Advanced Topics in Communication Networks

Internet Routing and Forwarding

Laurent Vanbever

nsg.ee.ethz.ch

17 Nov 2020

Lecture starts at **14:15**

Subsets of the materials inspired and/or coming from Olivier Bonaventure

Last week on

Advanced Topics in Communication Networks

Fast Convergence

How do we *quickly* retrieve connectivity
upon *sudden* failures?

Lecture Outline:

1. Introduction / Motivation:
   - 1.1. What do we mean by convergence?
   - 1.2. Why should we care?
   - 1.3. What conditions the convergence time?
2. Fast Convergence in IP networks:
   - 2.1. Fast detection
   - 2.2 Fast propagation
   - 2.3 Fast computation
   - 2.4 Fast updates
     - 2.4.1. Loop-Free Alternates
     - 2.4.2. Prefix-Independent Convergence
3. Fast Convergence in MPLS networks (next lecture)

Longer term solution:

Reorganize the FIB data structure so that it allows for fast incremental updates.

Step 1: Pre-compute the backup state.

Step 2: Pre-load it in the reorganized FIB.

Step 3: Activate the pre-loaded backup state upon detecting a failure.

IGP → Loop-Free Alternates (LFA)

BGP → Prefix-Independent Convergence (PIC)

This week on

Advanced Topics in Communication Networks

Fast Convergence

How do we *quickly* retrieve connectivity upon *sudden* failures?

IP Multicast

How do we efficiently send traffic to a set of receivers?

Fast Convergence

IP Multicast

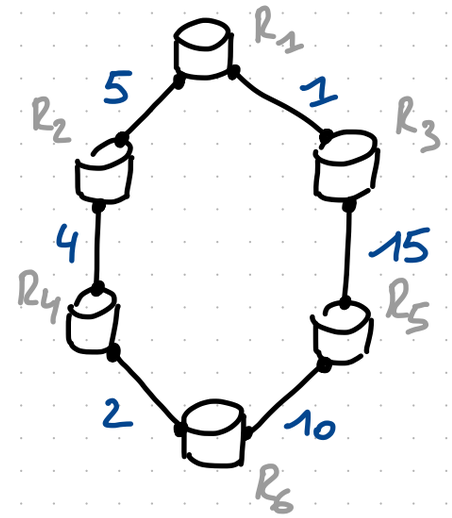
How do we *quickly* retrieve connectivity upon *sudden* failures?

Let's switch to
06_fast_convergence_notes.pdf

resuming from
page **25** / 32

Increasing LFA's coverage with Remote LFAs:

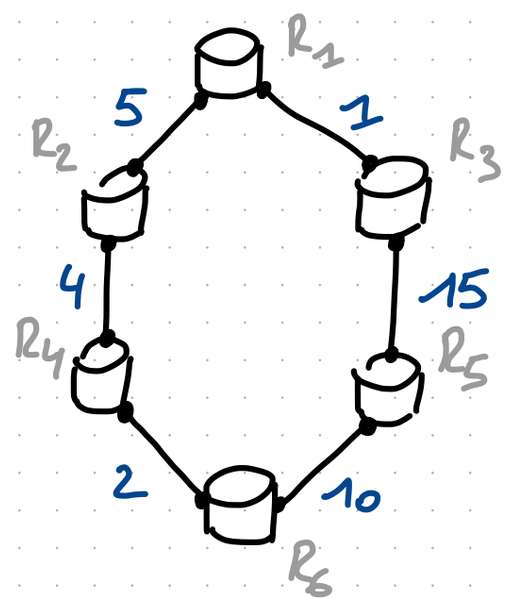
Let's look at another example:



In this topology, R_1 does not have a per-link, nor a per-prefix LFA to R_6. Why? Because R_3 uses R_1 to reach R_6 in the pre-convergence state.

Increasing LFA's coverage with Remote LFAs:

Let's look at another example:



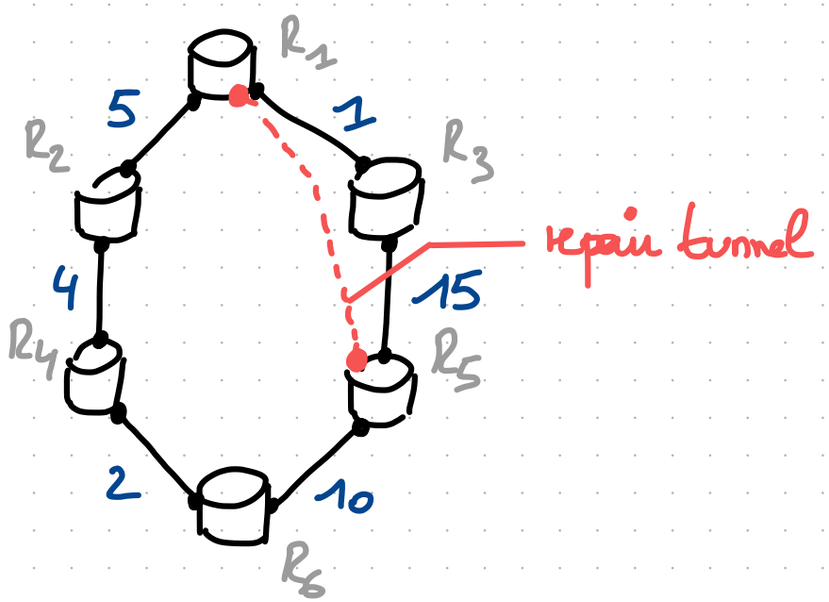
ring-based network

In this topology, R_1 does not have a per-link, nor a per-prefix LFA to R_6. Why? Because R_3 uses R_1 to reach R_6 in the pre-convergence state.

Remote LFAs enable to increase the LFA coverage by allowing a router to use remote, non-neighboring routers as repair nodes by tunneling to them.

IP-based, "with a twist" of LDP-based MPLS

Let's look at the previous example again:



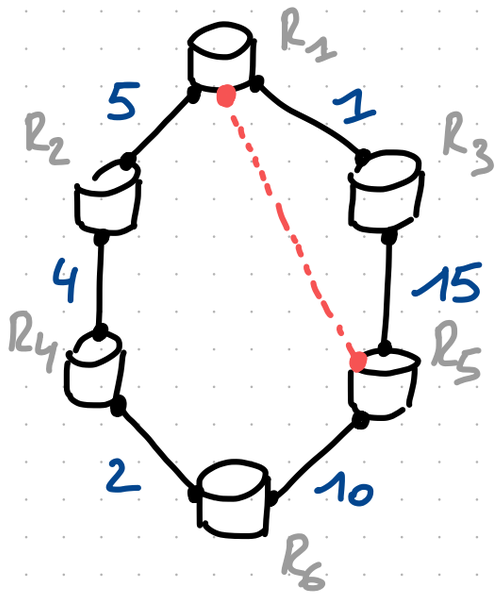
- While R_3 uses R_1 to reach R_6, it does not use R_1 to reach R_5.
- R_5 reaches R_6 directly (i.e. not via R_3)

=> By encapsulating its R_6-directed traffic to R_5 and sending that encapsulated traffic to R_3, R_1 can retrieve connectivity!

How do we compute remote LFAs?

Given $D_{opt}(a,b)$, a function that returns the shortest-path distance between a and b.

- On router X:
  - ∀ destination Y:
    - Let nh be the pre-converge next-hop used by X to reach Y (according to Dopt)
    - Let P be the set of nodes that X can reach <u>without</u> traversing (X, nh).
    - Let Q be the set of nodes that can reach Y <u>without</u> traversing (X, nh)
    - Let candidates_RLFA = $P \cap Q$
    - return candidates_RLFA.



- Considering R_1 and R_6 again:
  - $P = \{R_3, R_5\}$ the set of nodes R_1 can reach NOT going via (R_1, R_2).
  - $Q = \{R_2, R_4, R_5\}$ the set of nodes which reach R_6 NOT via (R_1, R_2)
  - $P \cap Q = \{R_5\}$ → the only RLFA available to protect R_6 is R_5.

Not RFLA do NOT guarantee full coverage.

Consider what happens to P and Q with (R_3, R_5) set to 23 instead of 15...

(R)LFA gives us a simple condition a router can check locally to know to which neighbor it can safely redirect traffic to.

The next question now becomes:

"How do we organize the FIB to allow for a fast activation of the backup state?"

One possible solution:

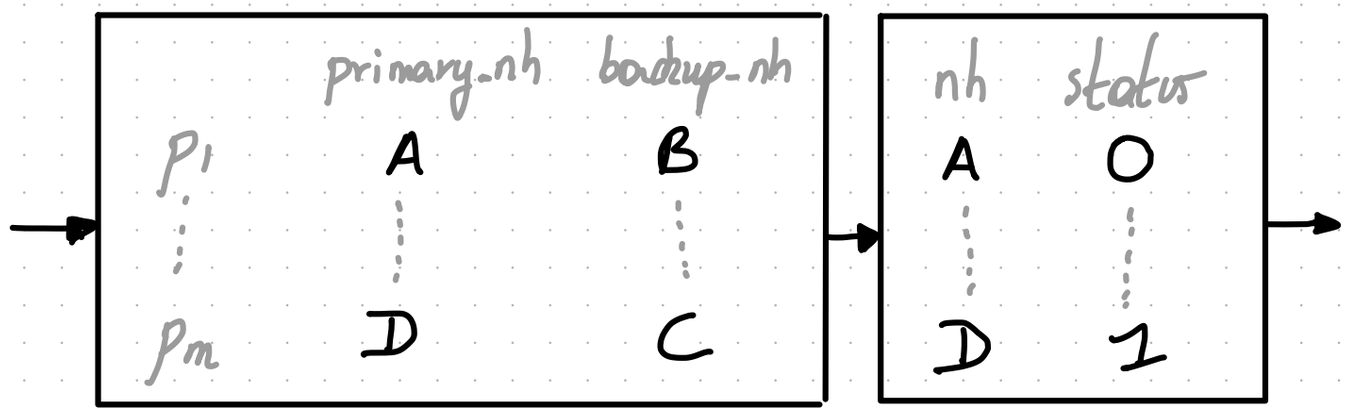


TABLE I

put primary and backup nh in metadata

Implementable in P4!

