Institut Mines-Télécom

6LoWPAN: Fragmentation & Reassembly, Frame Delivery Modes, & Fragment Forwarding

Georgios Z. PAPADOPOULOS, Associate Professor, IMT Atlantique
georgios.papadopoulos@imt-atlantique.fr
www.georgiospapadopoulos.com
www.youtube.com/c/gzpapadopoulos

SOMMAIRE

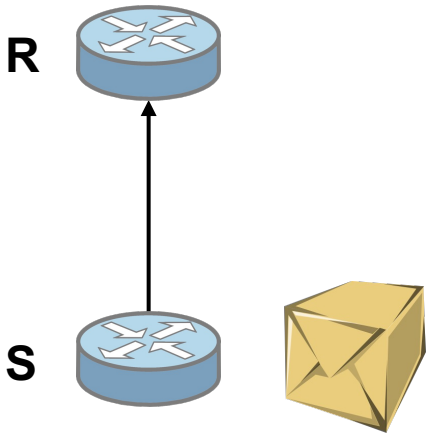
Institut Mines-Télécom

Chapter 1

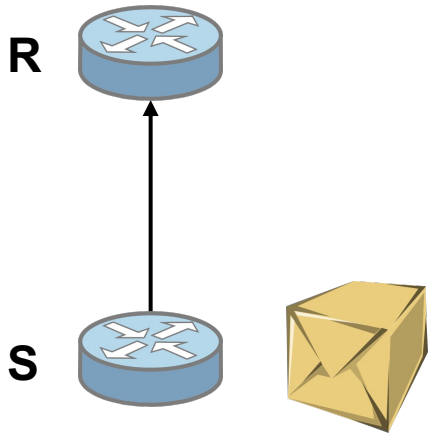
RFC 4944: 6LoWPAN Fragmentation & Reassembly

Institut Mines-Télécom

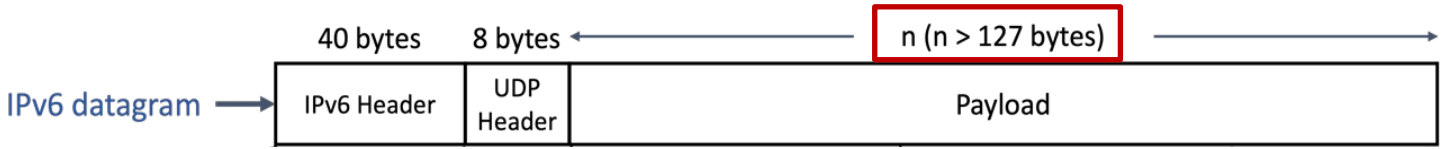
RFC 4944



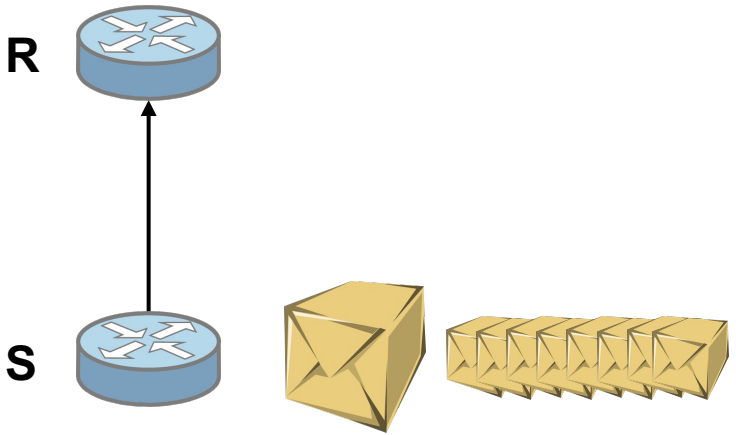
RFC 4944



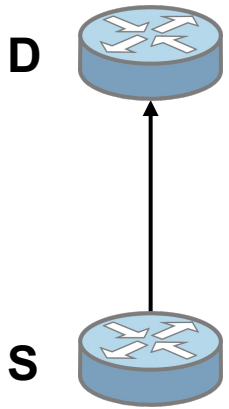
When an IPv6 packet is larger than IEEE 802.15.4 MTU (127 bytes)



