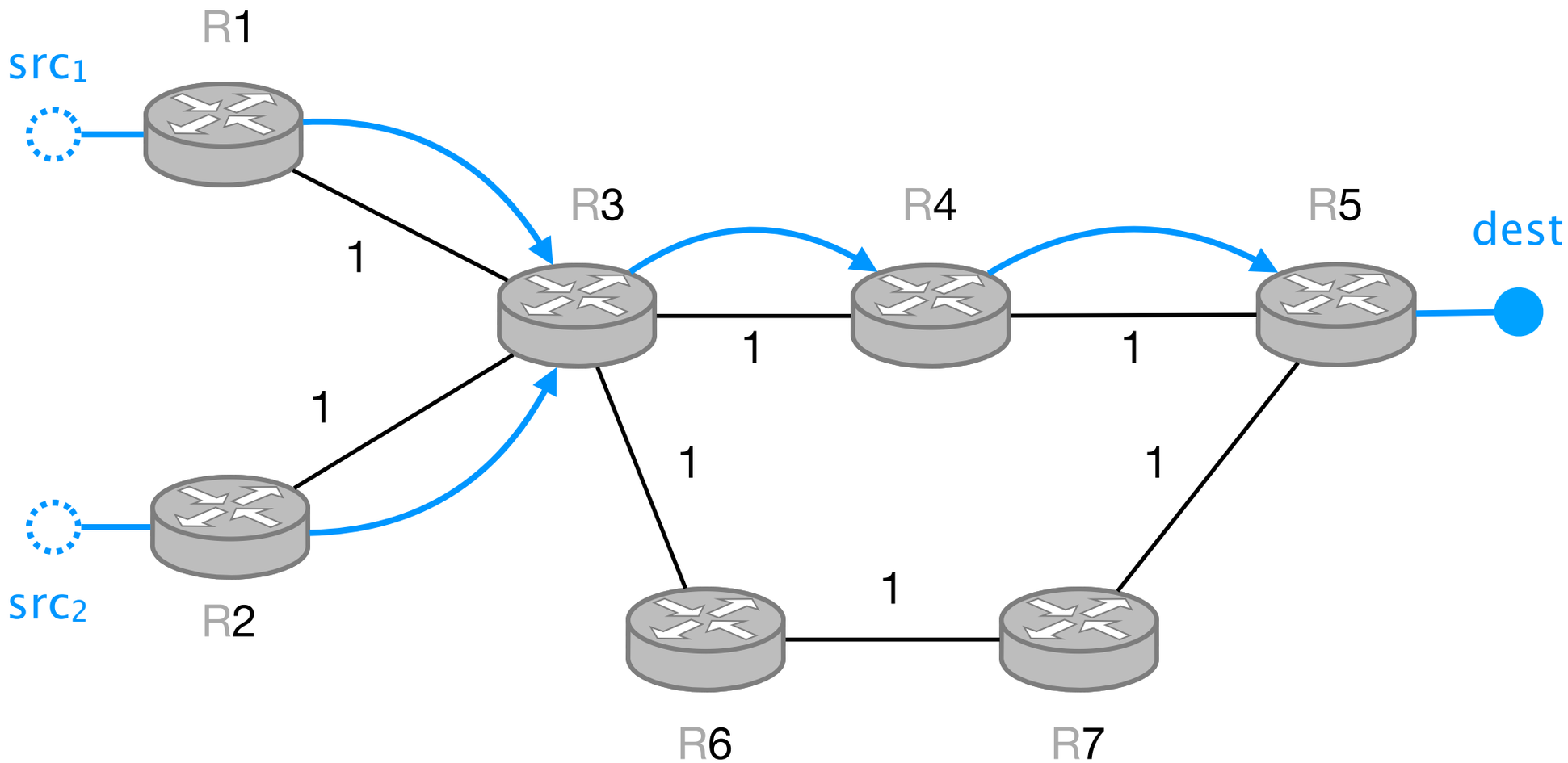


Consider this 10-Gbps network running vanilla OSPF

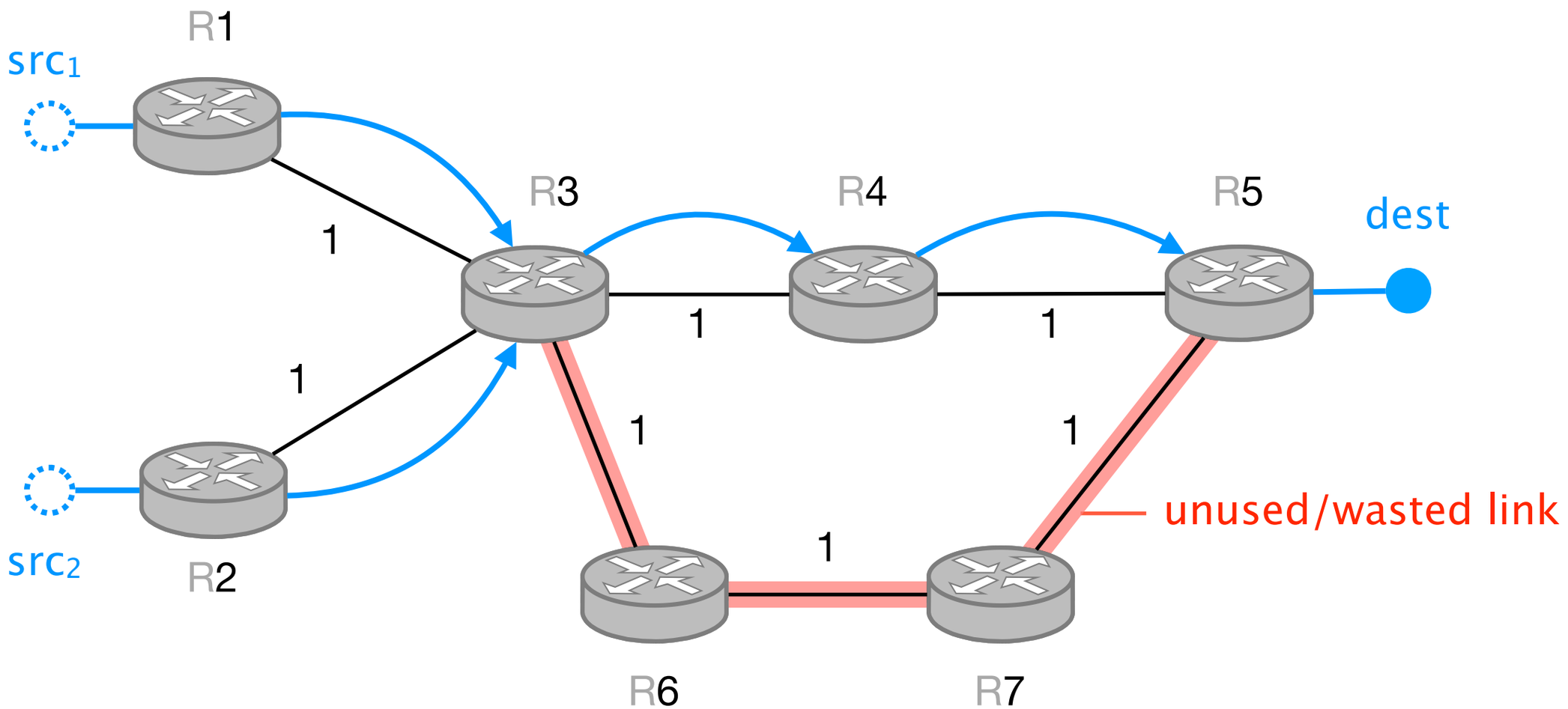
src_1 and src_2 send traffic to dest, 1 TCP flow each



What's the max. throughput (src_1, dest) and (src_2, dest) can achieve?

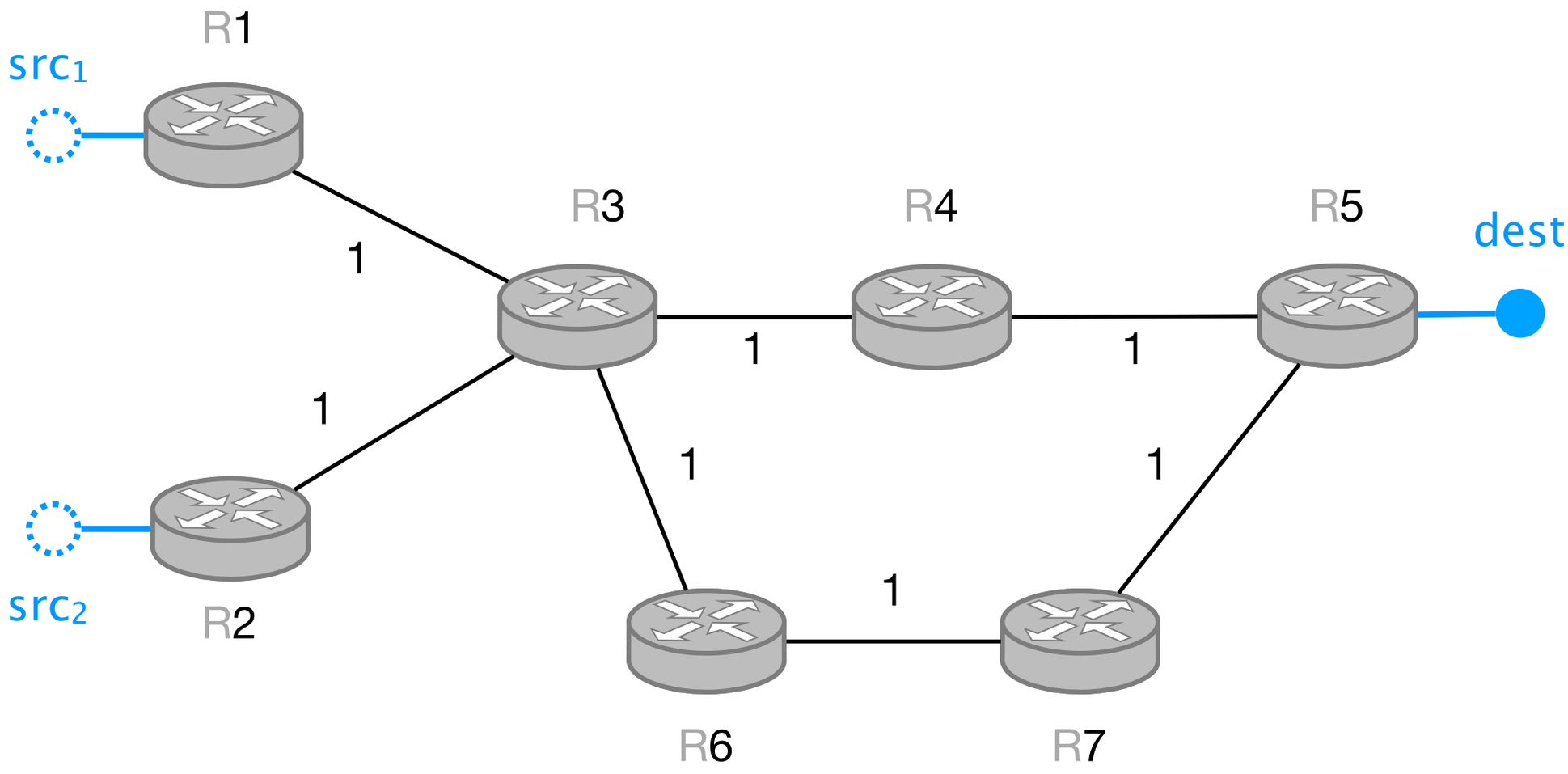


What's the max. throughput (src_1, dest) and (src_2, dest) can achieve? 5 Gbps



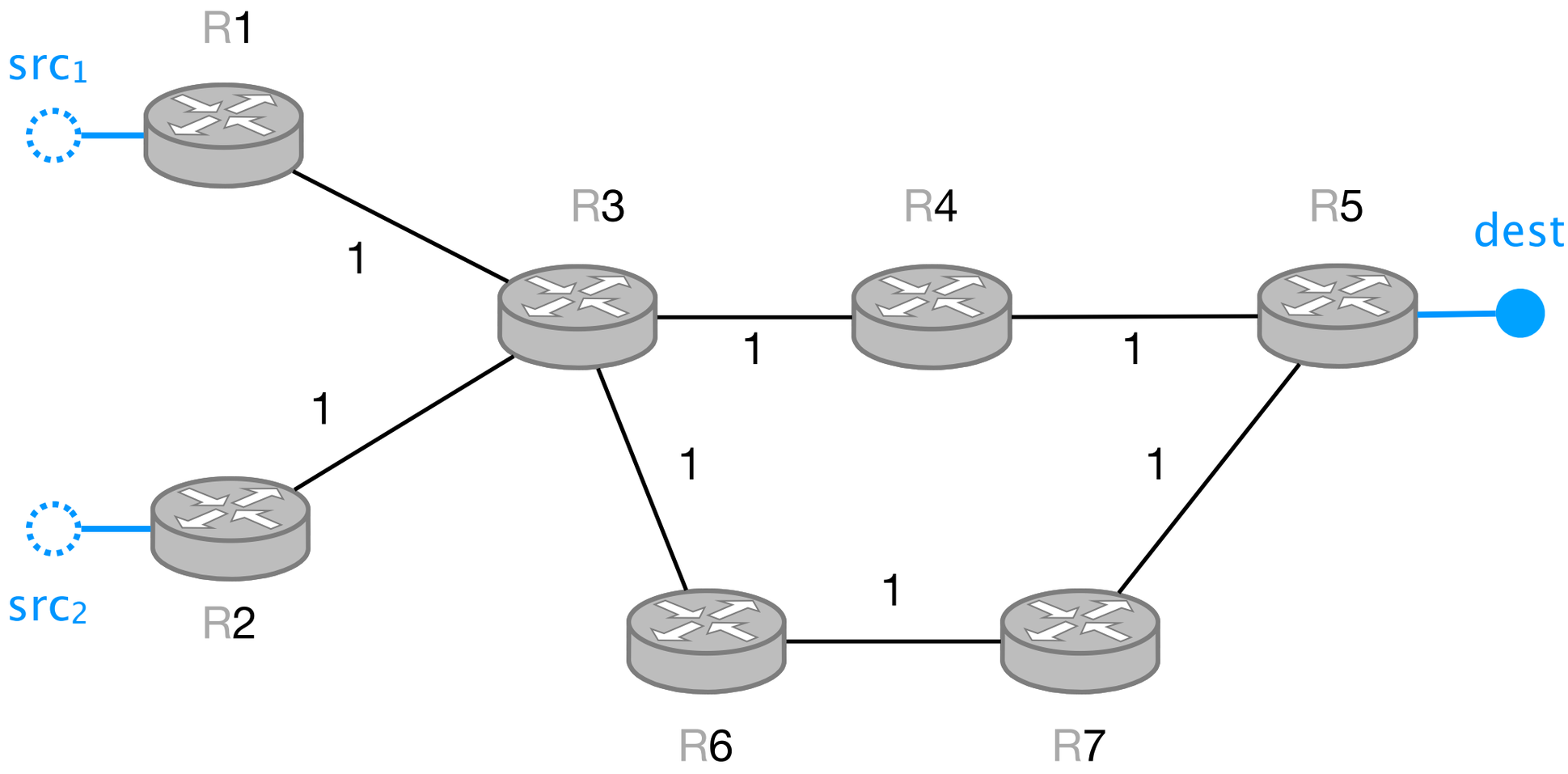
What's the max. throughput (src_1, dest) and (src_2, dest) can achieve? 5 Gbps

What can we do better?



What can we do better?

What about better weights?



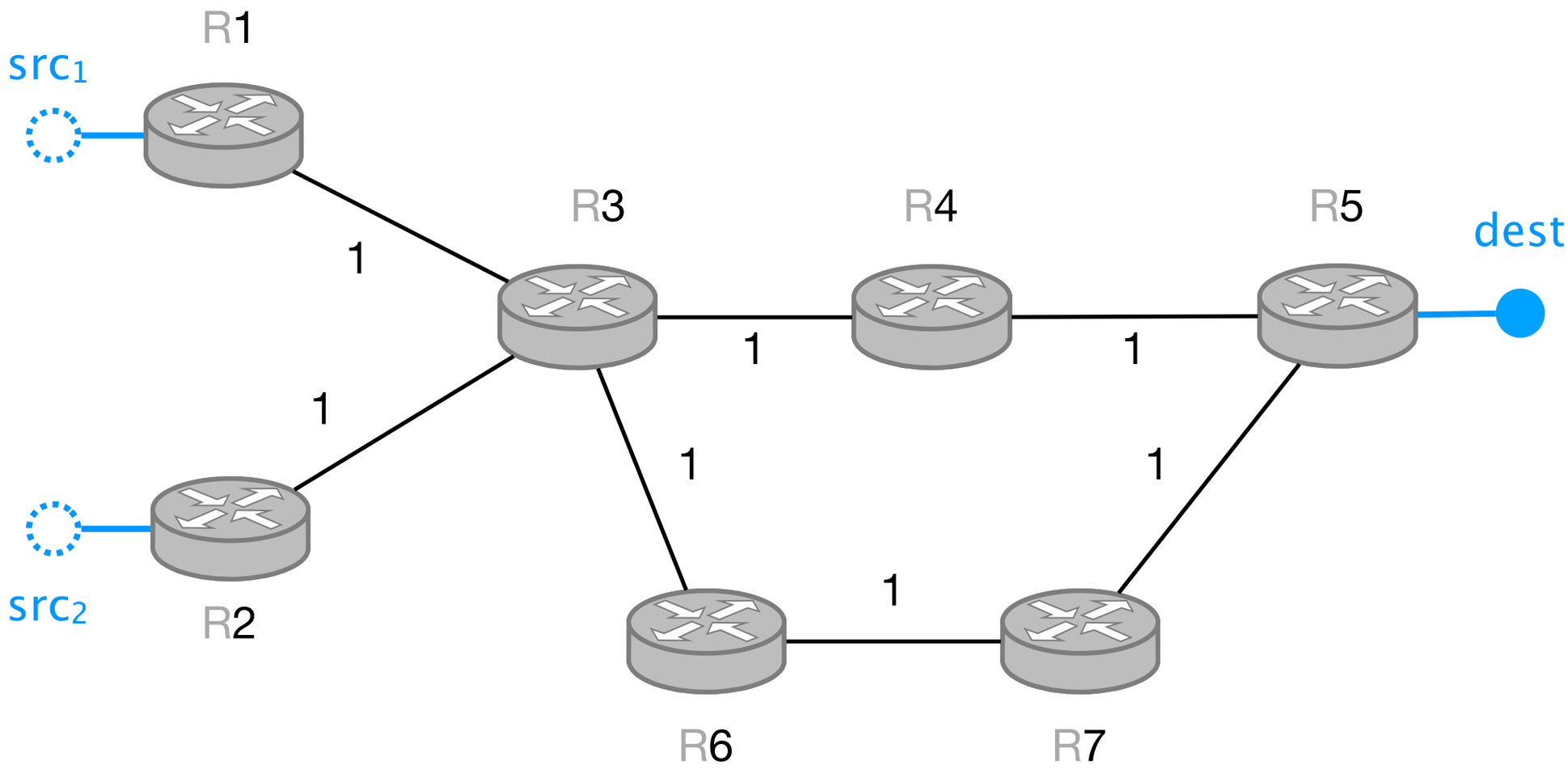
Equal-Cost Multi-Path (ECMP) is a routing strategy in which IP routers splits traffic over all best paths

Equal-Cost Multi-Path (ECMP) is a routing strategy in which IP routers splits traffic over all best paths

in OSPF, best == shortest

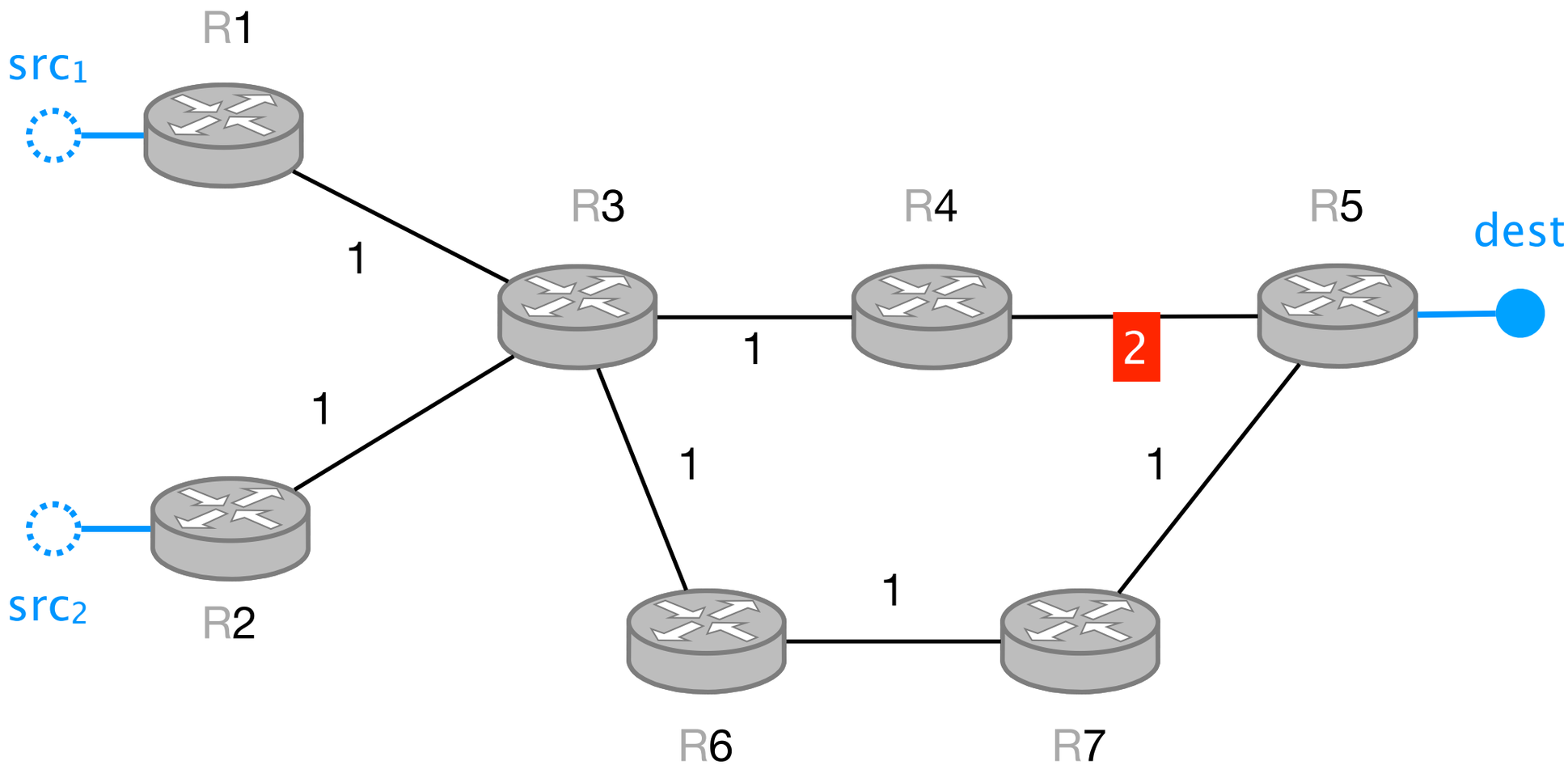
What can we do better?

What about better weights?