TABLE II

```
if status==0:
  egress_port=backup;
else
  egress_port=primary;
```

cf. last week's exercises session

"Cheap trick": Prioritize FIB updates according to how much traffic each prefix sees.

Longer term solution:

Reorganize the FIB data structure so that it allows for fast incremental updates.

Step 1: Pre-compute the backup state.

Step 2: Pre-load it in the reorganized FIB.

Step 3: Activate the pre-loaded backup state upon detecting a failure.

IGP → Loop-Free Alternates (LFA)

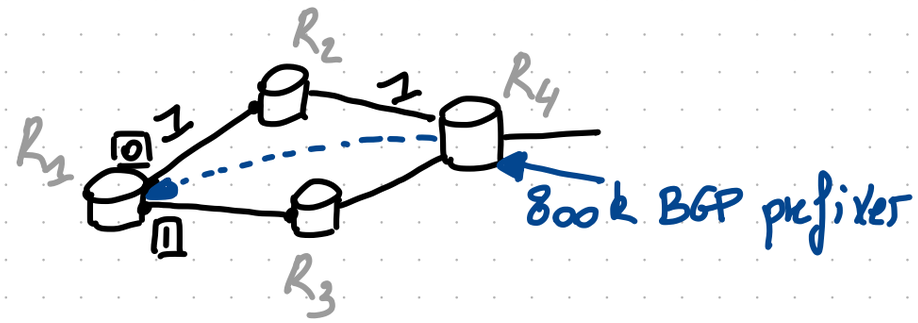
BGP → Prefix-Independent Convergence (PIC) — 2.4.2

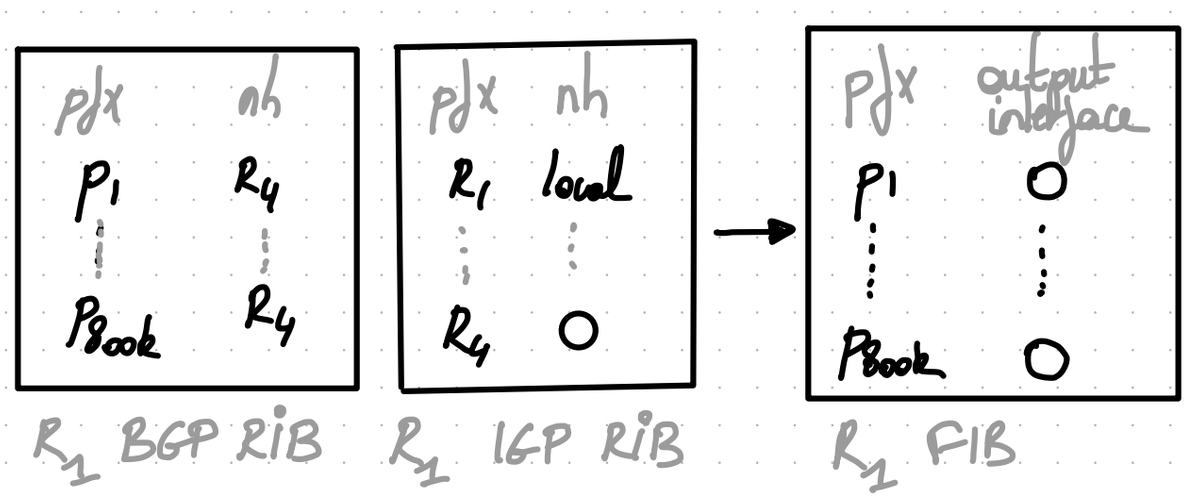
2.4.2. BGP Prefix Independent Convergence (PIC):

Goal: Enable routers to quickly switchover to pre-installed alternate paths upon failures that affect BGP routes.

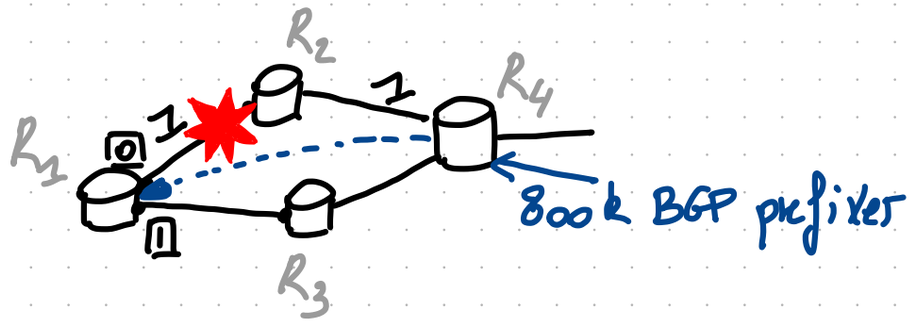
(Make BGP as fast to converge as the IGP)

Problem: "Flat" FIBs





Problem: "Flat" FIBs



pfx	nh
P_1	R_4
⋮	⋮
P_{800k}	R_4

R_1 BGP RIB

pfx	nh
R_1	local
⋮	⋮
R_4	1

R_1 IGP RIB

pfx	output interface
P_1	1
⋮	⋮
P_{800k}	1

R_1 FIB

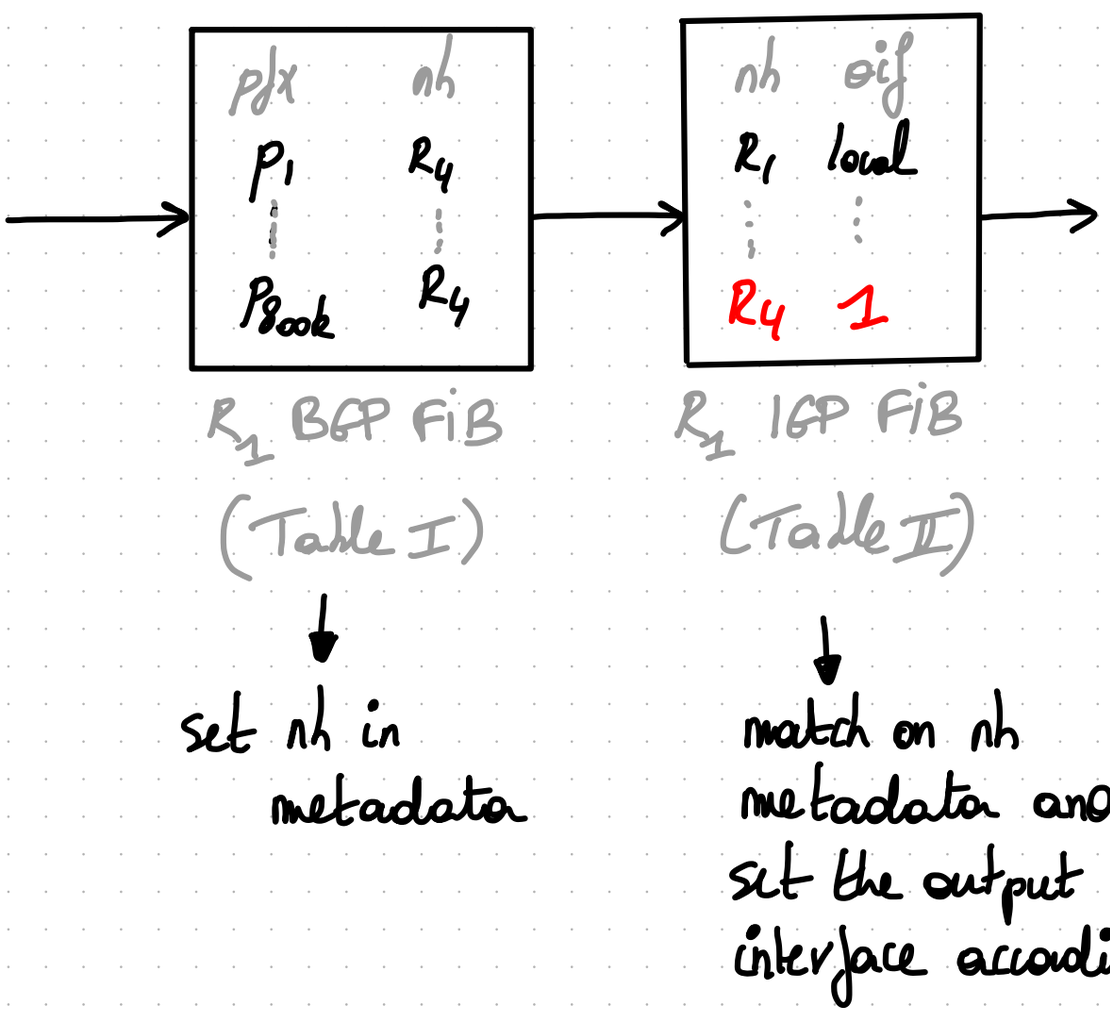
Upon the failure of (R_1, R_2) link, R_1 has to perform 800k updates to its FIB...