RFC 4944



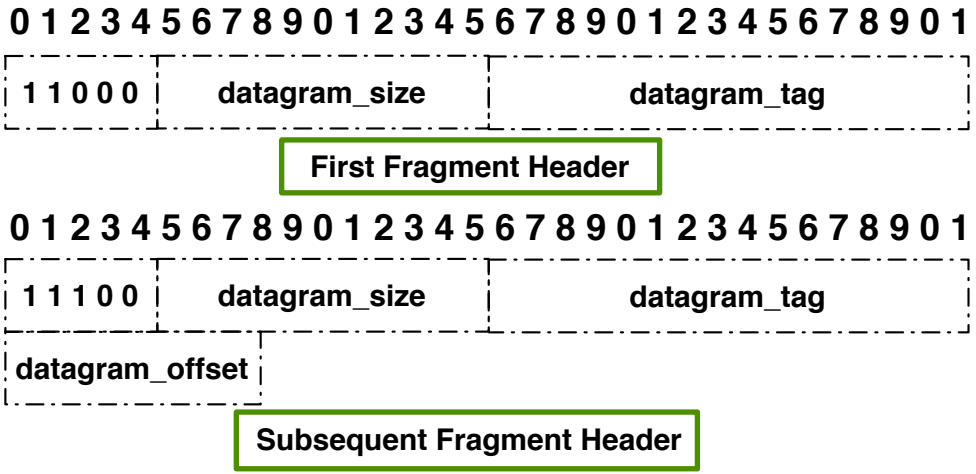
RFC 4944



= 127 bytes

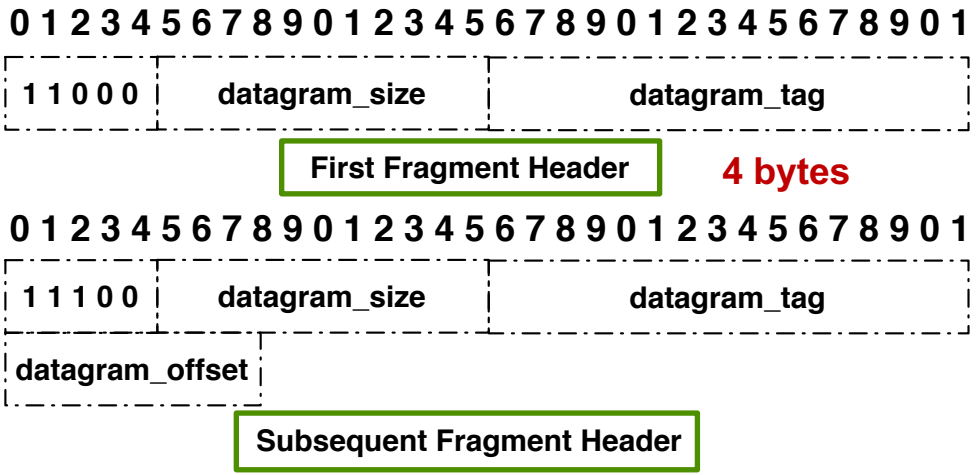
Fragmentation Operation

RFC 4944



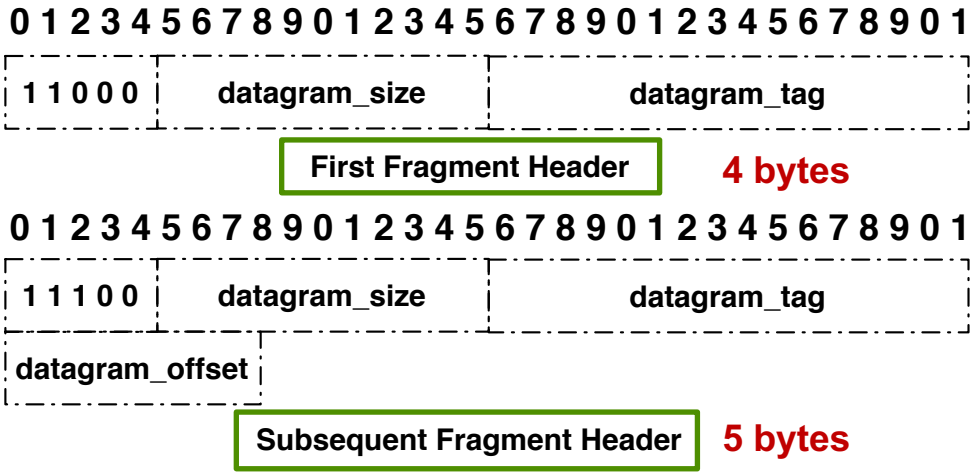
Fragmentation Operation

RFC 4944



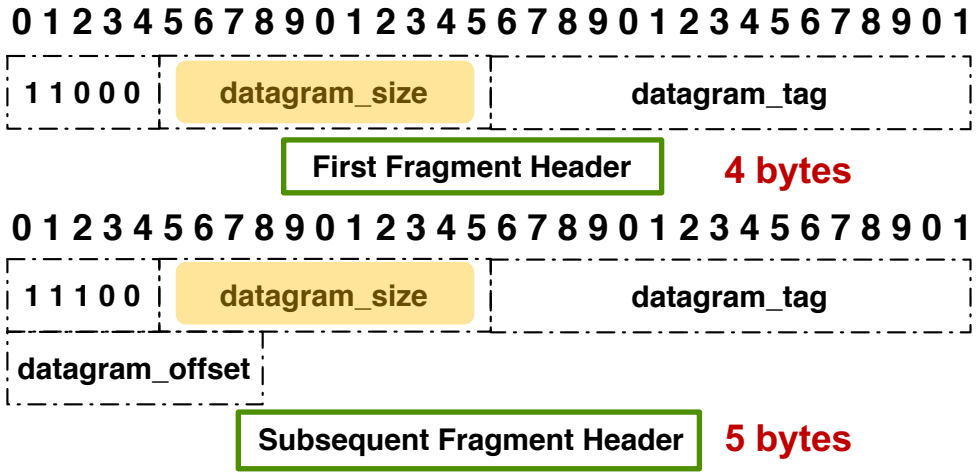
Fragmentation Operation

RFC 4944



Fragmentation Operation

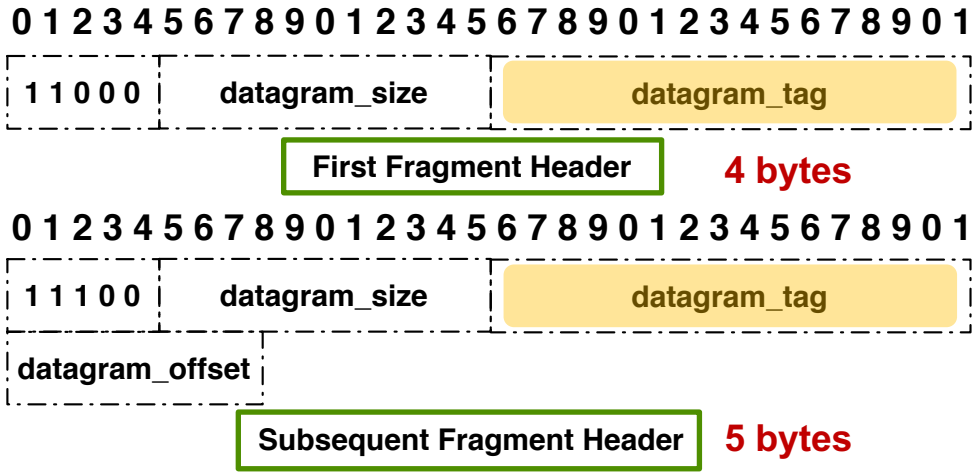
RFC 4944



1. The ***datagram_size*** to identify the size of the IPv6 datagram.

Fragmentation Operation

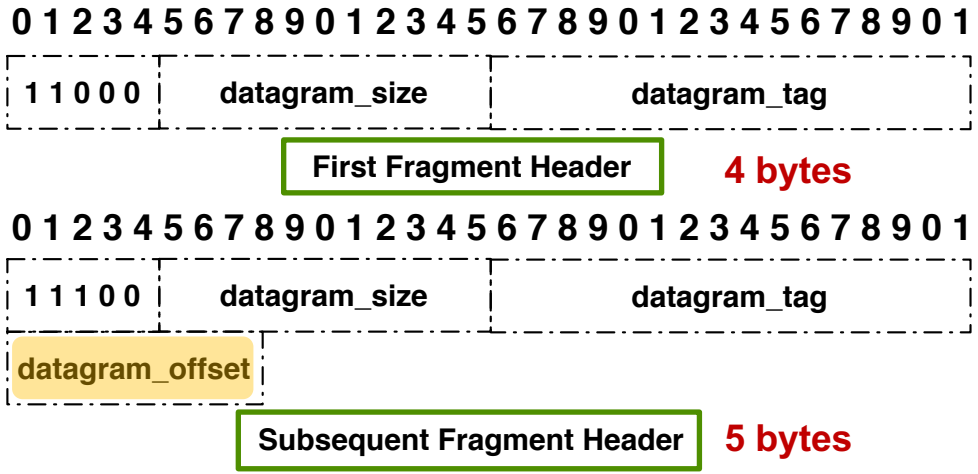
RFC 4944



2. The ***datagram_tag*** to identify all fragments of a single datagram.

Fragmentation Operation

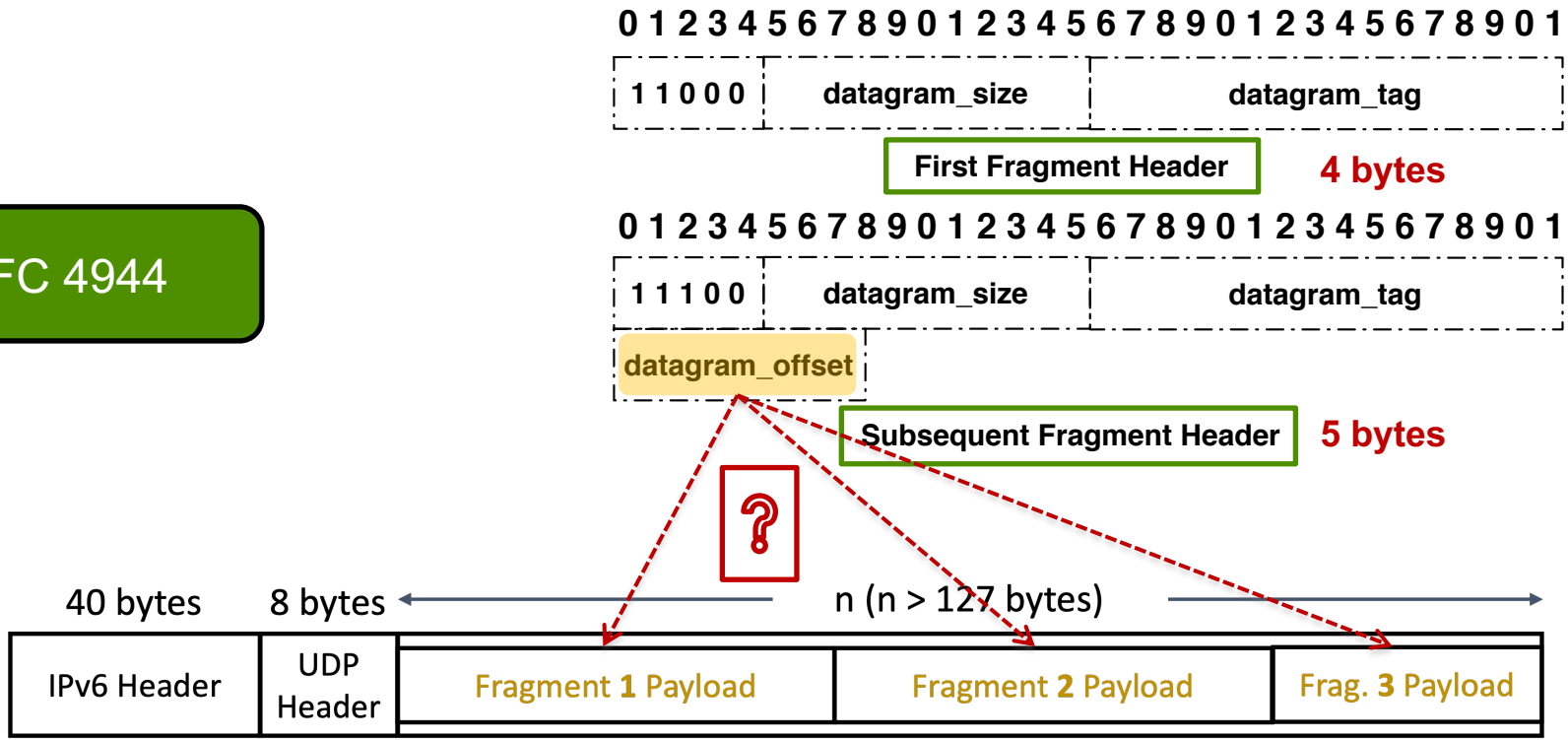
RFC 4944



3. The ***datagram_offset*** to identify the location of the received fragment.

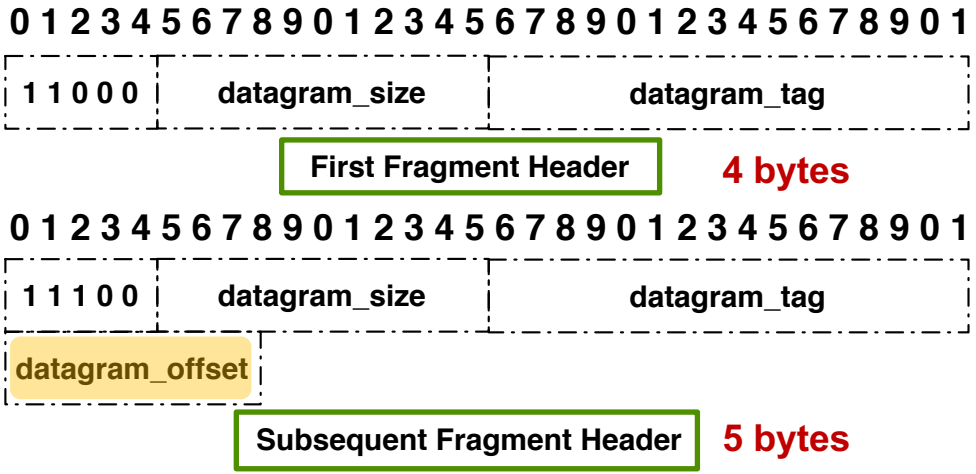
Fragmentation Operation

RFC 4944



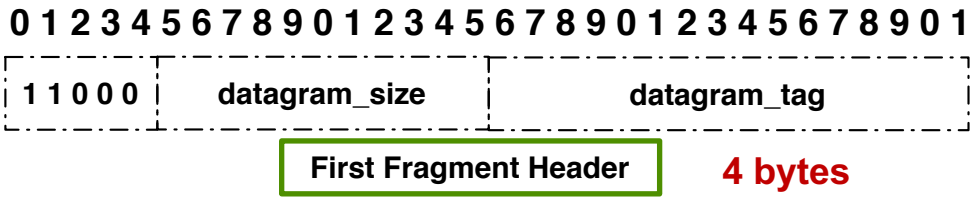
Fragmentation Operation

RFC 4944



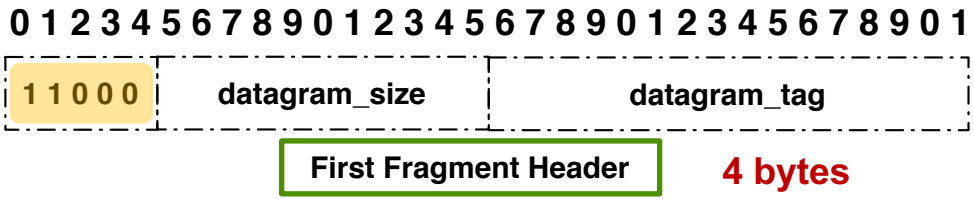
Fragmentation Operation

RFC 4944



Fragmentation Operation

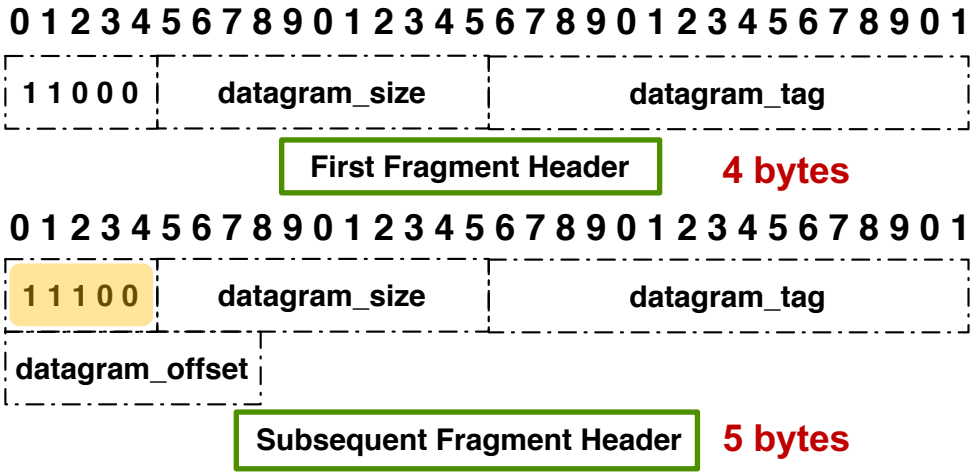
RFC 4944

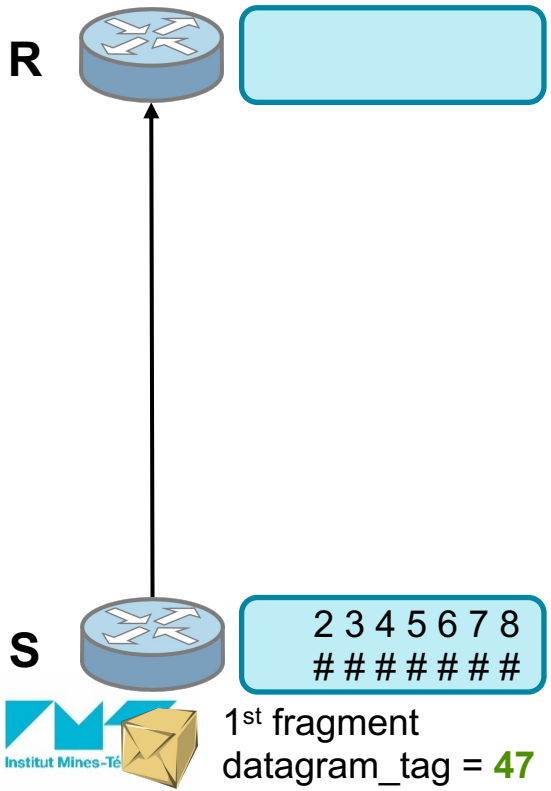


Dispatch Value Bit Pattern → the Fragmentation Type

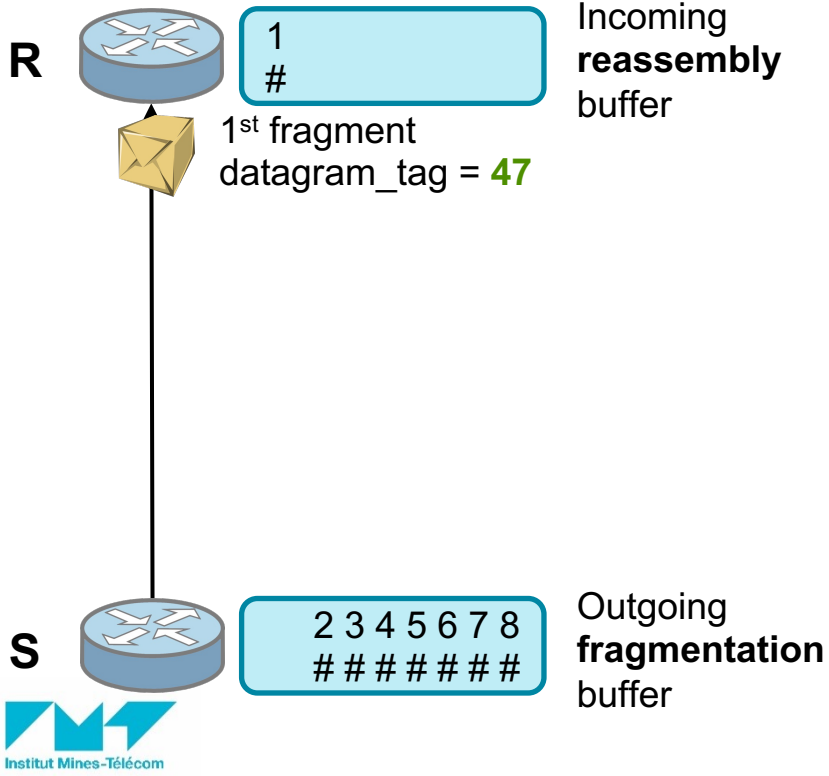
Fragmentation Operation

RFC 4944



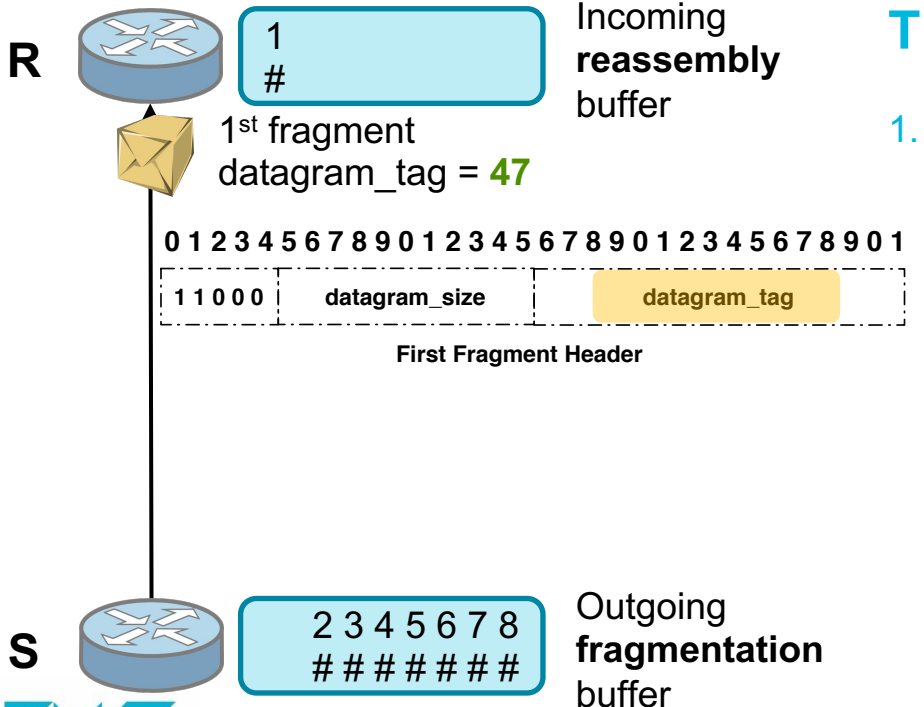


The receiving node R:



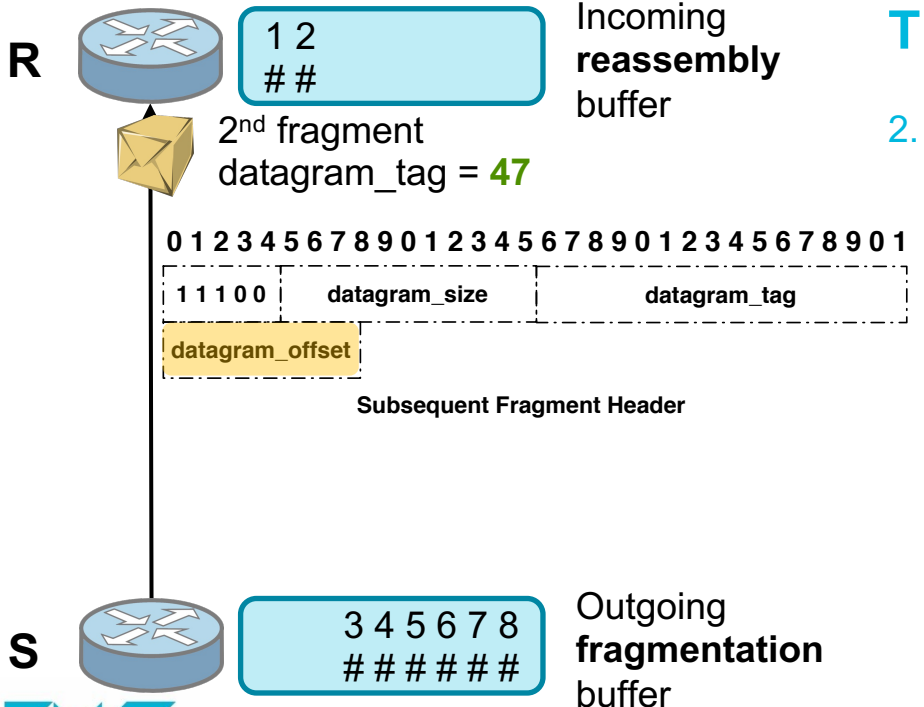
The receiving node R:

- Initiates the reassembly operation **to reconstruct the original IPv6 datagram**.



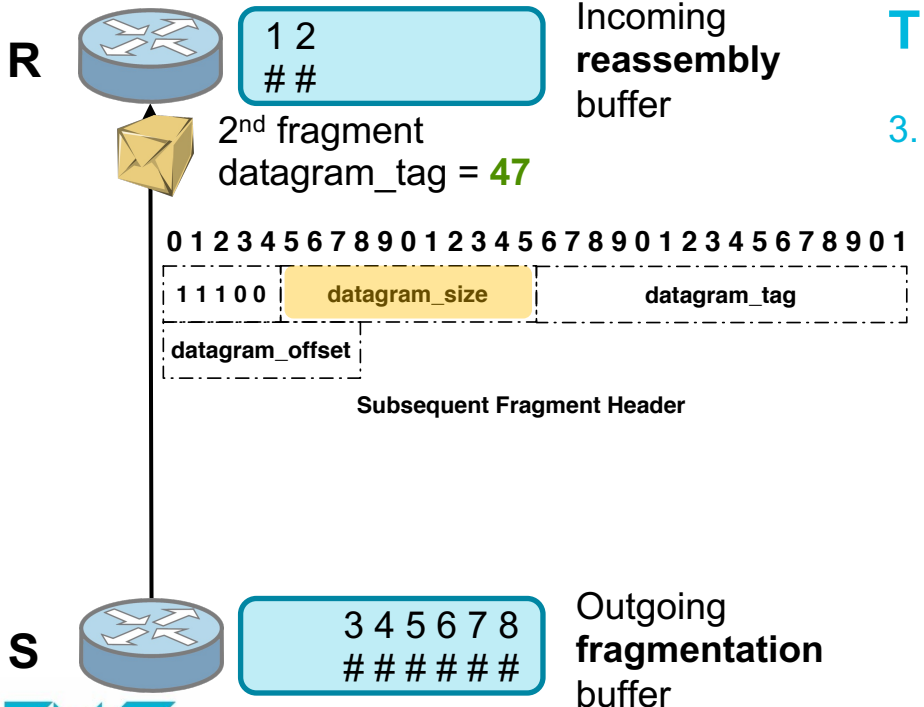
The receiving node R:

1. Checks the ***datagram_tag*** to identify the fragments that belong to a given IPv6 datagram.



The receiving node R:

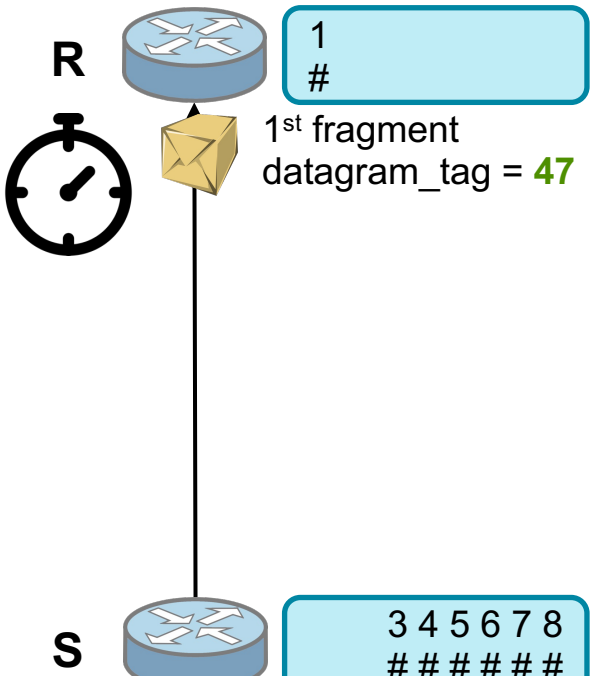
2. Checks the ***datagram_offset*** to determine the location of the received individual fragment.



The receiving node R:

3. Checks the ***datagram_size*** to identify the size of the original unfragmented datagram.

Institut Mines-Télécom

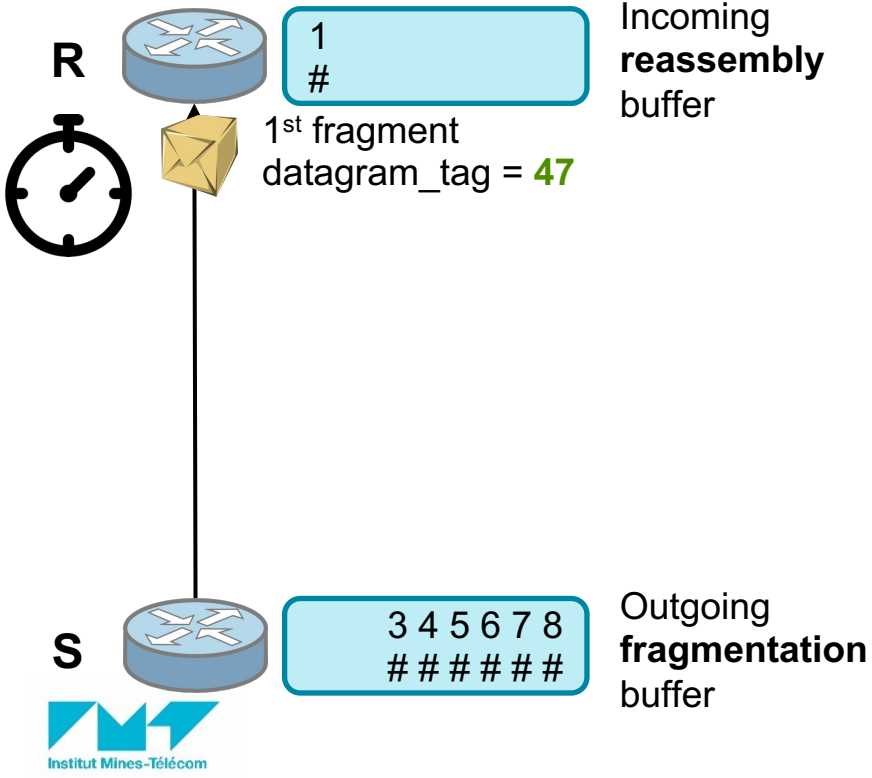


Incoming **reassembly** buffer

Outgoing **fragmentation** buffer

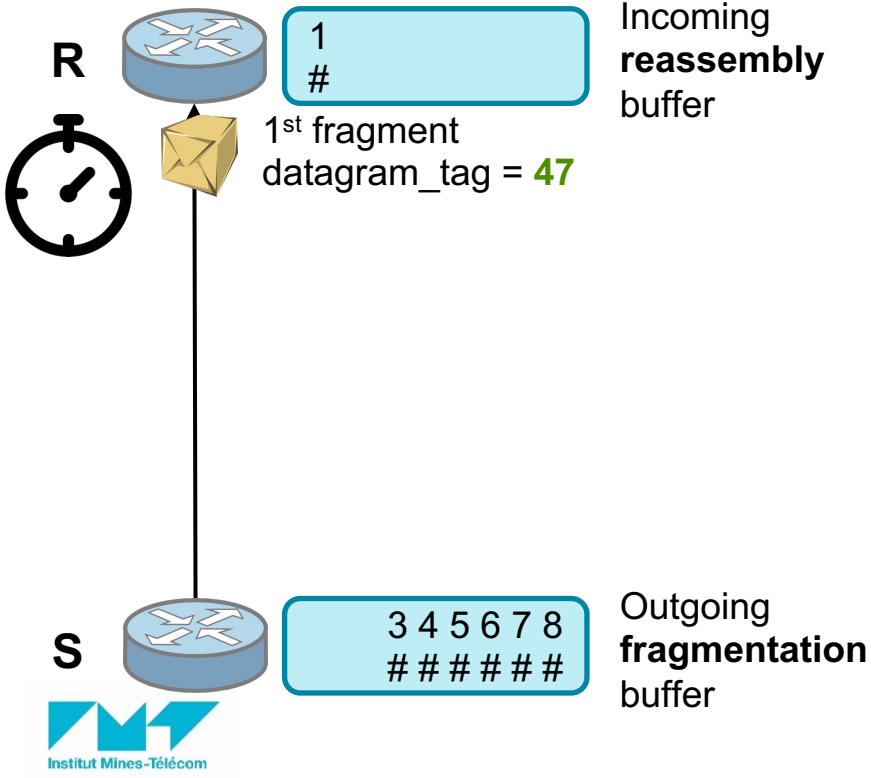
The receiving node R:

- ***Sets a reassembly timer***



The receiving node R:

- Sets a reassembly timer, ***which is configured to a maximum of 60 seconds***



The receiving node R:

- Sets a reassembly timer, which is configured to a maximum of 60 seconds ***to allow discarding the partially reassembled packet after the timeout***.

