Advanced Topics in Communication Networks

Internet Routing and Forwarding

Laurent Vanbever

nsg.ee.ethz.ch

6 October 2020

Lecture starts at **14:15**

Materials inspired and/or coming from Olivier Bonaventure, Mike Freedman, Nick Feamster, Alex Snoeren, Jennifer Rexford, and p4.org

Thanks for your 3/2/1 input on padlet!

Let me answer your lecture-related questions https://padlet.com/romainjacob42/qxhdeqy7cd6nm05t

1 How should we compare a software-programmable (P4) switch with more traditional L2/L3 switches or L3 routers?

2 Can I also do NAT using P4?

3 How can I make a P4 program modular?

4 Are there libraries of classical P4 programs?

5 How does the ternary match work?

6 Which operations may slow down the program and hence slow down switching?

7 What is the point of penultimate popping?

8 How do MPLS routers deal with MTU to fit extra MPLS headers?

Ternary match in P4

search_word	action	priority
0*	a1	1
1*	a2	2
10*	a3	3
111*	a4	4
101*	a5	5

```
table ternary_table {
    key = {
        hdr.ipv4.dstAddr: ternary;
    }
    actions = {
        ipv4_forward;
        drop;
        NoAction;
    }
    size = 1024;
    default_action = NoAction();
}
```

```
table_set_default ternary_table drop
table_add ternary_table ipv4_forward 0x00000000&&&0x80000000 => 00:00:00:00:00:01 2 5
table_add ternary_table ipv4_forward 0x80000000&&&0x80000000 => 00:00:00:00:00:02 2 4
table_add ternary_table ipv4_forward 0x80000000&&&0xc0000000 => 00:00:00:00:00:03 2 3
table_add ternary_table ipv4_forward 0xe0000000&&&0xe0000000 => 00:00:00:00:00:04 2 2
table_add ternary_table ipv4_forward 0xa0000000&&&0xe0000000 => 00:00:00:00:00:05 2 1
```

Last week on

Advanced Topics in Communication Networks

We *finished* to dive in the P4 ecosystem and *continued* to look at Multiprotocol Label Switching

P4 environment	P4 language	label switching
What is needed to program in P4?	Deeper-dive into the language constructs	the basics

P4
environment

What is needed to
program in P4?

P4
language

Deeper-dive into
the language constructs

label
switching

Stateful objects in P4

Table	managed by the control plane
Register	store arbitrary data
Counter	count events
Meter	rate-limiting
…	…

externs in v1model

Summary

	Data plane interface		Control plane interface	
Object	read	modify/write	read	modify/write
Table	`apply()`	—	yes	yes
Register	`read()`	`write()`	yes	yes
Counter	—	`count()`	yes	reset
Meter	`execute()`		configuration only	

P4
environment

P4
language

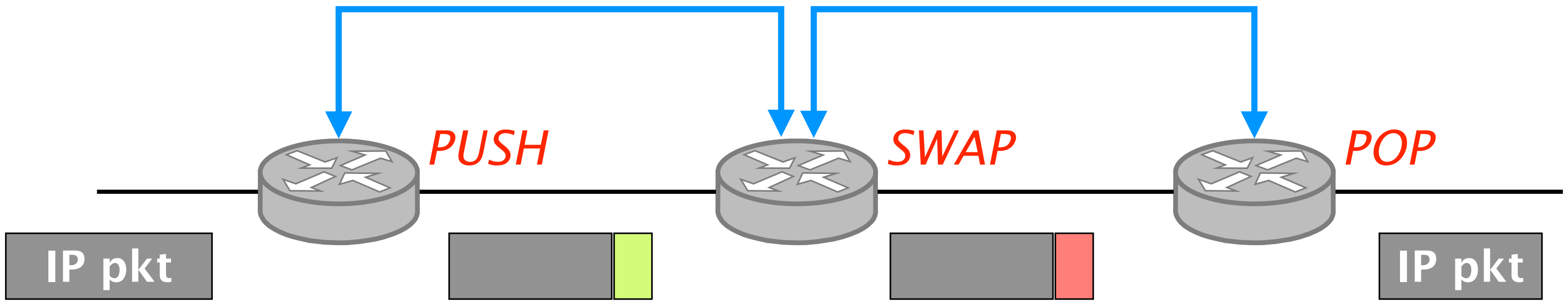
label
switching

the basics

Multiprotocol Label Switching

"IP meets virtual circuits"