The fundamental problem is that the dependency between BGP next hops and the IGP one is NOT maintained in the FIB. It is "flattened".

The fundamental problem is that the dependency between BGP next hops and the IGP one is NOT maintained in the FIB. It is "flattened".

Solution: Maintain the hierarchy between BGP next-hops and IGP next-hops in the FIB as well.

(again)
easy to implement
in P4



Let's now switch to
06b_fast_convergence_mpls_notes.pdf

3. Fast convergence in MPLS networks:

Principles

- Pre-establish secondary LSPs to protect for the failures of important primary LSPs.

  These LSPs don't carry traffic unless there is a failure.

- Switch to using secondary LSPs upon detecting the failure!

  This can be done immediately without any coordination with neighboring routers, provided the secondary LSP exist and is NOT impacted by the failure...

Existing solutions can be divided into:

3.1. End-to-end LSP protection.

3.2. Local LSP protection.

3. Fast convergence in MPLS networks:

Principles

- Pre-establish secondary LSPs to protect for the failures of important primary LSPs.

  These LSPs don't carry traffic unless there is a failure.

3. Fast convergence in MPLS networks:

Principles

- Pre-establish secondary LSPs to protect for the failures of important primary LSPs.

  These LSPs don't carry traffic unless there is a failure.

- Switch to using secondary LSPs upon detecting the failure!

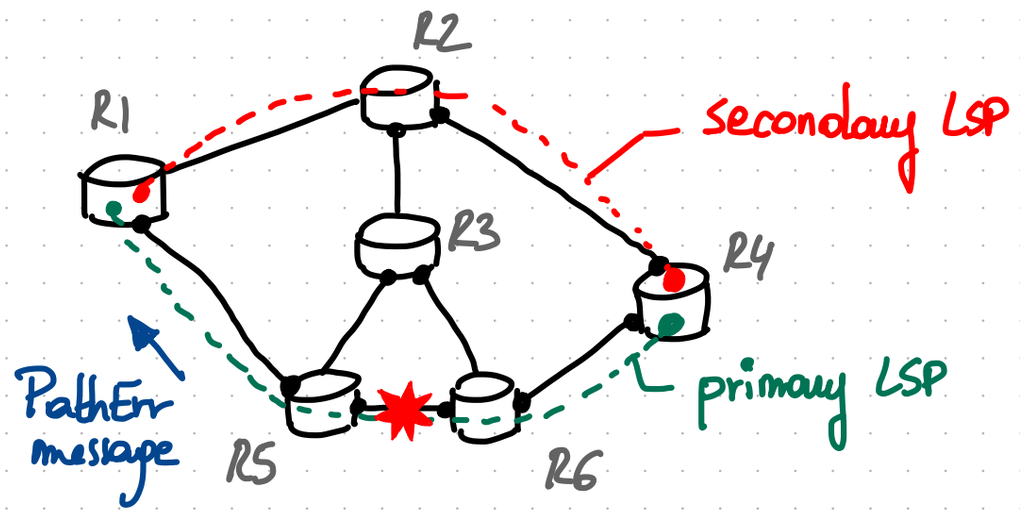
  This can be done immediately without any coordination with neighboring routers, provided the secondary LSP exist and is NOT impacted by the failure ...

Existing solutions can be divided into :

3.1. End-to-end LSP protection.

3.2. Local LSP protection.

3.1. End-to-end LSP protection :

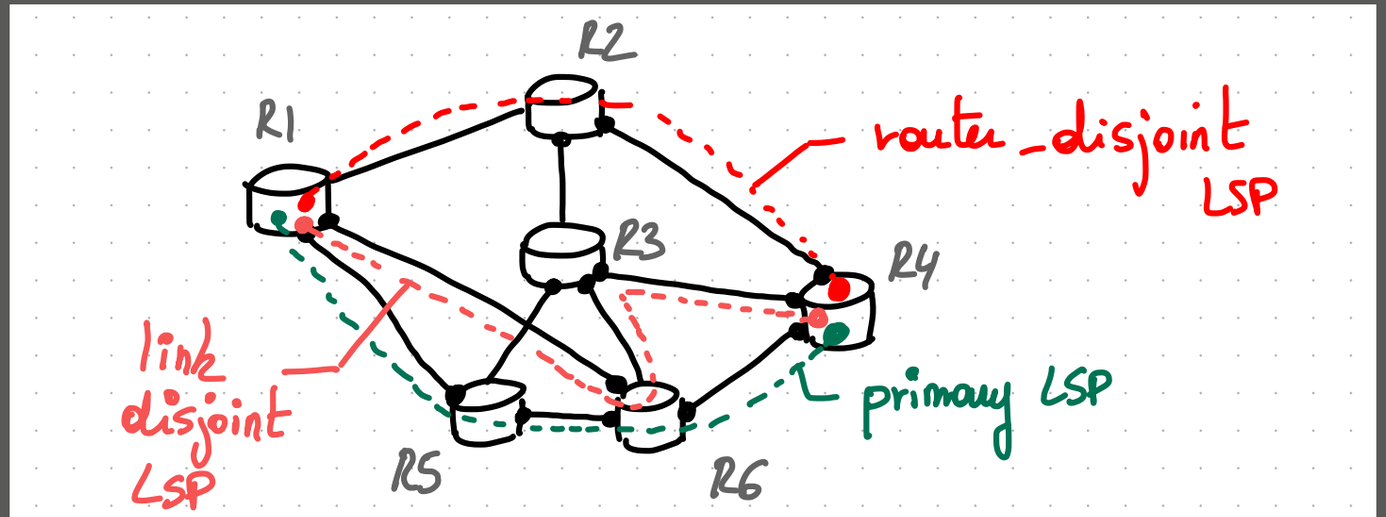


In this mode, a secondary LSP (here, in red) is established between the ingress and the egress LSR. When a failure happens, the adjacent router sends a PathErr message to the ingress which triggers the switch.

For this to work, the secondary LSP must rely on disjoint physical ressources...

Typical protection schemes include:

- Router_disjoint protection LSP which do not use any of the same routers as the primary LSP.
- Link_disjoint protection LSP which do not use any of the same links as the primary LSP.



The ingress LSR can compute these LSRs using its path selection algorithm and signal them using RSVP-TE (using the Explicit Route Object / ERO).

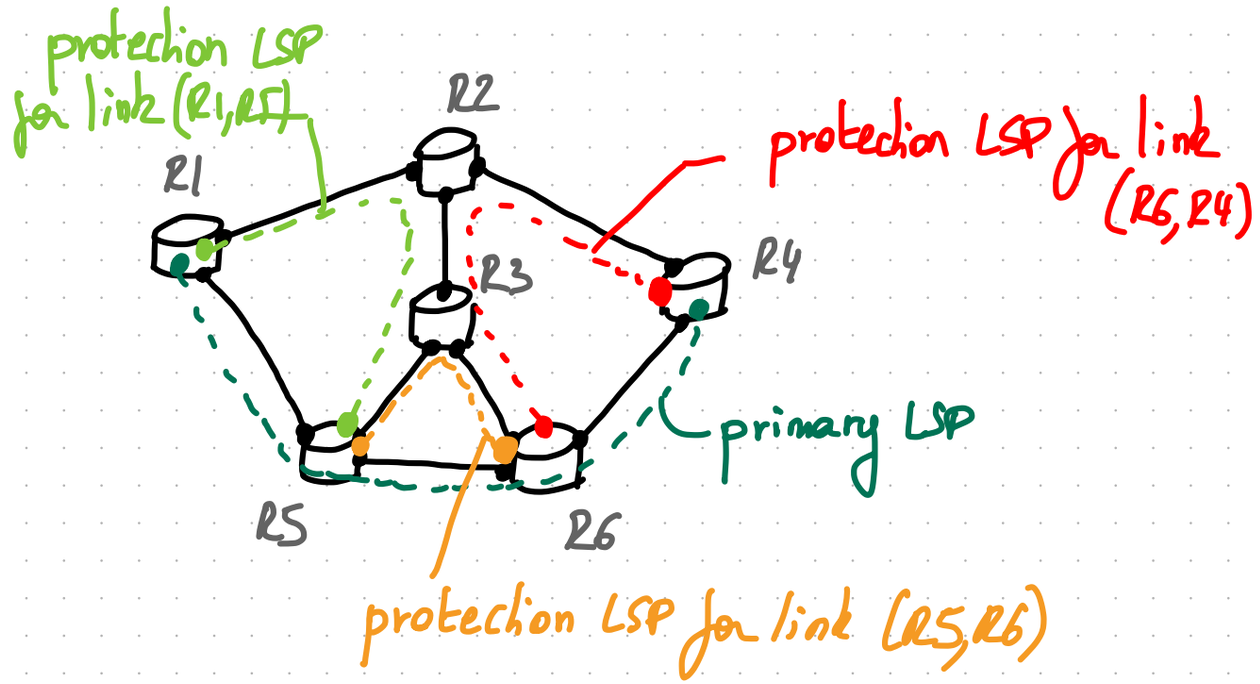
ingress is driving the convergence

Pros: Ingress can IMMEDIATELY activate the secondary LSP, without any coordination.