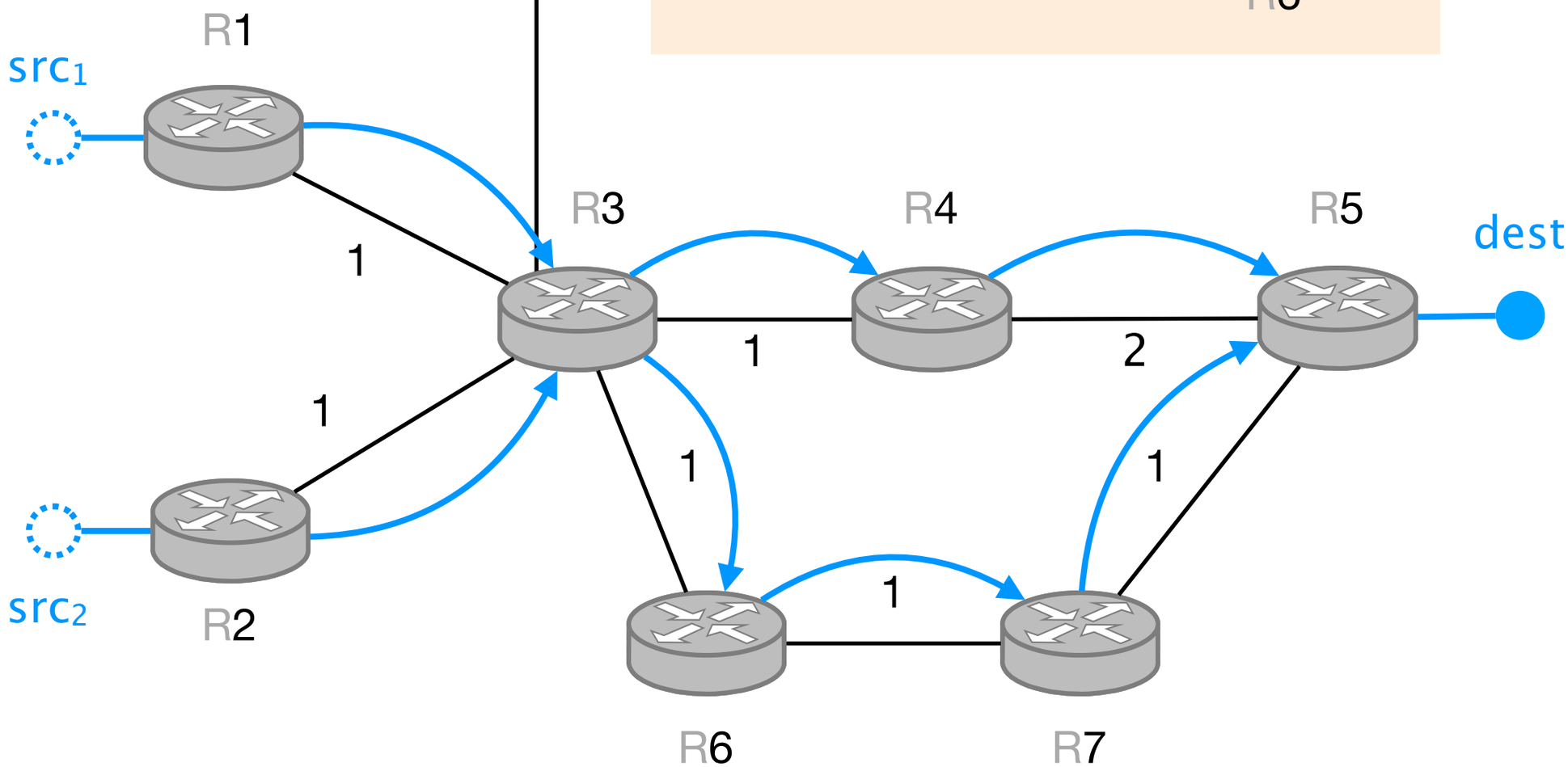
What can we do better?

What about better weights?



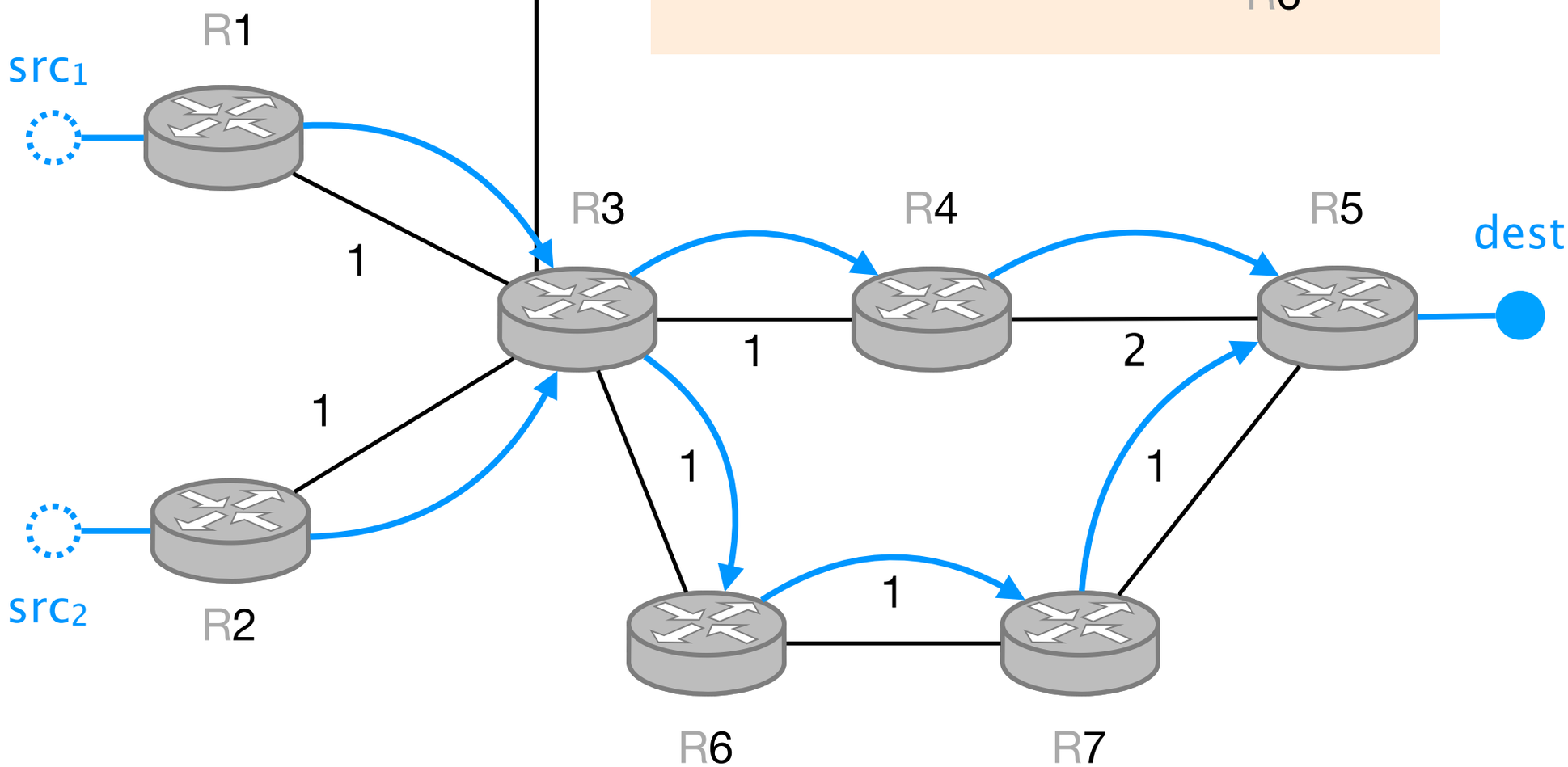
R3's routing table

pfx	min_cost	next_hop
dest	3	R4
		R6



R3's routing table

pfx	min_cost	next_hop
dest	3	R4
		R6



Is it enough to guarantee we fully utilize the network? Nope…

Equal-Cost Multi-Path (ECMP) is a routing strategy in which IP routers splits traffic over all best paths

how?

Equal-Cost Multi-Path (ECMP) typically relies on **stateless hash functions** to split traffic

$$\text{ECMP next\_hop} = \text{hash}\begin{bmatrix}\text{src\_ip}\\ \text{dst\_ip}\\ \text{src\_port}\\ \text{dst\_port}\\ \text{proto}\end{bmatrix} \% \text{\# next\_hops}$$

Property

All packets belonging to the same TCP flows go over the same path

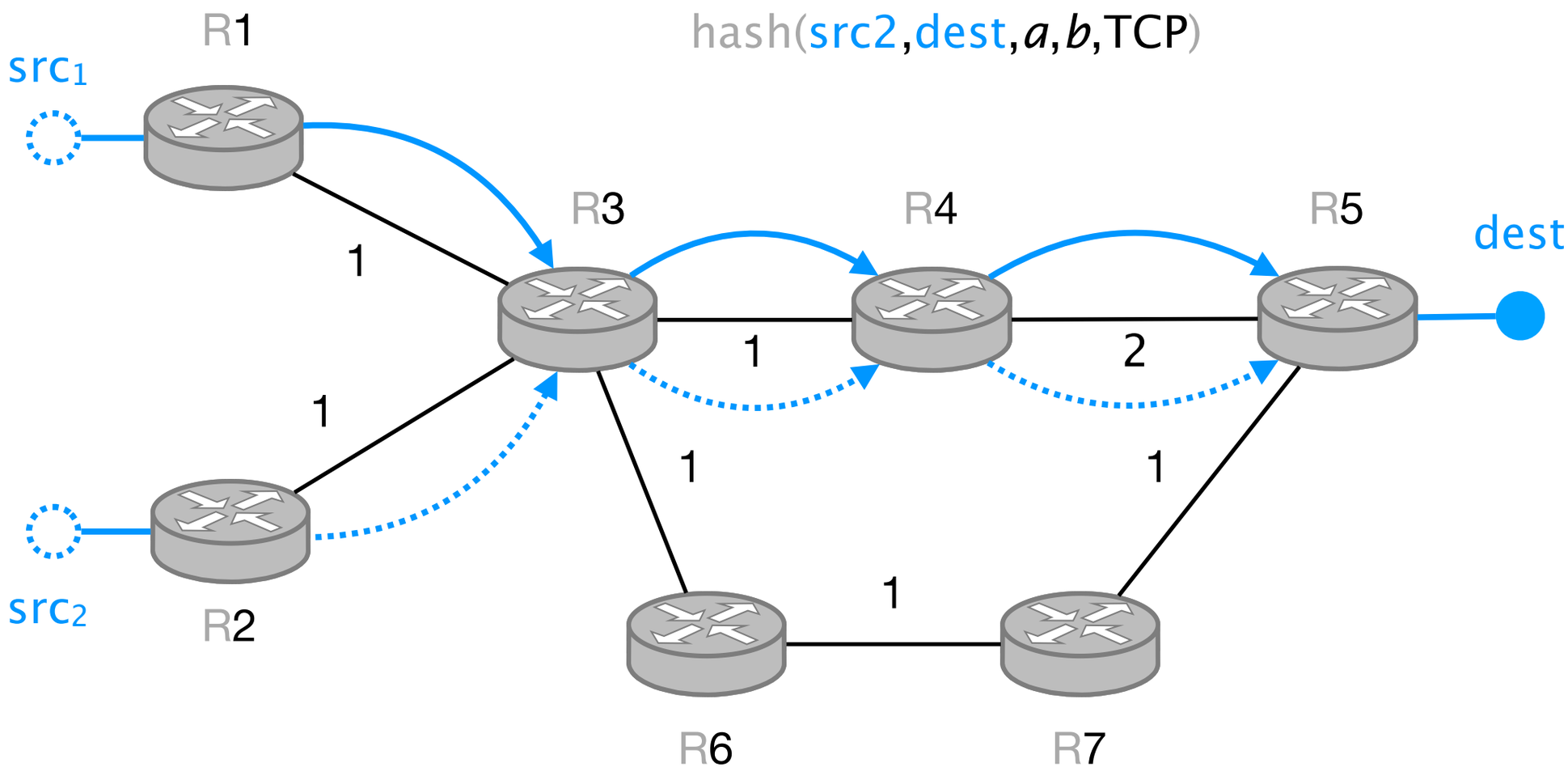
(doing so avoids reordering)

What can go wrong?

Hash collisions!

hash(src1,dest,*x*,*y*,TCP) = 0

hash(src2,dest,*a*,*b*,TCP)

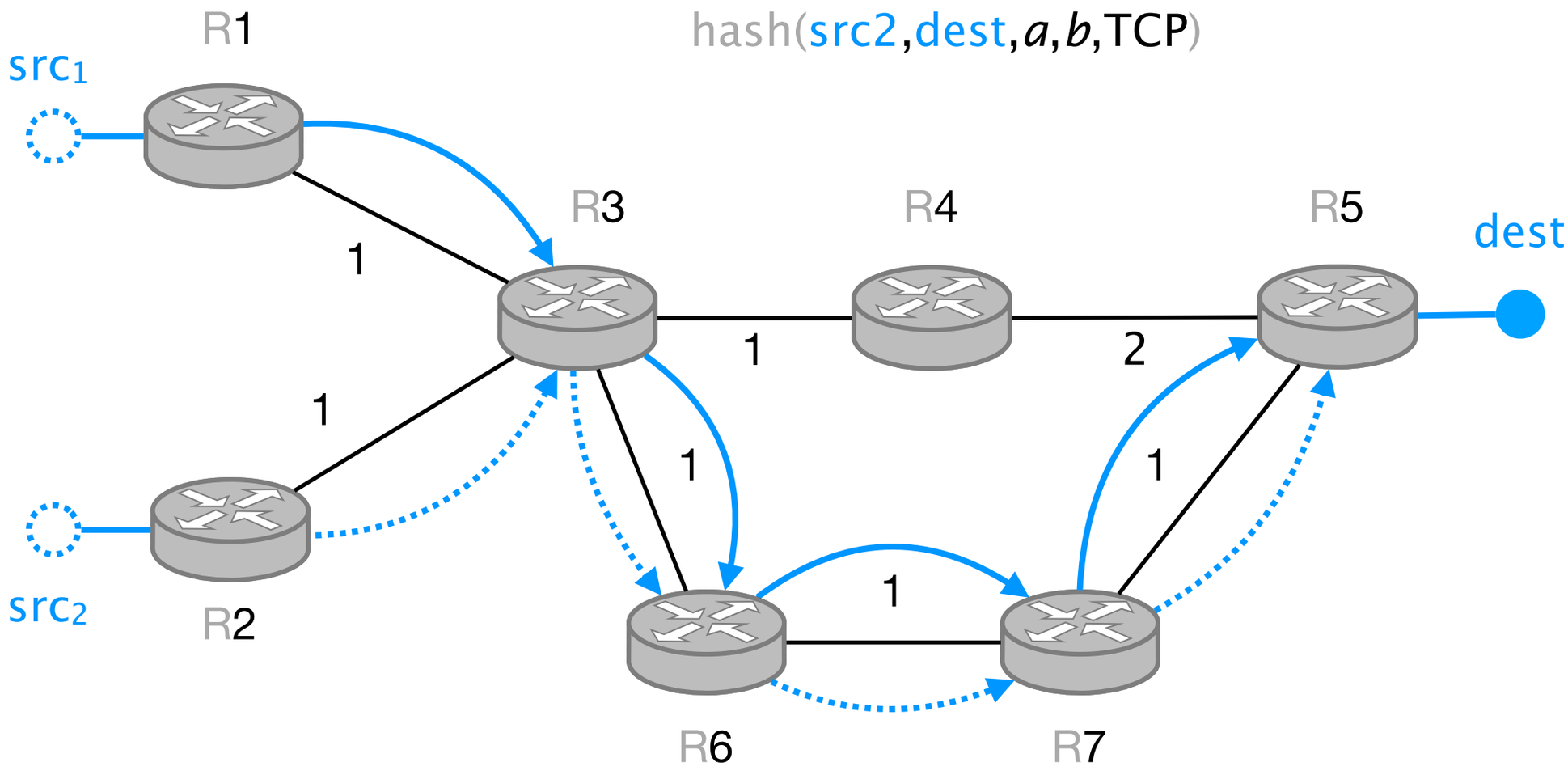


What can go wrong?

Hash collisions!

hash(src1,dest,*x*,*y*,TCP) = 1

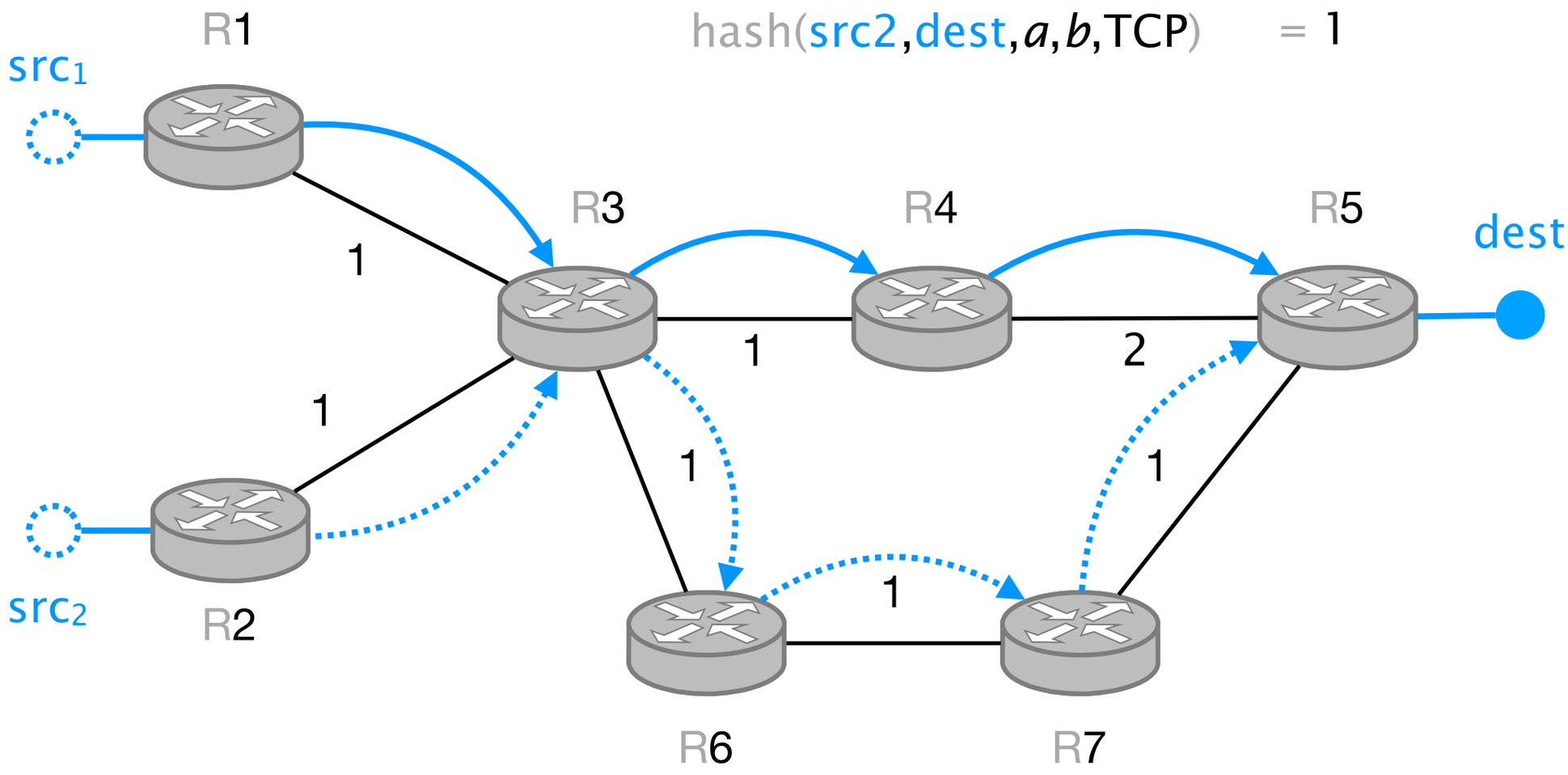
hash(src2,dest,*a*,*b*,TCP)



Traffic is effectively load balanced when the hash don't match

hash(src1,dest,*x*,*y*,TCP) = 0 or vice-versa

hash(src2,dest,*a*,*b*,TCP) = 1



We'll see different ways to solve this problem including:

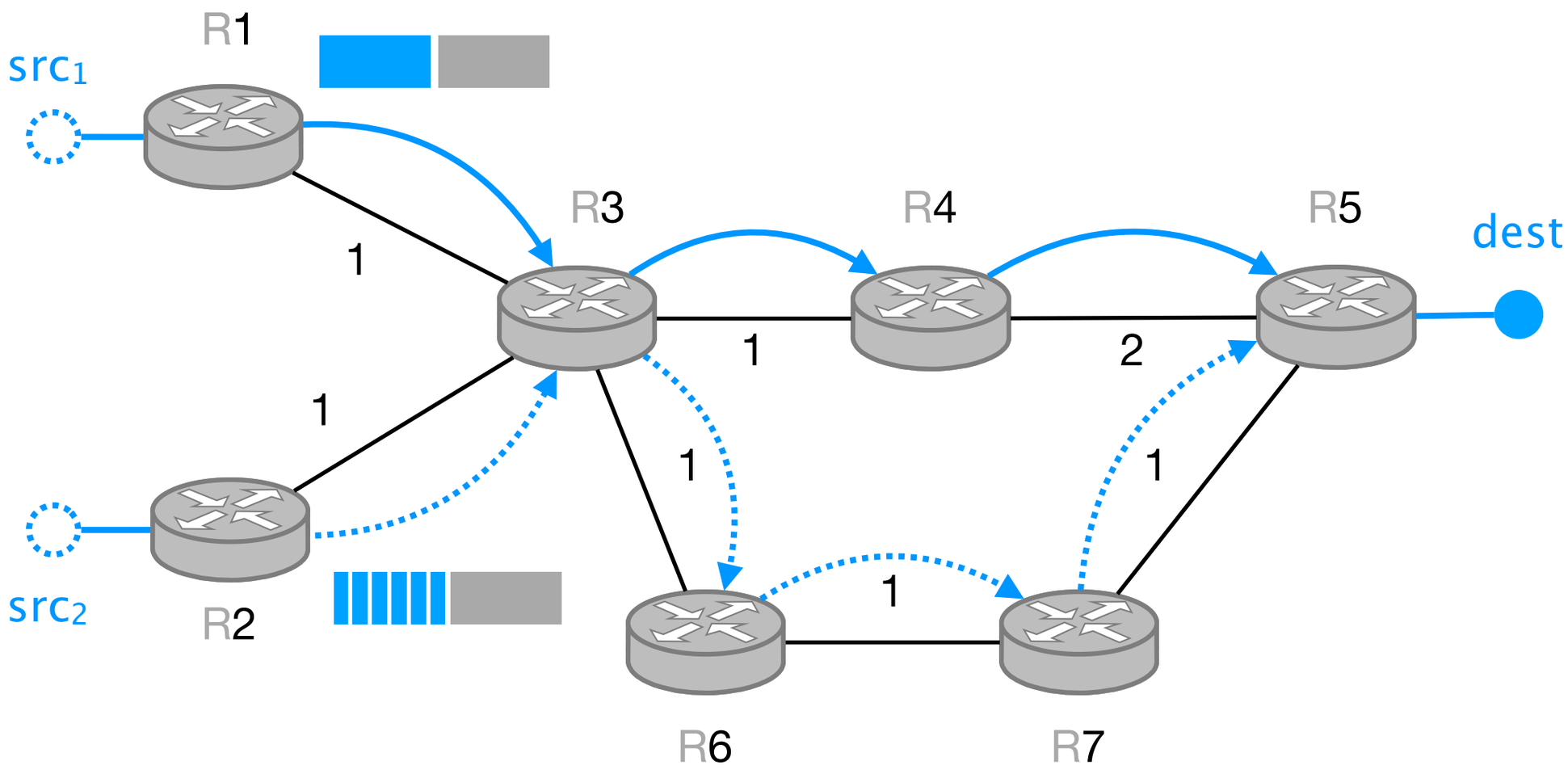
smarter load-balancing functions and label switching

We'll see different ways to solve this problem including:
smarter load-balancing functions and label switching

next_hop = function

- queue occupancy
- downstream path load
- packet interarrival time
- random
- ...