Label Distribution Protocol



PUSH

SWAP

POP

IP pkt

IP pkt

Ingress LSR

map FEC to 🟩

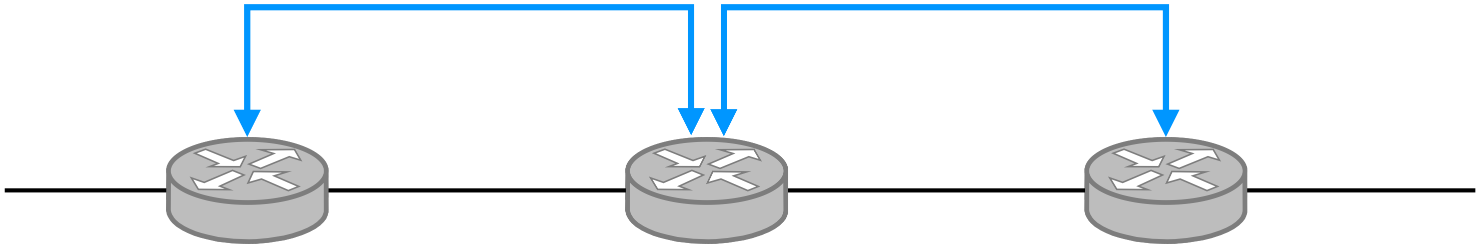
Core LSR

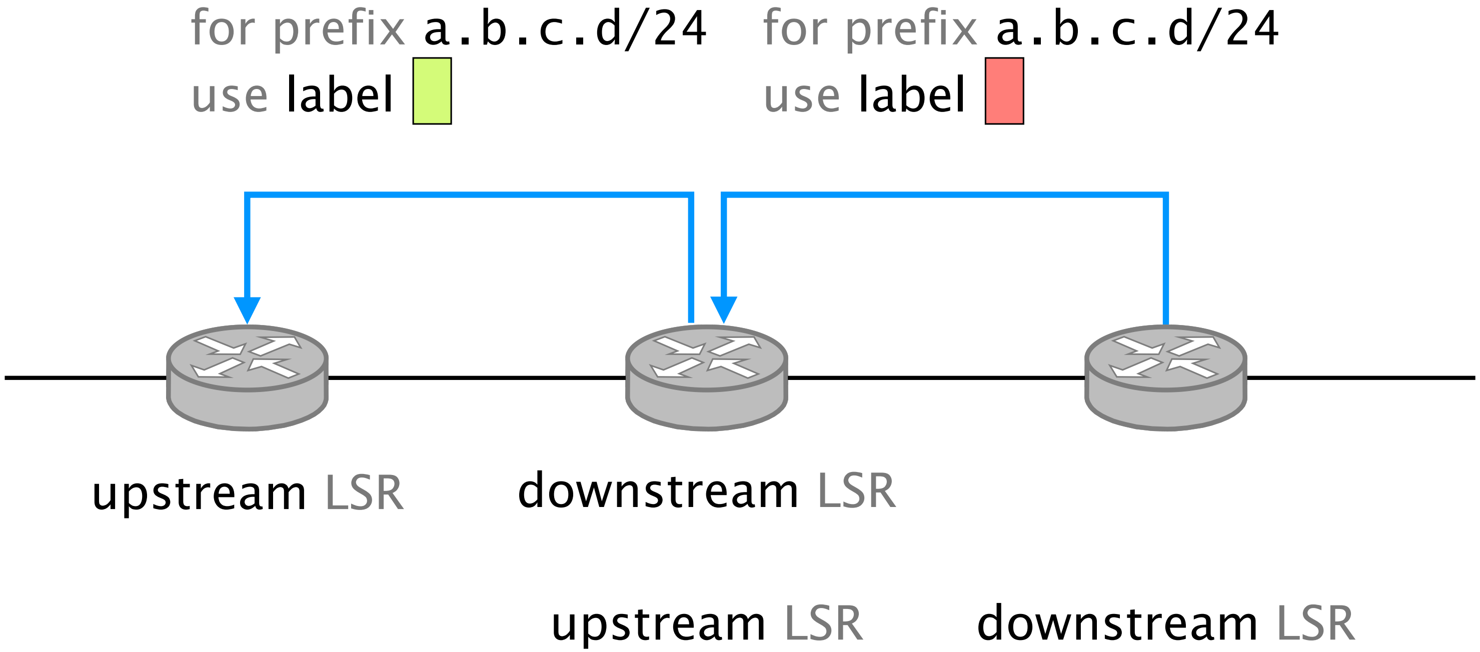
lookup 🟩

Egress LSR

lookup 🟥

Label Distribution Protocol





for prefix a.b.c.d/24
use label 🟩

for prefix a.b.c.d/24
use label 🟥

upstream LSR

downstream LSR

upstream LSR

downstream LSR

We'll see two label distribution protocols:
LDP and RSVP-TE

	LDP	RSVP-TE
Who initiates LSP creation?		
What types of LSP are signaled?		
Can LSPs follow arbitrary paths?		
How easy is it to manage?		
Does it scale?		

	LDP	RSVP-TE
Who initiates LSP creation?	egress	ingress
What types of LSP are signaled?	unidirectional & multi-point-to-point "many heads, one tail"	unidirectional & point-to-point "one head, one tail"