Cons:

- One protection LSP must be established for each primary LSP. This effectively doubles the amount of memory needed.
- The failure information (PathErr) must travel all the way to the ingress before connectivity can be retrieved. This is slow...

3.2. Local LSP protection



In this mode, each LSR crossed by the primary LSP will signal a protection LSP to cover for the failure of each link used by the primary LSP.

Pros: Traffic can be immediately switched onto a protection LSP by the router detecting the failure (not only the ingress).

Cons: Depending on the network, a large number of protection LSPs might be required.

Fast Convergence

How do we *quickly* retrieve connectivity upon *sudden* failures?

IP Multicast

How do we efficiently send traffic to a set of receivers?

Fast Convergence

IP Multicast

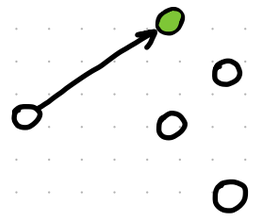
How do we efficiently send traffic to a set of receivers?

Let's switch to
07_ip_multicast_notes.pdf

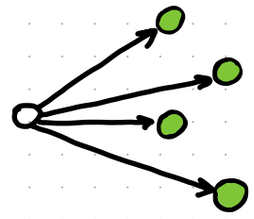
Advanced Topics in Communication Networks

L9: IP MULTICAST | 17.11.2020

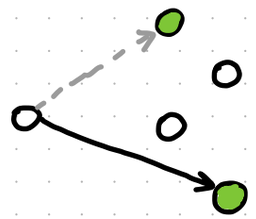
Prof. Laurent VANBEVER - nsg.ee.ethz.ch



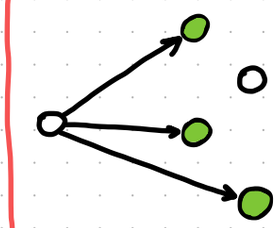
UNICAST
one-to-one



BROADCAST
one-to-all



ANYCAST
one-to-one-of-many



MULTICAST
one-to-many
many-

How do we efficiently transmit data to a set of receivers?

How do we efficiently transmit data to a set of receivers?

today's fundamental question

Let's consider two possible solutions to this question

Let's consider two possible solutions to this question

1. Source-based solution :
   - Sender simply sends as many copies of each IP packet as there are receivers.
   - Easy to implement / Waste a lot of bandwidth, not efficient!

Let's consider two possible solutions to this question

1. Source-based solution:
   - Sender simply sends as many copies of each IP packet as there are receivers.
   - Easy to implement / Waste a lot of bandwidth, not efficient!

2. Network-based solution:
   - Sender transmits one copy of each IP packet (as in IP unicast).
   - Network (i.e. the routers) take care of distributing this information to all the receivers.
   - Efficient (each packet crosses each link only once) / Hard to implement (need new protocols and forwarding mechanisms).

Let's consider two possible solutions to this question

1. Source-based solution:
   - Sender simply sends as many copies of each IP packet as there are receivers.
   - Easy to implement / Waste a lot of bandwidth, not efficient!

2. Network-based solution: — our winner
   - Sender transmits one copy of each IP packet (as in IP unicast).
   - Network (i.e. the routers) take care of distributing this information to all the receivers.
   - Efficient (each packet crosses each link only once) / Hard to implement (need new protocols and forwarding mechanisms).

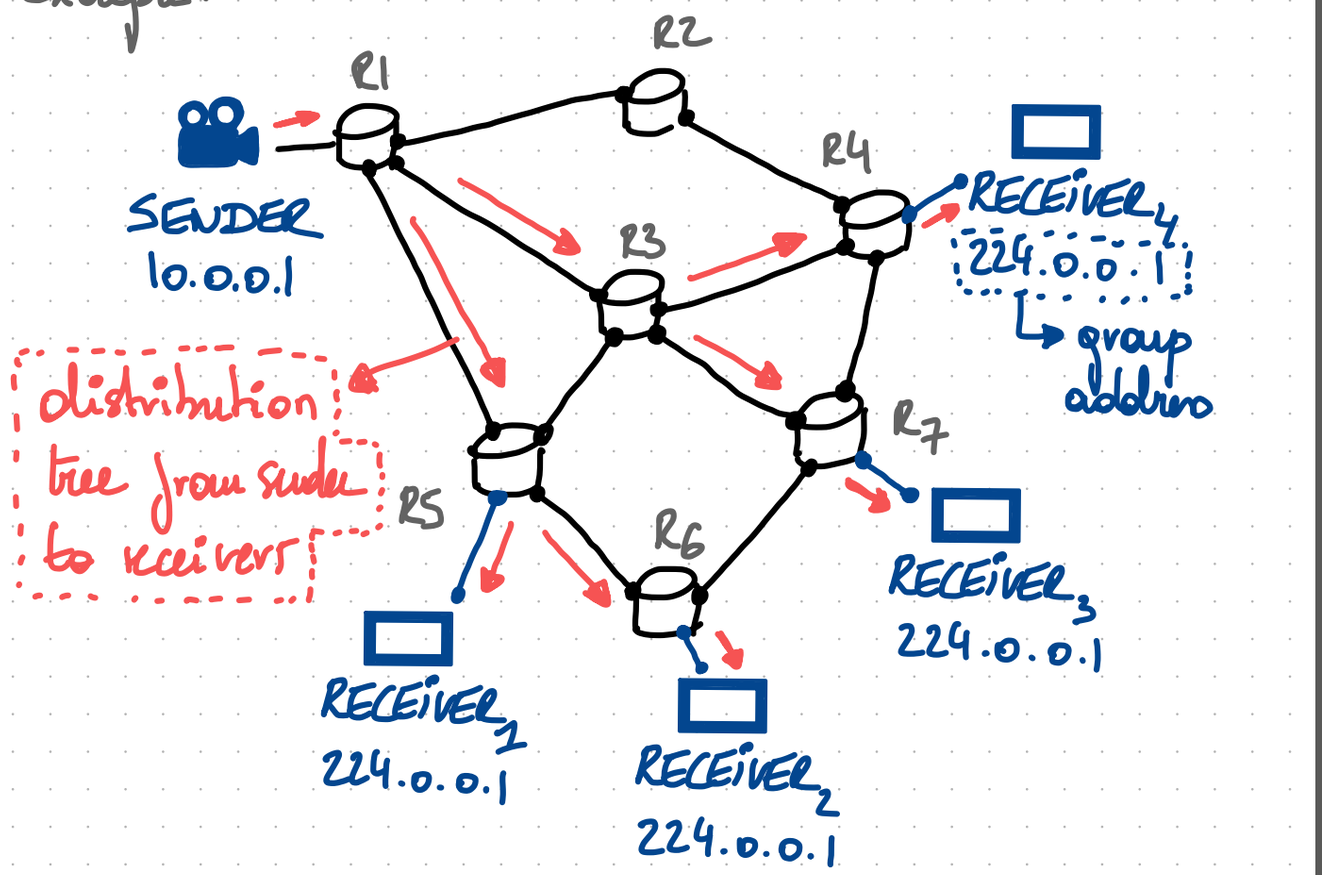
Network-based Multicast:

Principle: Sender sends multicast packets towards group of receivers identified by a group address.

Intermediate routers retransmit each received multicast packet such that:

- packets reach all receivers;
- packets traverse each link only once.

Example:


R1
R2
R3
R4
R5
R6
R7
SENDER
10.0.0.1
RECEIVER4
224.0.0.1
group address
distribution tree from sender to receivers
RECEIVER1
224.0.0.1
RECEIVER2
224.0.0.1
RECEIVER3
224.0.0.1

IP Multicast: Outline

1. How do we address a group of receivers?
2. How does a host receive traffic destined to a multicast address?
3. How do routers figure out which hosts belong to which group?
4. How do routers dynamically construct efficient distribution trees from sender to receiver?
   - 4.1. Pro-active solutions
   - 4.2. Reactive solutions
     - 4.2.1. "Flood and Prune"
     - 4.2.2. Rendez-vous Point
   - 4.3. Protocol Independent Multicast (PIM)

IP Multicast: Outline

1. How do we address a group of receivers?
2. How does a host receive traffic destined to a multicast address?
3. How do routers figure out which hosts belong to which group?
4. How do routers dynamically construct efficient distribution trees from sender to receiver?
   - 4.1. Pro-active solutions
   - 4.2. Reactive solutions
     - 4.2.1. "Flood and Prune"
     - 4.2.2. Rendez-vous Points
   - 4.3. Protocol Independent Multicast (PIM)

1/ How do we address a group of multicast receiver?

A subset of the IP space is reserved for multicast:

224.0.0.0 - 239.255.255.255

(leading bits "1110")

Some of these addresses have pre-defined allocations:

224.0.0.1 : All hosts
224.0.0.2 : All multicast routers
224.0.0.5 : All OSPF routers
...

Some of these addresses can only be used within an AS (they are not publicly routed)

239.0.0.0 - 239.255.255.255
(typically used for IPTV).

A multicast sender simply sends to a multicast IP address.