Chapter 2

RFC 4944: 6LoWPAN Frame Delivery Modes

RFC 4944

6LoWPAN Frame Delivery modes:

- Mesh-Under.
- Router-Over.

RFC 4944

6LoWPAN Frame Delivery modes:

- Mesh-Under: Layer 2 based on MAC.
- Router-Over.

RFC 4944

6LoWPAN Frame Delivery modes:

- Mesh-Under: Layer 2 based on MAC.
- Router-Over: Layer 3 based on IP.

RFC 4944

Mesh-Under:

- The ***routing*** and ***forwarding*** operations are performed at 6LoWPAN Adaptation layer.

RFC 4944

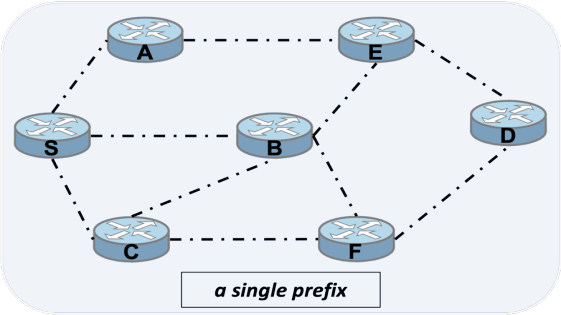
Mesh-Under:

- The ***routing*** and ***forwarding*** operations are performed at 6LoWPAN Adaptation layer.
- The IPv6 header ***does not*** need to be unpacked.

RFC 4944

Mesh-Under:

- The ***routing*** and ***forwarding*** operations are performed at 6LoWPAN Adaptation layer.
- The IPv6 header ***does not*** need to be unpacked.
- All nodes share the same prefix.



RFC 4944

To forward the received fragments, a node requires:

- The ***final destination address***, the link-layer address of the Final Destination.

RFC 4944

To forward the received fragments, a node requires:

- The ***final destination address***, the link-layer address of the Final Destination.
- The ***originator address***, the link-layer address of the Originator.

RFC 4944

To forward the received fragments, a node requires:

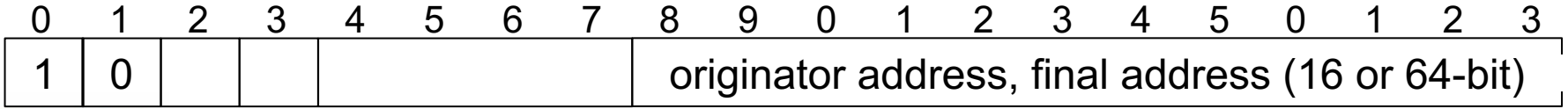
- The ***final destination address***, the link-layer address of the Final Destination.
- The ***originator address***, the link-layer address of the Originator.

originator address, final address

RFC 4944

To forward the received fragments, a node requires:

- The ***final destination address***, the link-layer address of the Final Destination.
- The ***originator address***, the link-layer address of the Originator.

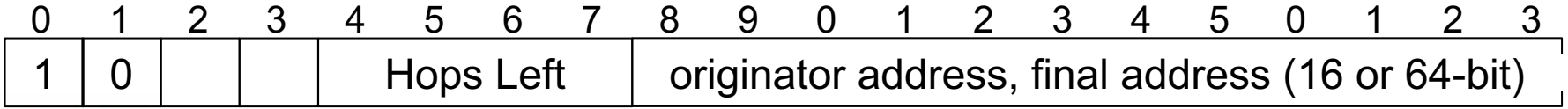


Mesh Addressing Type & Header: It is defined by 1 and 0 as the first two bits.

RFC 4944

To forward the received fragments, a node requires:

- The ***final destination address***, the link-layer address of the Final Destination.
- The ***originator address***, the link-layer address of the Originator.
- ***Hops Left*** (4 bits), a layer-2 equivalent of an IPv6 Hop Limit.

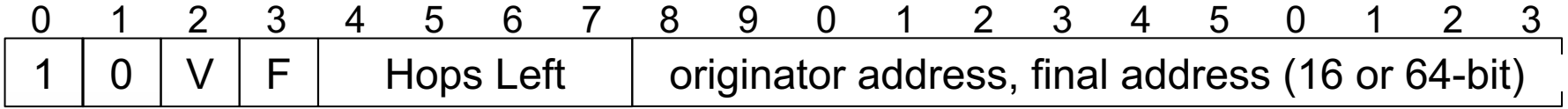


Mesh Addressing Type & Header: It is defined by 1 and 0 as the first two bits.

RFC 4944

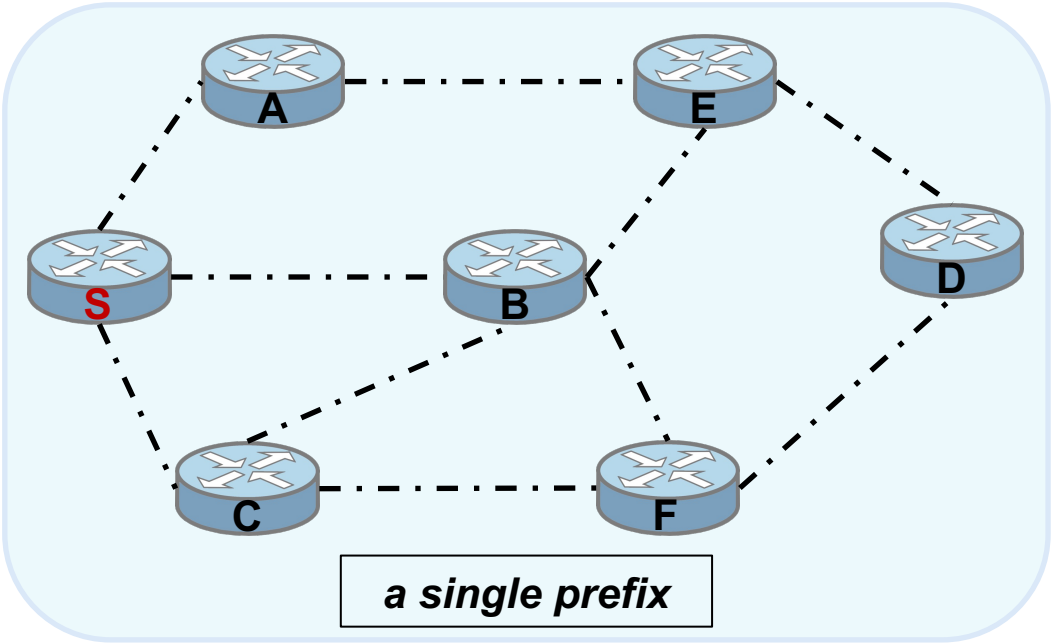
To forward the received fragments, a node requires:

- The ***final destination address***, the link-layer address of the Final Destination.
- The ***originator address***, the link-layer address of the Originator.
- ***Hops Left*** (4 bits), a layer-2 equivalent of an IPv6 Hop Limit.
- ***V*** and ***F bits*** to indicate whether the addresses are 16-bit short or 64-bit EUI-64.

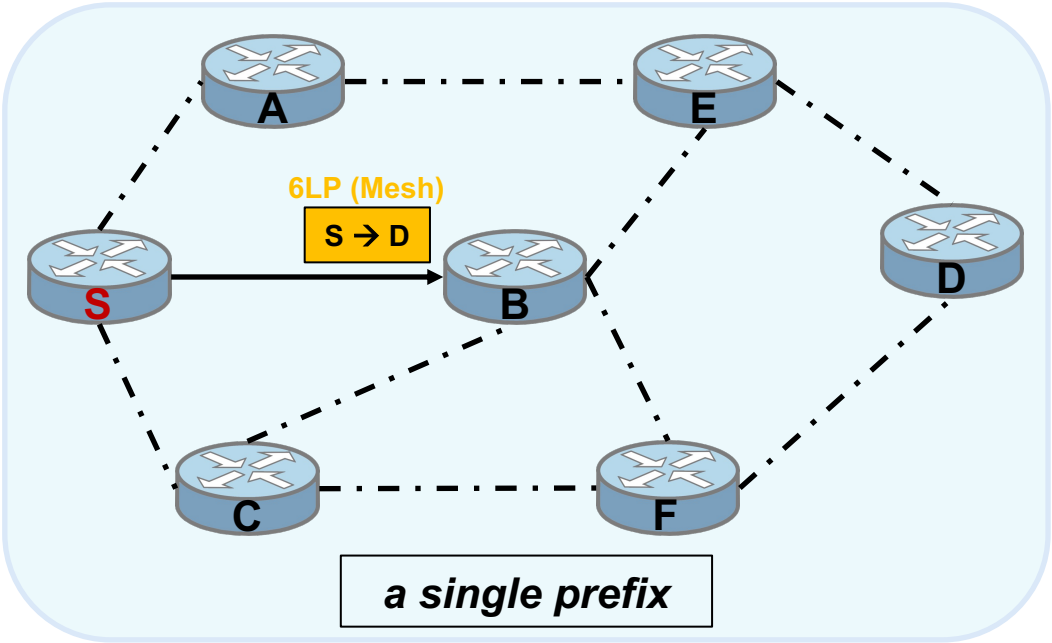


Mesh Addressing Type & Header: It is defined by 1 and 0 as the first two bits.

Mesh-Under: Operation Example

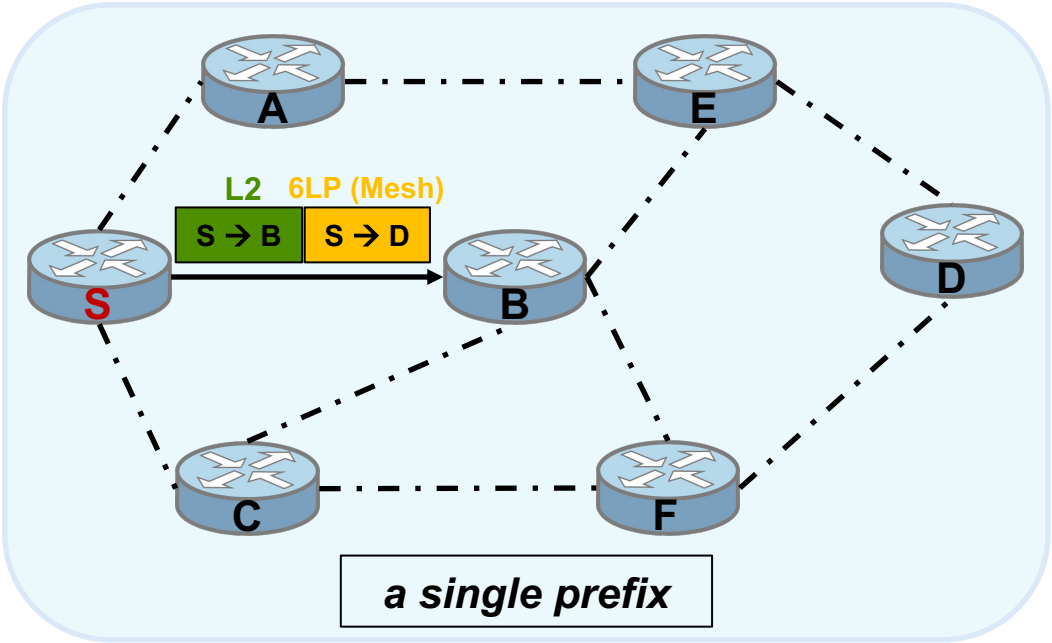


Mesh-Under: Operation Example



6LP: **6LoWPAN Adaptation Layer** (i.e., Mesh Addressing Type & Header)

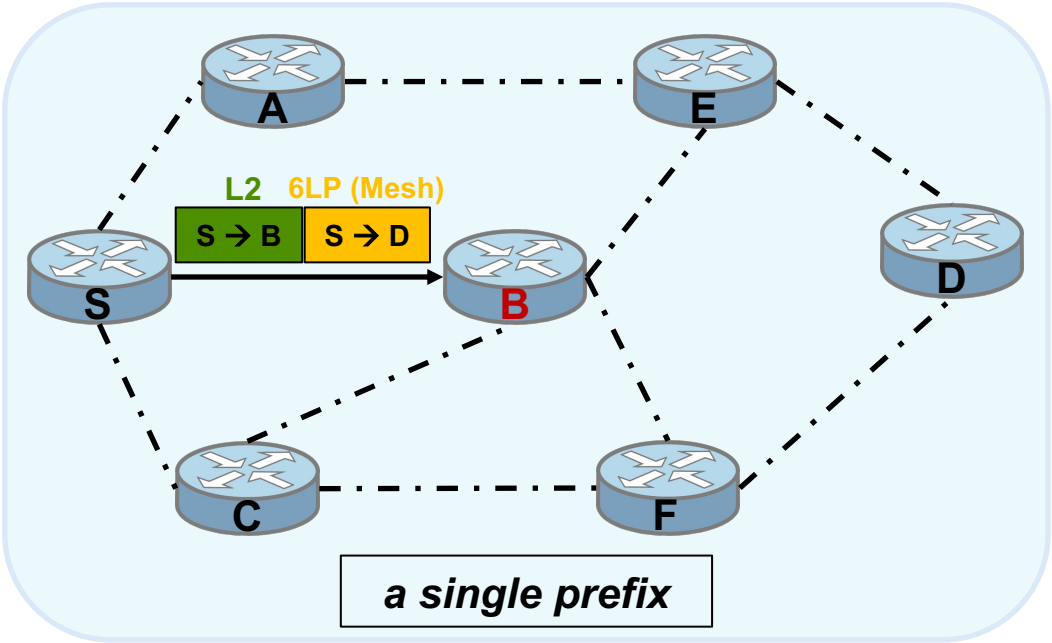
Mesh-Under: Operation Example



6LP: **6LoWPAN Adaptation Layer** (i.e., Mesh Addressing Type & Header)

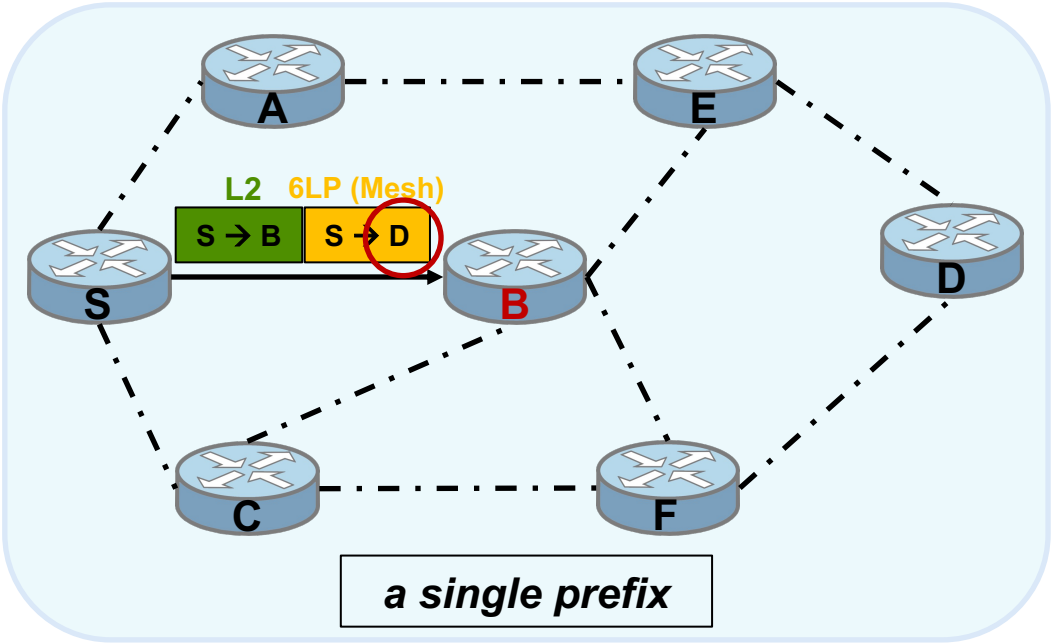
L2: **Layer 2** (e.g., IEEE Std 802.15.4)