We'll see different ways to solve this problem including:
smarter load-balancing functions and label switching



Techniques

Performance

Traffic Engineering

Load Balancing

Quality of Service

Multicast

Flexibility

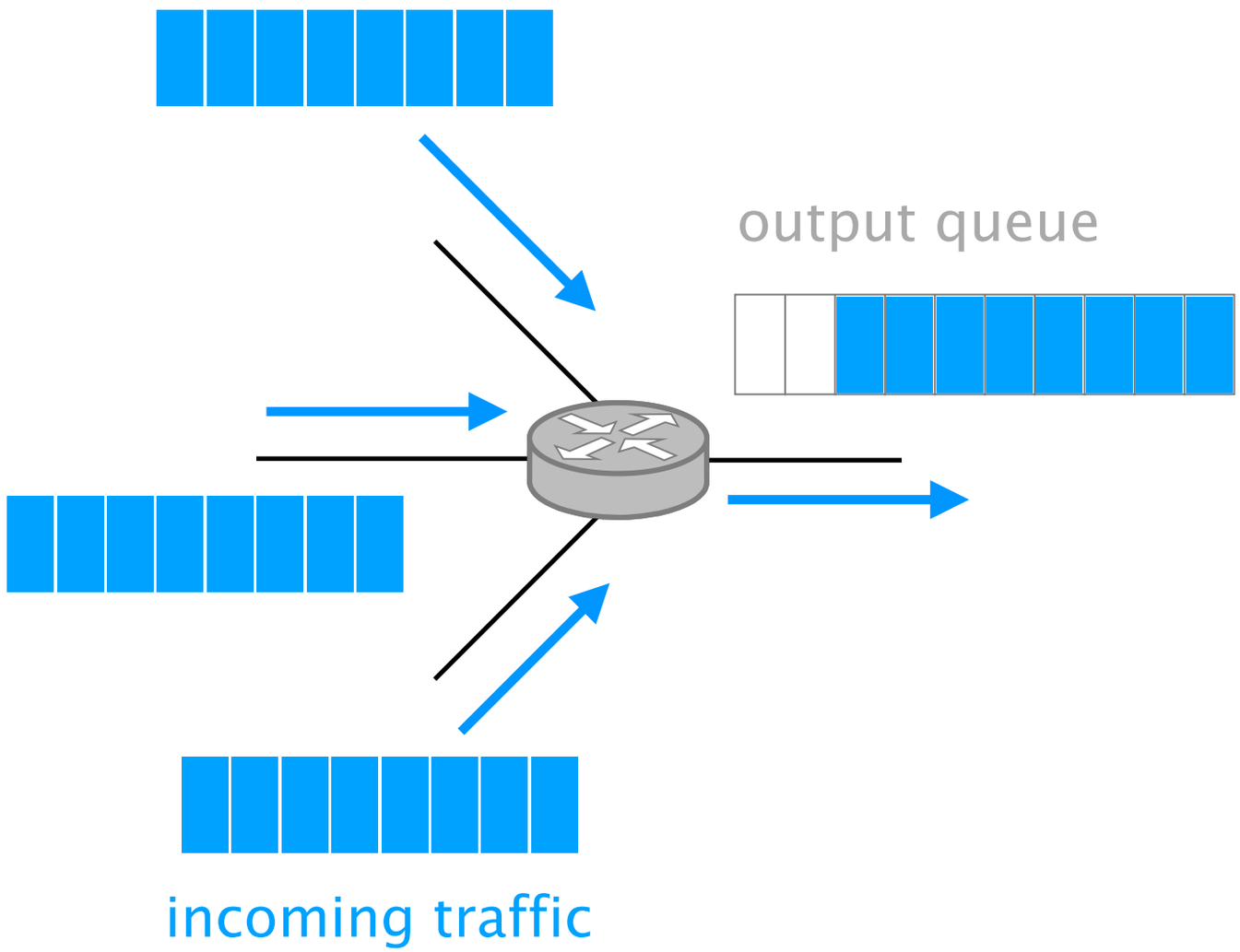
Virtual Private Networks

Reliability

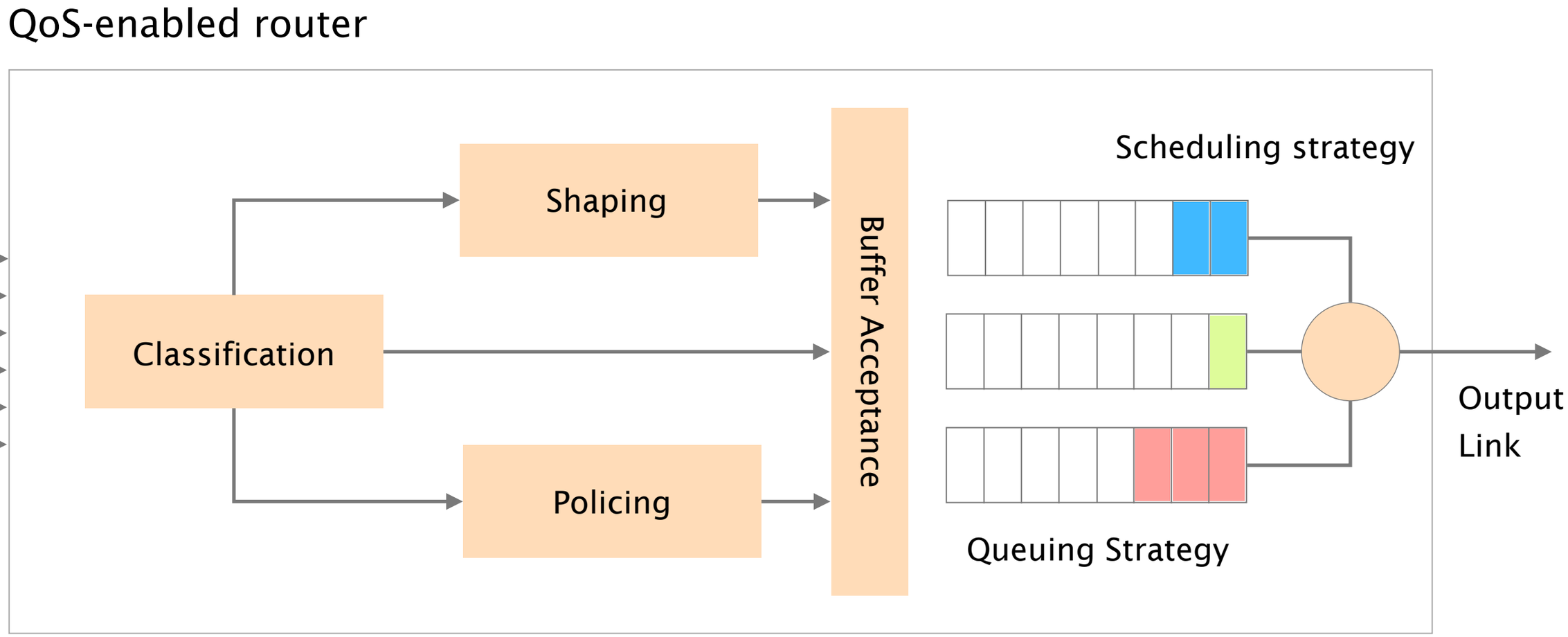
Fast Convergence

While traffic engineering helps,

it cannot always prevent congestion



We'll see different ways to manage congestion using Quality of Service (QoS)



Techniques

Performance

Traffic Engineering

Load Balancing

Quality of Service

Multicast

Flexibility

Virtual Private Networks

Reliability

Fast Convergence

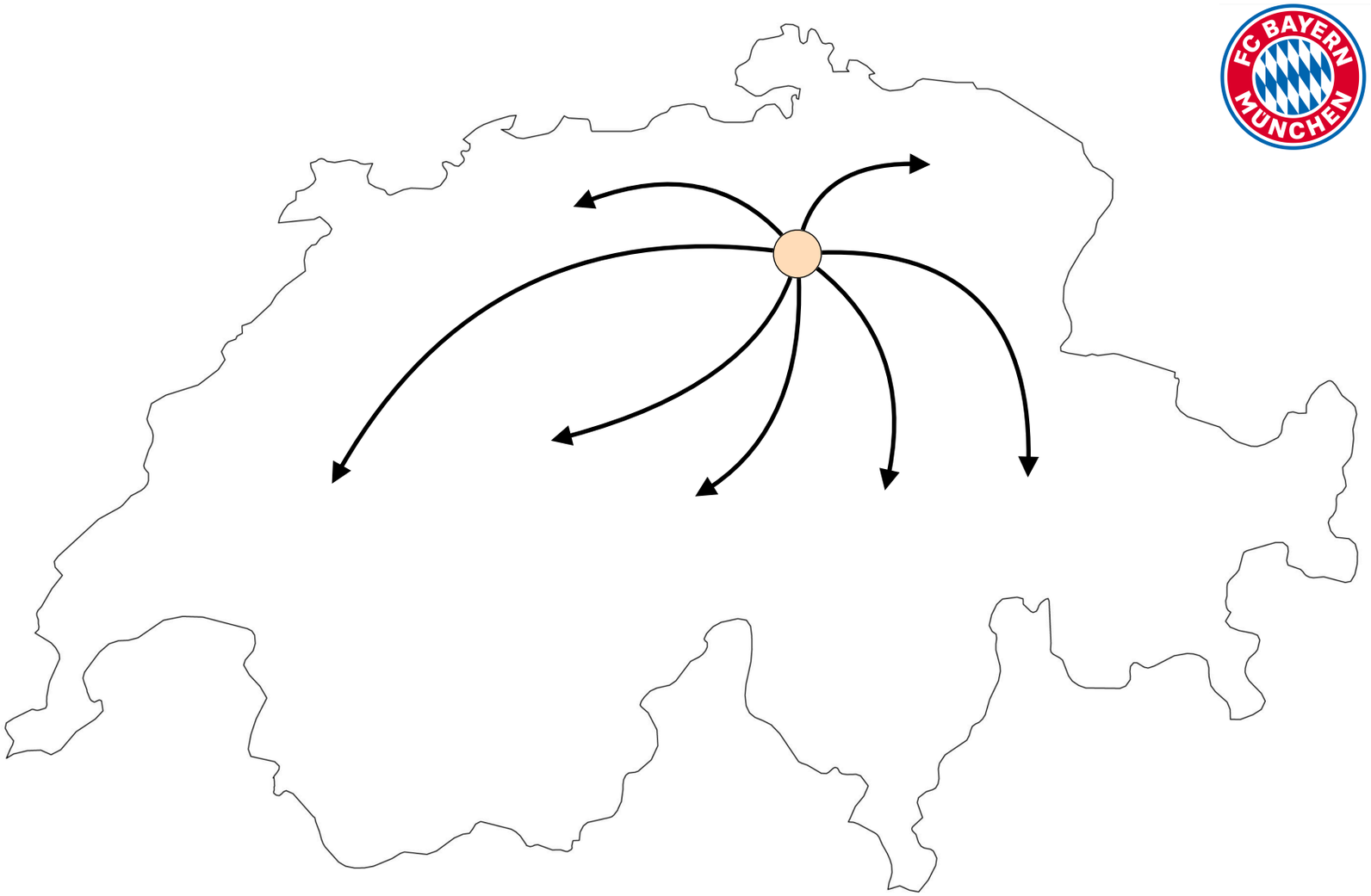
How do we deliver live videos to millions of clients?

Champions League final

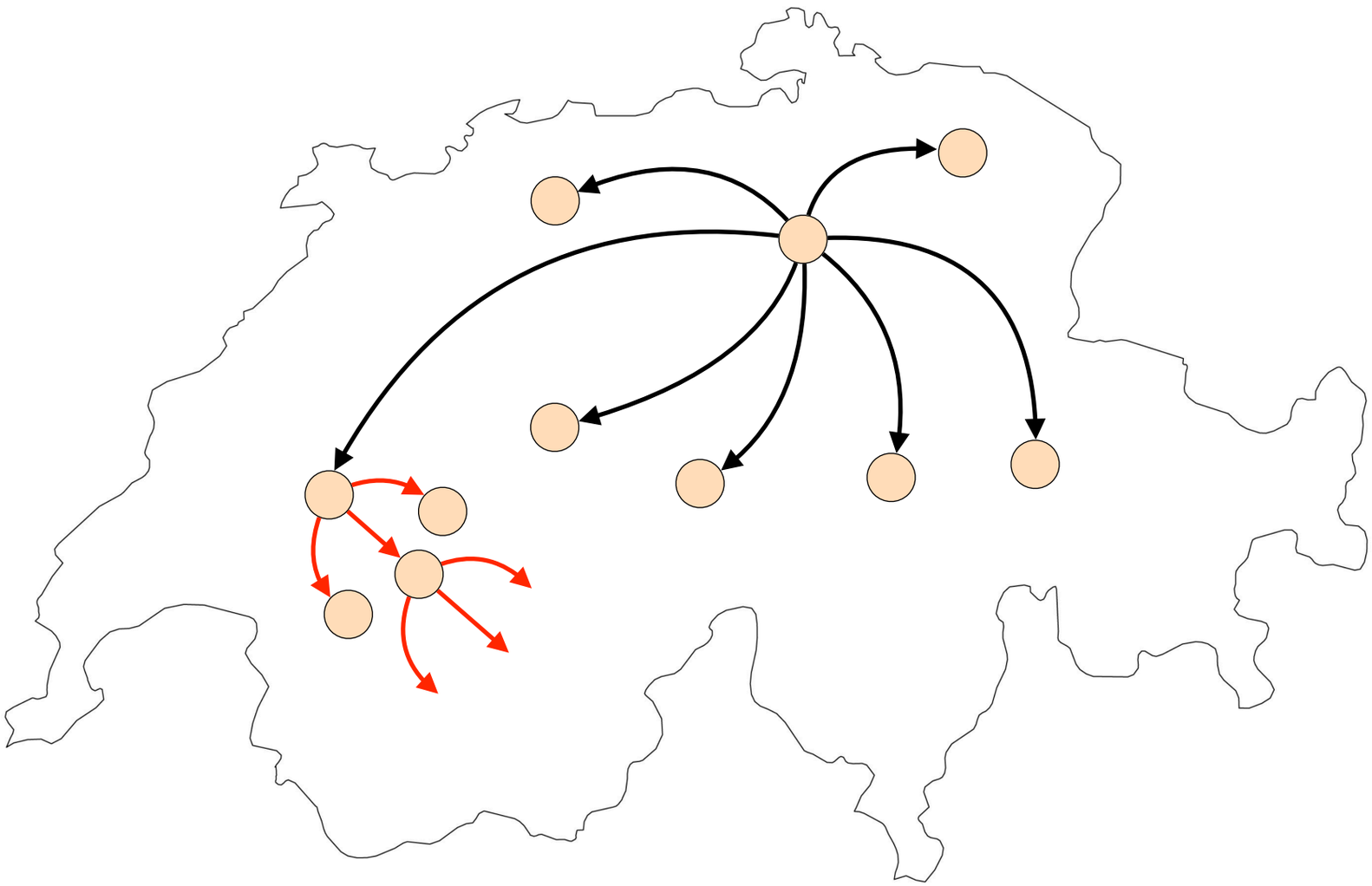
vs

	million viewers average		Mbps/feed {HD, UHD}		Tbps
France	11.1	x	{8, 25}	=	{89, 279}
Germany	13.84				{110, 346}

How do we deliver live videos to millions of clients?



We'll see how the network itself can help doing that using IP multicast



Techniques

Performance

Traffic Engineering

Load Balancing

Quality of Service

Multicast

Flexibility

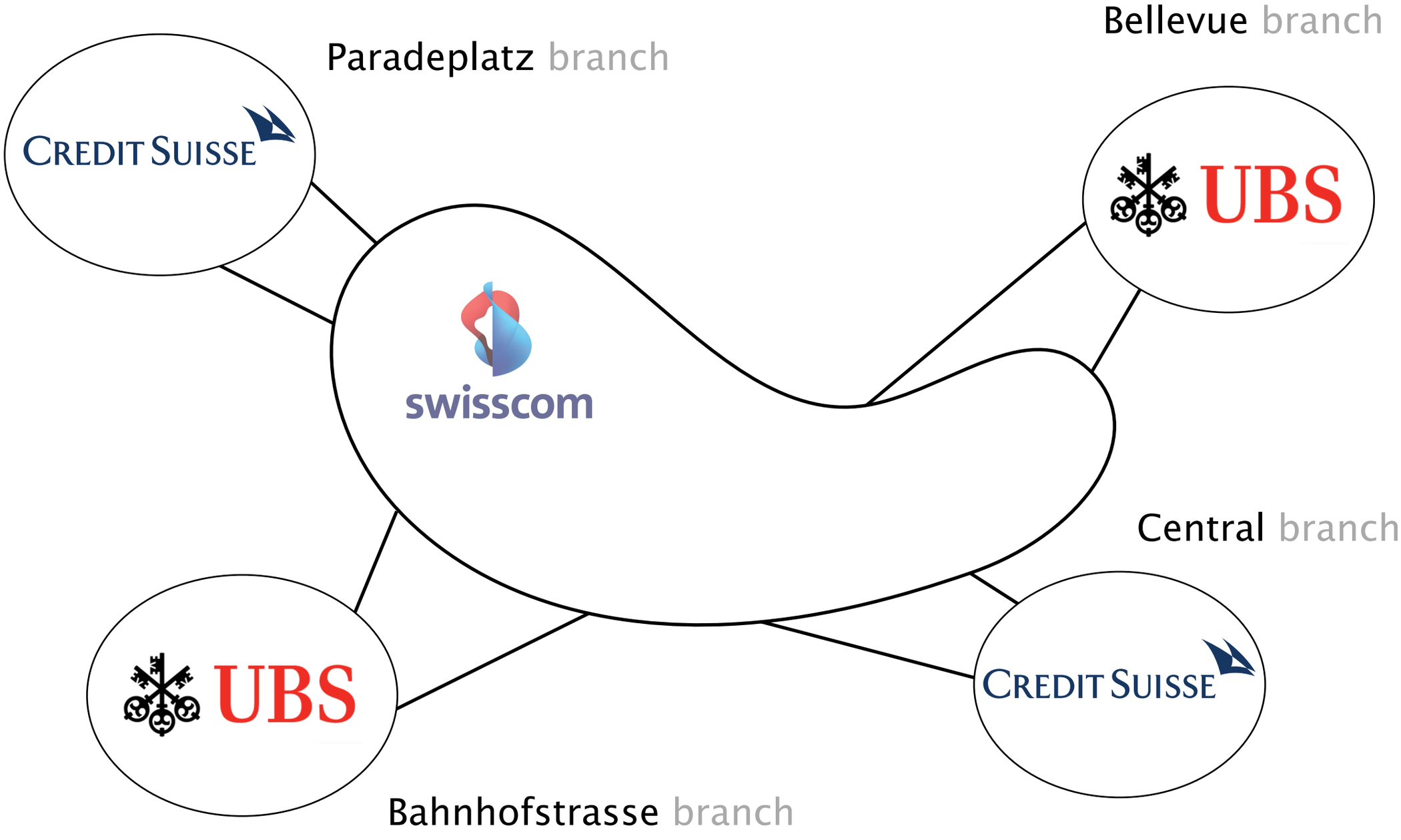
Virtual Private Networks

Reliability

Fast Convergence

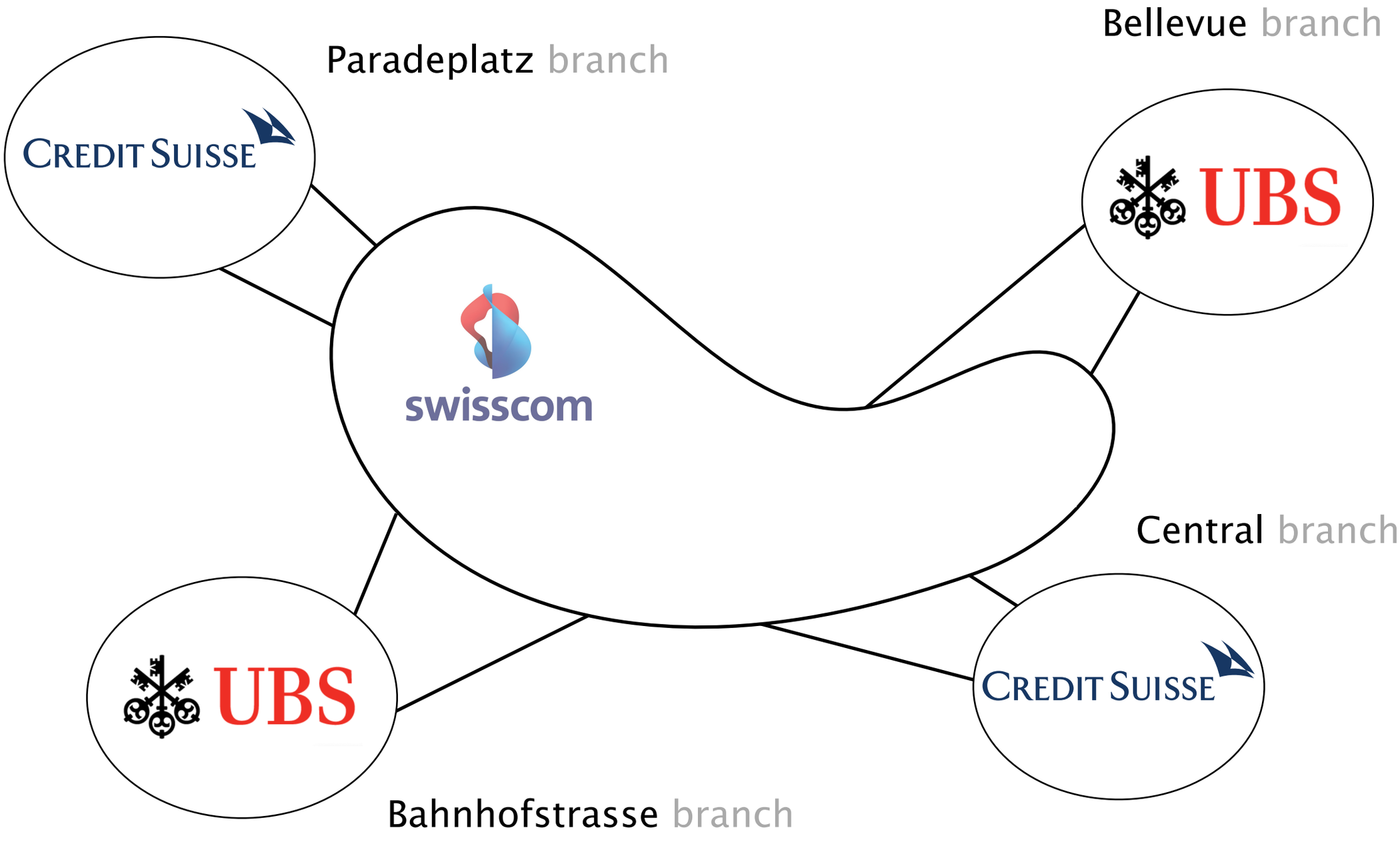
How do we virtualize a network infrastructure?