Check out
iptv-ch.github.io
for example of
Swiss-based Multicast IPs

239.186.68.1 for SRF1 in Swisscom
239.186.68.1 for SRF2

233.35.254.125 for SRF1 in Sunrise
233.35.254.126 for SRF2

pro-tip Open the RTP's URL with VLC…

README.md

IPTV m3u Playlists for Swiss Providers

This repository contains M3U playlist files for Swiss IPTV Providers. Here are the files available:

IPTV open channels from Netplus + Sunrise + Swisscom as language french only, english only, german only, italian only.

https://iptv-ch.github.io/tvopenchfr.m3u

https://iptv-ch.github.io/tvopenchen.m3u

https://iptv-ch.github.io/tvopenchde.m3u

https://iptv-ch.github.io/tvopenchit.m3u

CityCable TV Lausanne ftth

https://iptv-ch.github.io/citycable.m3u

special thanks to bendreth for their update for community and customers of CityCable TV Lausanne over ftth date 2019-12-18

Netplus TV HD + SD

https://iptv-ch.github.io/netplus.m3u

Many of the channels available on Netplus TV partners.
This will only work on your home network if netplus partners like Citycable (old BoisyTV) is your broadband provider.
This file only lists the HD channels in the case where a channel is available on both HD and non-HD.
EPG information from https://xmltv.ch/ are included.
More information.

Swisscom TV SD only

https://iptv-ch.github.io/swisscom-sd.m3u

Many of the channels available on Swisscom TV.
This will only work on your home network if Swisscom is your broadband provider.
This file only lists the SD channels.
EPG information from https://xmltv.ch/ is included.
More information.

Swisscom TV HD + SD

https://iptv-ch.github.io/swisscom-hd.m3u

Many of the channels available on Swisscom TV.
This will only work on your home network if Swisscom is your broadband provider.
This file only lists the HD channels in the case where a channel is available on both HD and non-HD.
EPG information from https://xmltv.ch/ is included.
More information.

IP Multicast: Outline

1. How do we address a group of receivers?
2. How does a host receive traffic destined to a multicast address?
3. How do routers figure out which hosts belong to which group?
4. How do routers dynamically construct efficient distribution trees from sender to receiver?
   - 4.1. Pro-active solutions
   - 4.2. Reactive solutions
     - 4.2.1. "Flood and Prune"
     - 4.2.2. Rendez-vous Point
   - 4.3. Protocol Independent Multicast (PIM)

2/ How do the hosts receive multicast traffic?

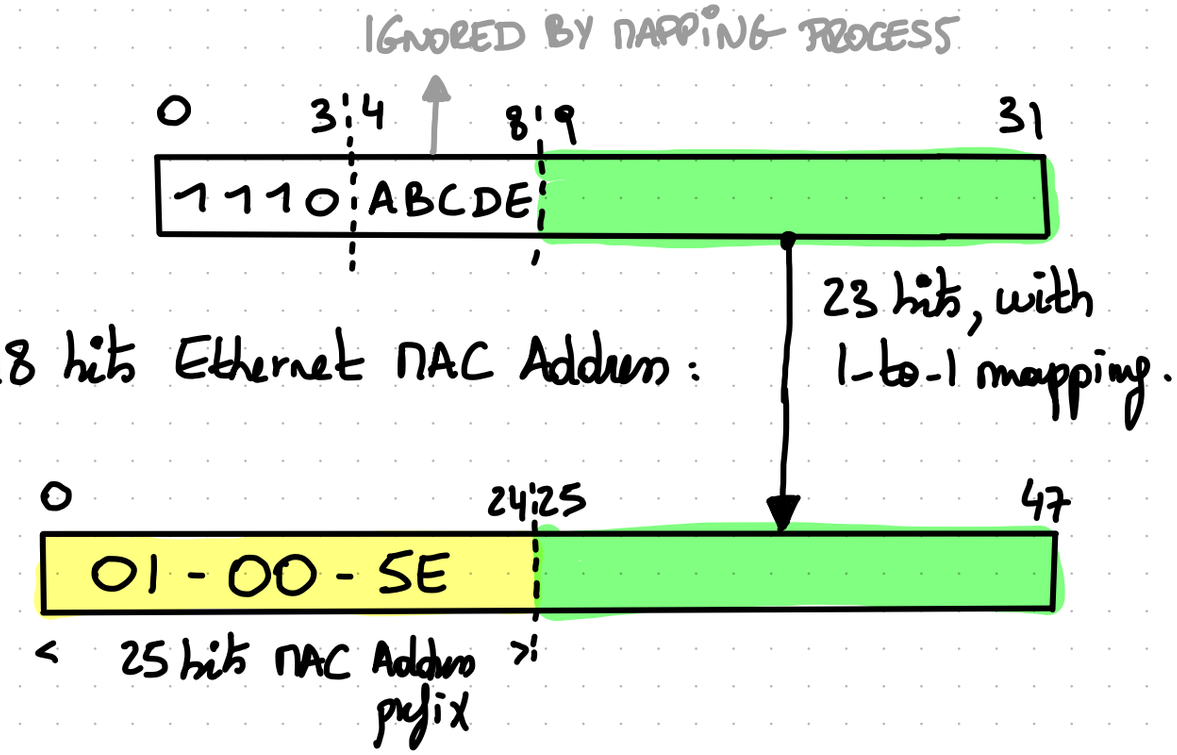
Ethernet addresses are of 2 kinds:

1. Physical addresses which identify one Ethernet adapter.
   -> These addresses start with "0" in the first byte.
2. Logical addresses which identify a group of Ethernet destinations.
   -> These addresses start with "1" in the first byte.

Ethernet adapters can be configured to capture frames whose destination is a set of their logical addresses. (in addition to their unicast address).

Hosts automatically figure out the logical MAC address from the IP multicast directly.

32 bits IPv4 Multicast Address:



Note that the mapping is not lossless: 32 IP multicast addresses are mapped to the same Ethernet logical address.

→ This can lead to unwanted traffic.

IP Multicast: Outline

1. How do we address a group of receivers?
2. How does a host receive traffic destined to a multicast address?
3. How do routers figure out which hosts belong to which group?
4. How do routers dynamically construct efficient distribution trees from senders to receivers?
   - 4.1. Pro-active solutions
   - 4.2. Reactive solutions
     - 4.2.1. "Flood and Prune"
     - 4.2.2. Rendez-vous Points
   - 4.3. Protocol Independent Multicast (PIM)

Internet Group Management Protocol (IGMP):

IGMP is a protocol used by hosts and adjacent routers to create multicast group membership.

Internet Group Management Protocol (IGMP):

IGMP is a protocol used by hosts and adjacent routers to create multicast group memberships.

- Hosts request membership to a group (i.e. a multicast IP address).
- Adjacent routers listen and keep track of these requests. They also periodically send out subscription queries. (one router is elected to do that)
- Adjacent routers then use Protocol Independent Multicast (PIM) to direct traffic from hosts sending multicast traffic to the hosts that have registered for it.

IP Multicast: Outline

1. How do we address a group of receivers?
2. How does a host receive traffic destined to a multicast address?
3. How do routers figure out which hosts belong to which group?
4. How do routers dynamically construct efficient distribution trees from sender to receiver?
   - 4.1. Pro-active solutions
   - 4.2. Reactive solutions
     - 4.2.1. "Flood and Prune"
     - 4.2.2. Rendez-vous Point
   - 4.3. Protocol Independent Multicast (PIM)

4/ How do we dynamically construct efficient shortest-path distribution tree from senders to receiver?

Challenge: How do routers learn about where the various receivers are and keep track of those over time.

Two possible solutions:

4.1. Pro active: A routing protocol is used to distribute group membership so that each router knows the exact location of each group member. One can extend link-state protocols for that, e.g. MOSPF.

4.2. Reactive: Assume that group members are everywhere initially → Broadcast the traffic. If a router receives unwanted traffic, it asks the upstream router to stop.

IP Multicast: Outline

1. How do we address a group of receivers?
2. How does a host receive traffic destined to a multicast address?
3. How do routers figure out which hosts belong to which group?
4. How do routers dynamically construct efficient distribution trees from sender to receiver?
   - 4.1. Pro-active solutions
   - 4.2. Reactive solutions
     - 4.2.1. "Flood and Prune"
     - 4.2.2. Rendez-vous Point
   - 4.3. Protocol Independent Multicast (PIM)

4.1. Pro-actively building distribution trees using link-state routing protocols

Principle:

1. Routers collect membership info using IGMP.
2. Group membership is flooded by link-state protocols using a new type of messages.
3. Each router computes the shortest path tree (S,G) for each source S and each group G.

Note the shortest path tree (S,G) is built on demand, whenever the router receives a packet destined to G.

Pros: Routers have full knowledge.

Cons:
- Possibly important memory overhead on ALL routers, independently on whether or not they see any multicast traffic.
- The flooding of group membership msgs compete with normal link-state msgs (e.g. link down...)
- As the number of sources and groups grows, computing shortest path trees can become problematic.

IP Multicast: Outline

1. How do we address a group of receivers?
2. How does a host receive traffic destined to a multicast address?
3. How do routers figure out which hosts belong to which group?
4. How do routers dynamically construct efficient distribution trees from sender to receiver?
   - 4.1. Pro-active solutions
   - 4.2. Reactive solutions
     - 4.2.1. "Flood and Prune"
     - 4.2.2. Rendez-vous Point
   - 4.3. Protocol Independent Multicast (PIM)

4.2. Reactively building distribution trees:

2 possible solutions:

4.2.1 "Flood-and-Prune"

Principles:

1. Flood the multicast traffic in the entire network.
2. Prune branches when there is no receivers.

4.2. Reactively building distribution trees:

2 possible solutions:

4.2.1 "Flood-and-Prune"

Principles:
1. Flood the multicast traffic in the entire network.
2. Prune branches when there is no receivers.