Mesh-Under: Operation Example



6LP: **6LoWPAN Adaptation Layer** (i.e., Mesh Addressing Type & Header)

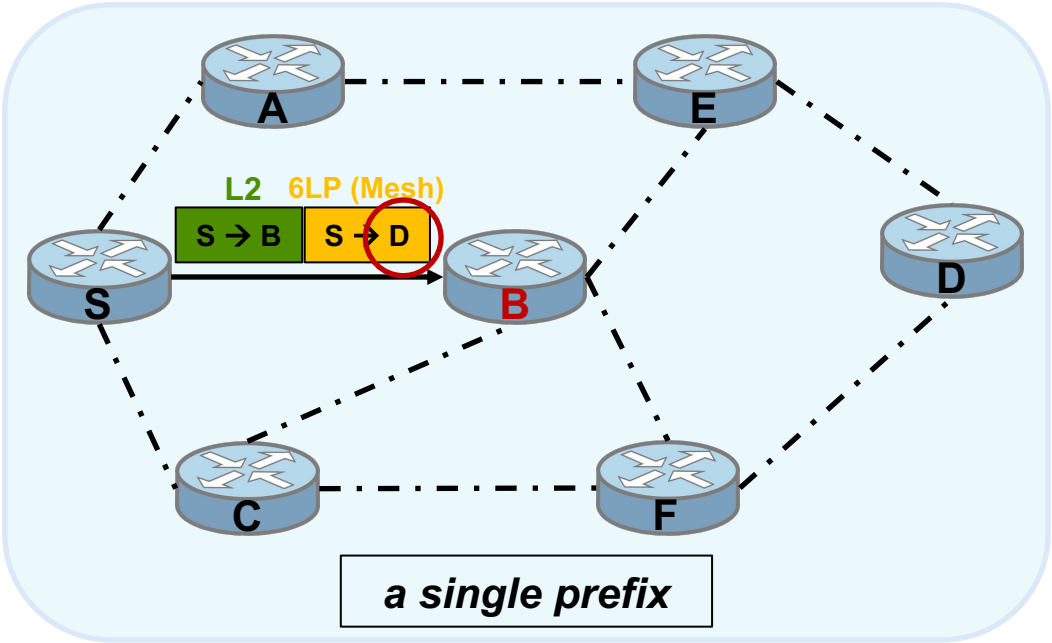
L2: **Layer 2** (e.g., IEEE Std 802.15.4)



Institut Mines-Télécom

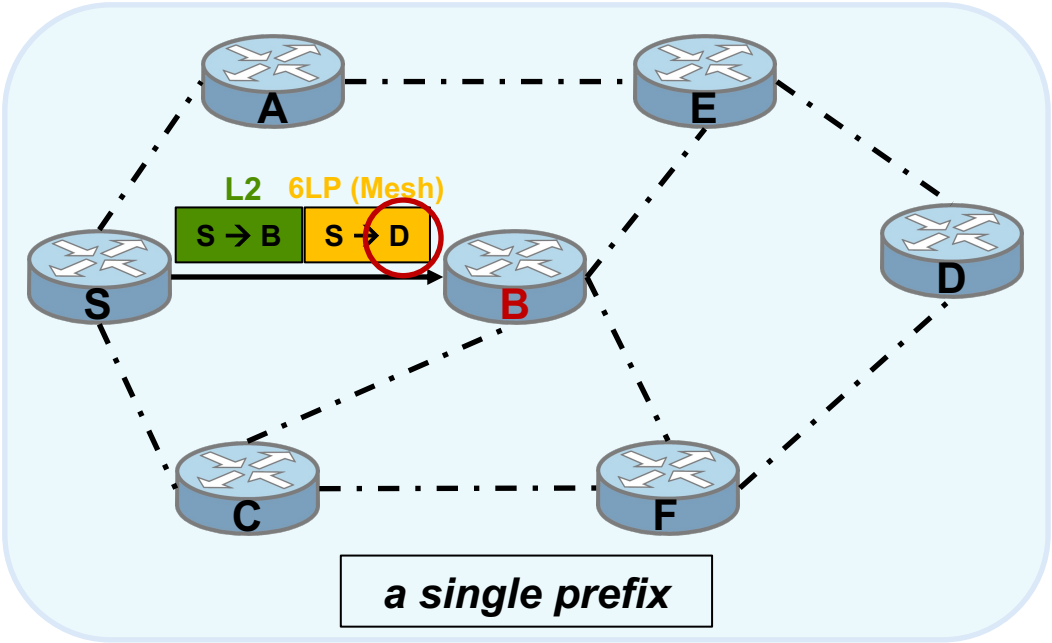
6LP: 6LoWPAN Adaptation Layer (i.e., Mesh Addressing Type & Header)

L2: Layer 2 (e.g., IEEE Std 802.15.4)



1. If the node ***is*** the final destination, → it initiates the reassembly of the IPv6 packet.

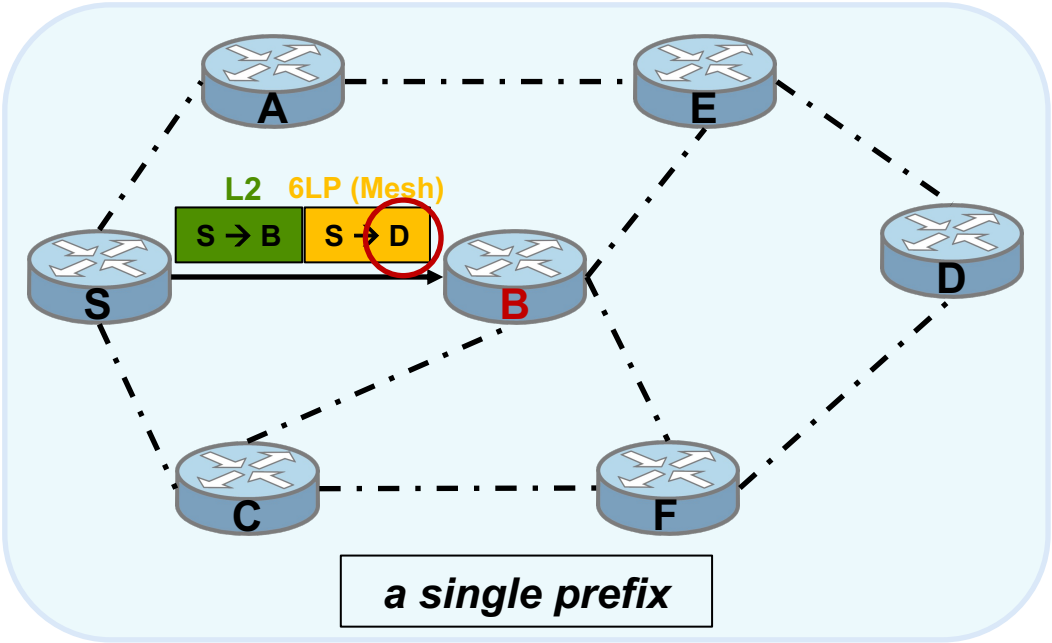
6LP: **6LoWPAN Adaptation Layer** (i.e., Mesh Addressing Type & Header)
L2: **Layer 2** (e.g., IEEE Std 802.15.4)



2. If it ***is not***, →
   it reduces the "Hops Left",

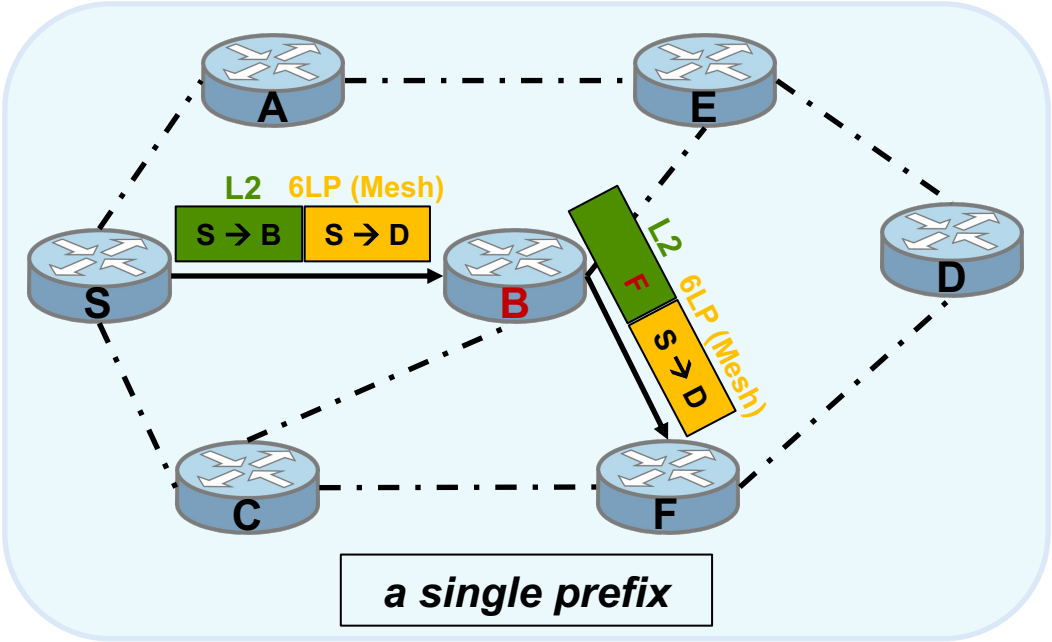
6LP: **6LoWPAN Adaptation Layer** (i.e., Mesh Addressing Type & Header)
L2: **Layer 2** (e.g., IEEE Std 802.15.4)

Mesh-Under: Operation Example



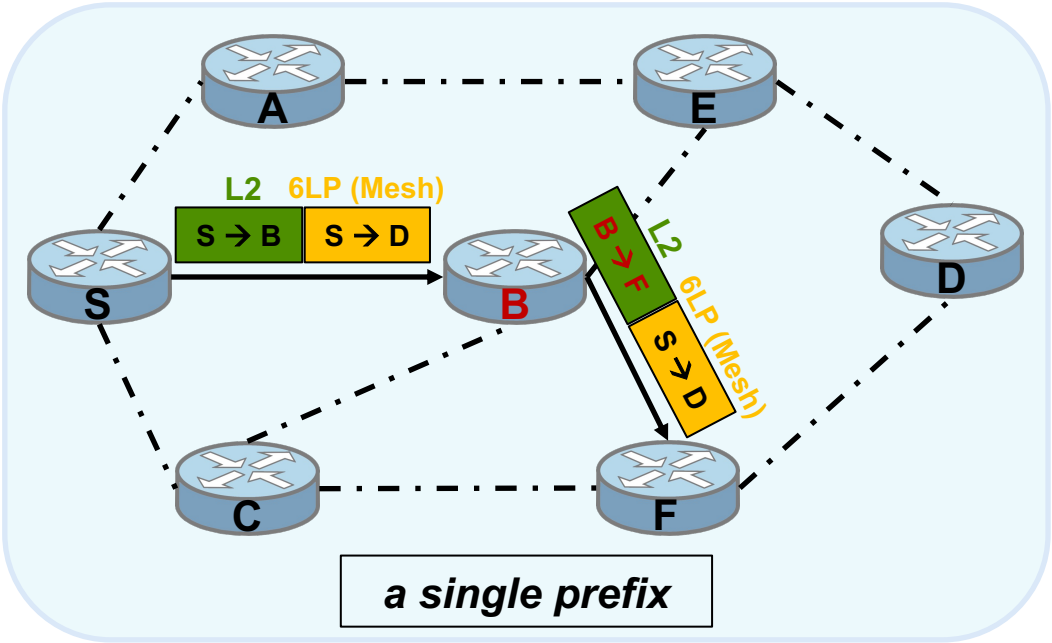
2. If it ***is not***, →
   it reduces the "Hops Left",
   and if it ***decremented to zero***, →
   it **discards** the frame.

6LP: **6LoWPAN Adaptation Layer** (i.e., Mesh Addressing Type & Header)
L2: **Layer 2** (e.g., IEEE Std 802.15.4)



3. Otherwise,
   - it sets **F** in the destination address field **of the L2 header**.

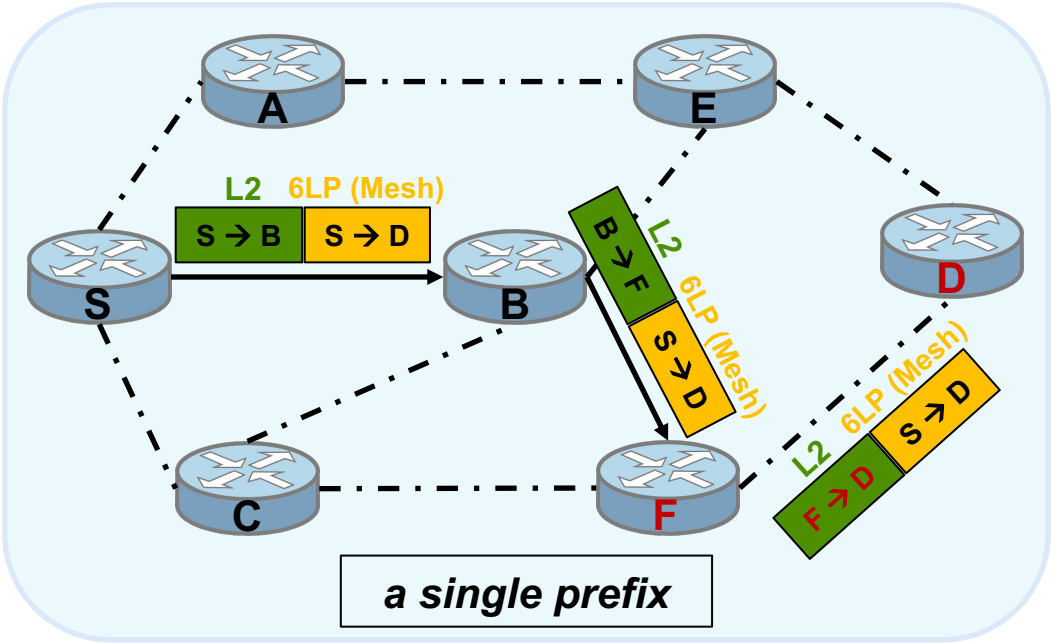
6LP: **6LoWPAN Adaptation Layer** (i.e., Mesh Addressing Type & Header)
L2: **Layer 2** (e.g., IEEE Std 802.15.4)



3. Otherwise,
   - it sets **F** in the destination address field **of the L2 header**.
   - Finally, it sets the **L2 source address to its own** and transmits the frame.

6LP: 6LoWPAN Adaptation Layer (i.e., Mesh Addressing Type & Header)
L2: Layer 2 (e.g., IEEE Std 802.15.4)

Mesh-Under: Operation Example



6LP: **6LoWPAN Adaptation Layer** (i.e., Mesh Addressing Type & Header)

L2: **Layer 2** (e.g., IEEE Std 802.15.4)

Institut Mines-Télécom

RFC 4944

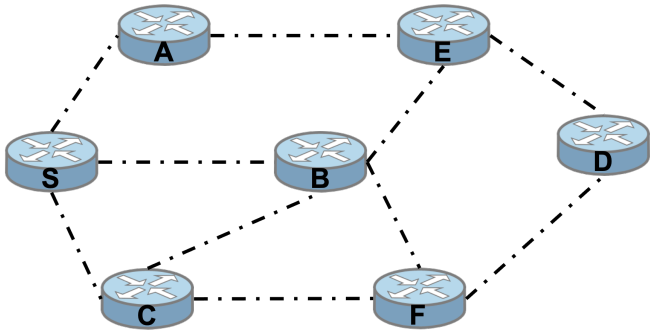
Route-Over:

- The ***routing*** and ***forwarding*** operations are performed at Layer 3 based on IP.