Can LSPs follow arbitrary paths?	nope only shortest-paths	yes
How easy is it to manage?	simple, "automatic"	hard, manual
Does it scale?	yep	not-so-much

	LDP	RSVP-TE
Can LSPs follow arbitrary paths?	nope only shortest-paths	yes

	LDP	RSVP-TE
Can LSPs follow arbitrary paths?	nope only shortest-paths	yes
What's the **main usage**?	virtual private network	traffic engineering fast convergence

This week on

Advanced Topics in Communication Networks

label switching

traffic engineering

the basics

(the end)

IP-, MPLS-based

(the beginning)

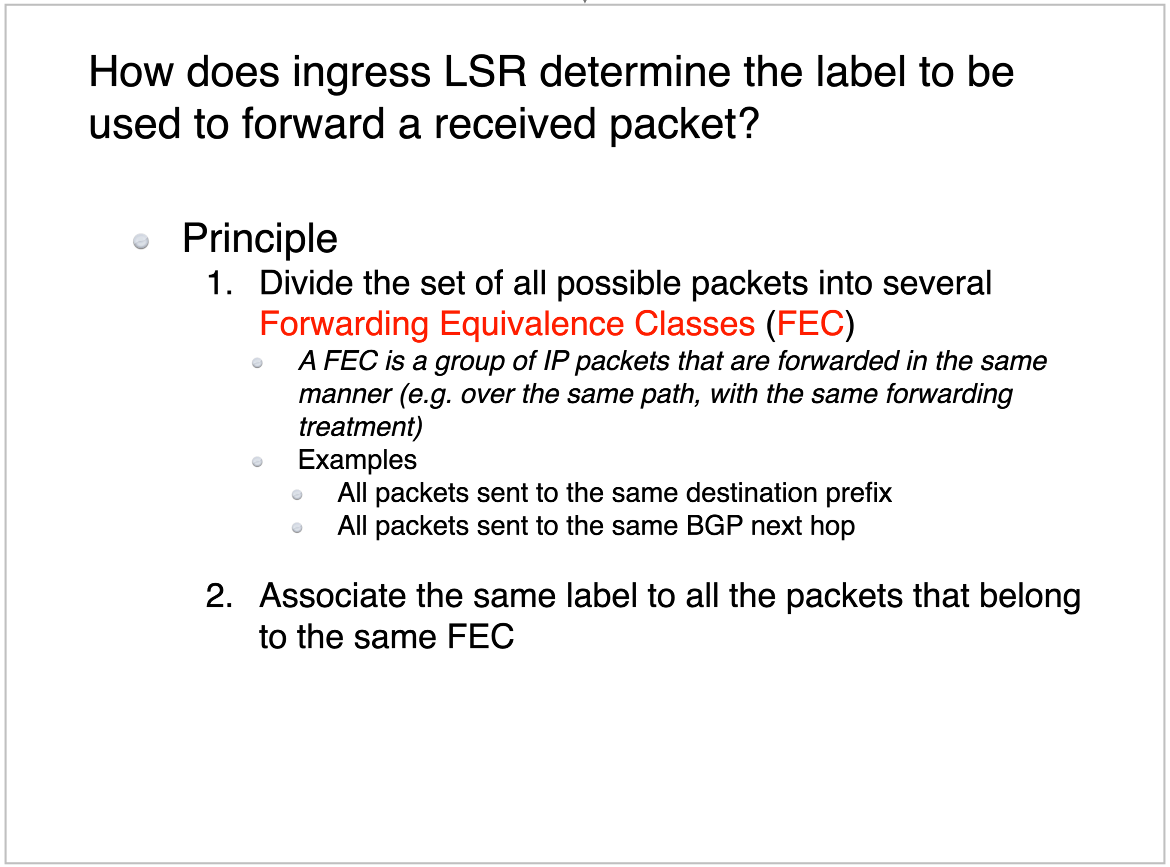
label switching

traffic engineering

the basics

(the end)

Switch to slides 94/117 from 22 Sep 2020



How does ingress LSR determine the label to be used to forward a received packet?