4.2.2. Use rendez-vous points.

Principles:
1. Have one router acts as root of a shared distribution tree per group.
2. Have the sources encapsulate the traffic to the root.
3. Have the root multicast the encapsulated traffic alongside the shared traffic.

4.2. Reactively building distribution trees:

2 possible solutions:

4.2.1 "Flood-and-Prune"

Principles:

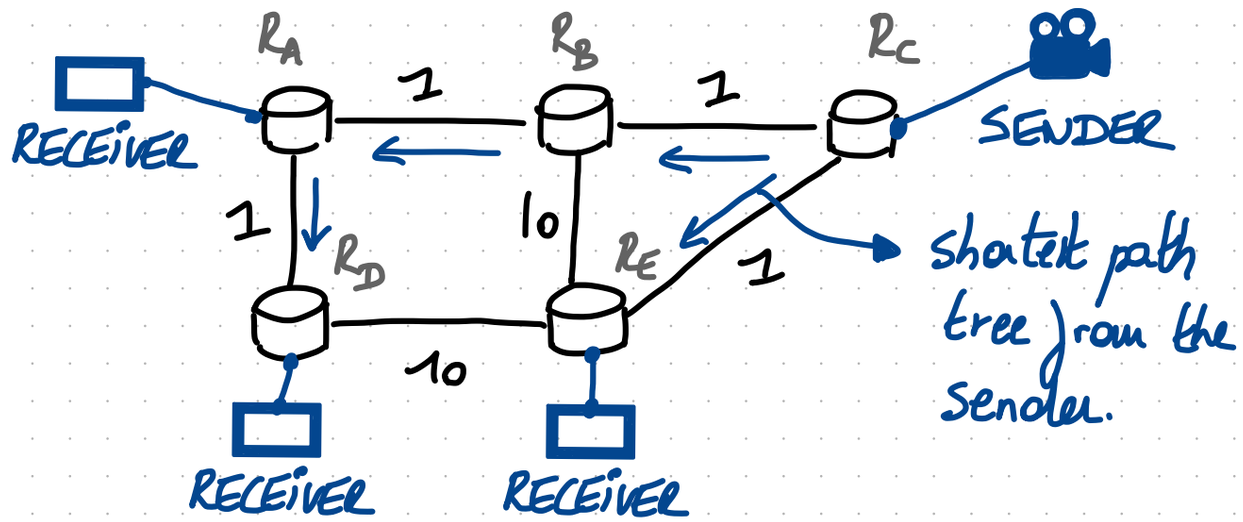
1. Flood the multicast traffic in the entire network.
2. Prune branches when there is no receivers.

let's look at this first

42.1. Flood and Prune

How do we broadcast traffic in a large network?

Goal: Broadcast traffic following the shortest path tree.



Insight: Flood traffic only when it arrives from the shortest-path upstream.

This strategy is known as:

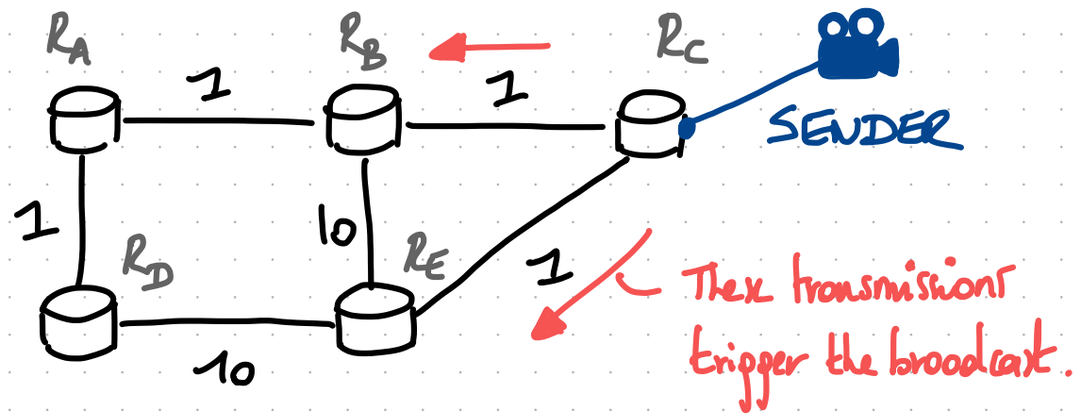
Reverse Path Filtering a RPF

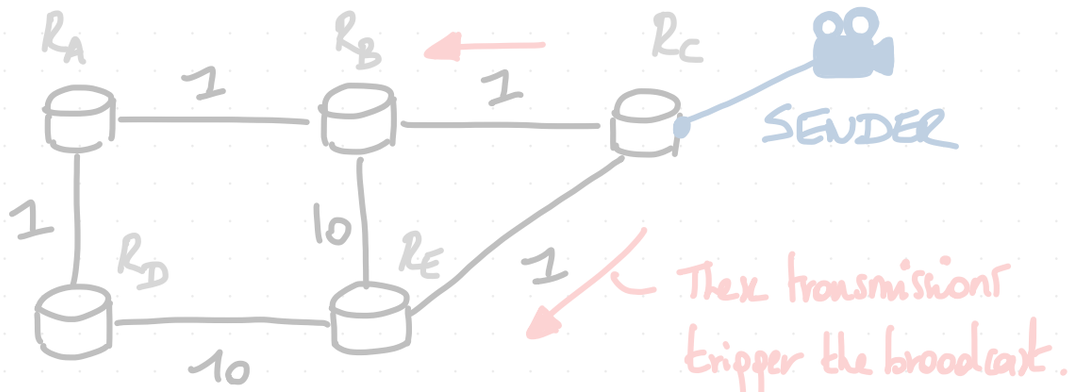
RPF Algorithm:

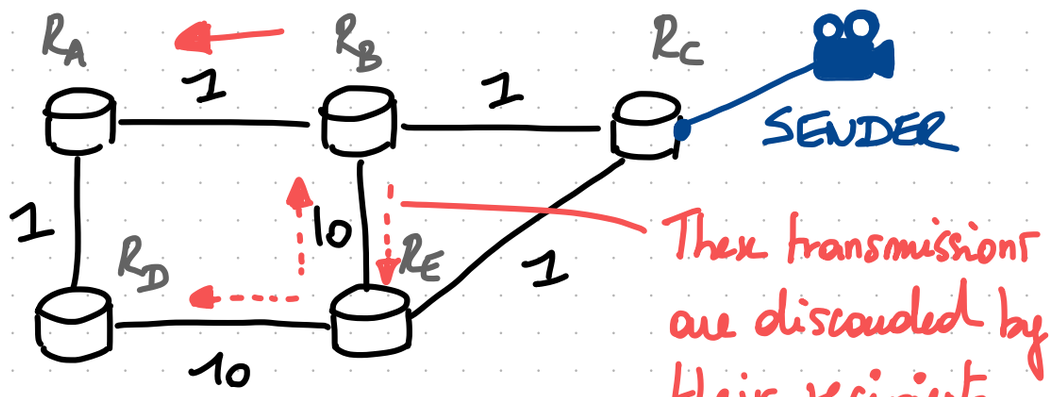
Upon receiving an IP packet from source S on interface in:

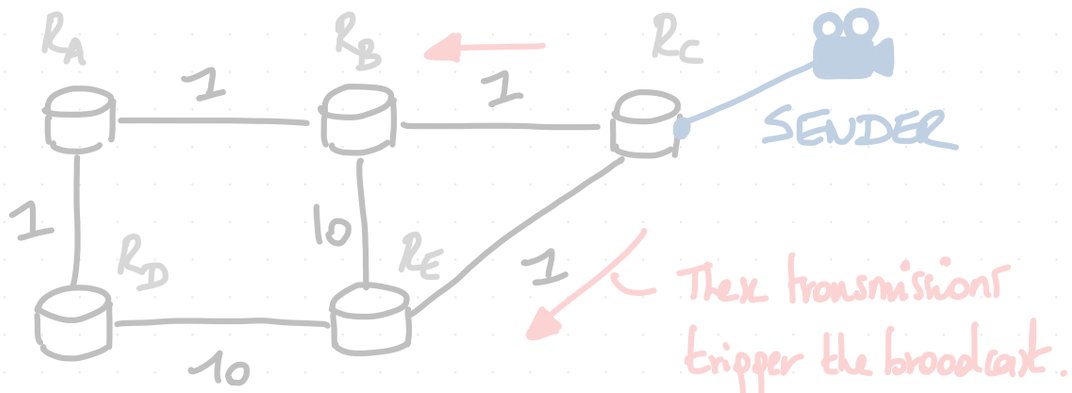
- if (in == next_hop_interface(S)):
  - for (interface ij_ in interfaces):
    - if (ij_ ≠ in):
      - send_packet(ij_)

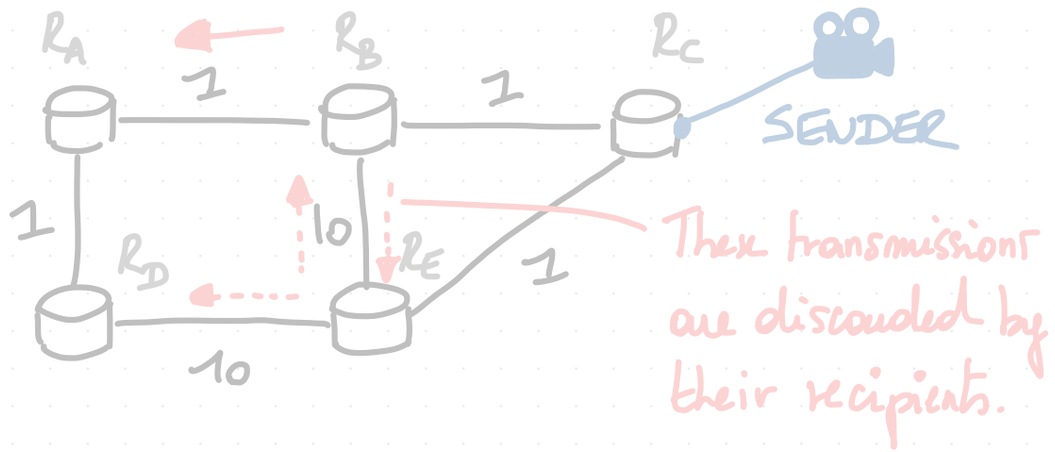
An interesting observation is that, unlike unicast routing which depends solely on the destination, multicast routing depends solely on the source address!