Consider this classical scenarios in which a provider (Swisscom)
would like to interconnect multiple sites with each other



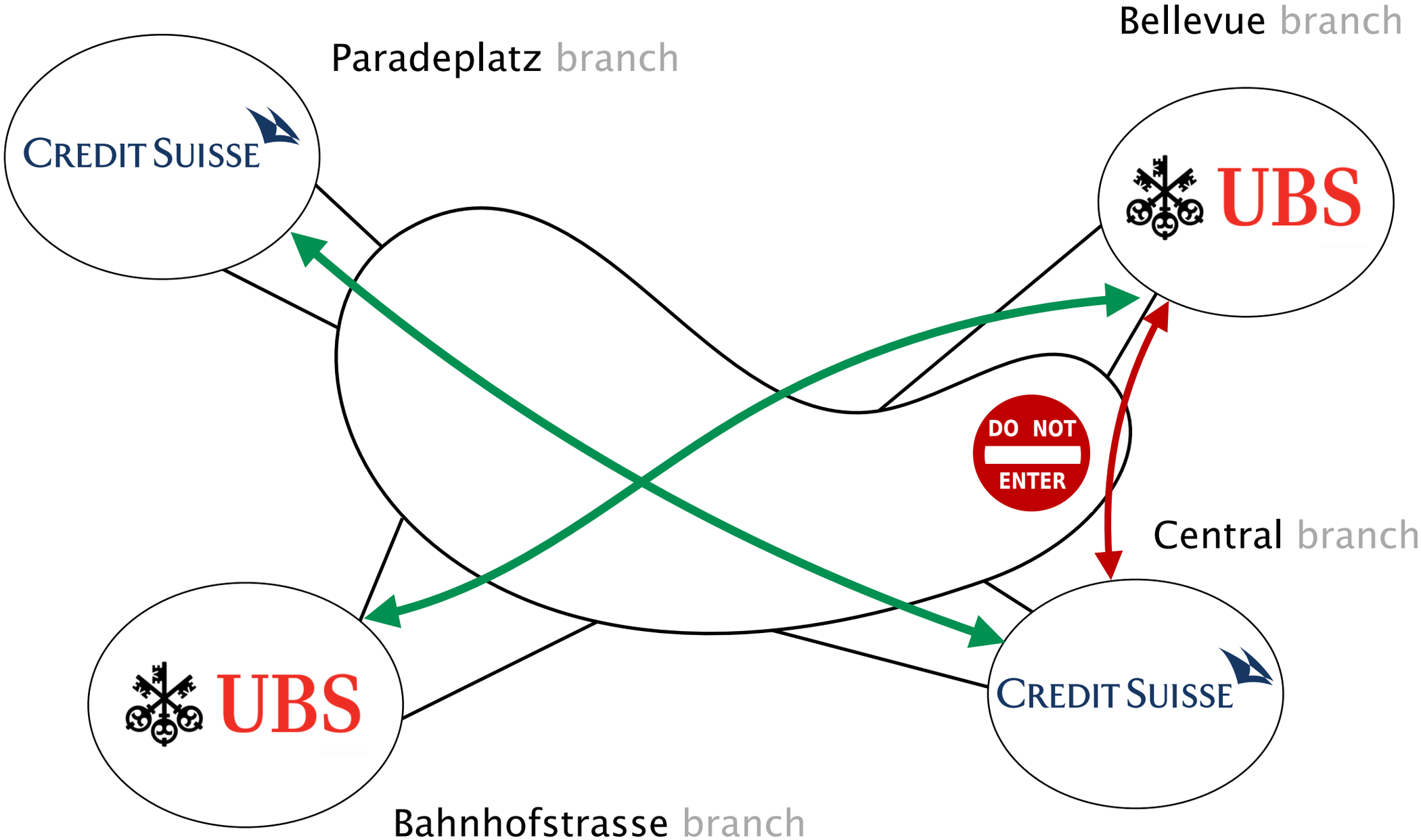
For obvious reasons…

Credit Suisse (resp. UBS) sites should *only* be able to talk to each other

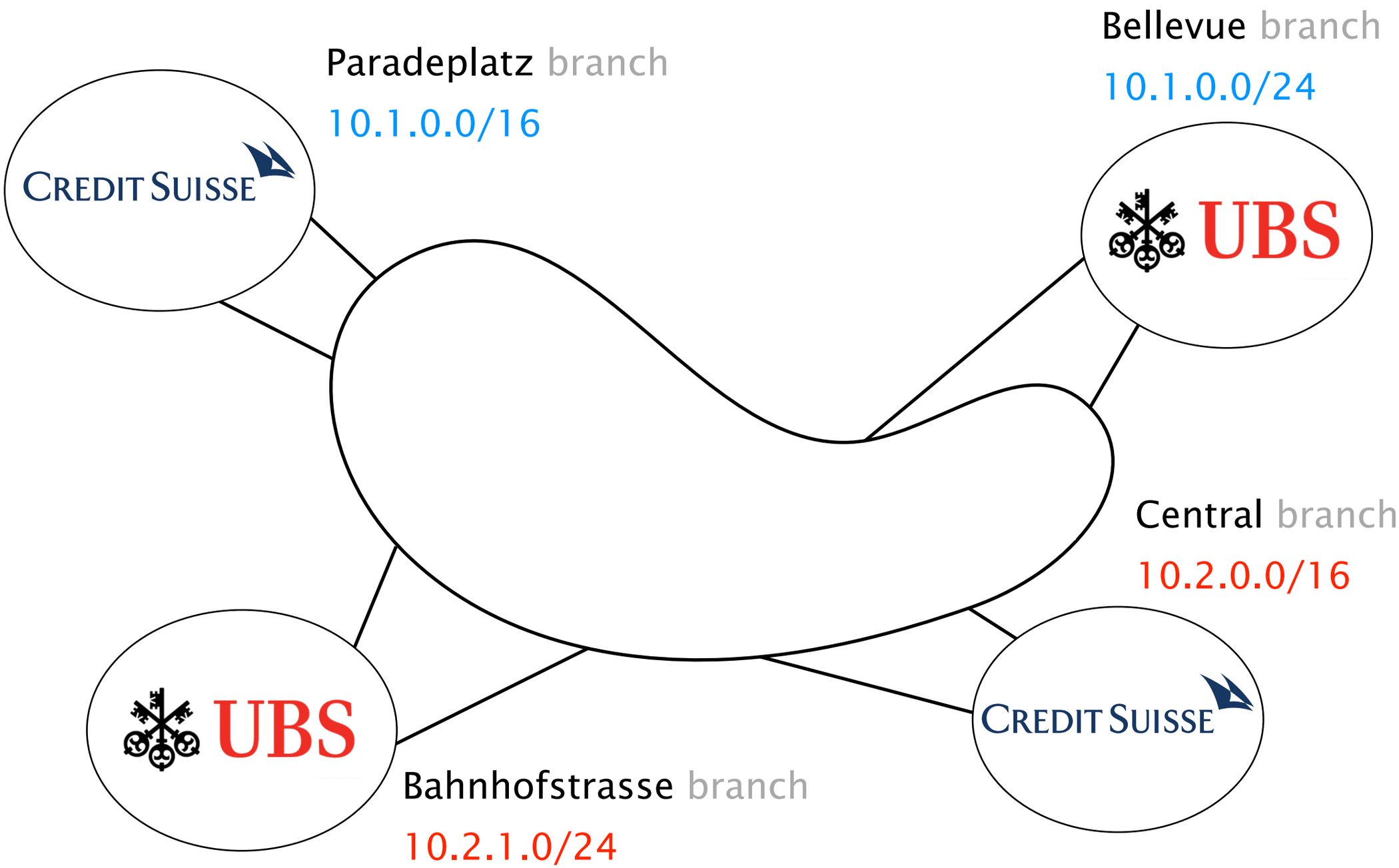


For obvious reasons…

Credit Suisse (resp. UBS) sites should *only* be able to talk to each other



How does Swisscom enable such service given that Credit Suisse and UBS address space can overlap?



Paradeplatz branch

10.1.0.0/16

Bellevue branch

10.1.0.0/24

IP packet to

10.1.0.1

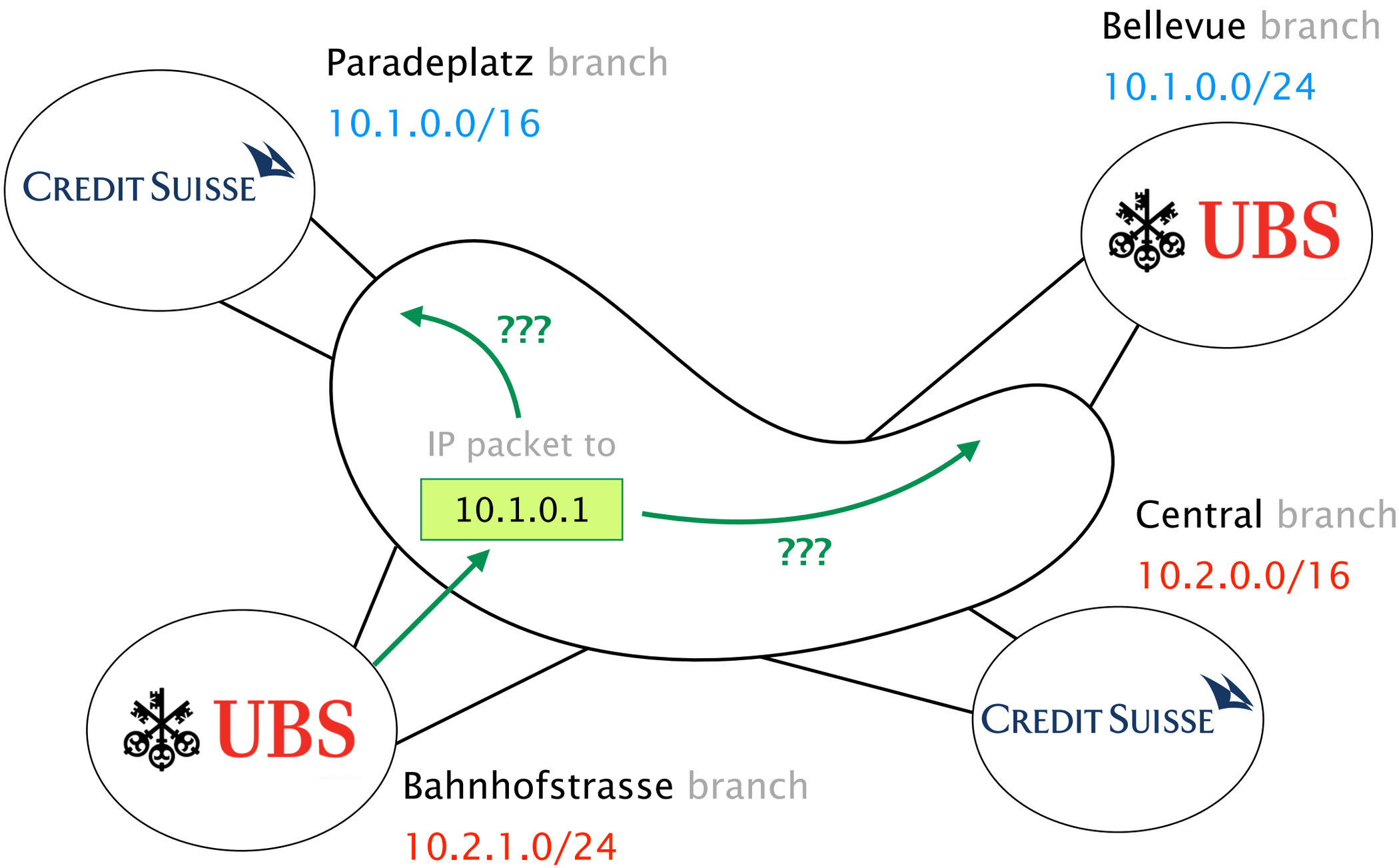
Central branch

10.2.0.0/16

Bahnhofstrasse branch

10.2.1.0/24

How does Swisscom enable such service given that Credit Suisse and UBS address space can overlap?



We'll see different how ISPs achieve this using label switching again

Paradeplatz branch

10.1.0.0/16

CREDIT SUISSE

label

Bellevue branch

10.1.0.0/24

UBS

label

Central branch

10.2.0.0/16

label

UBS

Bahnhofstrasse branch

10.2.1.0/24

label

CREDIT SUISSE

Paradeplatz branch

10.1.0.0/16

CREDIT SUISSE

label

Bellevue branch

10.1.0.0/24

UBS

label

IP packet to

10.1.0.1

Central branch

10.2.0.0/16

label

UBS

label

CREDIT SUISSE

Bahnhofstrasse branch

10.2.1.0/24

Techniques

Performance

Traffic Engineering

Load Balancing

Quality of Service

Multicast

Flexibility

Virtual Private Networks

Reliability

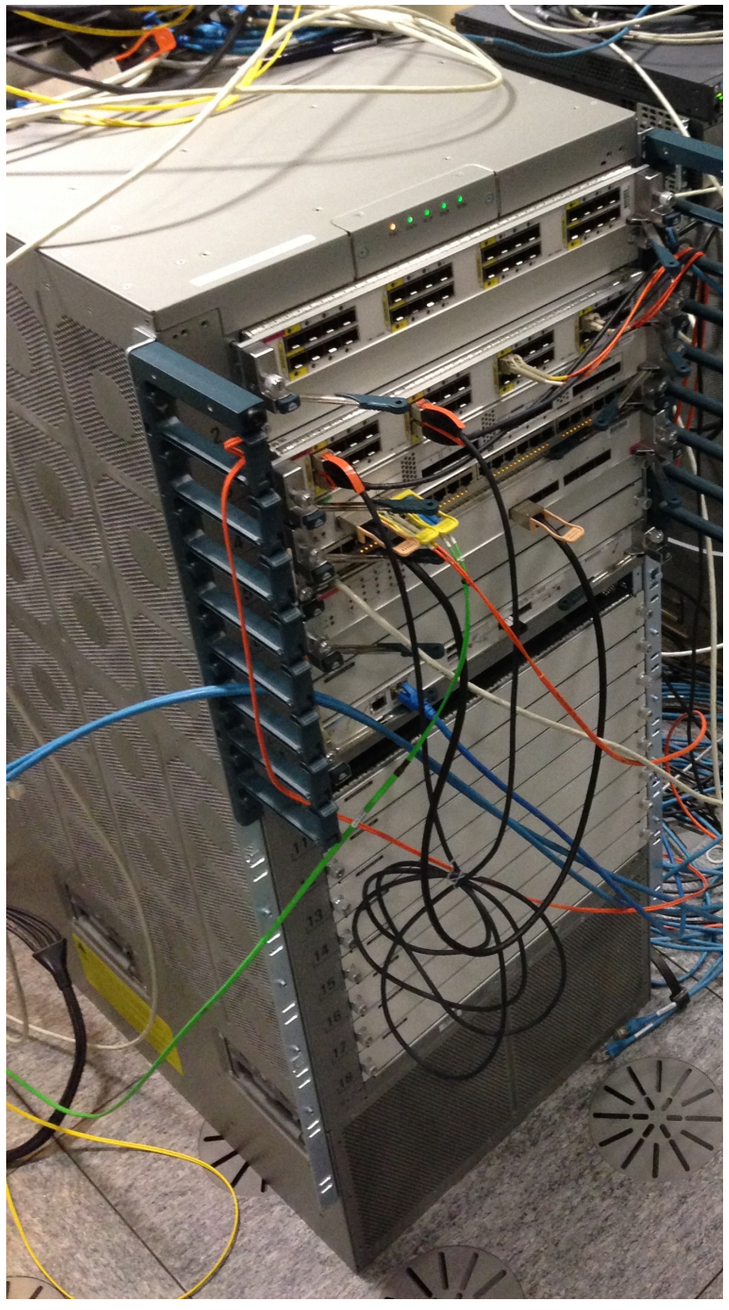
Fast Convergence

IP networks converge slowly upon failures

at least, by default

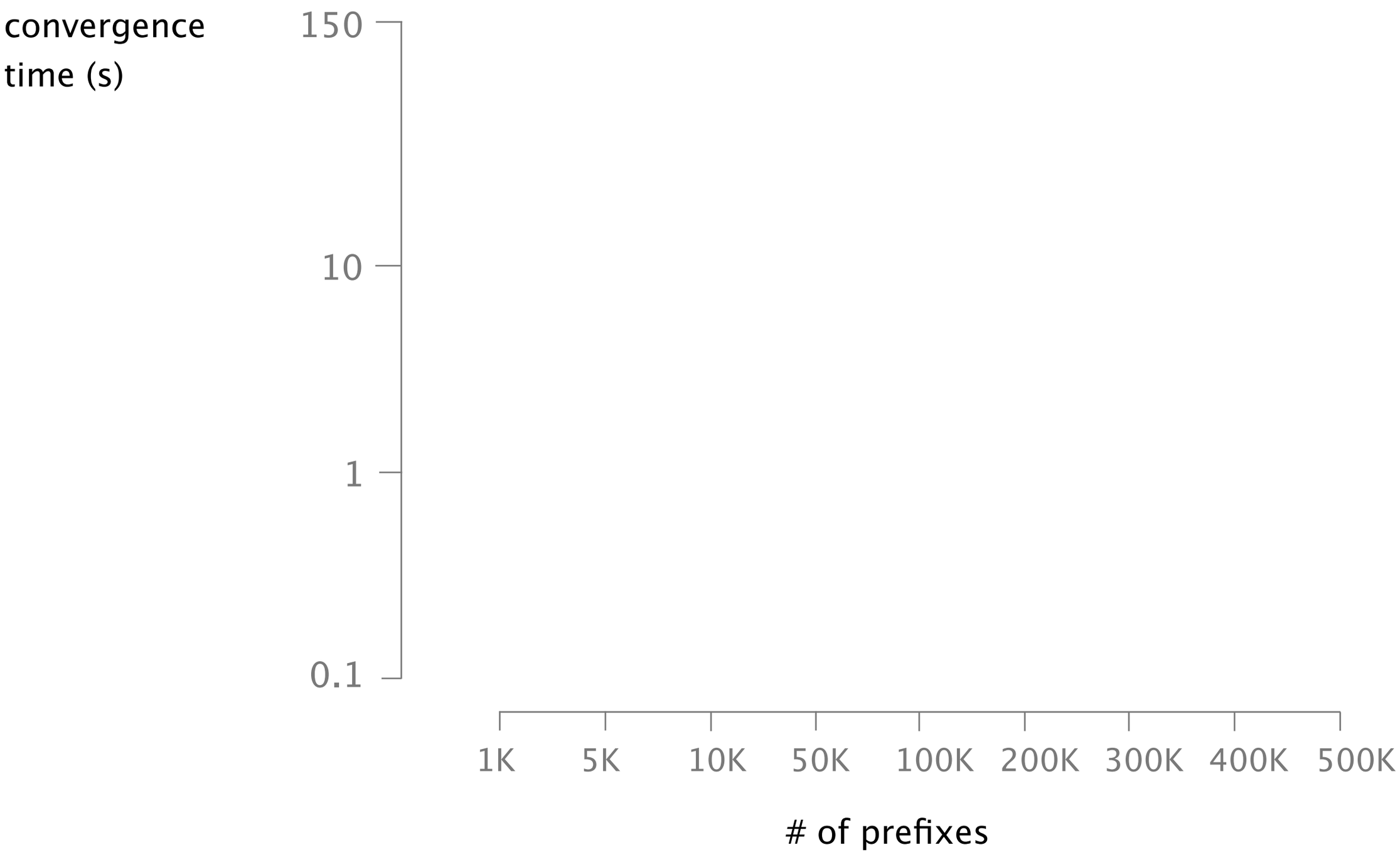
We measured how long it takes for

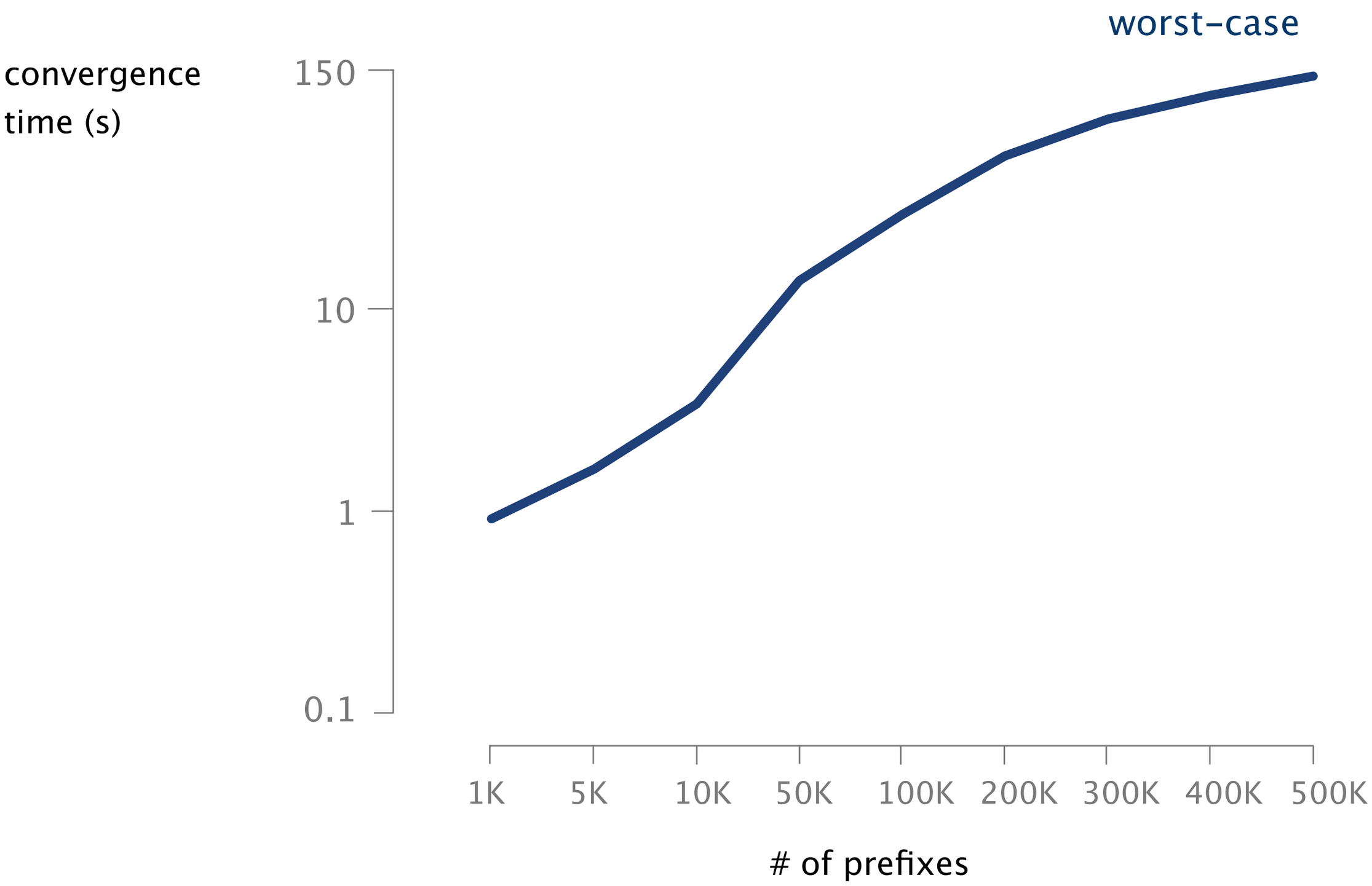
an ETH router to converge

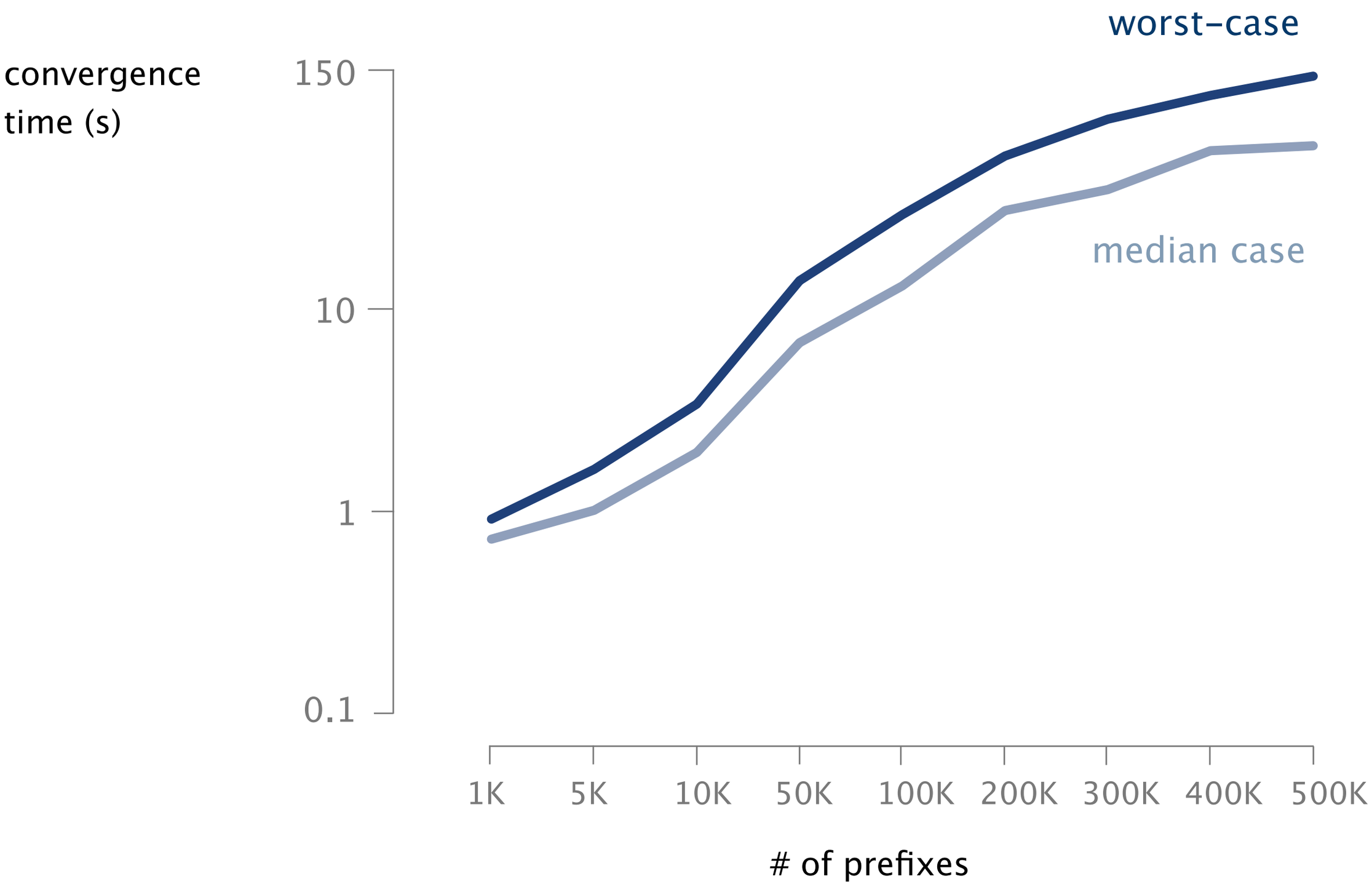


Cisco Nexus 9k

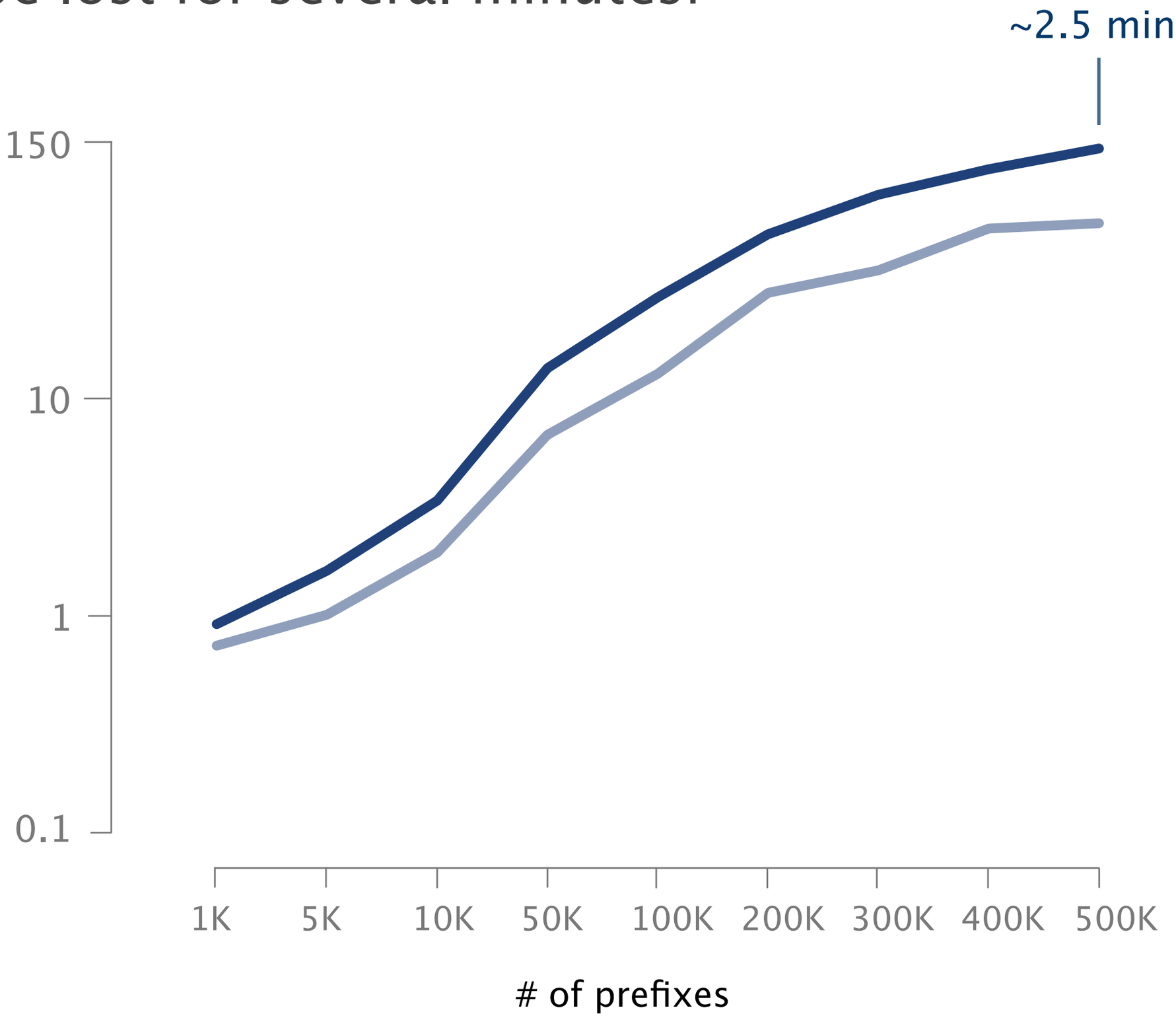
ETH's recent routers


convergence
time (s)
150
10
1
0.1
1K
5K
10K
50K
100K
200K
300K
400K
500K
# of prefixes


worst-case
convergence
time (s)
150
10
1
0.1
1K
5K
10K
50K
100K
200K
300K
400K
500K
# of prefixes


convergence
time (s)
150
10
1
0.1
worst-case
median case
1K
5K
10K
50K
100K
200K
300K
400K
500K
# of prefixes

Traffic can be lost for several minutes!