RFC 4944

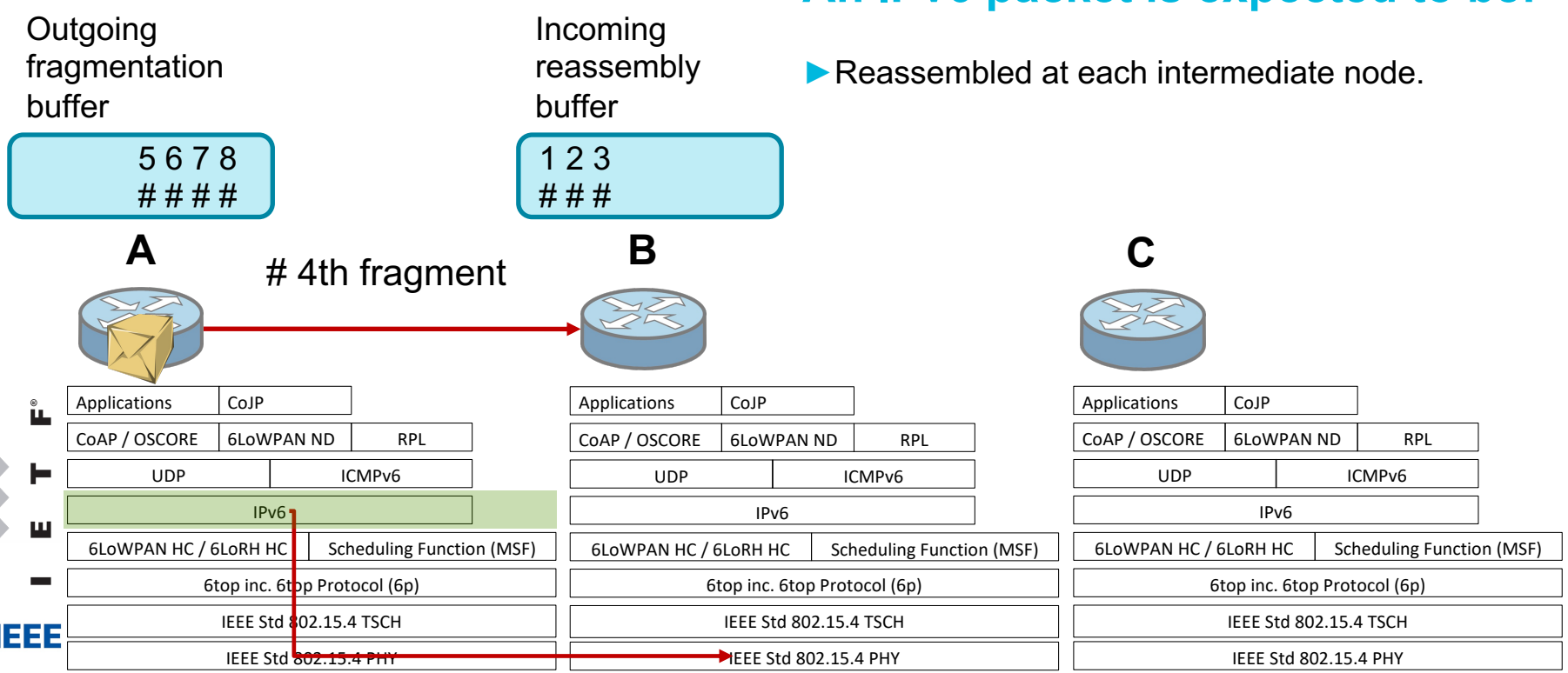
Route-Over:

- The ***routing*** and ***forwarding*** operations are performed at Layer 3 based on IP.
- Each link layer hop is an IP hop.



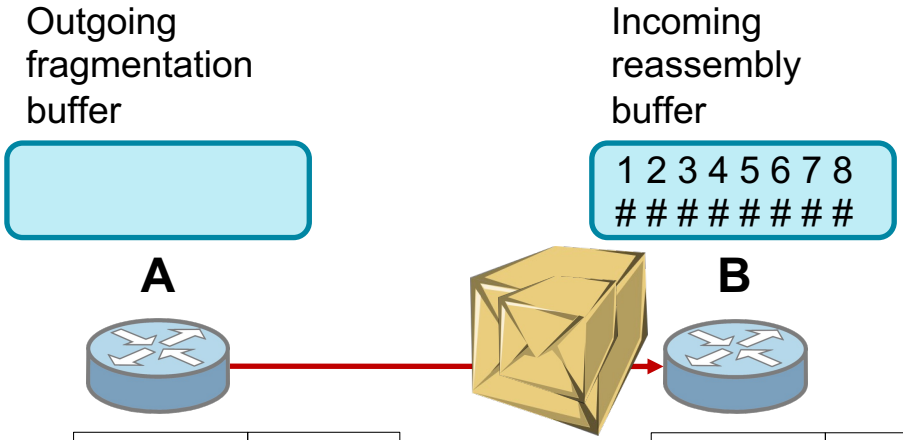
An IPv6 packet is expected to be:

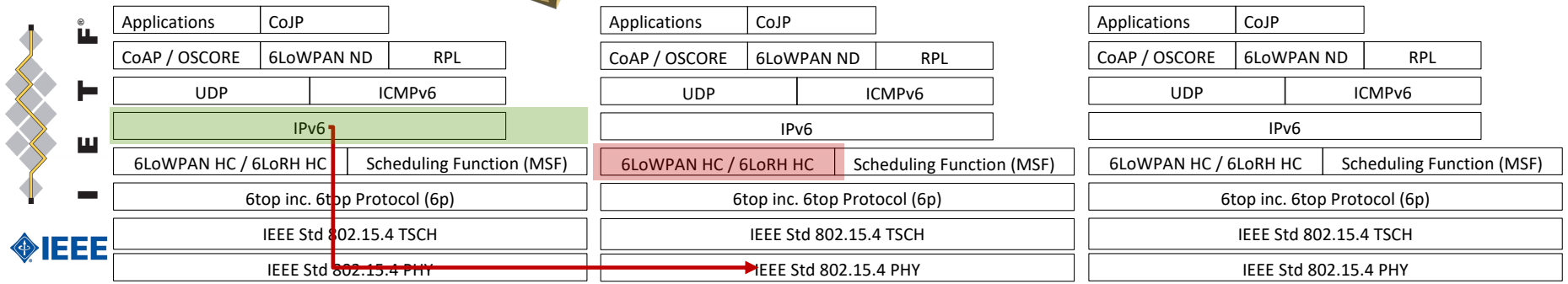
- Reassembled at each intermediate node.



An IPv6 packet is expected to be:

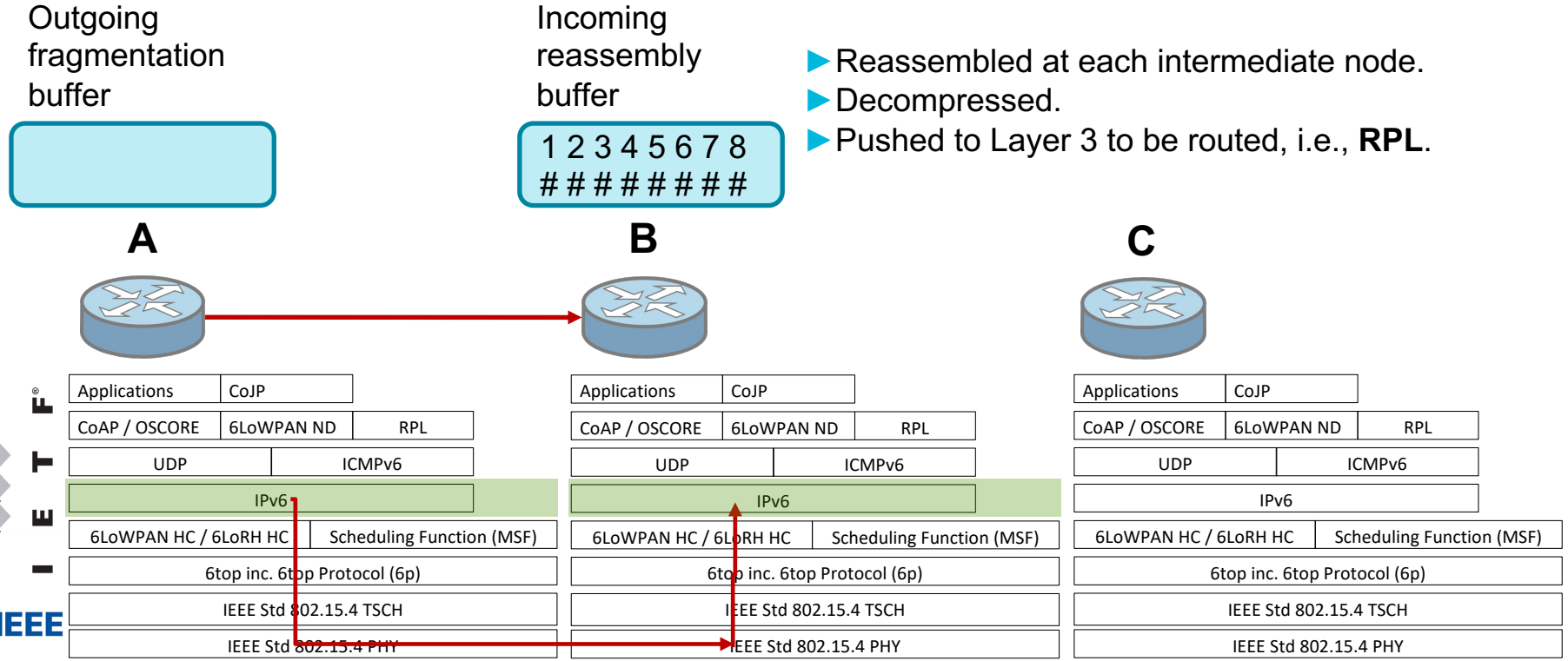
- Reassembled at each intermediate node.
- Decompressed.





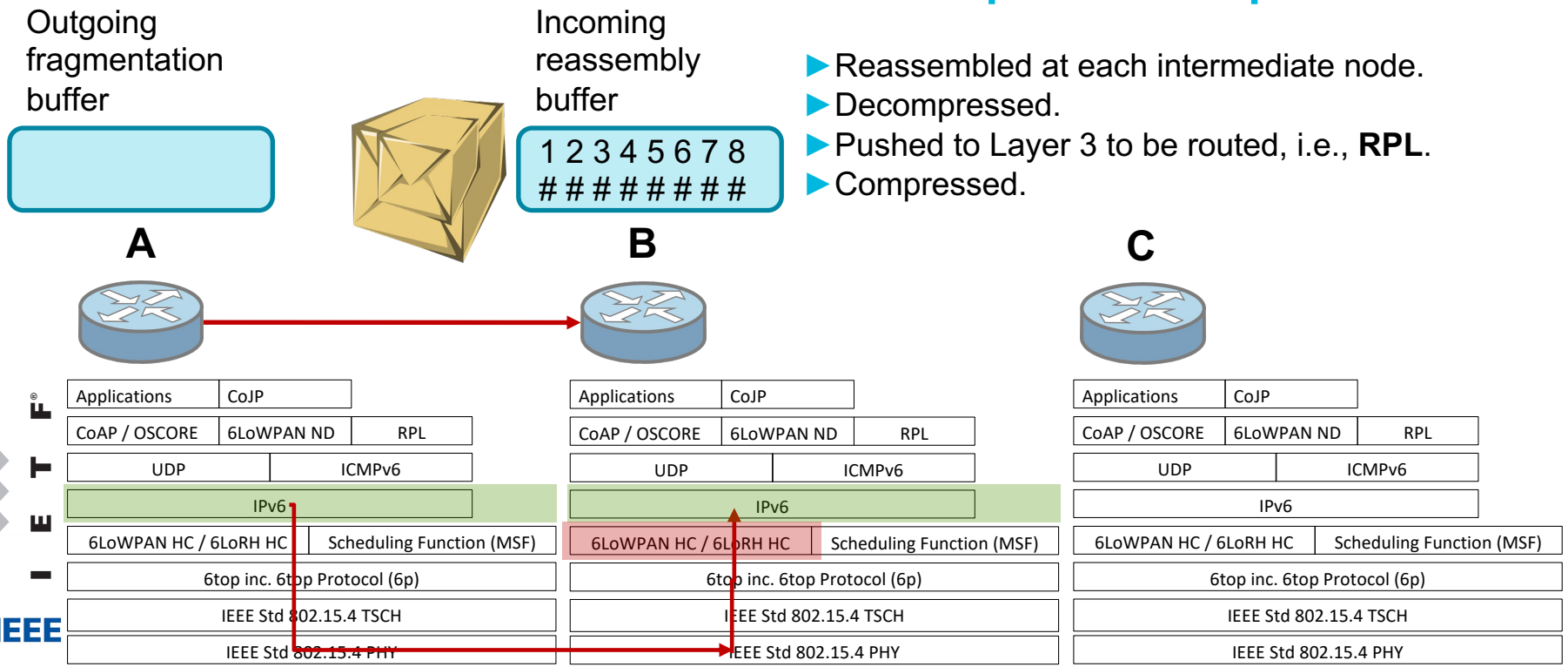
An IPv6 packet is expected to be:

- Reassembled at each intermediate node.
- Decompressed.
- Pushed to Layer 3 to be routed, i.e., **RPL**.



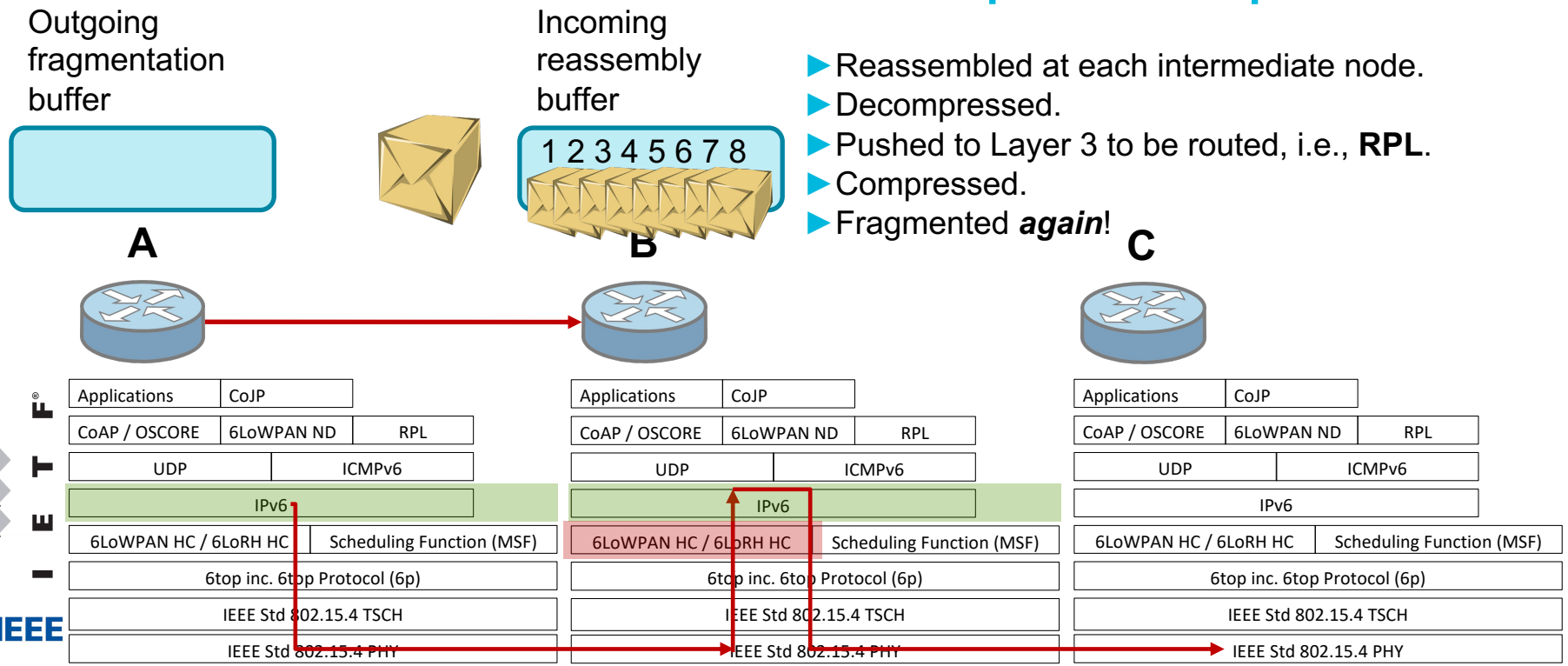
An IPv6 packet is expected to be:

- Reassembled at each intermediate node.
- Decompressed.
- Pushed to Layer 3 to be routed, i.e., **RPL**.
- Compressed.

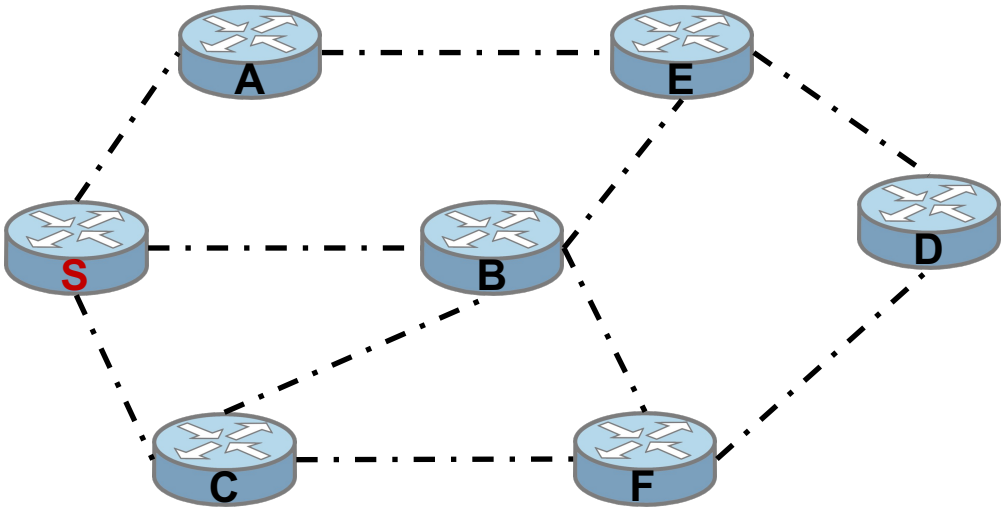


An IPv6 packet is expected to be:

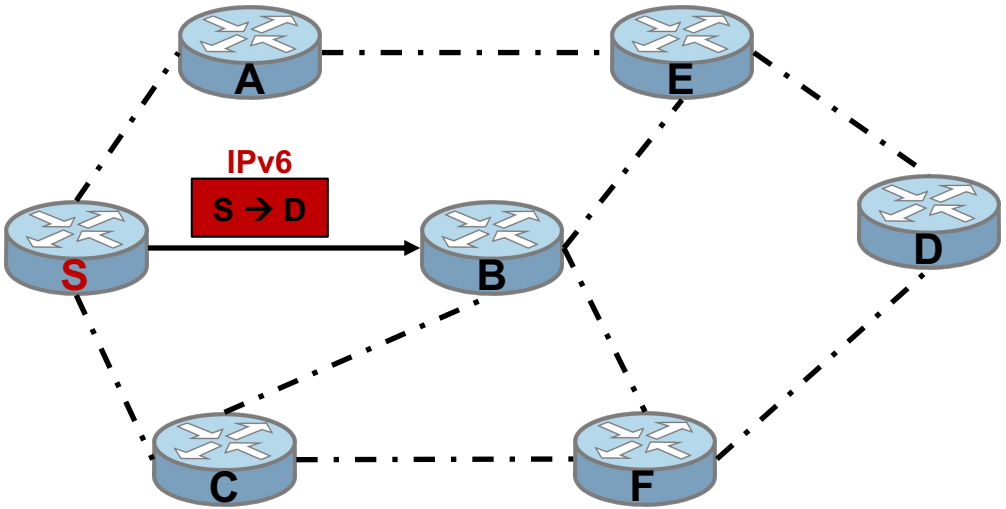
- Reassembled at each intermediate node.
- Decompressed.
- Pushed to Layer 3 to be routed, i.e., **RPL**.
- Compressed.
- Fragmented ***again***!