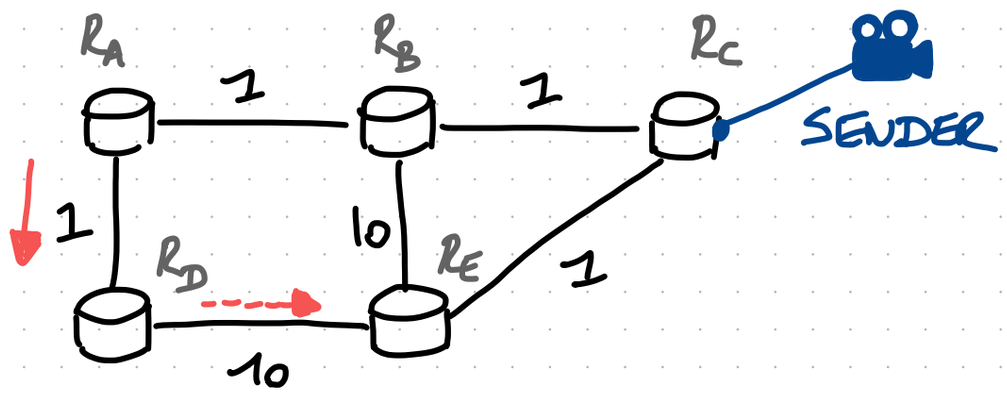

R_A
R_B
R_C
R_D
R_E
1
1
1
1
10
10
SENDER
These transmissions trigger the broadcast.


R_A
R_B
R_C
R_D
R_E
1
1
1
1
10
10
SENDER
These transmissions trigger the broadcast.


R_A
R_B
R_C
R_D
R_E
1
1
1
1
10
10
SENDER
These transmissions are discarded by their recipients.


R_A
R_B
R_C
SENDER
1
1
1
10
R_D
R_E
1
10
These transmissions trigger the broadcast.


R_A
R_B
R_C
SENDER
1
1
1
10
R_D
R_E
1
10
These transmissions are discarded by their recipients.


R_A
R_B
R_C
SENDER
1
1
1
10
R_D
R_E
1
10

How do we avoid unnecessary transmissions?

Sender-based solution:

For each possible source, each router computes the list of interfaces on which to broadcast such that it only includes the interfaces on the shortest path tree from the source.

How do we avoid unnecessary transmissions?

Sender-based solution:

For each possible source, each router computes the list of interfaces on which to broadcast such that it only includes the interfaces on the shortest path tree from the source.
For that routers can rely on the network topology learned through e.g. OSPF or IS-IS. Note that this is different from MOSPF. (It has nothing to do with group membership).

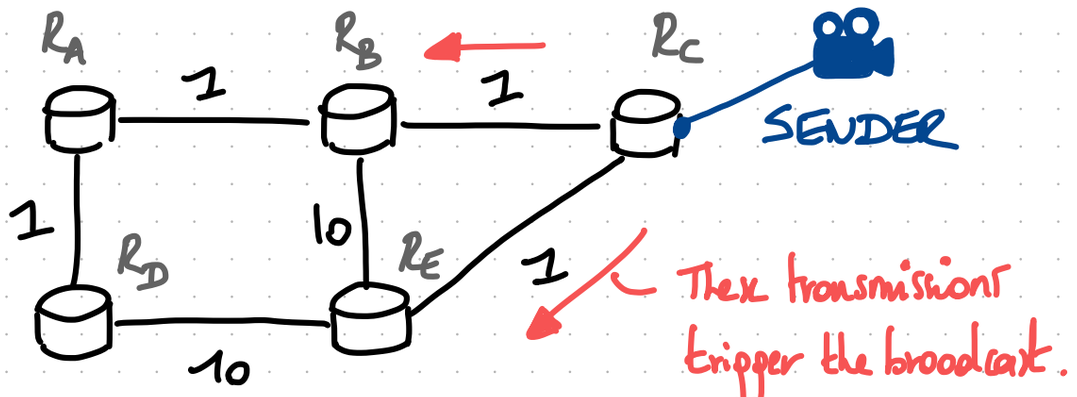
How do we avoid unnecessary transmissions?

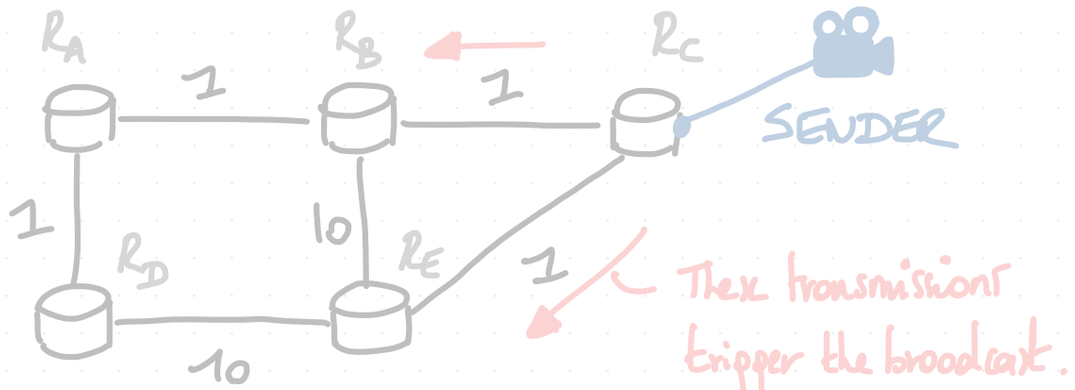
Sender-based solution:

For each possible source, each router computes the list of interfaces on which to broadcast such that it only includes the interfaces on the shortest path tree from the source.
For that routers can rely on the network topology learned through e.g. OSPF or IS-IS. Note that this is different from IGMP: it has nothing to do with group membership).

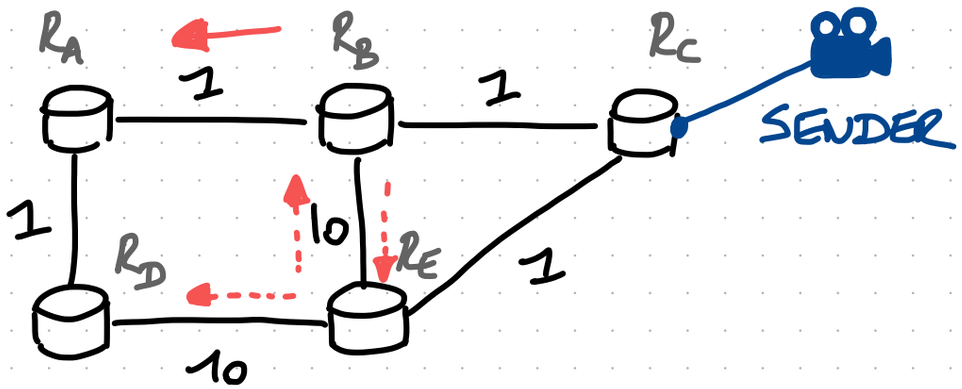
Receiver-based solution:

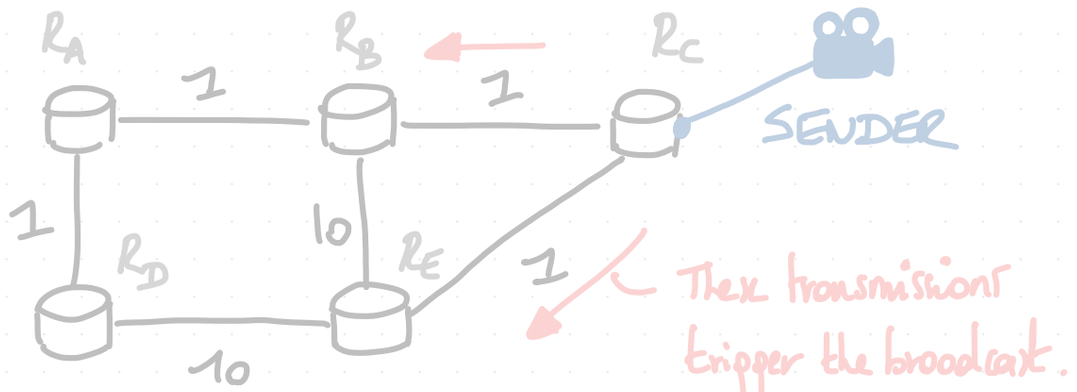
For each possible source, each router learns the list of outgoing interfaces on which to broadcast by having downstream routers tell them that a given interface is <u>not</u> on the shortest path.
Doing so is less intensive sender-based solution.