- Principle
  1. Divide the set of all possible packets into several Forwarding Equivalence Classes (FEC)
     - *A FEC is a group of IP packets that are forwarded in the same manner (e.g. over the same path, with the same forwarding treatment)*
     - Examples
       - All packets sent to the same destination prefix
       - All packets sent to the same BGP next hop
  2. Associate the same label to all the packets that belong to the same FEC

label switching

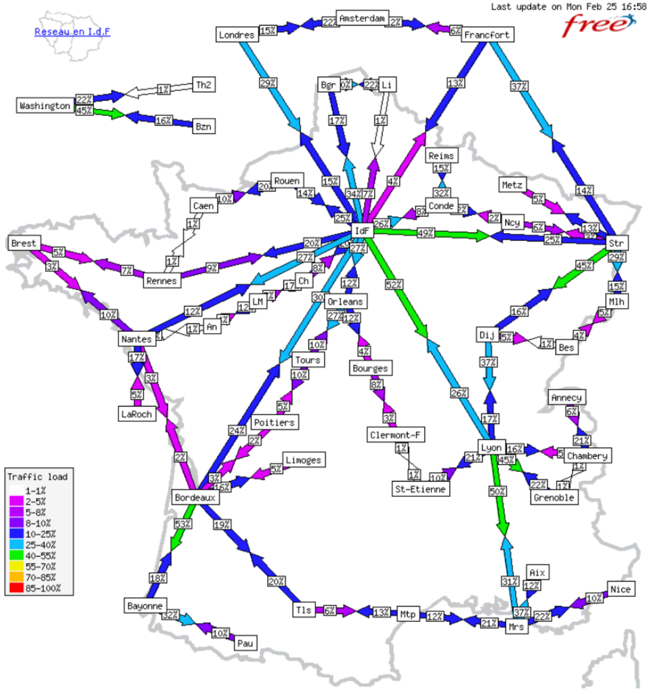
traffic engineering

IP-, MPLS-based

(the beginning)