Why does it take *so* long?

	detect the failure	ms
	report the failure	ms
	generate/flood routing messages	10s of ms
	recompute routing paths	10s of ms
bottleneck	communicate next-hops to the line cards	100s of ms
	install new next-hops	10s of ms

We'll see different Fast Reroute techniques
to speed up convergence time

Fast Reroute
technologies

detect the failure

report the failure

generate/flood routing messages

recompute routing paths

communicate next-hops to the line cards

install new next-hops

hw acceleration

pre-computation

pre-provisionning
fast activation

max. convergence time for *any* failure

<1 sec

Techniques

Performance

Flexibility

Reliability

Traffic Engineering

Load Balancing

Quality of Service

Multicast

Virtual Private Networks

Fast Convergence

Besides learning about the techniques per-se,
you'll learn how to operate and implement them

Besides learning about the techniques per-se,

you'll learn how to operate and implement them

operate

your own IP network

in virtual labs

implement

your own forwarding logic

in programmable networks

Until recently, network devices tended to be completely locked down

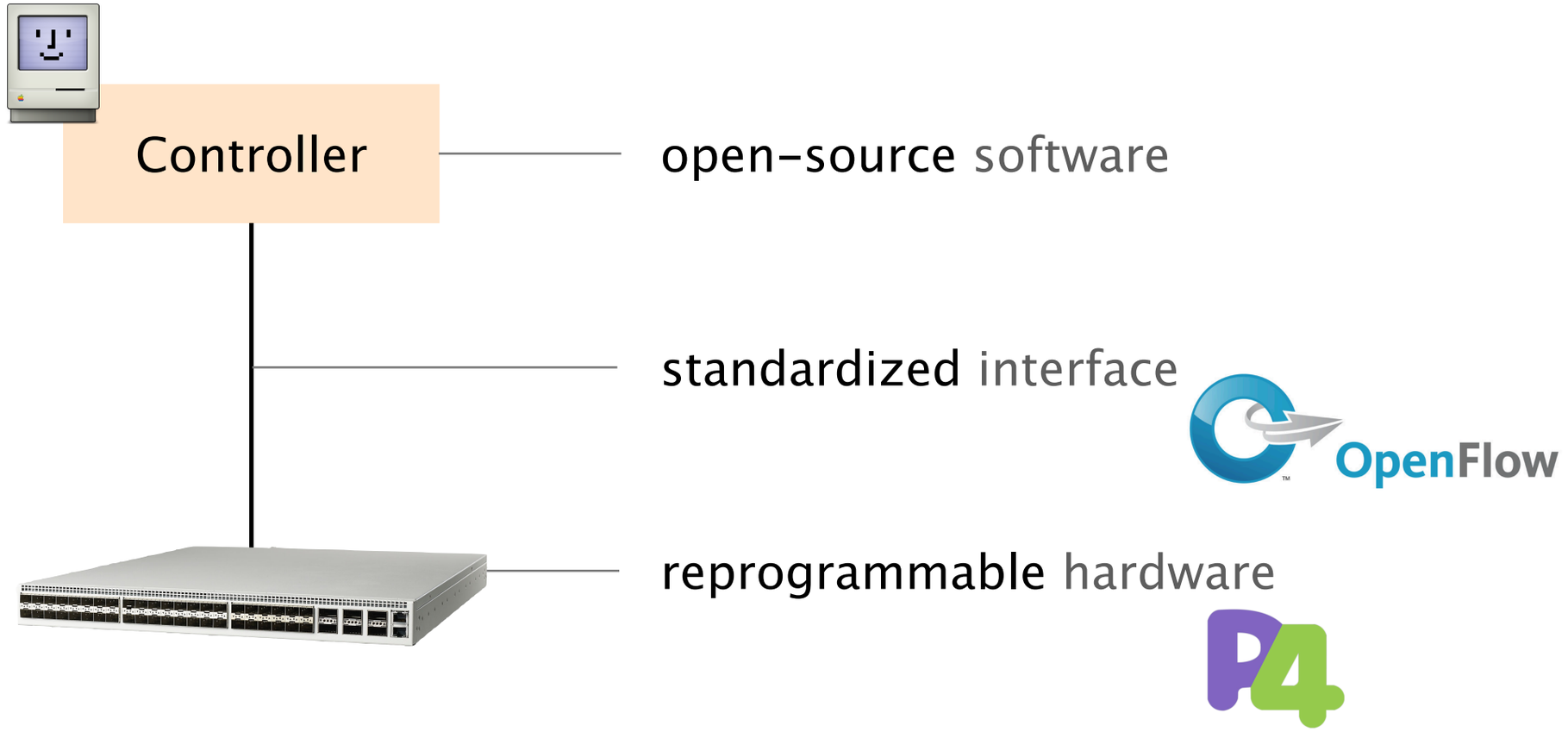
closed software

closed hardware

Cisco™ device

Things are changing though as

networks are becoming programmable



Network programmability is attracting tremendous industry interest…

VMware Acquires Once-Secretive Start-Up Nicira for $1.26 Billion

JULY 23, 2012 AT 1:25 PM PT

VMware, the software company best known for its virtualization technology that forms the backbones of so-called cloud computing today, said it will pay $1.26 billion for Nicira, a networking start-up that has sought to do to networks what VMware has done to computers.

The news comes on the same day that VMware was to report quarterly earnings. And while I don't usually cover VMware's earnings, I may as well mention the results: The company reported revenue for the quarter ended June rose to $1.12 billion, while earnings on a per-share basis were 68 cents. Analysts had been expecting sales of $1.12 billion and earnings of 66 cents.

Nicira had been running in stealth mode for quite awhile; I got to reveal its plans to the world last February.

The deal amounts to a nice payoff for Nicira's investors including Andreessen Horowitz, Lightspeed Venture Partners and NEA, as well as VMware founder Diane Greene and venture capitalist Andy Rachleff.

With $600M Invested in SDN Startups, the Ecosystem Builds

Scott Raynovich, June 10, 2014

More than $600 million has been invested in at least two dozen software-defined networking (SDN) startups so far, according to Rayno Report research. You can expect that to continue to climb. With the SDN ecosystem starting to take hold with a broad range of alliances and distribution partnerships, we're just getting started.

The Arista IPO will help build visibility for next-generation, software-driven networking. But Arista is selling its own hardware and is not an SDN pure-play. A new line of SDN startups, with a more radical approach to software-based networking, is building momentum. These newer SDN startups are just getting their gear into customers' hands and starting to build sales channels, so you can expect a long revenue ramp.

This excitement is boosting startup valuations, according to Rayno Report research. There are now at least ten SDN startups with valuations over $100 million. As I reported in April, a recent investment in Cumulus Networks pushed up the valuation of the private company north of $300 million, according to industry sources. Big Switch, which did a deal in 2012 valuing it near $170 million, took money from Intel in 2013, most likely boosting its valuation to over $200 million, according to several sources.

Related Articles

How to Effectively Embed SDN in the Enterprise

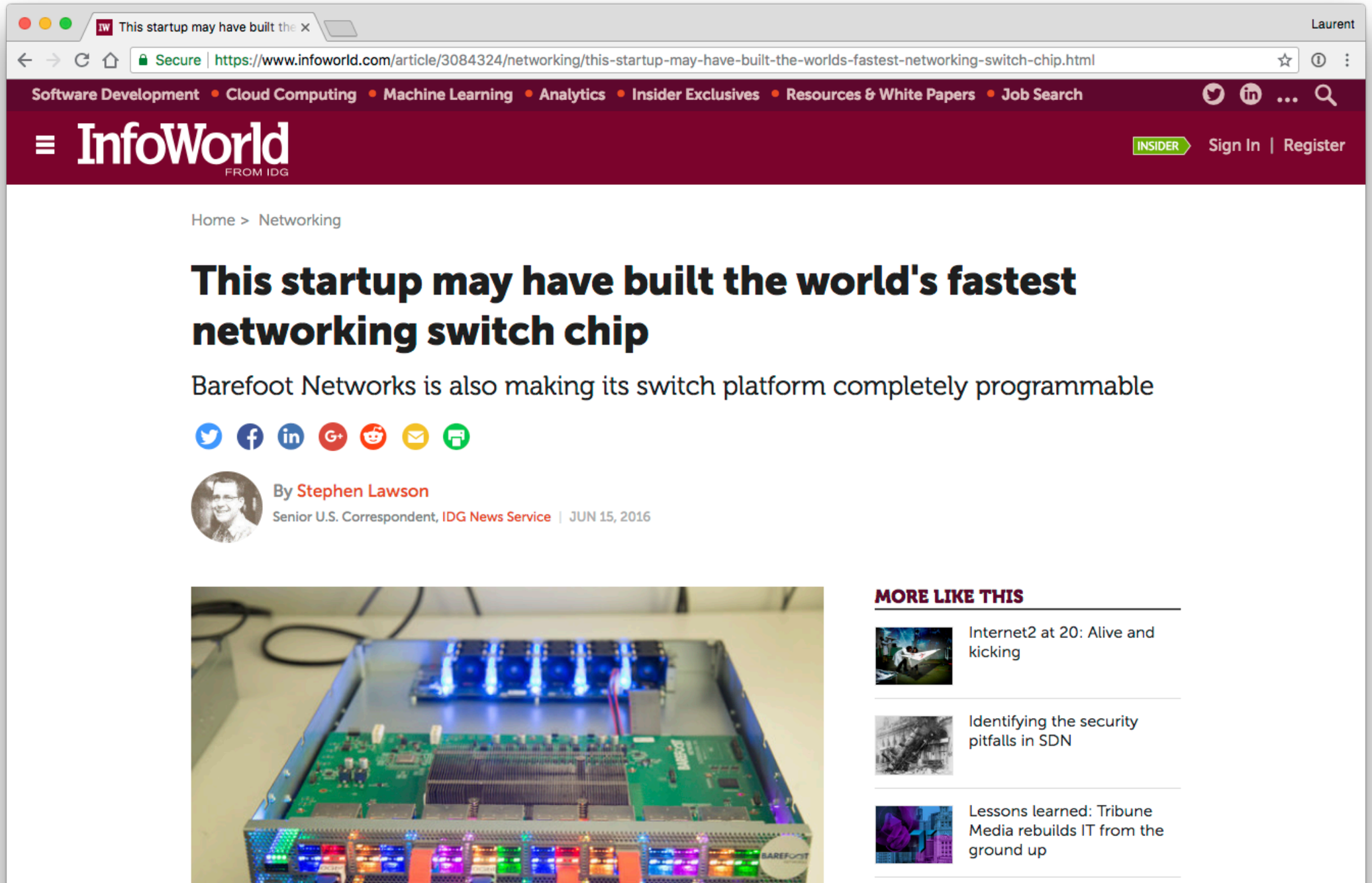
NFV and SDN: What's the Difference Two Years Later?

sFlow Creator Peter Phaal On Taming The Wilds Of SDN & Virtual Networking

Featured Article: Bringing Data-Driven SDN to the Network Edge

NFV Delivers Pervasive Intelligence for MNOs

Barefoot Networks (Stanford startup) started to produce re-programmable network hardware in 2013



In June 2019,

Barefoot Networks was acquired by Intel

THE WALL STREET JOURNAL.
Europe Edition ▾ | September 22, 2019 | Print Edition | Video

SHARE

TEXT

TECH

Intel Agrees to Acquire Networking Startup Barefoot Networks

Barefoot Networks is backed by Google, Alibaba, Tencent and Goldman Sachs

P4 is a domain-specific language which describes how a switch should process packets

https://p4.org

A sneak peek at our own networking lab

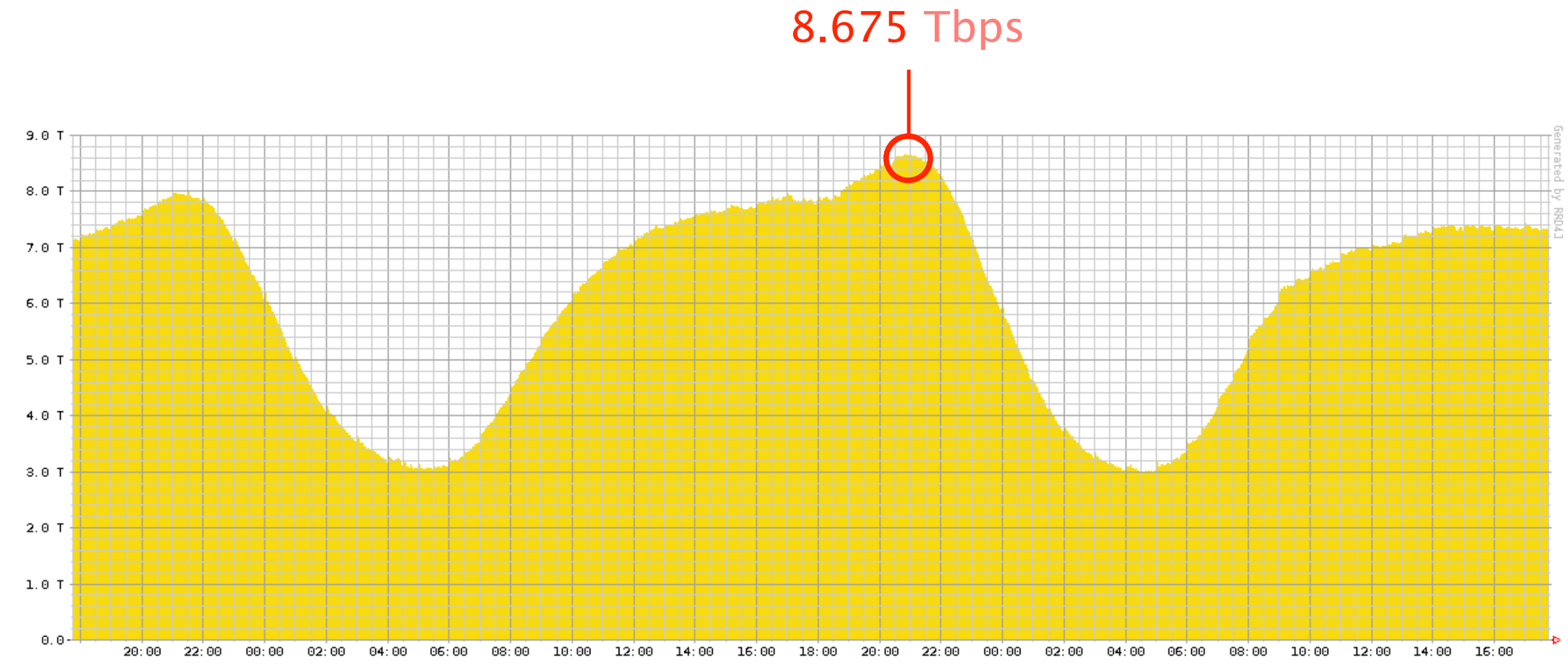
8x Wedge100BF-32X. Total capacity: 25.6 Tbps

32x QSFP28 ports

25/40/50/100 GbE

8x Wedge100BF-32X. Total capacity: 25.6 Tbps

~3x what DE-CIX sees at peak time!



https://www.de-cix.net/en/locations/germany/frankfurt/statistics

Course Organization

The course is gonna be divided in two blocks

Lectures/Exercices

~10 weeks

Group project

~4 weeks

in teams of 3

The course is gonna be divided in two blocks

Lectures/Exercices

~10 weeks

Group project

~4 weeks

in teams of 3

There will be 2h of lectures & 2h of exercises

Tue 14–16 Lecture

Tue 16–18 Practical exercises

Exercises are *not* graded *but* will help for the project & exam

Both will take place **online**

The course is gonna be divided in two blocks

Lectures/Exercices

~10 weeks

Group project

~4 weeks

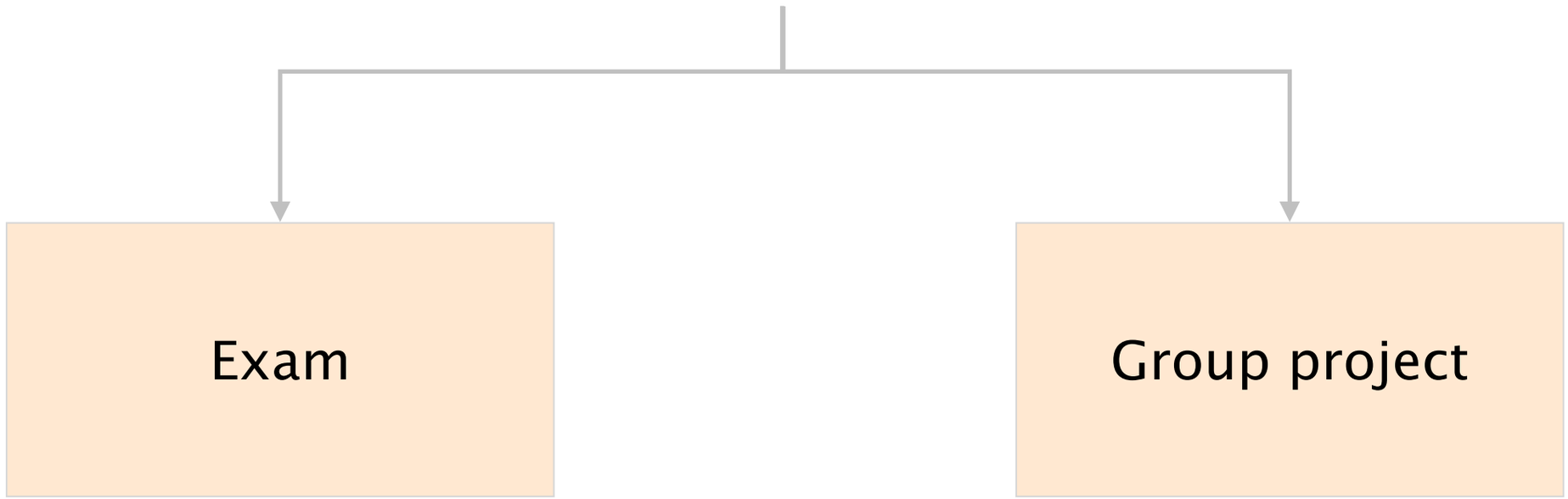
in teams of 3

In the project,
you'll partake in a class-wide challenge

Make ***your*** network the ***fastest*** *one*

We'll provide more information in the coming weeks

and, yes, *there will be prizes* for the best teams!



70%

written

30%

code, report, and presentation

Exam

70%

written

Examples (more in the lecture)

How would you solve problem <X>?

How would you optimize this network design for <X>?

Is the P4 program <X> correct?