R_A
R_B
R_C
R_D
R_E
1
1
1
1
10
10
1
SENDER
These transmissions trigger the broadcast.

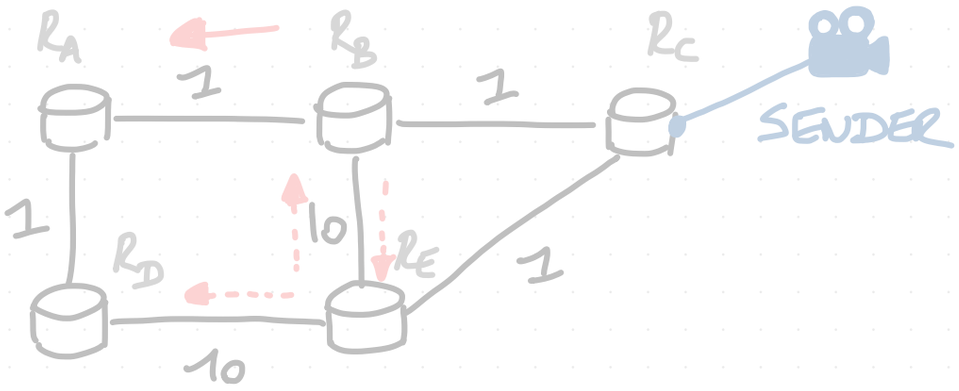


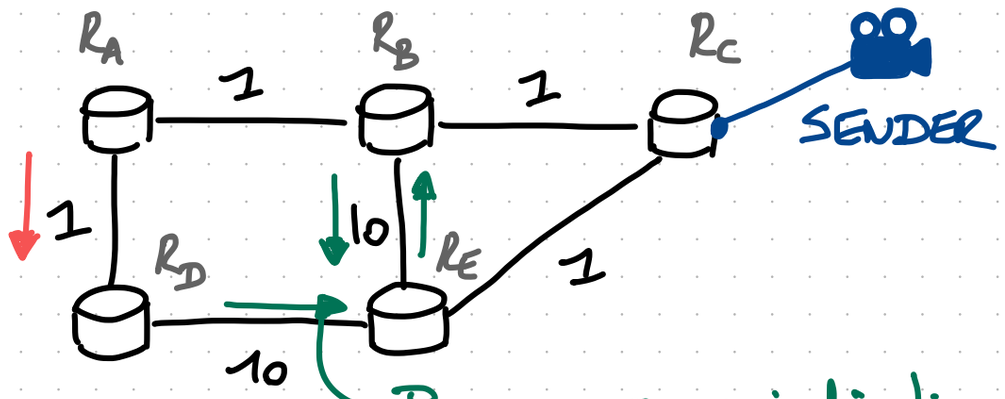
These transmissions trigger the broadcast.





These transmissions trigger the broadcast.





Prune message indicating to the upstream router that packets coming from SENDER should not be broadcasted on these interfaces anymore.

In practice, two types of PRUNE messages are sent:

1. Source-specific messages if packets are received on non shortest paths.
2. Group-specific messages if there is no group member downstream of a router.

The list of outgoing interfaces is periodically refreshed and flooding restarts.

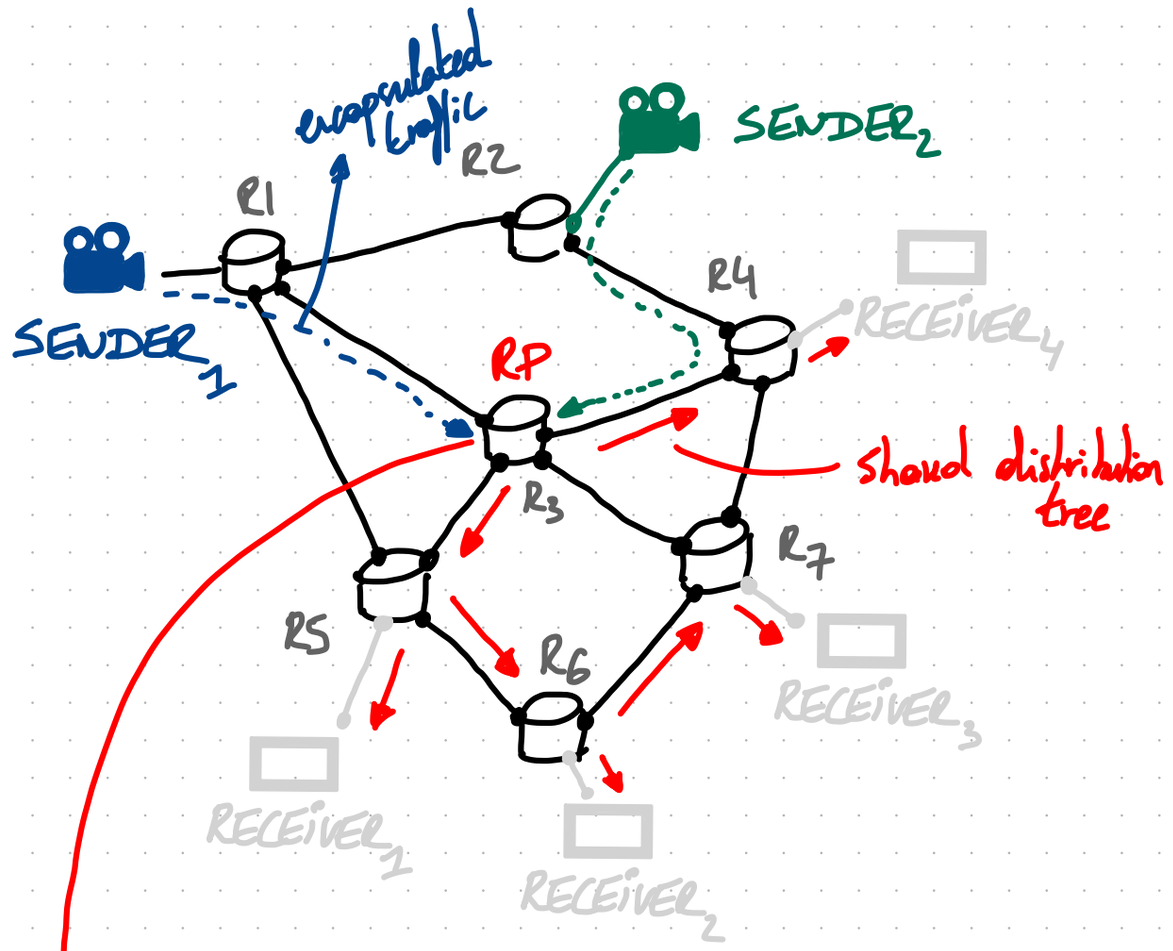
Pros: Approach is simple, "plug and play"

Cons: Approach is relatively costly:
- Routers need to maintain per (S,G) state
- Flooding is frequently reactivated.

4.2.2. Use shared trees and rendez-vous points

Principle :

- One router is configured as a Rendez-vous point (RP).
- All routers know the RP address.
- RP acts as the root of a shared tree for the group. This tree is denoted as (*, G).
- Multicast routers encapsulate packets sent by hosts to G and send them to the RP.
- RP redistributes these packets over the shared tree (*, G).
- Receivers dynamically join the shared tree.



RP's forwarding state $(*,G): \{R_5, R_7\}$

Note that RPs need to be configured in advance a a mechanism has to be put in place to discover them.

IP Multicast: Outline

1. How do we address a group of receivers?
2. How does a host receive traffic destined to a multicast address?
3. How do routers figure out which hosts belong to which group?
4. How do routers dynamically construct efficient distribution trees from sender to receiver?
   - 4.1. Pro-active solutions
   - 4.2. Reactive solutions
     - 4.2.1. "Flood and Prune"
     - 4.2.2. Rendez-vous Point
   - 4.3. Protocol Independent Multicast (PIM)

4.3. Protocol Independent Multicast (PIM):

PIM is the most widely-deployed multicast routing protocol.

PIM does not rely on a specific unicast routing protocol: it leverages the existing unicast routing table (which is populated by whatever protocol) to perform receiver-based optimization for RPF.

PIM has two "modes" :
- DENSE
- SPARSE

PIM DENSE works by periodically flooding and pruning so as to build source-specific distribution trees.

PIM SPARSE initially builds a shared distribution tree. The shared tree is rooted at a (typically pre-configured) Rendez-vous point.

If required, PIM SPARSE allows for the shared tree to be converted to a shortest-path tree.

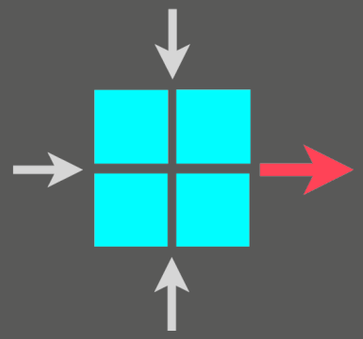
cf. today's exercises session!

We made it!

	Techniques
Performance	Traffic Engineering
	Load Balancing
	Quality of Service
	Multicast
Flexibility	Virtual Private Networks
Reliability	Fast Convergence

Advanced Topics in Communication Networks

Internet Routing and Forwarding



Laurent Vanbever

nsg.ee.ethz.ch

ETH Zürich (D-ITET)

17 Nov 2020