Switch to slides 51/83 from 29 Sep 2020

Traffic load in large IP networks



15

Addendum to slides of 29 Sep 2020

Let's look at an example in which ECMP underperforms because of collisions

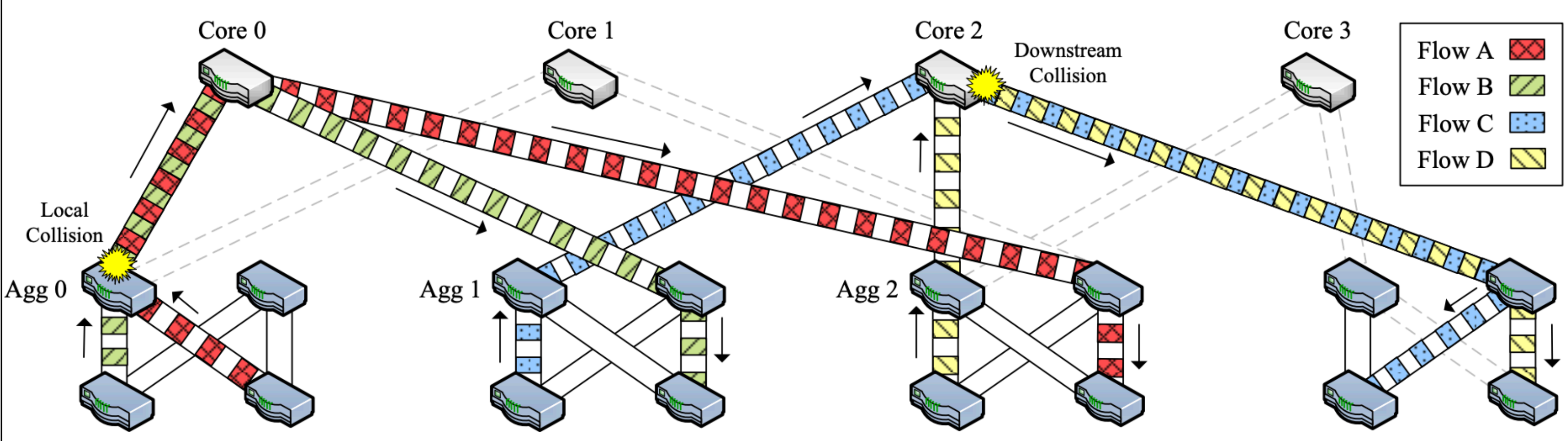
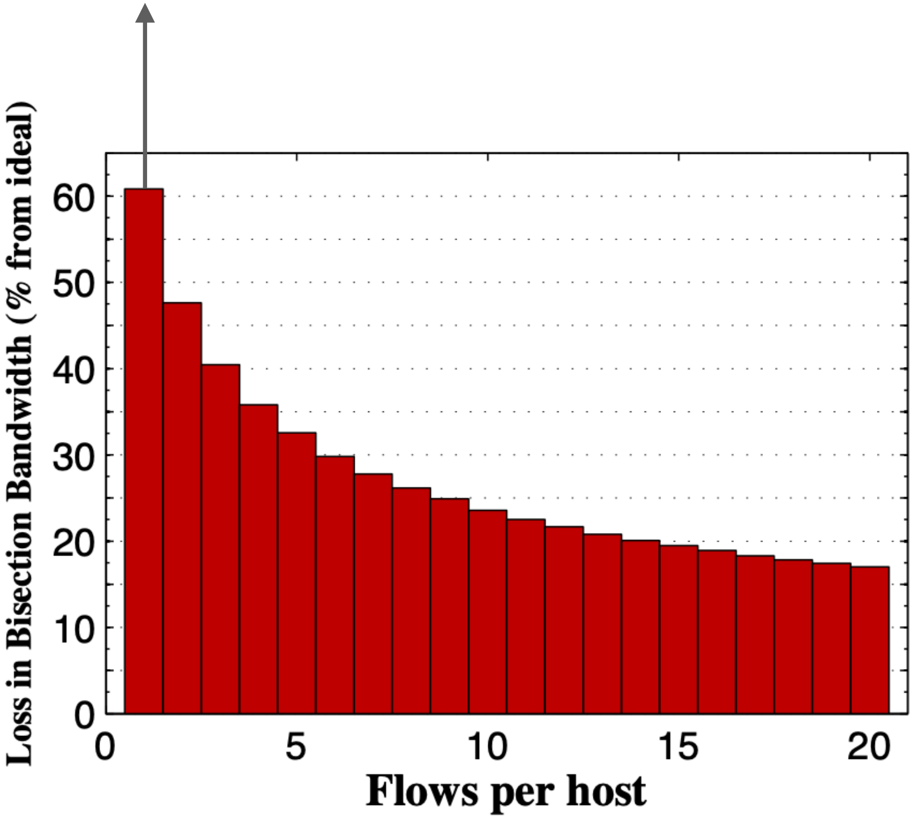


Figure 2: Examples of ECMP collisions resulting in reduced bisection bandwidth. Unused links omitted for clarity.

Hedera: Dynamic Flow Scheduling for Data Center Networks, USENIX NSDI 2010

If each host transfers an equal amount of data to all remote hosts one at a time, hash collisions reduce the network’s bisection bandwidth by an average of 60.8%



Hedera: Dynamic Flow Scheduling for Data Center Networks, USENIX NSDI 2010 across 1000 simulatenous flows

If each host transfers an equal amount of data to all remote hosts ~~one at a time,~~ hash collisions reduce the network’s bisection bandwidth by an average of ~~60.8%~~

only 2.5%

across 1000 simulatenous flows

If each host transfers an equal amount of data to all remote hosts ~~one at a time,~~ hash collisions reduce the network's bisection bandwidth by an average of ~~60.8%~~

only 2.5%

Intuition

The cost of a collision decreases with the number of flows

Here, each link has 1000 slots to fill

Performance only degrades if substantially
more than 1000 flows hash to the same link

Advanced Topics in Communication Networks

Internet Routing and Forwarding

Laurent Vanbever

nsg.ee.ethz.ch

ETH Zürich (D-ITET)

6 October 2020