… important to do the exercises

Your dream team of teaching assistants

Romain

head TA

Edgar

Roland

Thomas

Maria

Albert

Alexander

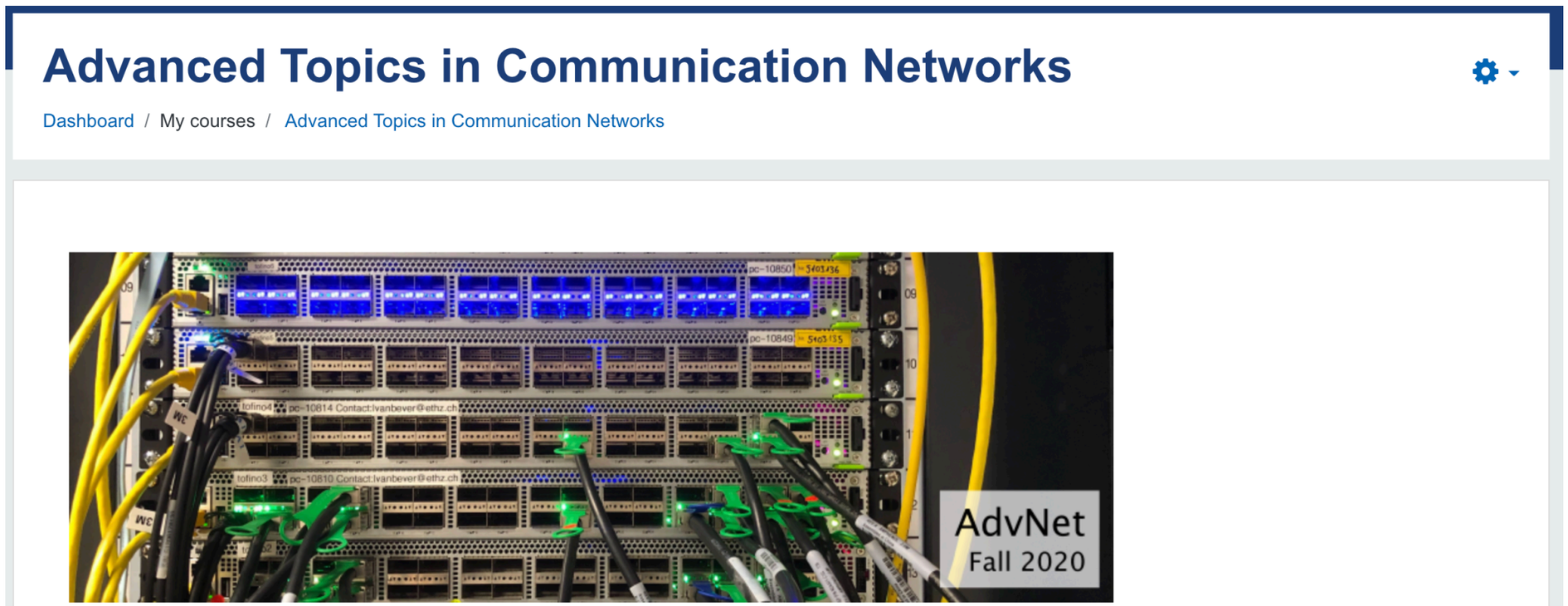
Ege

+ Eric, Yannick

followed lecture last year

We'll use Moodle as course platform

check it out regularly



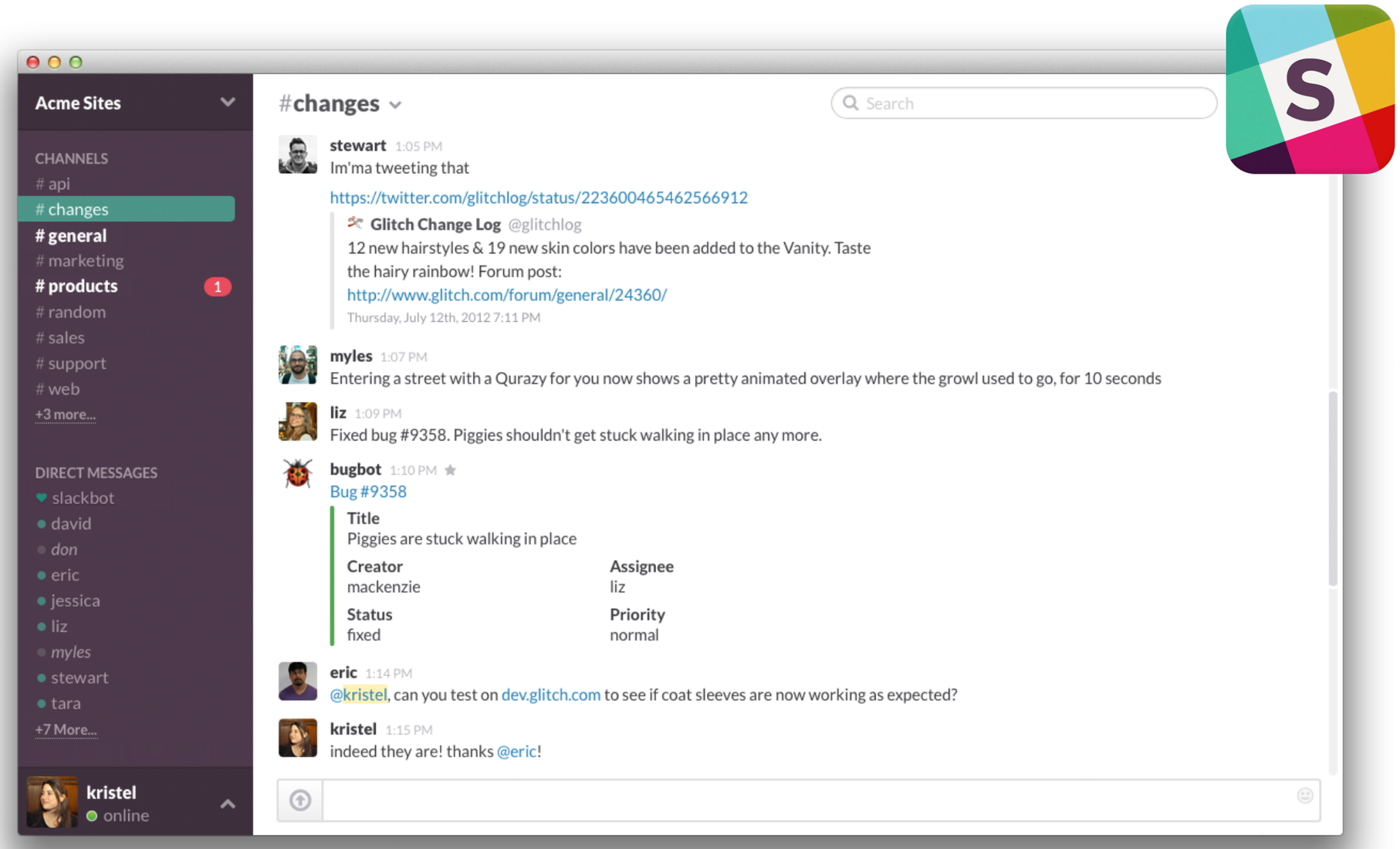
In previous courses, you have learned about the general Internet architecture and the main principles that make it work. But the reality is somewhat more complex. There are many ways to configure, use, and optimize networks; this is what we are going to cover in this course. This year, we will focus on the following questions.

> How do we optimize the
> - performance,
> - flexibility, and
> - reliability
>
> of (large) network infrastructures?

By the end of this course, you will know some of the most common methods used to address these questions and you will be able to apply them to configure an actual network, as if you were an actual network operator.

Class announcements

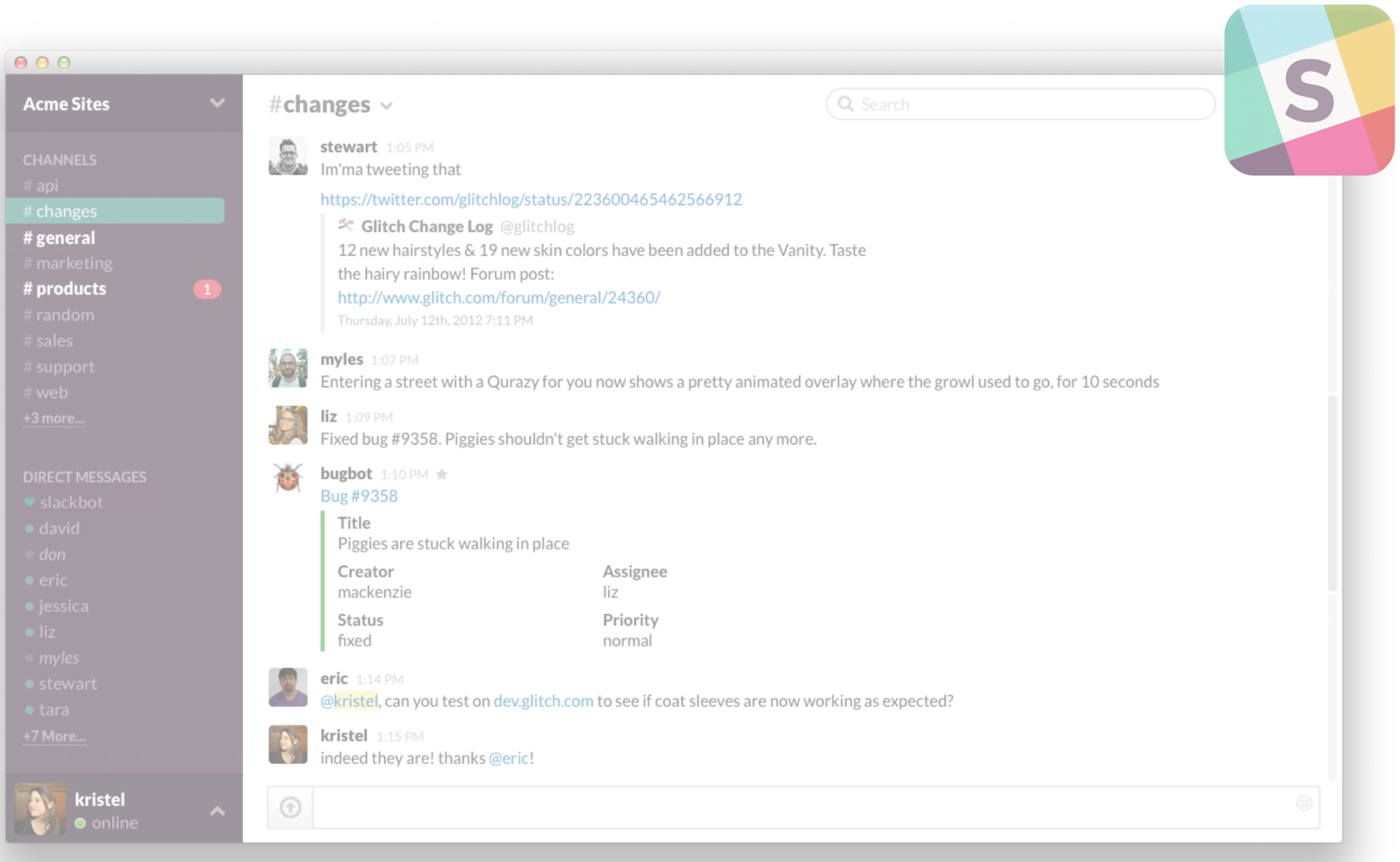
We'll use Slack to discuss about
the course, exercises, and projects



Web, smartphone and desktop clients available

Register today using your real name

https://adv-net20.slack.com/signup



Web, smartphone and desktop clients available

Should ***you*** take this course?

If you like computer networks, ***yes!***

that said…

You shouldn't take the course if

- you *hate* programming
- you can't work during the semester
- you expect 10+ years of exam history

We've almost completely revamped the lecture, some things will def. need adjusting. ***Please say so!***

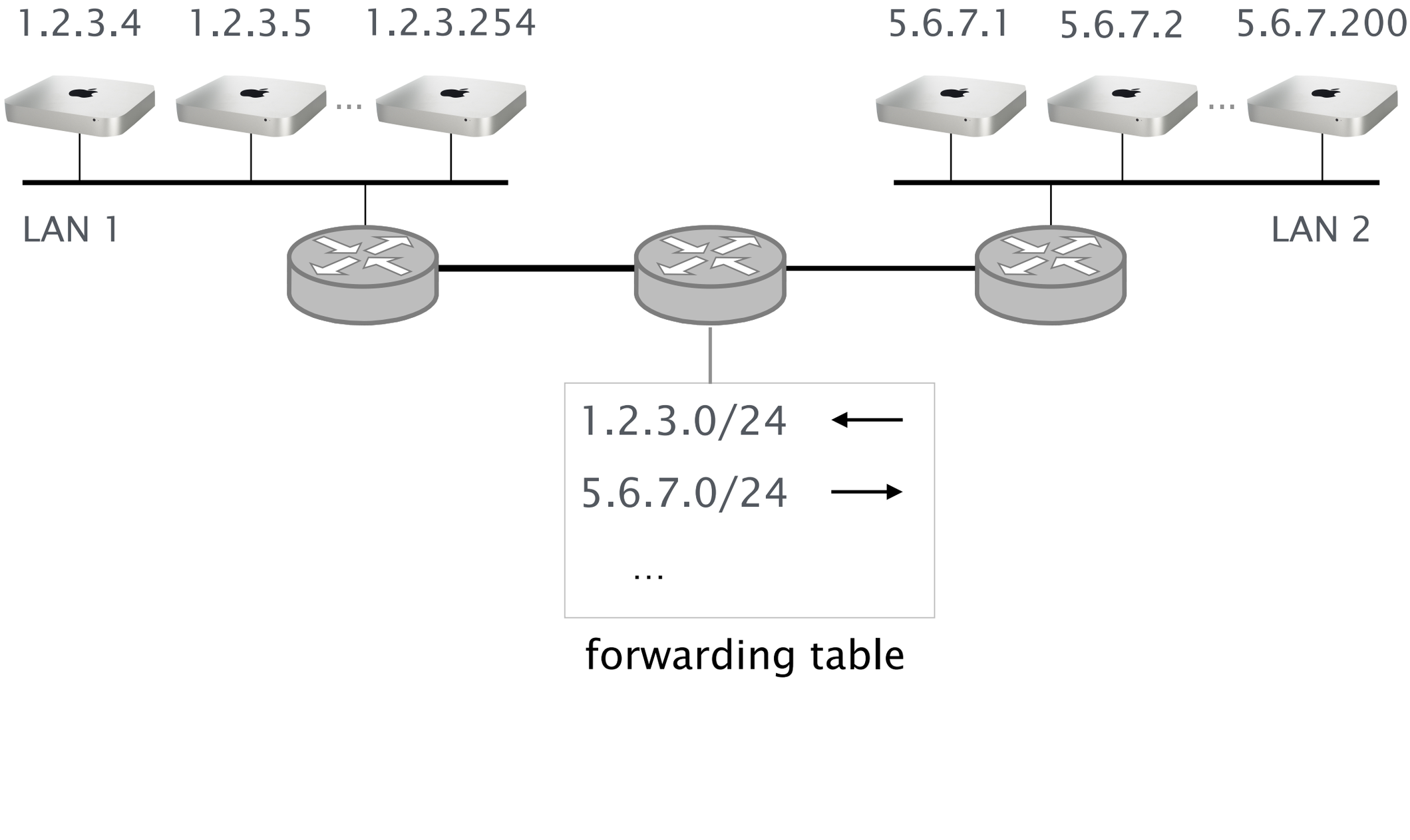
We do our best to take your feedback into account… so shoot!

Your first

program

IP forwarding

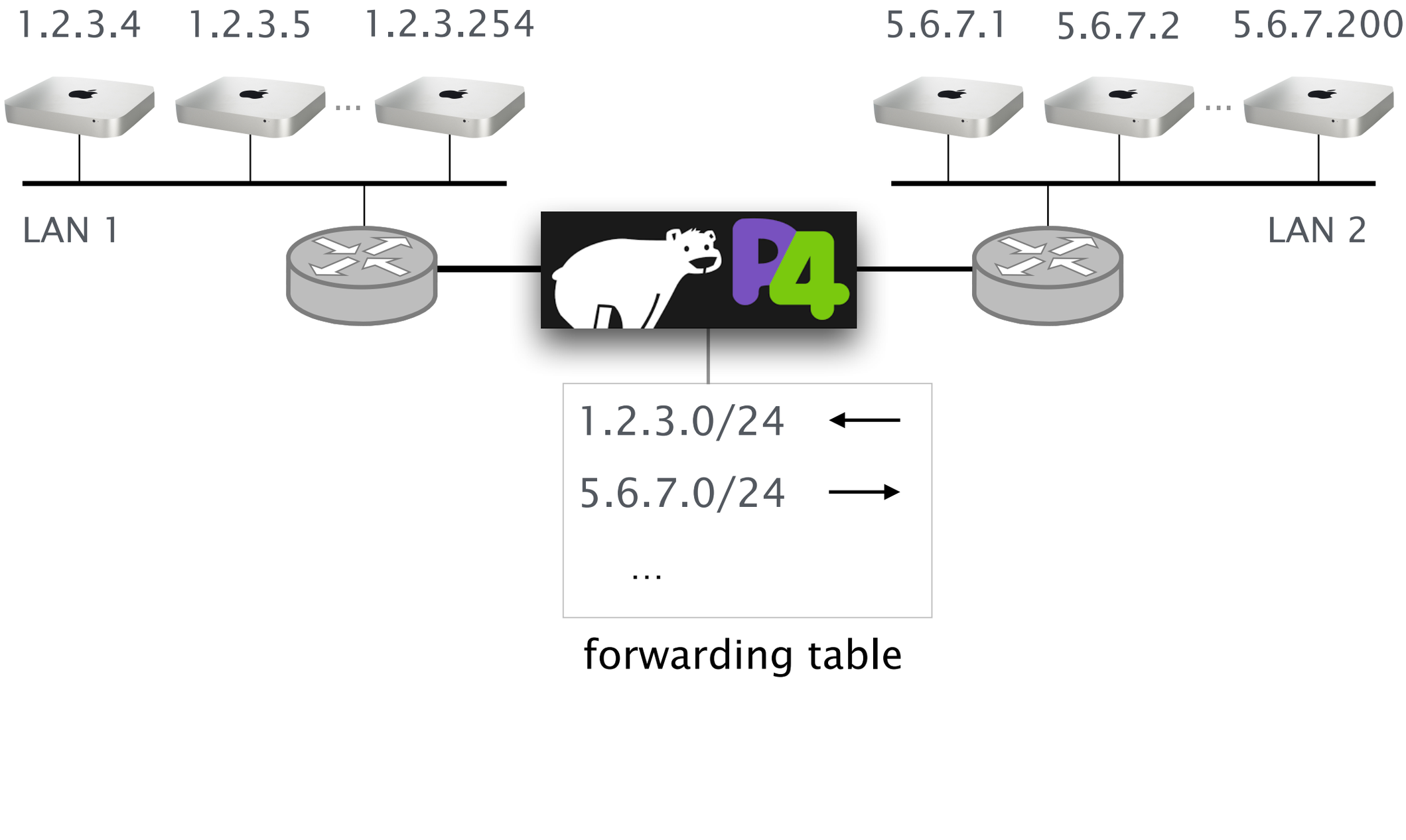
in a traditional router



forwarding table

IP forwarding

in P4?



When forwarding an IP packet,

an IP router performs four actions:

- Lookup the next hop(s) to use
- Update the MAC addresses
- Decrement TTL
- Forward packets to output port(s)

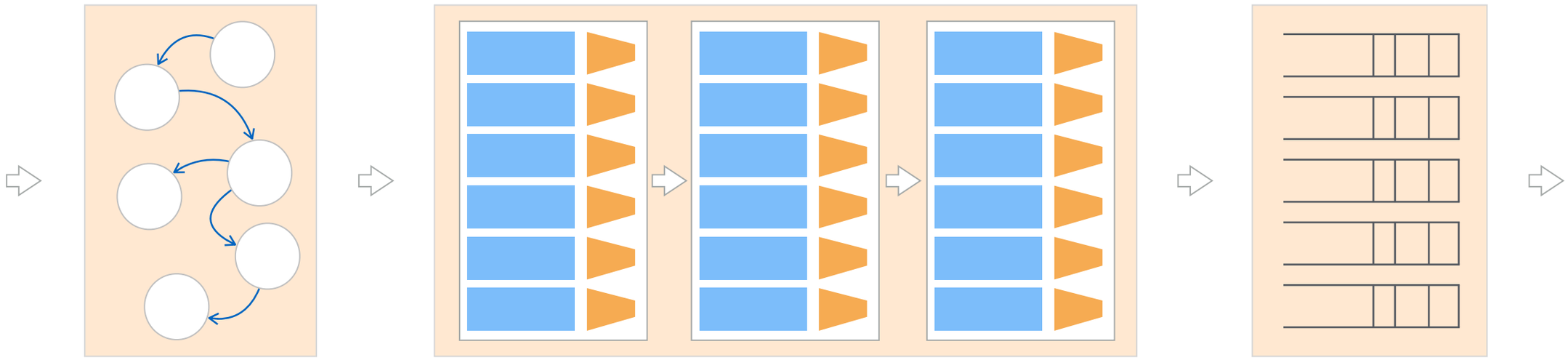
Each of them should be in implemented in P4

A P4 program consists of three basic parts

Parser

Match-Action Pipeline

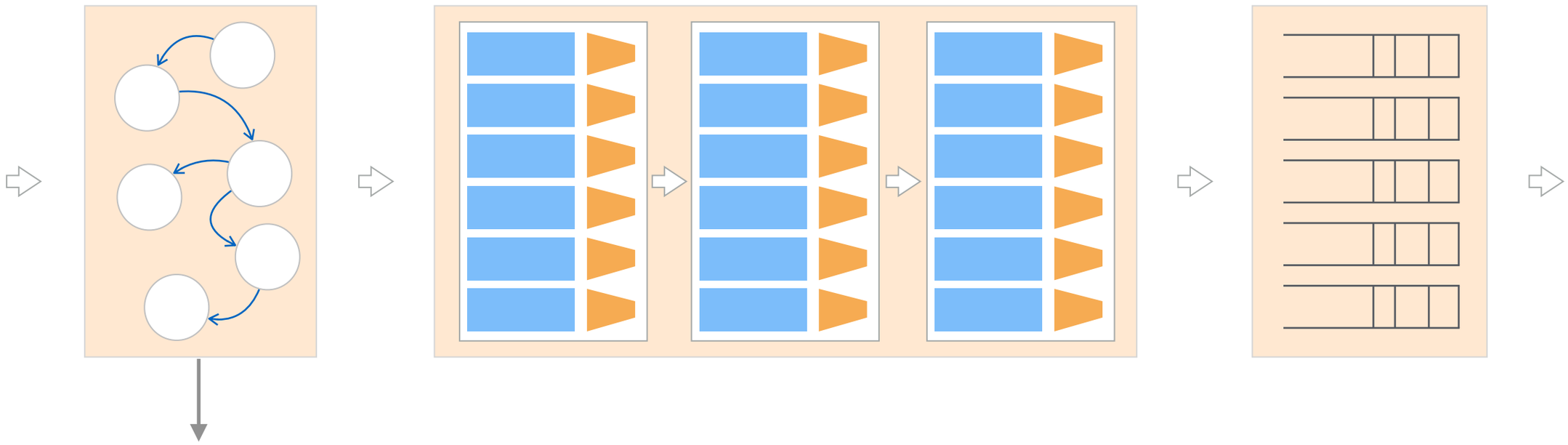
Deparser



Parser

Match-Action Pipeline

Deparser

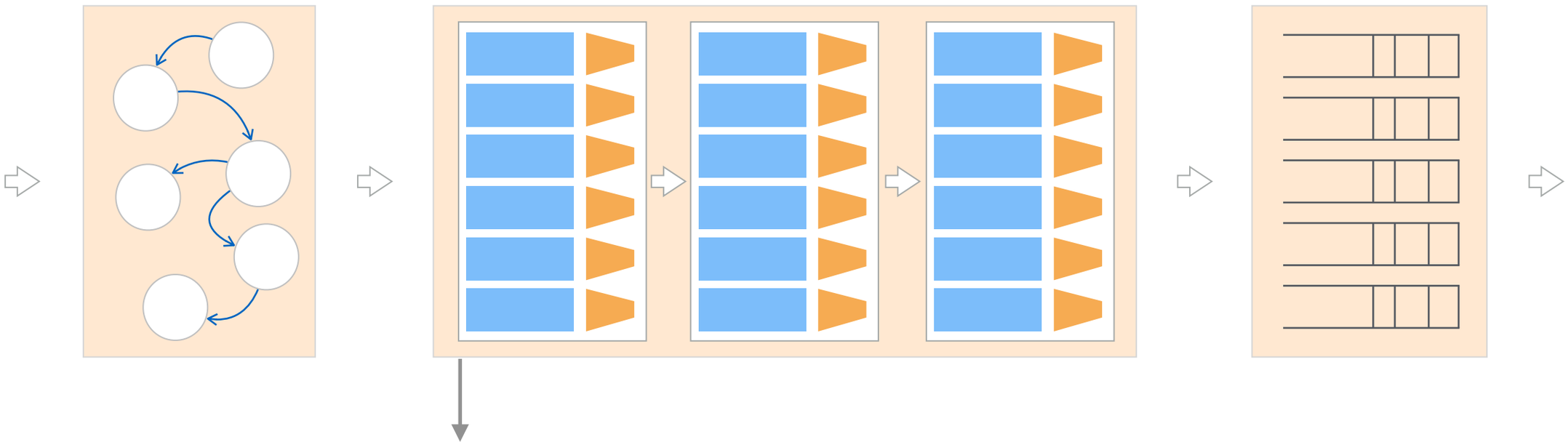


Declares the headers that
should be extracted from the packet

Parser

Match-Action Pipeline

Deparser

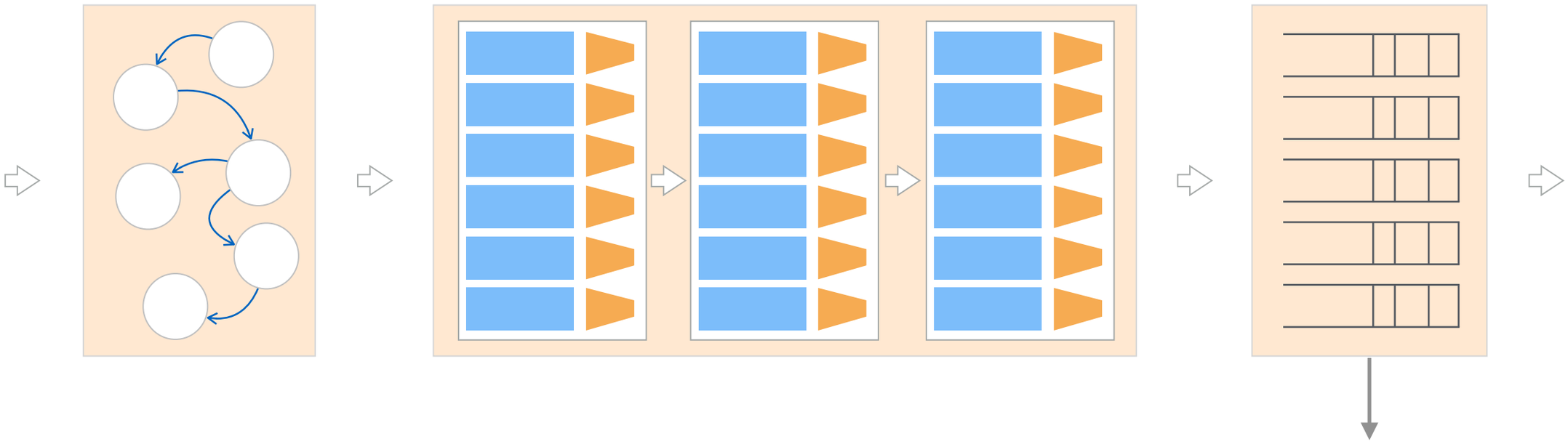


Defines the tables and
the processing logic

Parser

Match-Action Pipeline

Deparser

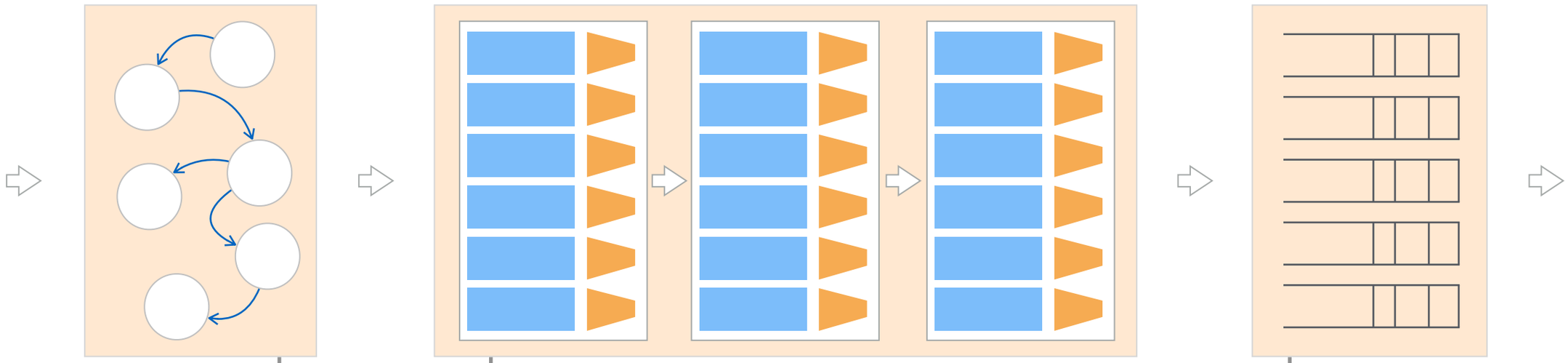


Declares how
the output packet
will look on the wire

Parser

Match-Action Pipeline

Deparser



```
V1Switch(
 MyParser(),
 MyVerifyChecksum(),
 MyIngress(),
 MyEgress(),
 MyComputeChecksum(),
 MyDeparser()
) main;
```

```
#include <core.p4>
#include <v1model.p4>
```