Route-Over: Operation Example

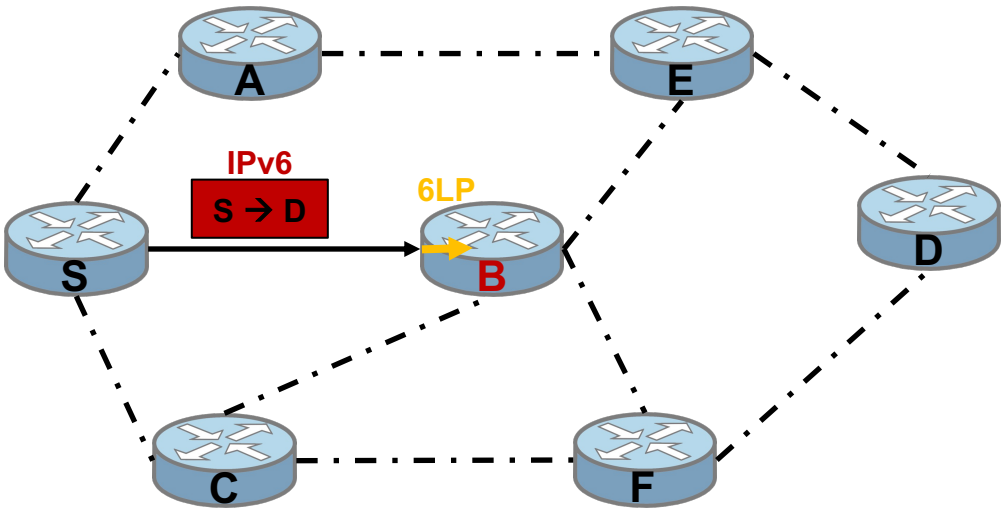


Route-Over: Operation Example



IPv6: IPv6 Layer

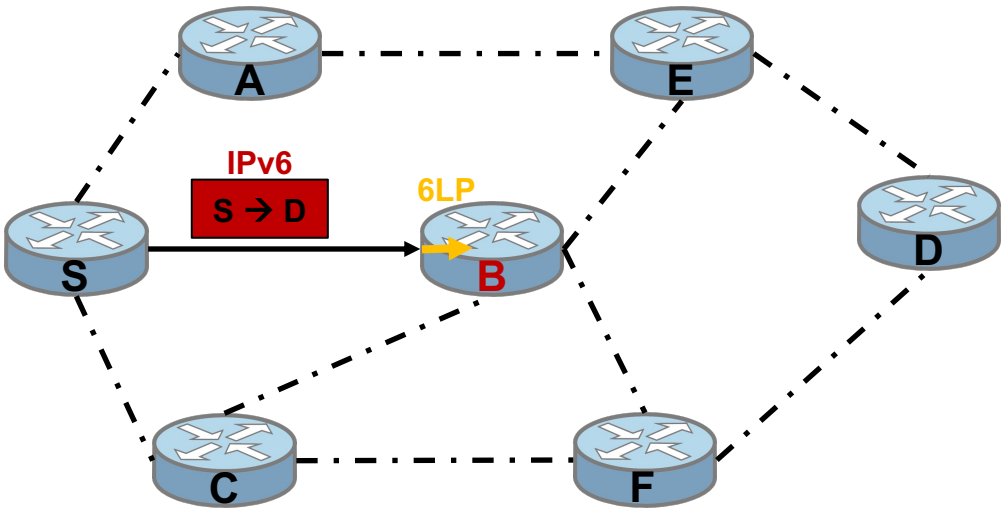
Route-Over: Operation Example



1. Upon the reception of the 1st fragment:
   - node B stores the received fragments.

IPv6: IPv6 Layer
6LP: 6LoWPAN Adaptation Layer

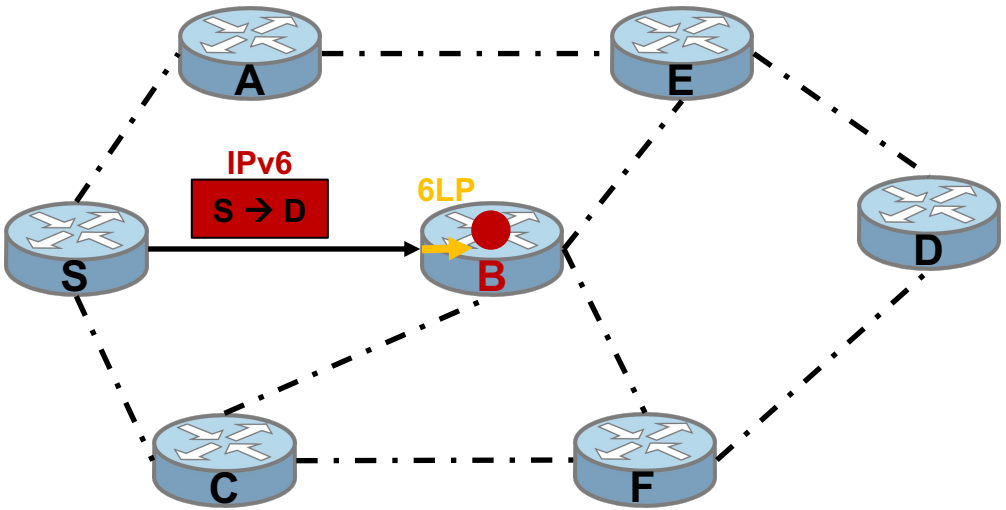
Route-Over: Operation Example



1. Upon the reception of the 1st fragment:
   - node B stores the received fragments.
   - then, it initiates the Reassembly and Decompression operations.

IPv6: IPv6 Layer
6LP: 6LoWPAN Adaptation Layer

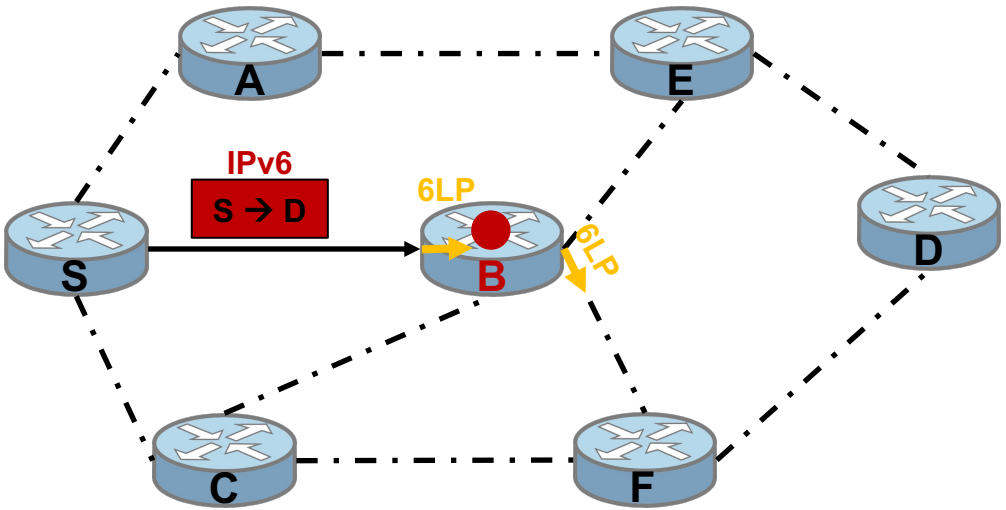
Route-Over: Operation Example



2. Then, it checks the IPv6 original source and final destination headers.

IPv6: IPv6 Layer
6LP: 6LoWPAN Adaptation Layer

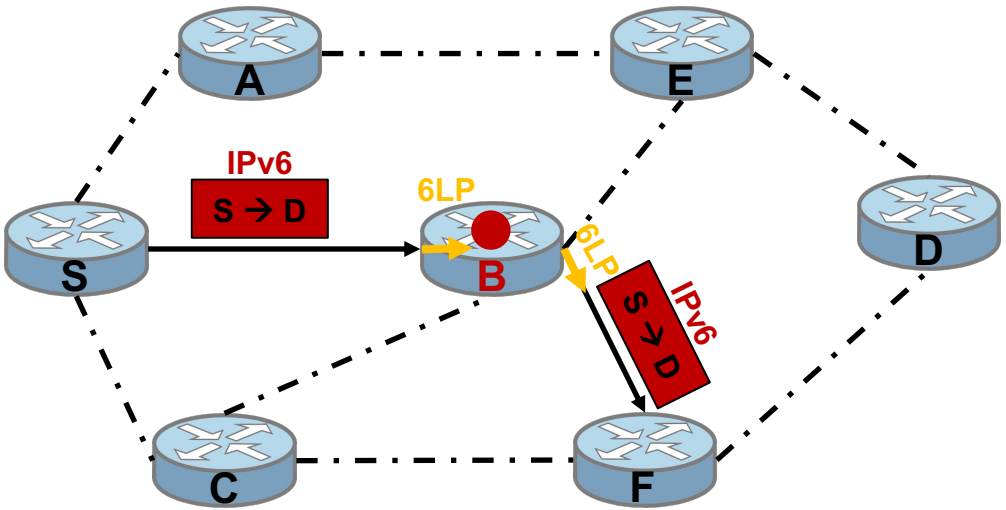
Route-Over: Operation Example



3. If node ***B is not*** the ***final destination***, it pushes to the 6LoWPAN layer:
   - for Compression and Fragmentation operations.

IPv6: IPv6 Layer
6LP: 6LoWPAN Adaptation Layer

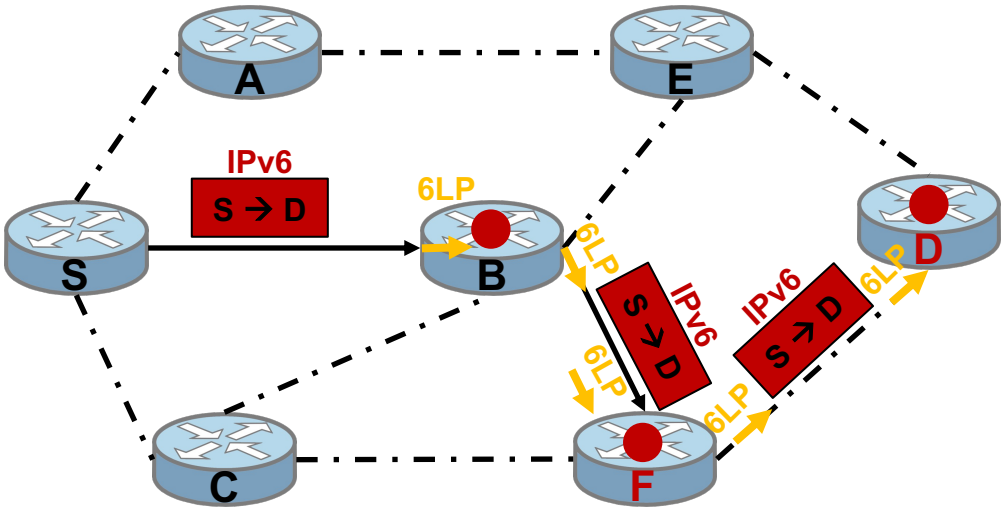
Route-Over: Operation Example



4. Based on its L3 routing table:
   - it will forward the packet to node F.

IPv6: IPv6 Layer
6LP: 6LoWPAN Adaptation Layer

Route-Over: Operation Example



Institut Mines-Télécom

IPv6: IPv6 Layer

6LP: 6LoWPAN Adaptation Layer

Chapter 3
Route Over (Per-hop Fragmentation & Reassembly): Issues

Institut Mines-Télécom

Route-Over: Issues

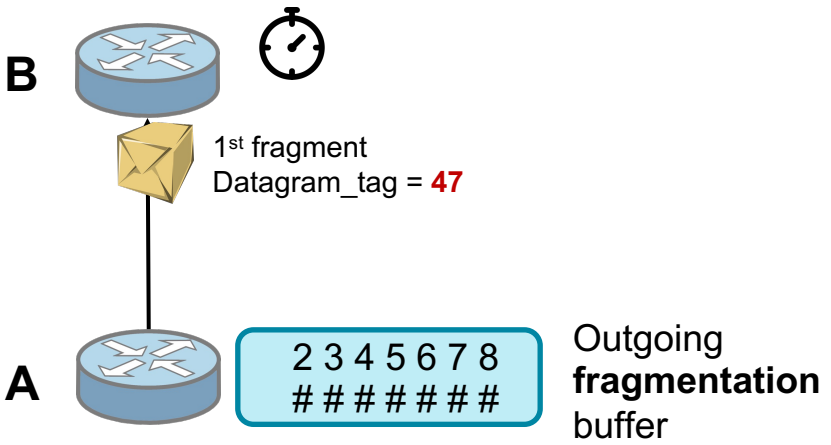
For more details, please check the following **articles**:

1. *G. Z. Papadopoulos, R. Jadhav, P. Thubert and N. Montavont, "Updates on RFC 4944: Fragment Forwarding and Recovery," In IEEE Communications Standards Magazine, vol. 3, pp. 54-59, June 2019.*
2. *G. Z. Papadopoulos, P. Thubert, S. Tsakalidis and N. Montavont, "RFC 4944: Per-hop Fragmentation and Reassembly Issues" In Proc. IEEE CSCN 2018 - Paris, France, October 2018*

RFC 4944

Reliability:

- When a node receives the 1st fragment, it initiates the ***reassembly timer*** (i.e., 60 seconds).



RFC 4944

Reliability:

- When a node receives the 1st fragment, it initiates the ***reassembly timer*** (i.e., 60 seconds).
- If **all fragments are not received** during this *reassembly period*:
  - **the reassembly** of the IPv6 packet **is not possible**.

RFC 4944

Latency:

- The reassembly at each hop increases the delay:
  - *each device is required to "wait" for 60 seconds*.

	Source	Forwarder 1	Forwarder 2	Destination
T = 0	F F F			
T = 1	F F (F)	F		
T = 2	F (F)	F F		
T = 3	(F)	F F F		
T = 4		F F (F)	F	
T = 5		F (F)	F F	
T = 6		(F)	F F F	
T = 7			F F (F)	F
T = 8			F (F)	F F
T = 9			(F)	F F F

RFC 4944

Latency:

- The reassembly at each hop increases the delay:
  - *each device is required to "wait" for 60 seconds*.
- There is an additional computation time for the fragmentation operation.

	Source	Forwarder 1	Forwarder 2	Destination
T = 0	F F F			
T = 1	F F (F)	F		
T = 2	F (F)	F F		
T = 3	(F)	F F F		
T = 4		F F (F)	F	
T = 5		F (F)	F F	
T = 6		(F)	F F F	
T = 7			F F (F)	F
T = 8			F (F)	F F
T = 9			(F)	F F F

RFC 4944

Latency:

- The reassembly at each hop increases the delay:
  - *each device is required to "wait" for 60 seconds*.
- There is an additional computation time for the fragmentation operation.
- Thus, **the more hops** in the path toward the destination is the **higher** the end-to-end **delay**.

	Source	Forwarder 1	Forwarder 2	Destination
T = 0	F F F			
T = 1	F F (F)	F		
T = 2	F (F)	F F		
T = 3	(F)	F F F		
T = 4		F F (F)	F	
T = 5		F (F)	F F	
T = 6		(F)	F F F	
T = 7			F F (F)	F
T = 8			F (F)	F F
T = 9			(F)	F F F

RFC 4944

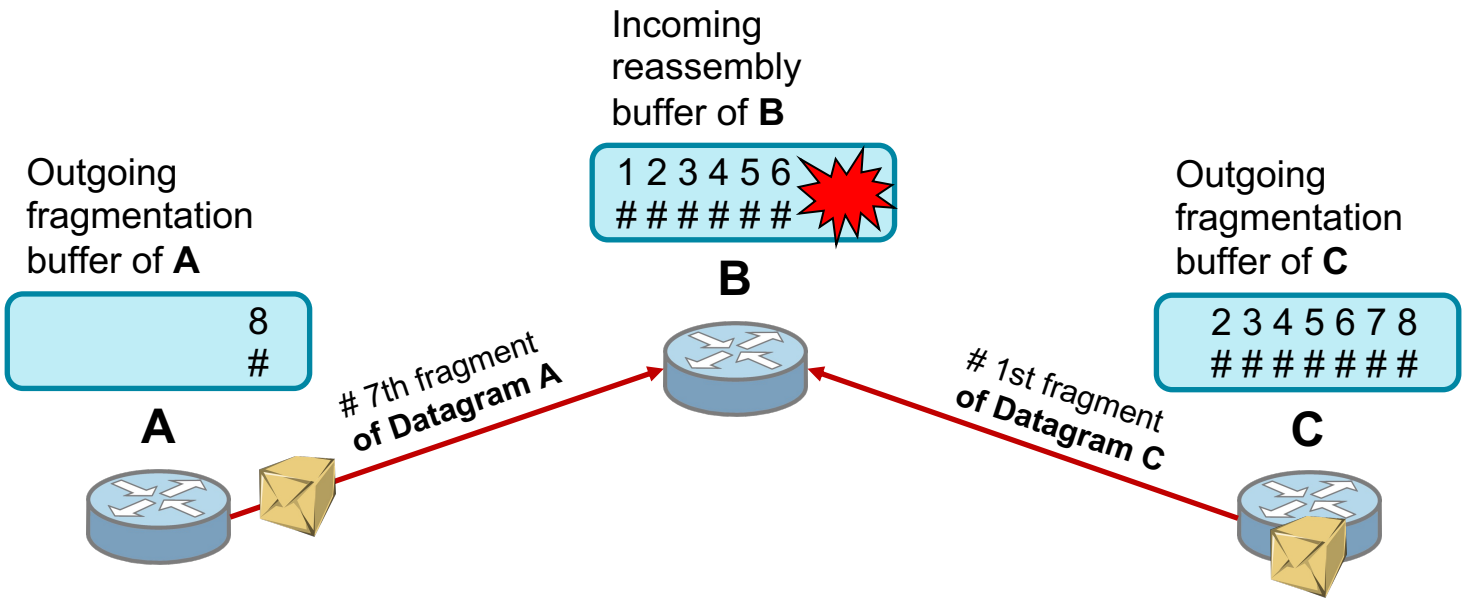
Resource Usage:

- The fragmentation process requires **large** incoming reassembly **buffers** (1280 bytes or more).