Libraries

```
const bit<16> TYPE_IPV4 = 0x800;
typedef bit<32> ip4Addr_t;
header ipv4_t {…}
struct headers {…}
```

Declarations

```
parser MyParser(…) {
    state start {…}
    state parse_ethernet {…}
    state parse_ipv4 {…}
}
```

Parse packet headers

```
control MyIngress(…) {

    action ipv4_forward(…) {…}

    table ipv4_lpm {…}

    apply {
        if (…) {…}
    }
}
```

Control flow
to modify packet

```
control MyDeparser(…) {…}
```

Assemble
modified packet

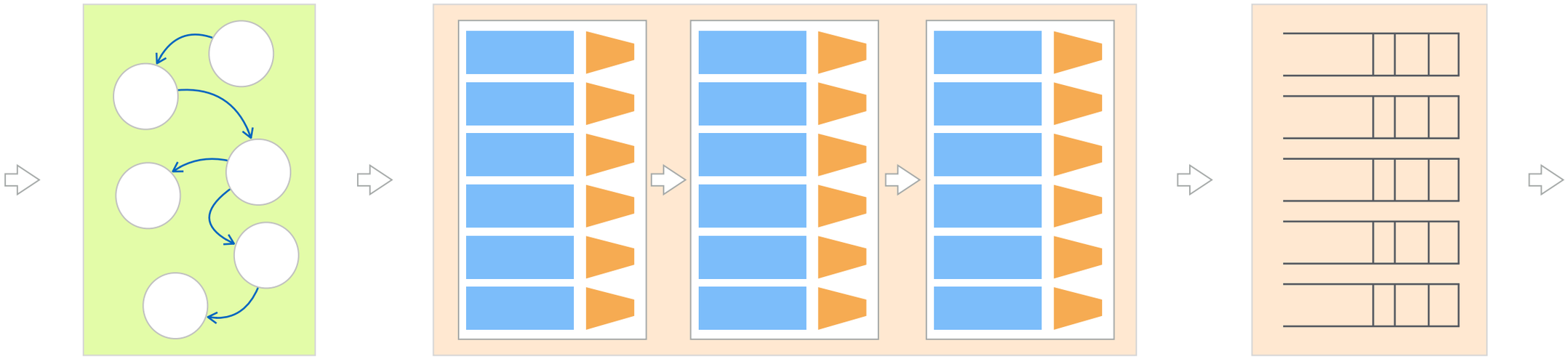
```
V1Switch(
 MyParser(),
 MyVerifyChecksum(),
 MyIngress(),
 MyEgress(),
 MyComputeChecksum(),
 MyDeparser()
) main;
```

“main()”

Parser

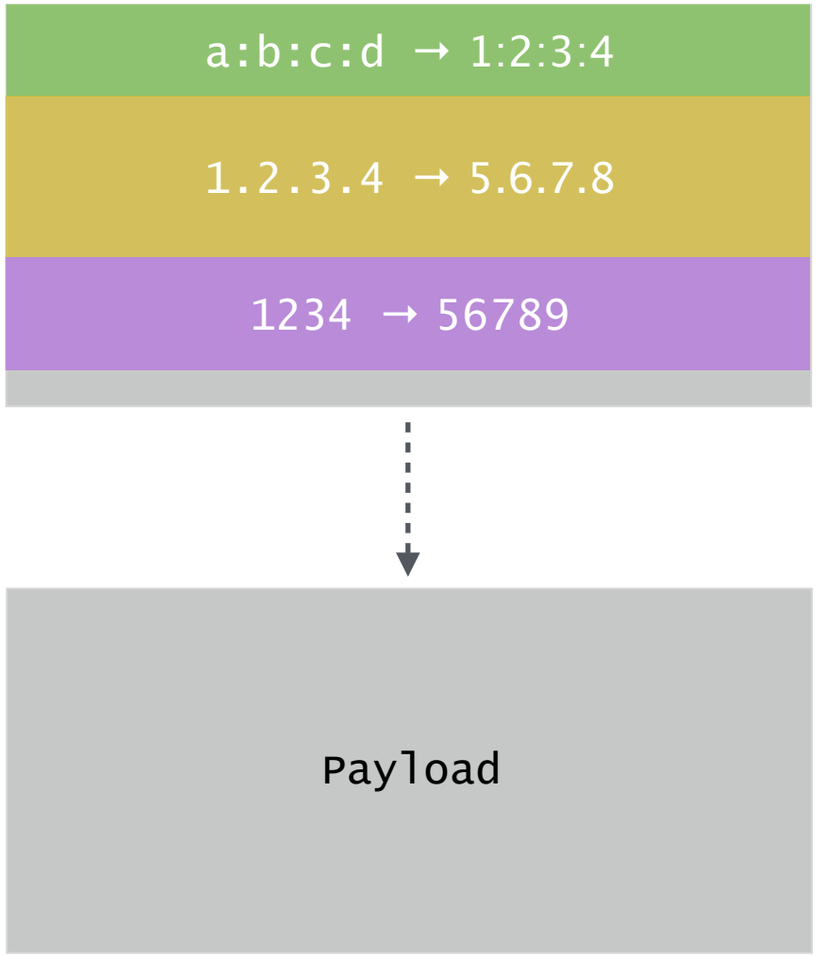
Match-Action Pipeline

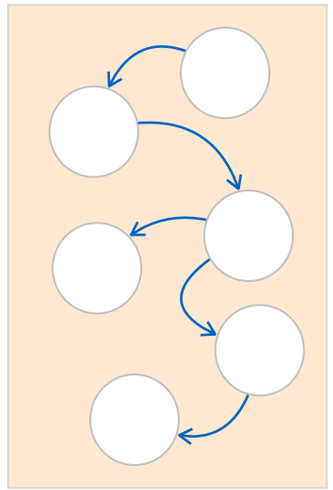
Deparser



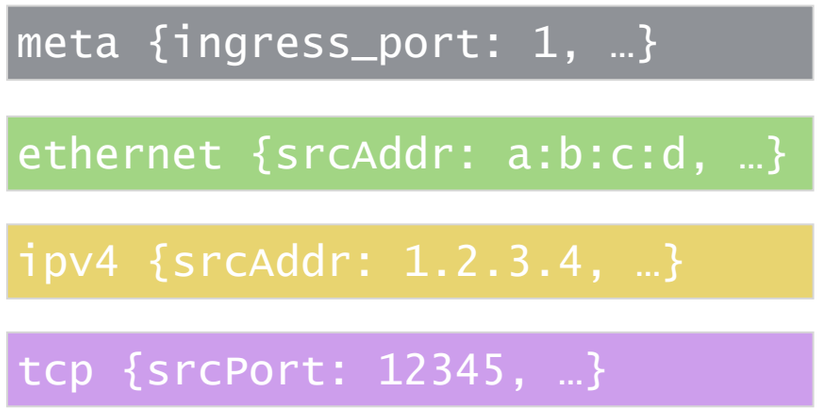
The parser uses a state machine to map packets into headers and metadata

Packet

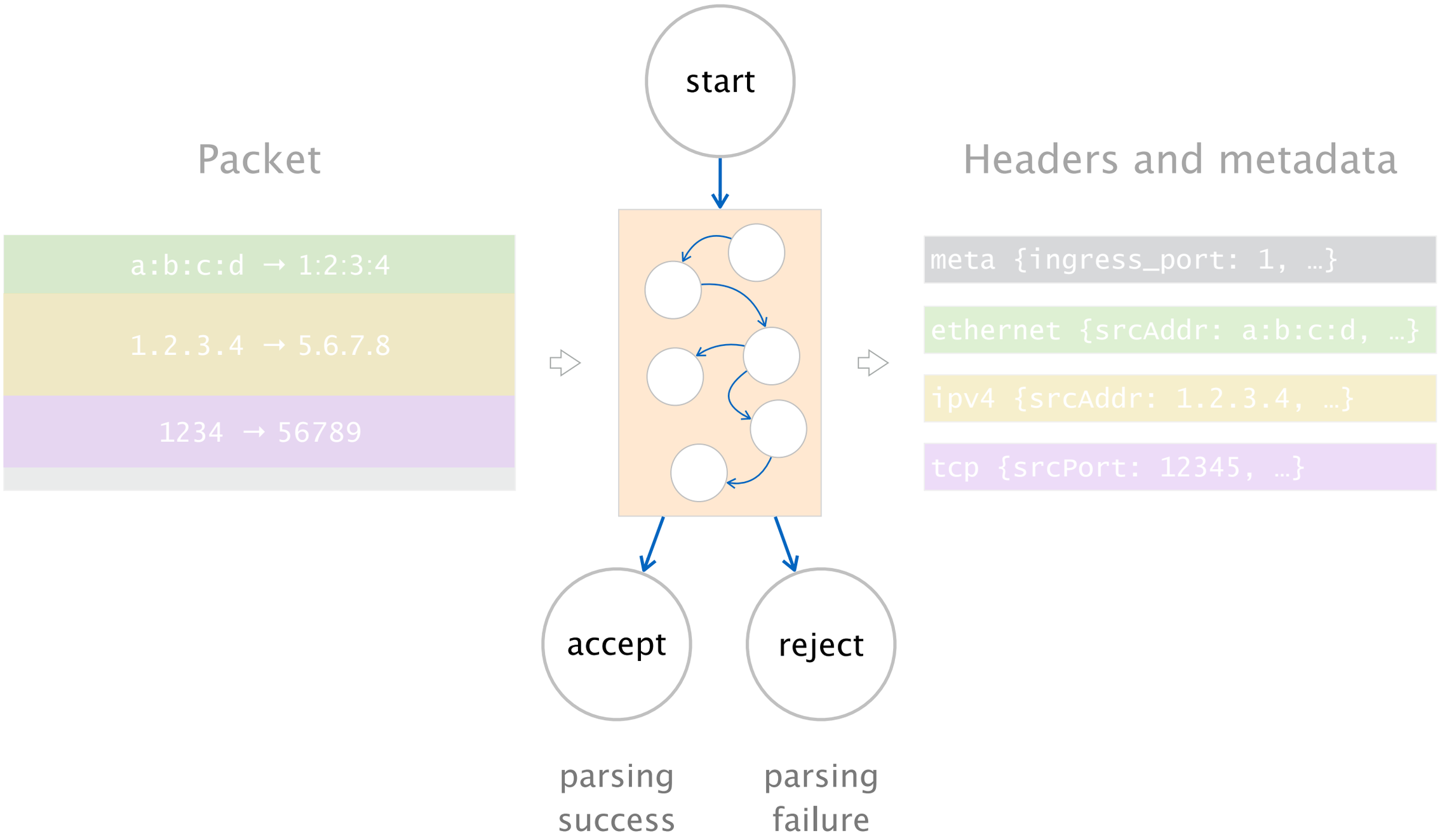


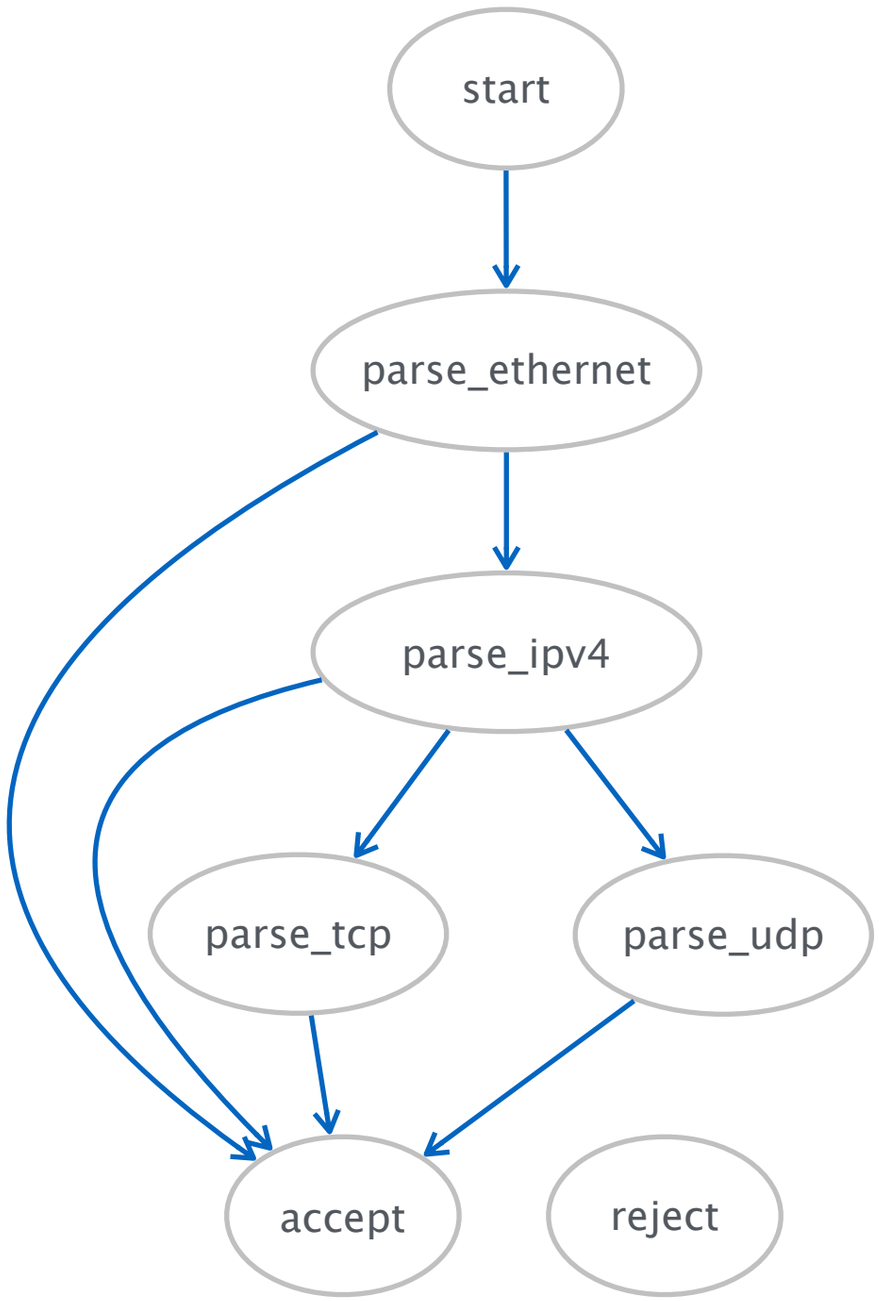


Headers and metadata



The parser has three predefined states: start, accept and reject



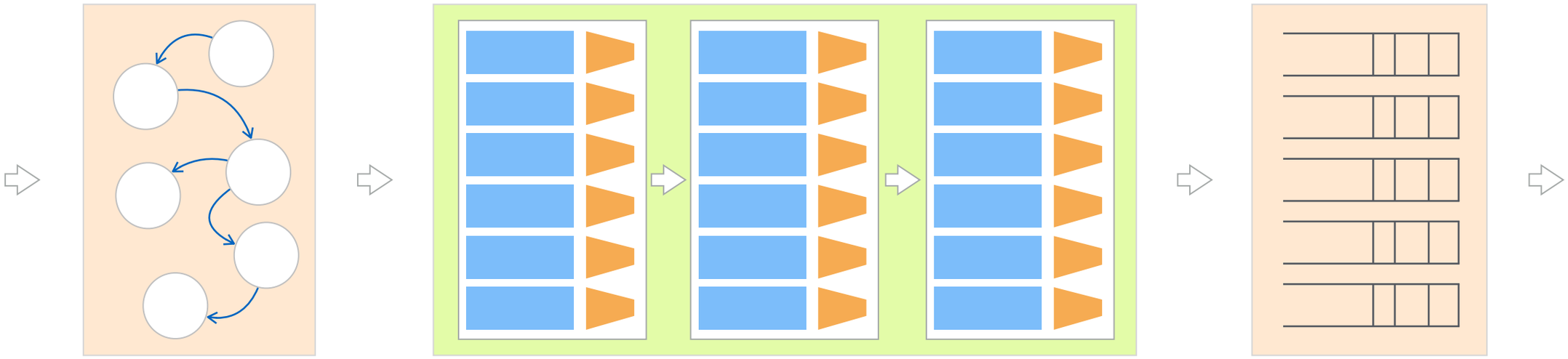



```
parser MyParser(…) {
 state start {
  transition parse_ethernet;
 }
 state parse_ethernet {
  packet.extract(hdr.ethernet);
  transition select(hdr.ethernet.etherType) {
    0x800: parse_ipv4;
    default: accept;
  }
 }
 state parse_ipv4 {
  packet.extract(hdr.ipv4);
  transition select(hdr.ipv4.protocol) {
   6: parse_tcp;
   17: parse_udp;
   default: accept;
  }
 }
 state parse_tcp {
  packet.extract(hdr.tcp);
   transition accept;
 }
 state parse_udp {
  packet.extract(hdr.udp);
   transition accept;
 }
}
```

Parser

Match-Action Pipeline

Deparser



Basic building blocks of P4 programs

Control

Control flow

describes how headers should be processed

Actions

fragments manipulating headers/metadata

Tables

map user-defined keys with actions

Control

Control flow

describes how headers should be processed

Actions

fragments manipulating headers/metadata

Tables

map user-defined keys with actions

Control flow expresses an imperative program which describes how packets are processed

Headers and metadata from parser

```
control MyIngress(inout headers hdr,
                  inout metadata meta,
                  inout standard_metadata_t std_meta) {

 bit<9> port;

 apply {
  port = 1
  std_meta.egress_spec = port;
  hdr.ethernet.srcAddr = hdr.ethernet.dstAddr;
  hdr.ethernet.dstAddr = 0x2;
  hdr.ipv4.ttl = hdr.ipv4.ttl - 1;
 }
}
```

Variable declaration

Control flow

Control

Control flow

describes how headers should be processed

Actions

fragments manipulating headers/metadata

Tables

map user-defined keys with actions

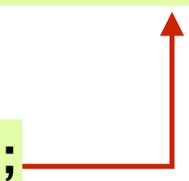
Actions allow to re-use code

similar to functions in C

```
control MyIngress(inout headers hdr,
                  inout metadata meta,
                  inout standard_metadata_t std_meta) {

 action ipv4_forward(macAddr_t dstAddr,
                     egressSpec_t port) {
  std_meta.egress_spec = port;
  hdr.ethernet.srcAddr = hdr.ethernet.dstAddr;
  hdr.ethernet.dstAddr = dstAddr;
  hdr.ipv4.ttl = hdr.ipv4.ttl - 1;
 }

 apply {
  ipv4_forward(0x123, 1);
 }
}
```



Control

Control flow

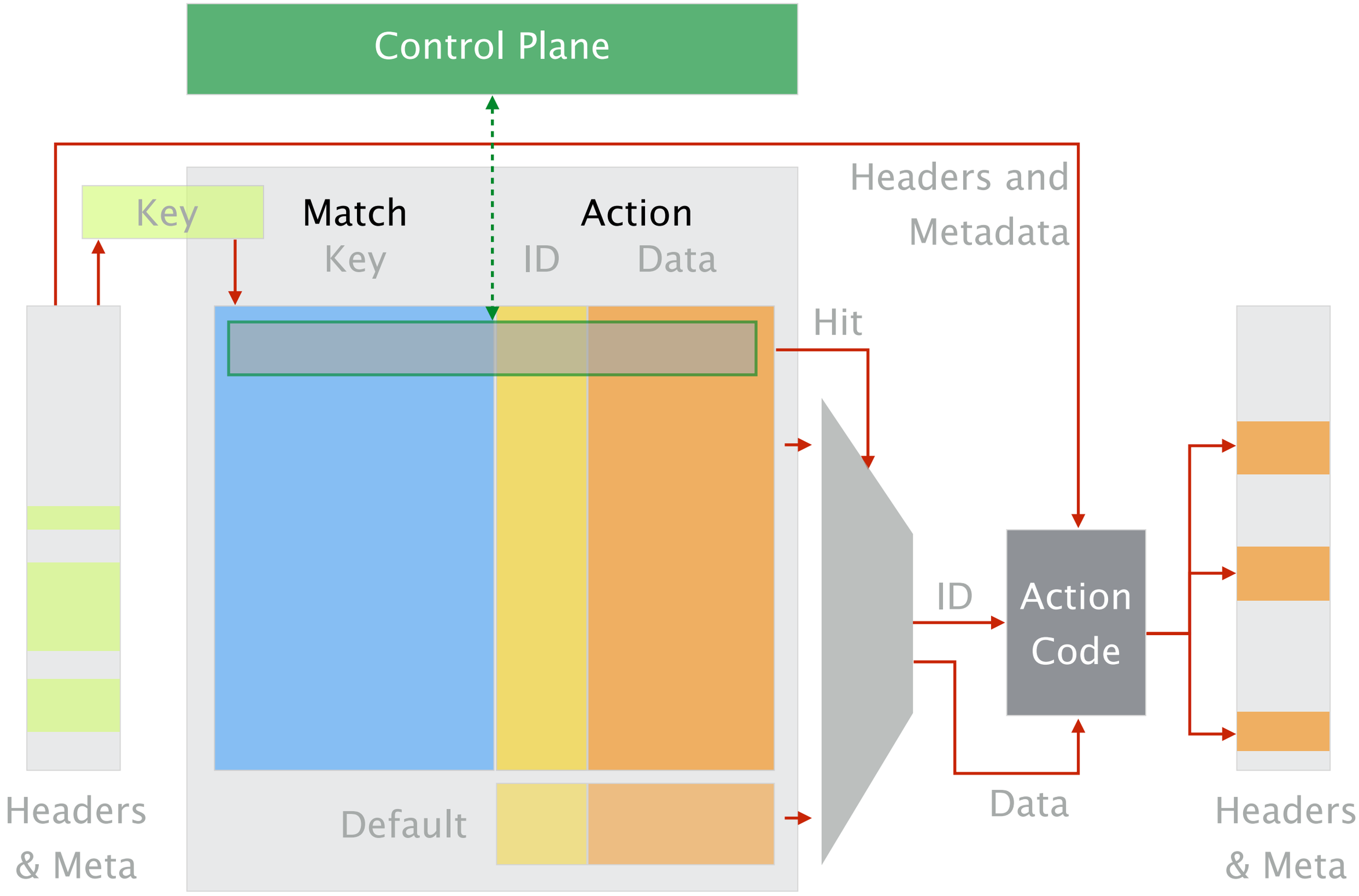
describes how headers should be processed

Actions

fragments manipulating headers/metadata

Tables

map user-defined keys with actions


Control Plane
Headers and
Metadata
Key
Match
Key
Action
ID
Data
Hit
Default
ID
Action
Code
Data
Headers
& Meta
Headers
& Meta

```
table                    {
 key = {
                         :    ;
 }
 actions = {
 }

 size =      ;

 default_action =      ;
}
```

Table name

Field(s) to match

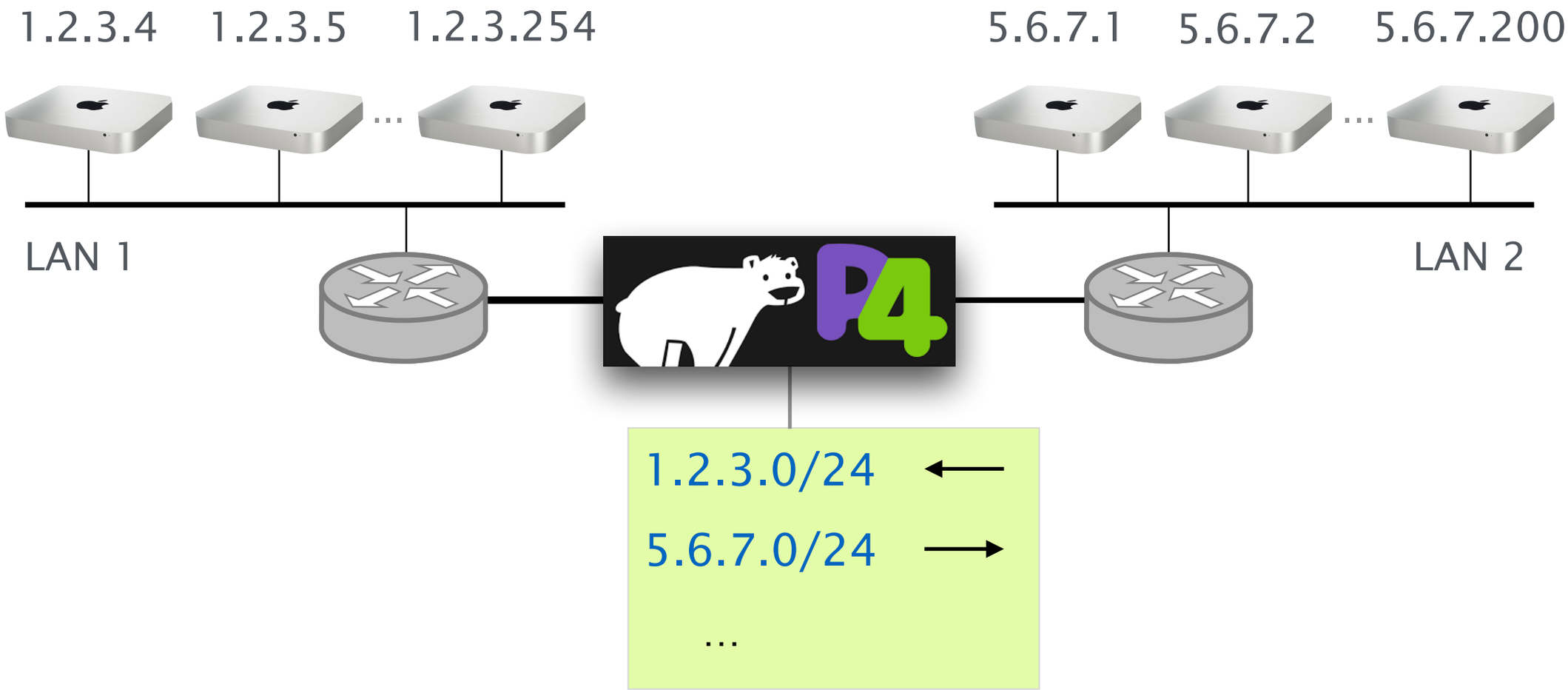
Match type

Possible actions

Max. # entries in table

Default action

Example: IP forwarding table



forwarding table

Control Plane

Destination IP address

Key

Match
Key

Action
ID Data

Headers and
Metadata

Hit

Default

ID

Action
Code

Data

Headers
& Meta

Headers
& Meta

Control Plane

Longest prefix match

Match
Key

Key

Action
ID Data

Headers and
Metadata

Hit

ID

Action
Code

Data

Headers
& Meta

Default

Headers
& Meta

1: ipv4_forward(mac,port)
2: drop()

Action

ID Data

Key

Match Key

Headers and Metadata

Hit

ID

Action Code

Data

Headers & Meta

Default

Headers & Meta

```
table ipv4_lpm {
 key = {
  hdr.ipv4.dstAddr: lpm;
 }
 actions = {
  ipv4_forward;
  drop;
 }

 size = 1024;

 default_action = drop();
}
```

Table name

Destination IP address

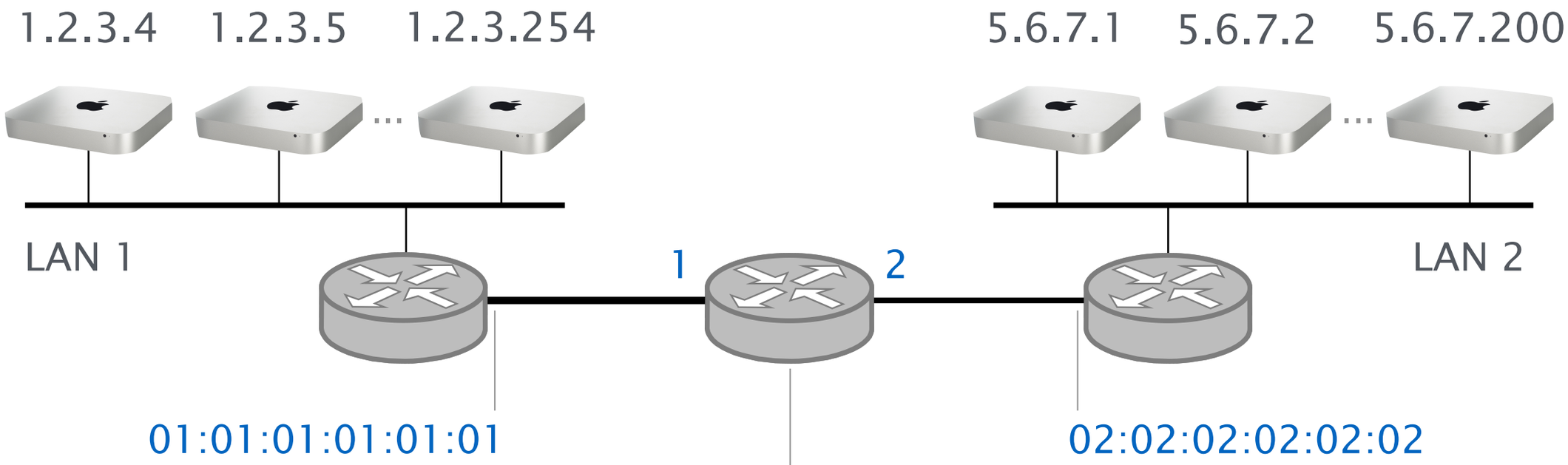
Longest prefix match

Possible actions

Max. # entries in table

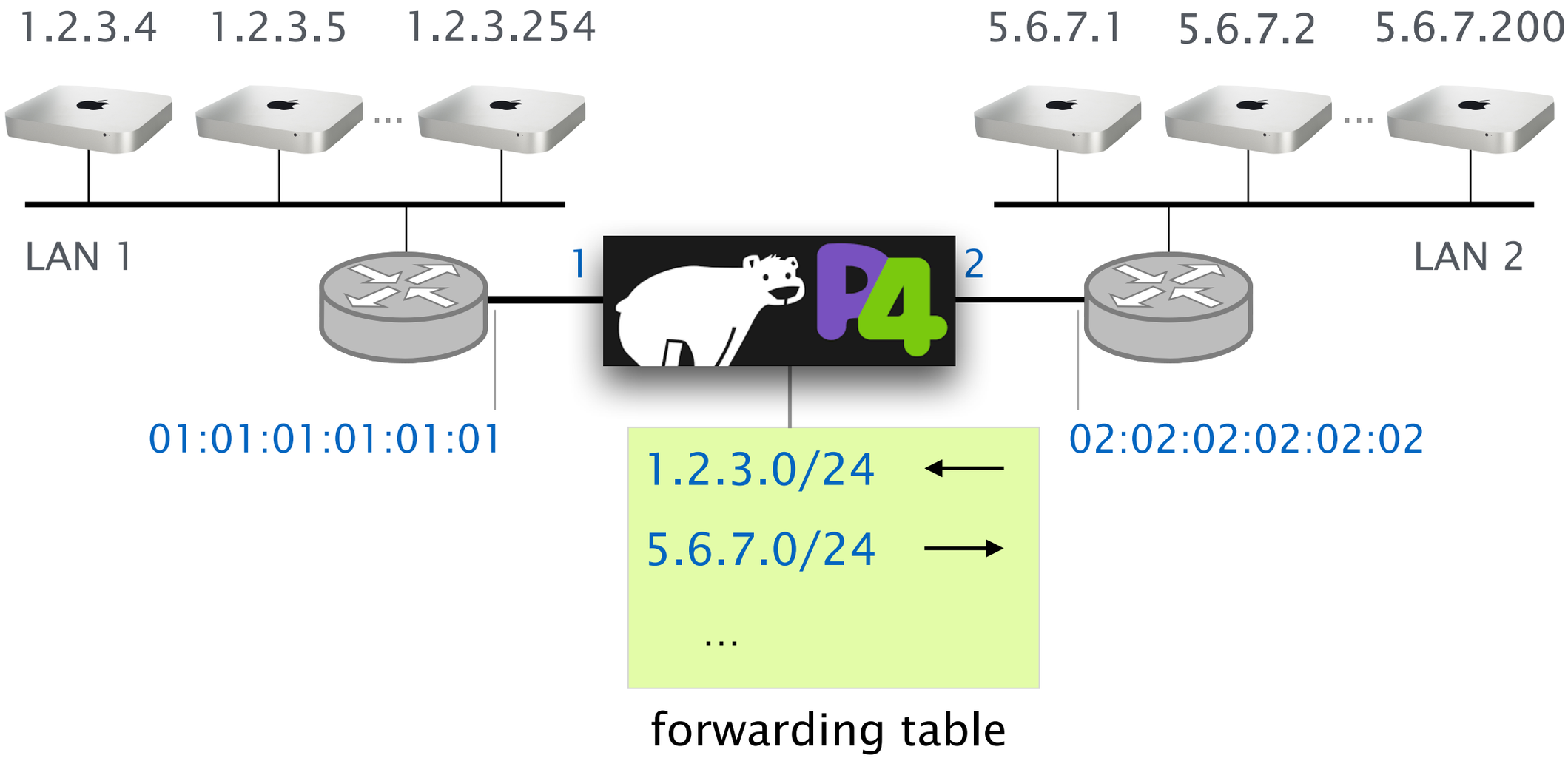
Default action

Example: IP forwarding table




```
action ipv4_forward(macAddr_t dstAddr, egressSpec_t port) {
    standard_metadata.egress_spec = port;
    hdr.ethernet.srcAddr = hdr.ethernet.dstAddr;
    hdr.ethernet.dstAddr = dstAddr;
    hdr.ipv4.ttl = hdr.ipv4.ttl - 1;
}
```

Example: IP forwarding table



Control Plane

```
table_add ipv4_lpm ipv4_forward 1.2.3.0/24 => 01:01:01:01:01:01 1
table_add ipv4_lpm ipv4_forward 5.6.7.0/24 => 02:02:02:02:02:02 2
```

Key	ID	Data
1.2.3.0/24	1	01:..., 1
5.6.7.0/24	1	02:..., 2
Default	2	

Hit

ID

Action Code

Data

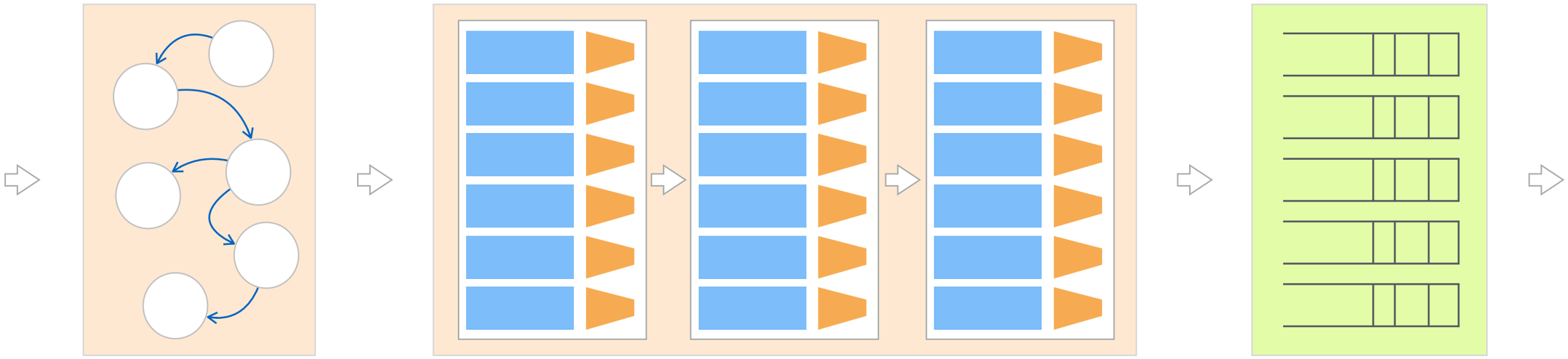
Headers & Meta

Headers & Meta

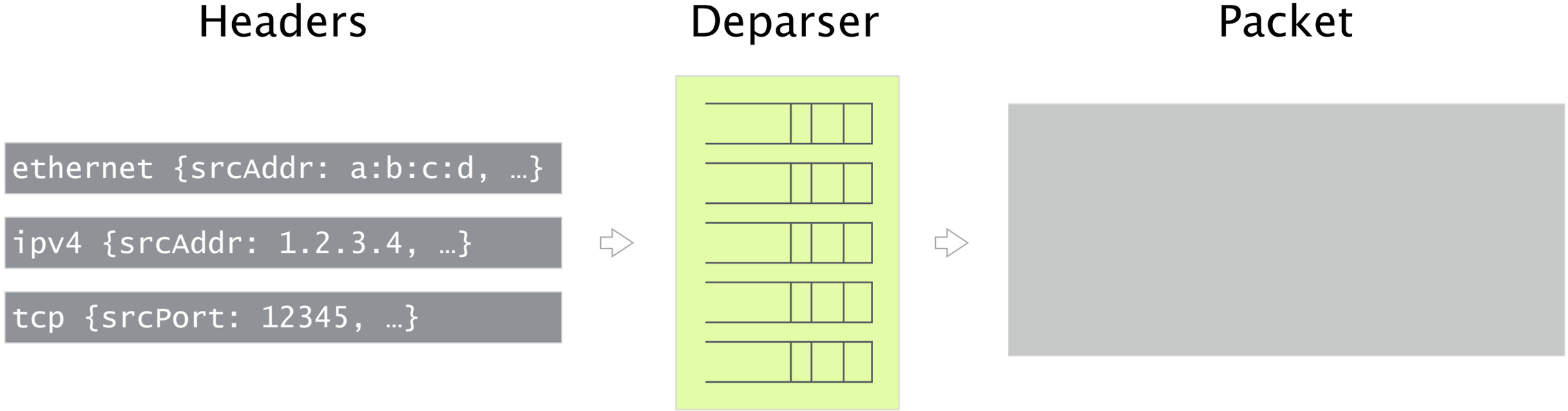
Parser

Match-Action Pipeline

Deparser



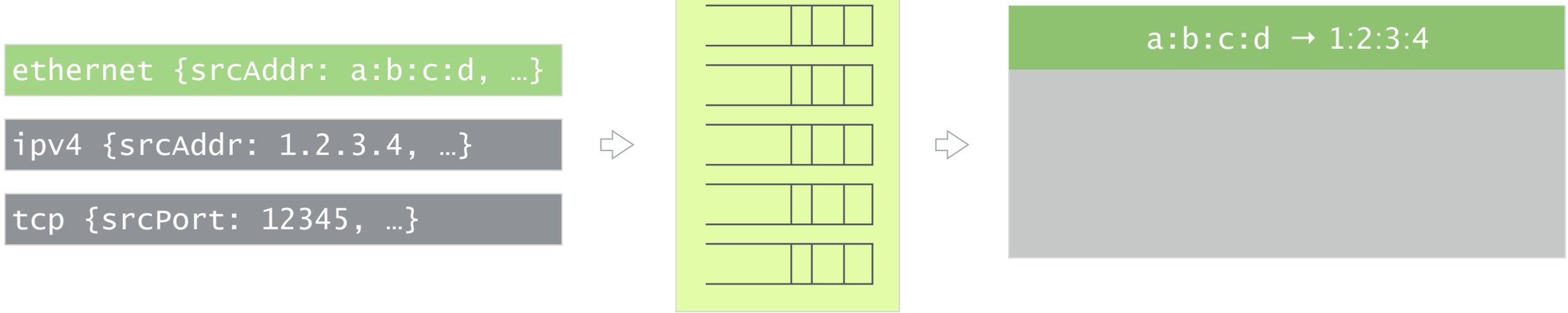
The Deparser assembles the headers back into a well-formed packet



Headers

Deparser

Packet



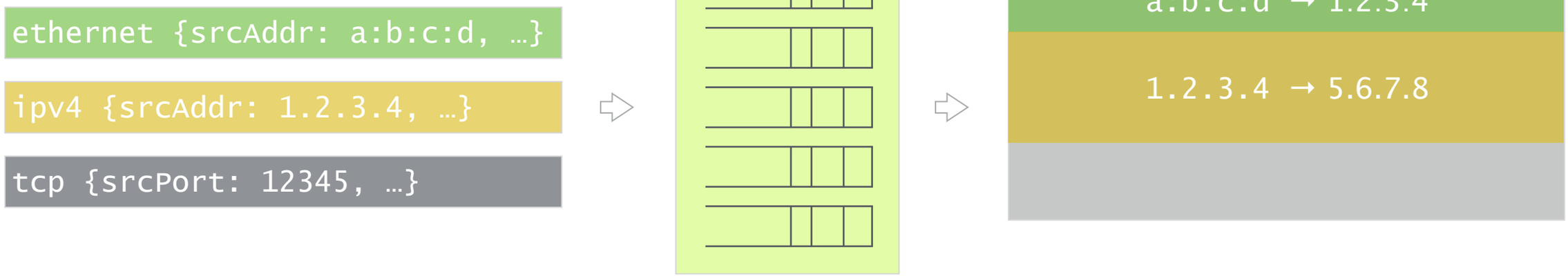

```
control MyDeparser(packet_out packet, in headers hdr) {
    apply {
        packet.emit(hdr.ethernet);


    }
}
```

Headers

Deparser

Packet



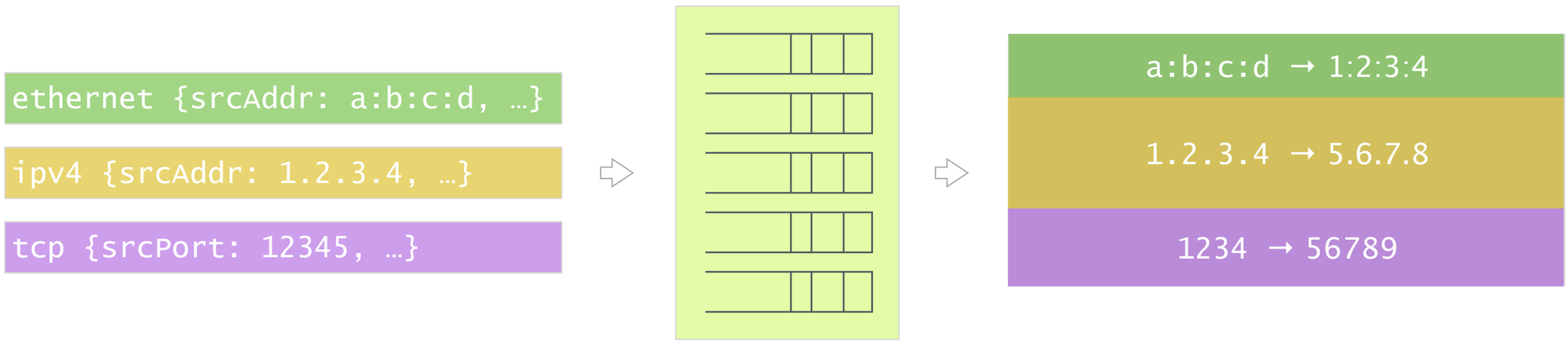

```
control MyDeparser(packet_out packet, in headers hdr) {
    apply {
        packet.emit(hdr.ethernet);
        packet.emit(hdr.ipv4);

    }
}
```

Headers

Deparser

Packet

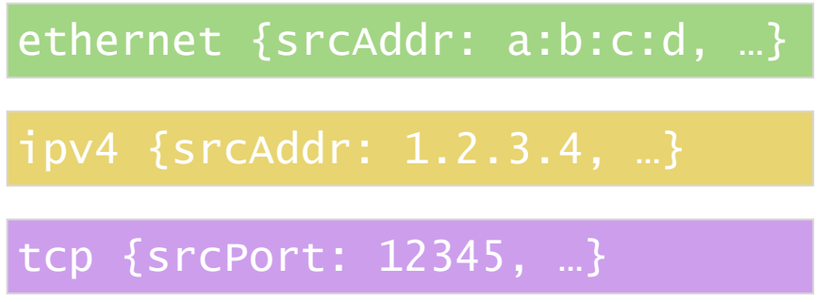


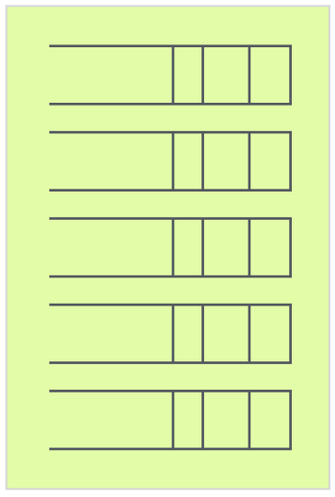

```
control MyDeparser(packet_out packet, in headers hdr) {
    apply {
        packet.emit(hdr.ethernet);
        packet.emit(hdr.ipv4);
        packet.emit(hdr.tcp);
    }
}
```

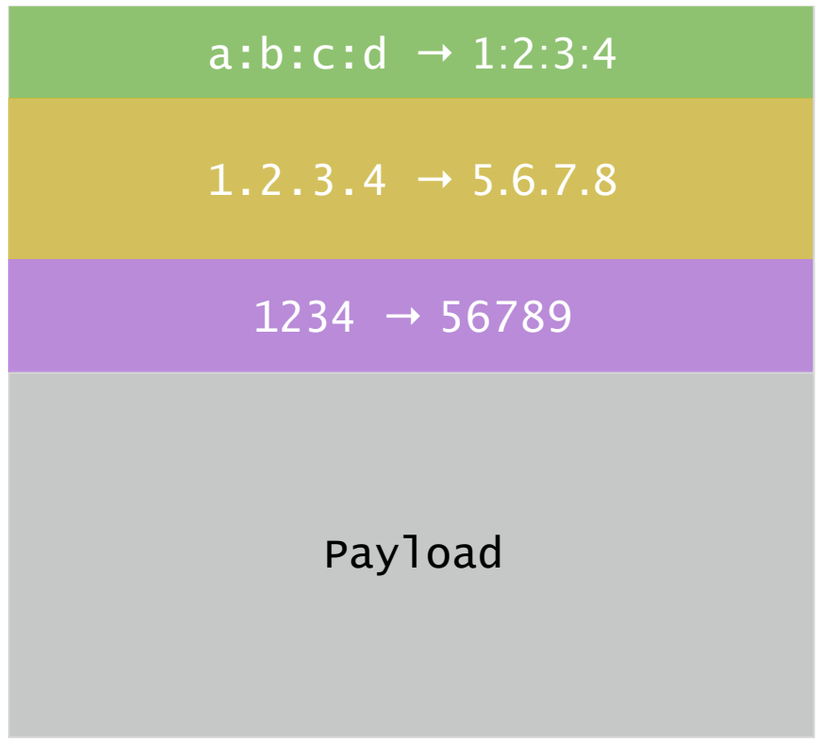
Headers

Deparser

Packet







Advanced Topics in Communication Networks

Internet Routing and Forwarding

Laurent Vanbever

nsg.ee.ethz.ch

ETH Zürich (D-ITET)

15 Sep 2020