RFC 4944

Resource Usage:

- The fragmentation process requires **large** incoming reassembly **buffers** (1280 bytes or more).
- It will be possible to reassemble only very few complete IPv6 packets.


Incoming
reassembly
buffer of **B**
1 2 3 4 5 6
# # # # # #
Outgoing
fragmentation
buffer of **A**
8
#
Outgoing
fragmentation
buffer of **C**
2 3 4 5 6 7 8
# # # # # # #
B
A
C
7th fragment
of Datagram A
1st fragment
of Datagram C

RFC 4944

Resource Usage:

- The fragmentation process requires **large** incoming reassembly **buffers** (1280 bytes or more).
- It will be possible to reassemble only very few complete IPv6 packets.
- Thus, such issues introduces more losses in a multi-hop network.

Route-Over: Issues (Implementation or *datagram_tag*)

RFC 4944

Implementation:

- Only the **1st fragment** of an IPv6 packet **contains** the source and destination **IPv6 addresses**:

Route-Over: Issues (Implementation or *datagram_tag*)

RFC 4944

Implementation:

- Only the **1st fragment** of an IPv6 packet **contains** the source and destination **IPv6 addresses**:
- The subsequent fragments are routed based on the *datagram_tag*.

RFC 4944

Implementation:

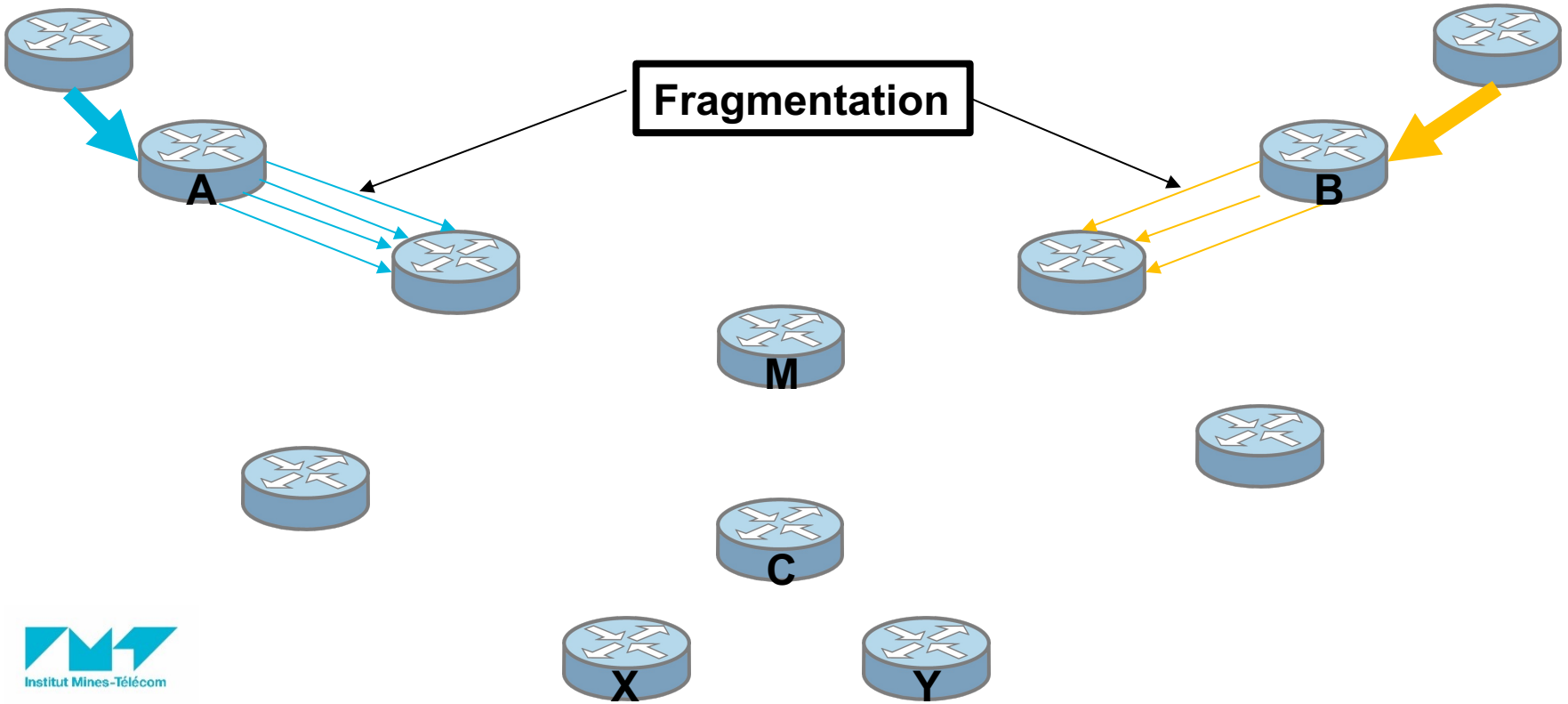
- Only the **1st fragment** of an IPv6 packet **contains** the source and destination **IPv6 addresses**:
- The subsequent fragments are routed based on the *datagram_tag*.
- A *datagram_tag* **is unique only** to the 6LoWPAN original **source** and final **destination nodes**.

RFC 4944

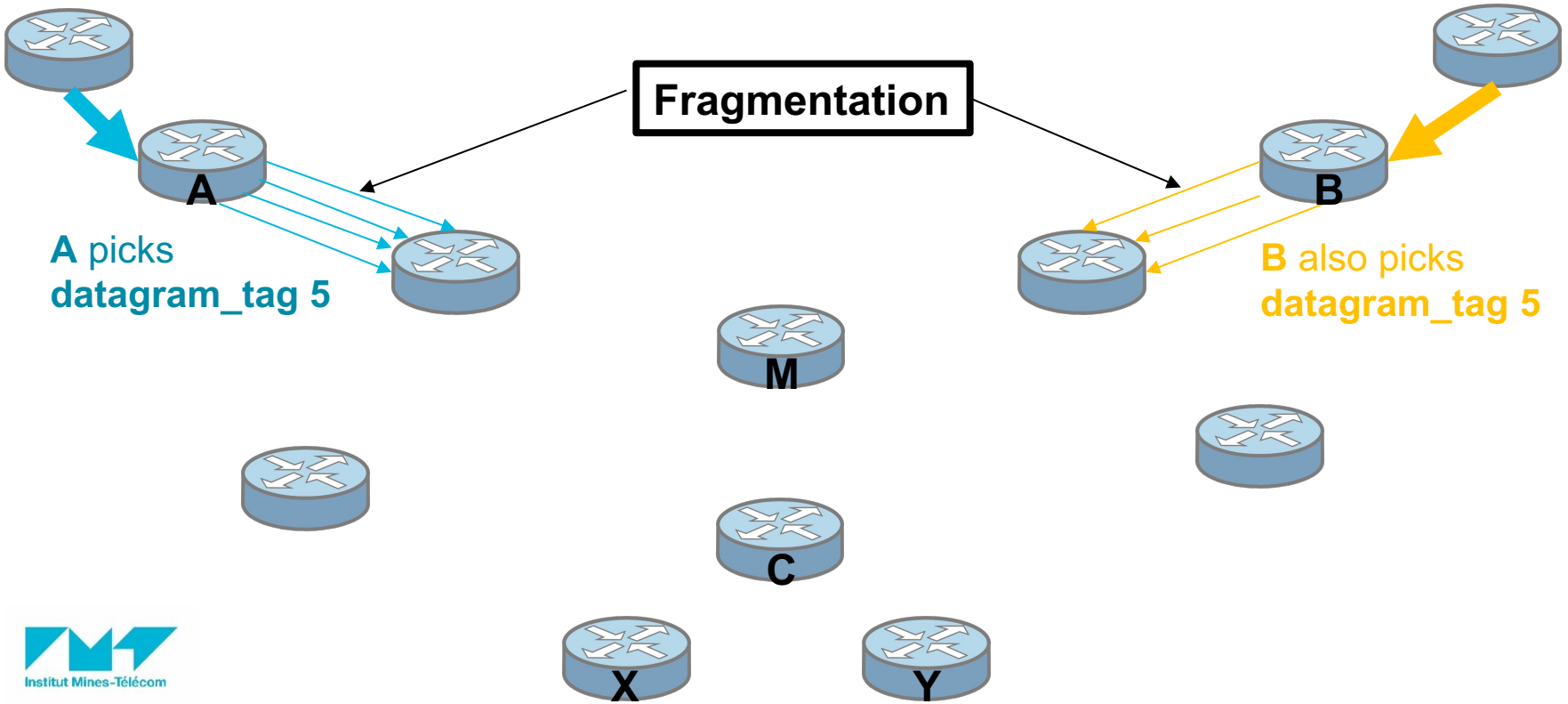
Implementation:

- Only the **1st fragment** of an IPv6 packet **contains** the source and destination **IPv6 addresses**:
- The subsequent fragments are routed based on the *datagram_tag*.
- A *datagram_tag* **is unique only** to the 6LoWPAN original **source** and final **destination nodes**.
- Thus, two different traffic flows may be tagged with the same value.

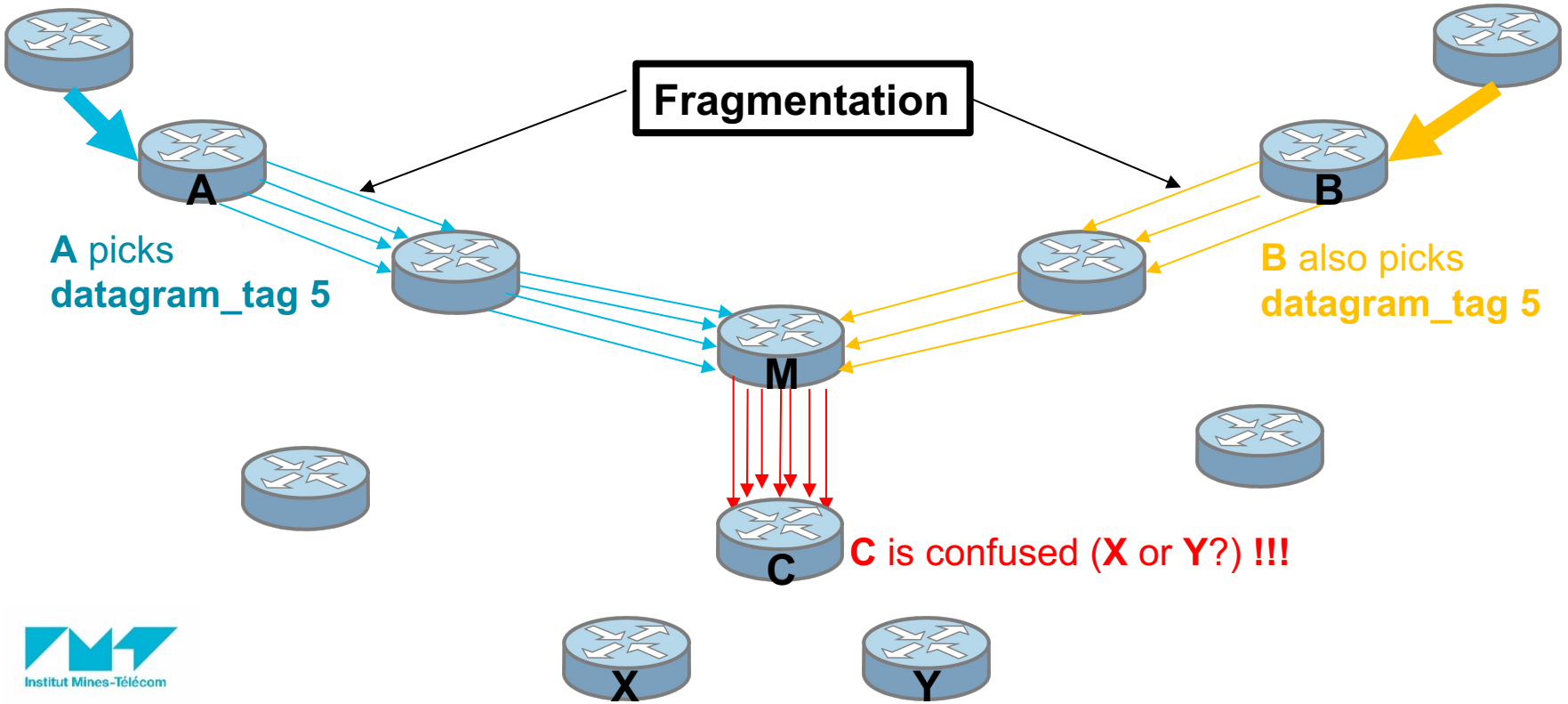
Route-Over: Issues (Implementation or *datagram_tag*)



Route-Over: Issues (Implementation or *datagram_tag*)

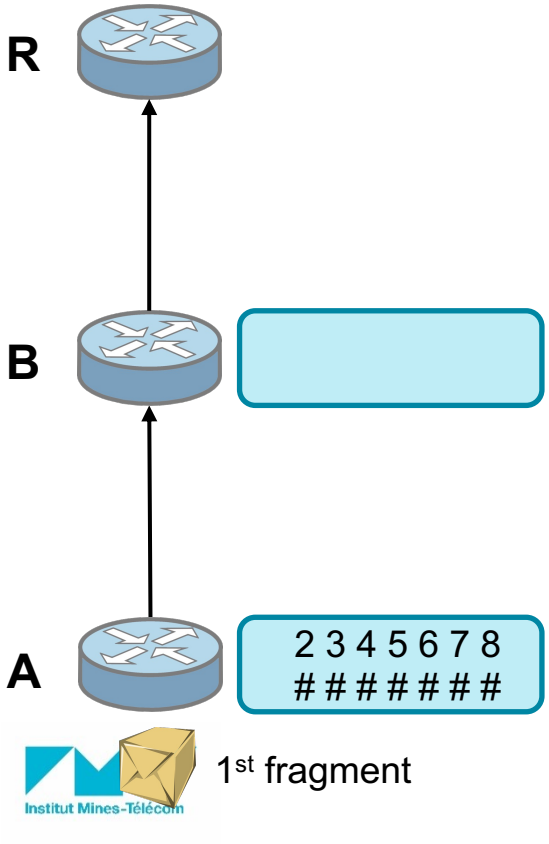


Route-Over: Issues (Implementation or *datagram_tag*)



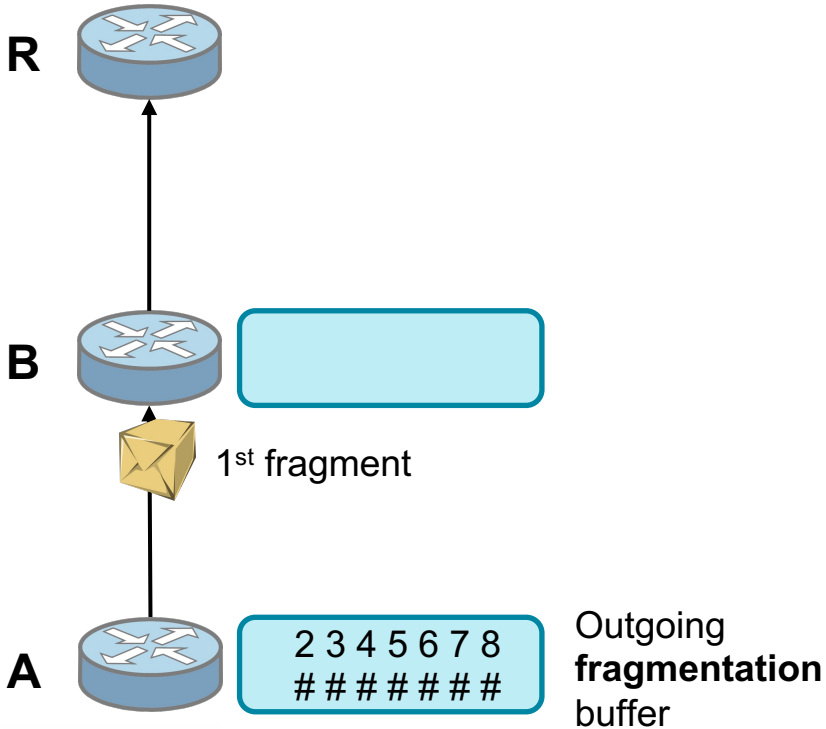
Chapter 4

RFC 8930: 6LoWPAN Fragment Forwarding (6LFF)



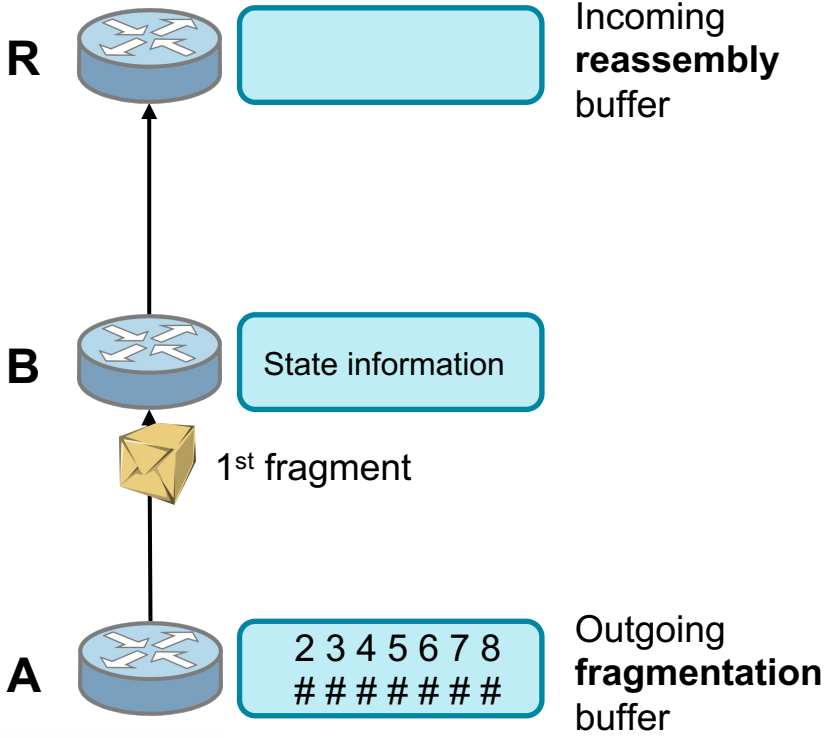
6LFF at a glance:

- The routing decision is made at the 1st fragment.



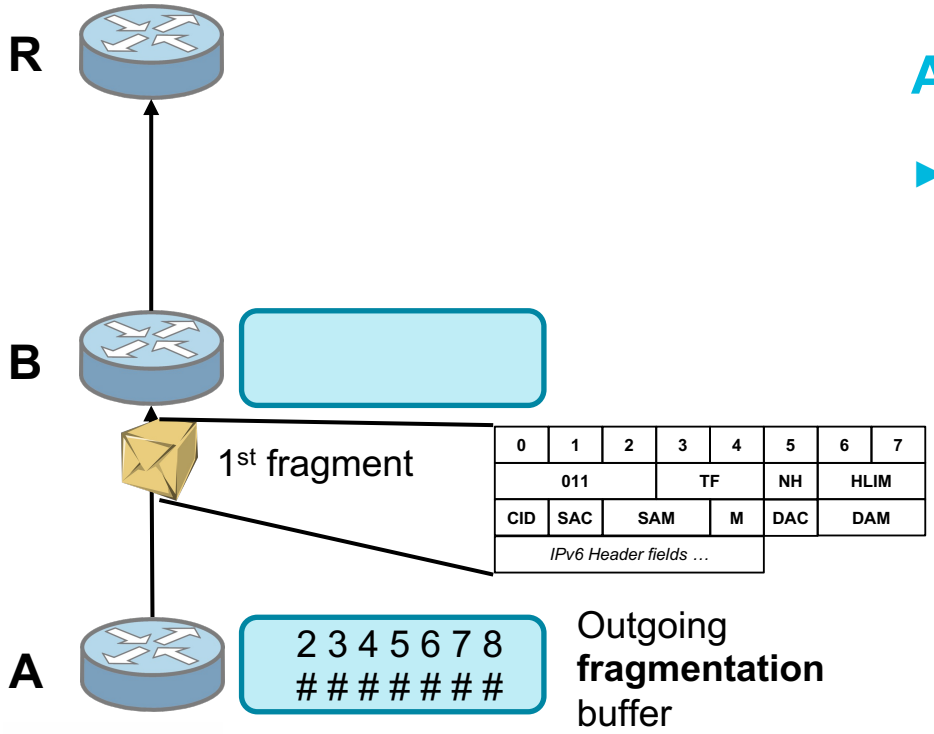
6LFF at a glance:

- The routing decision is made at the 1st fragment.
- The 1st fragment carries the ***IPv6 compressed header ✉ destination address***.



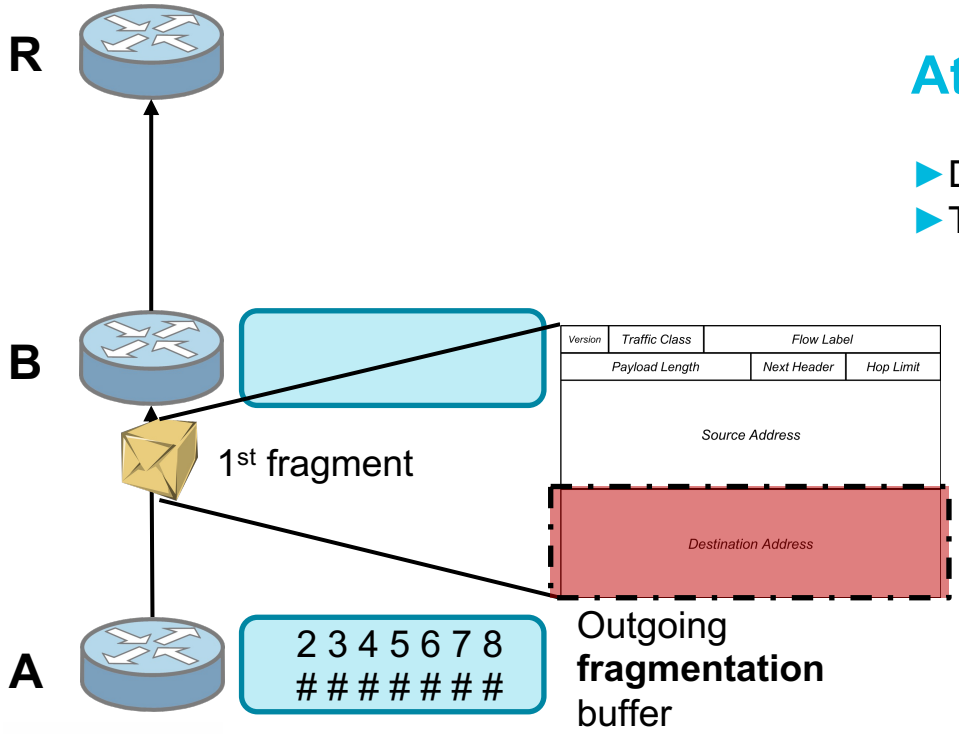
6LFF at a glance:

- The routing decision is made at the 1st fragment.
- The 1st fragment carries the ***IPv6 compressed header ✉ destination address***.
- The received fragments are forwarded immediately!



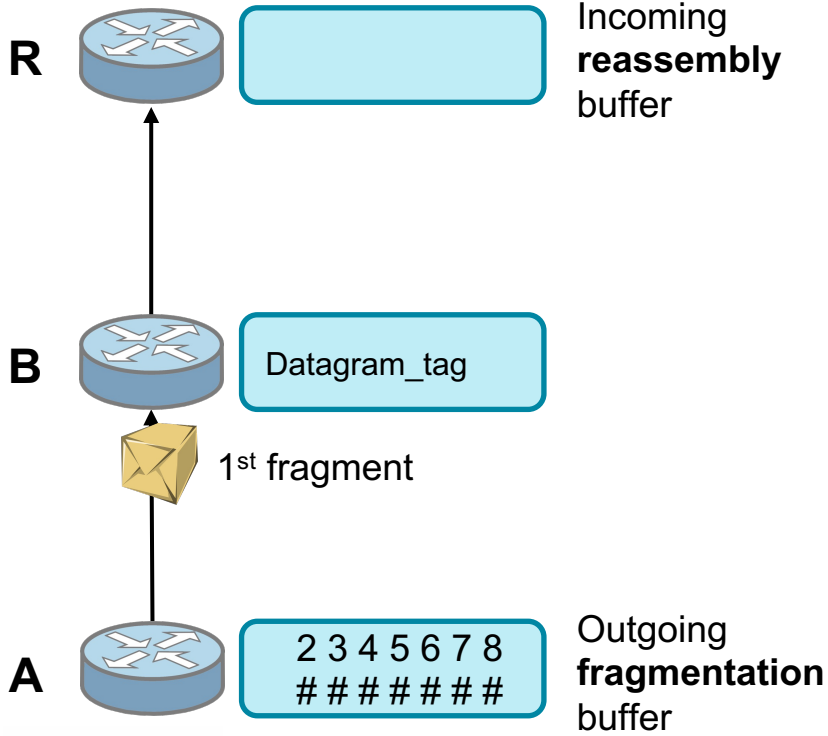
At the reception of the 1st fragment:

- Decompress the ***IPv6 compressed header***.



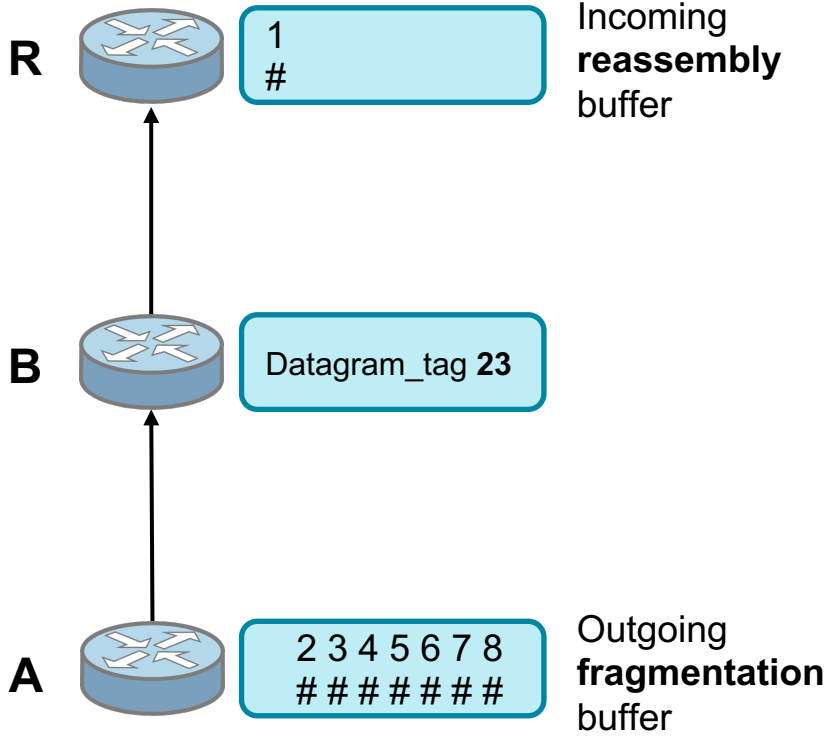
At the reception of the 1st fragment:

- Decompress the ***IPv6 compressed header***.
- To identify the IPv6 address of the next hop.



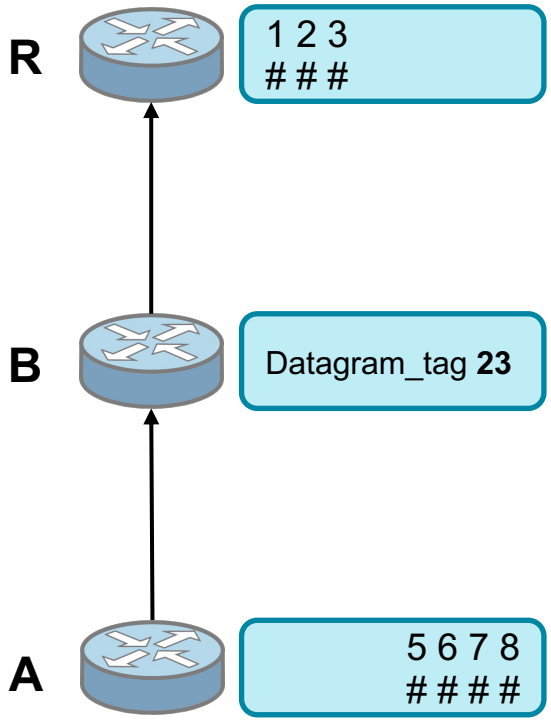
At the reception of the 1st fragment:

- Decompress the ***IPv6 compressed header***.
- To identify the IPv6 address of the next hop.
- Forward the fragment to the next hop.

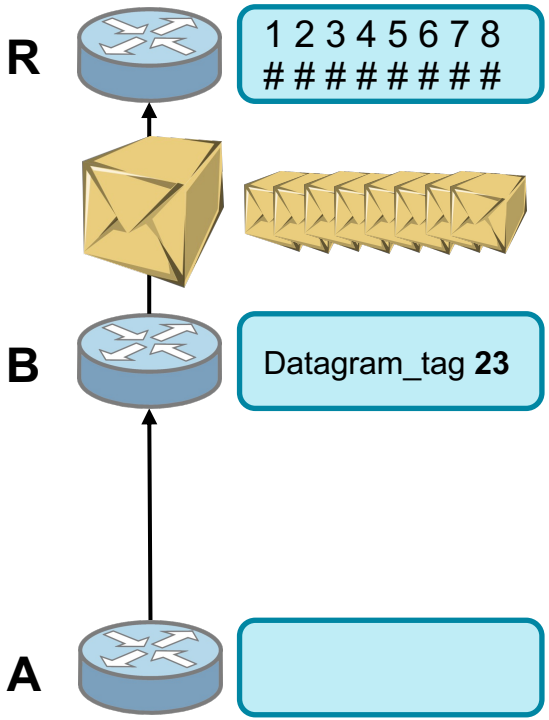


At the reception of the 1st fragment:

- Decompress the ***IPv6 compressed header***.
- To identify the IPv6 address of the next hop.
- Forward the fragment to the next hop.
- Register the ***datagram_tag*** of the fragment.

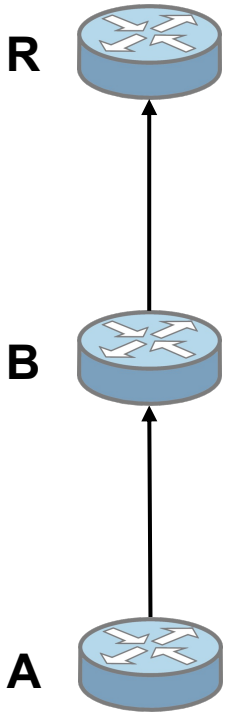


At the reception of the subsequent fragments:



At the reception of the subsequent fragments:

- The IPv6 packet is reassembled when **all** the fragments have arrived at the destination.

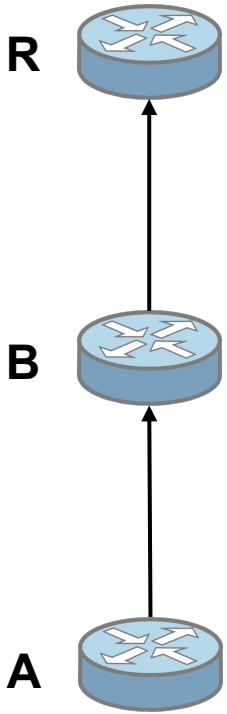


Node **B**'s VRB table.

incoming		outgoing	

VRB at a glance:

- The operation is similar to ***switching table***.

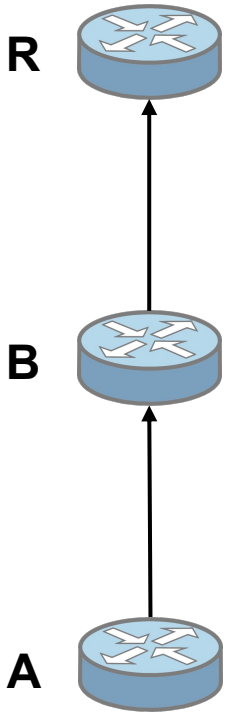


Node **B**'s VRB table.

incoming		outgoing	
entry 1 (IPv6 packet 1)			
entry 2 (IPv6 packet 2)			
entry … (IPv6 packet …)			
entry *n* (IPv6 packet *n*)			

VRB at a glance:

- The operation is similar to ***switching table***.
- The entries correspond to IPv6 packets.

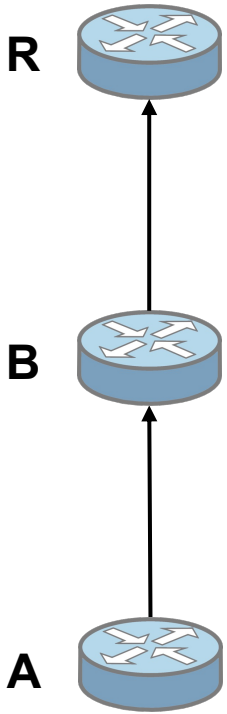


Node **B**'s VRB table.

incoming		outgoing	
entry 1 (empty)			
entry 2 (empty)			
entry … (empty)			
entry *n* (empty)			

VRB at a glance:

- The operation is similar to ***switching table***.
- The entries correspond to IPv6 packets.
- The VRBs have a maximum pre-allocated memory.

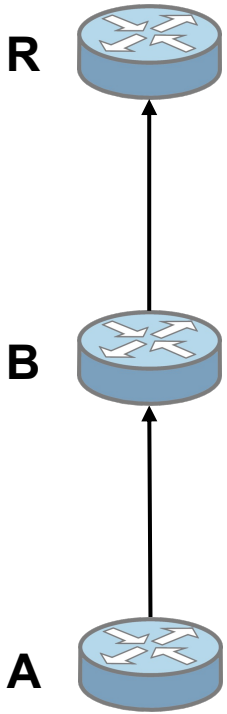


Node **B**'s VRB table.

incoming		outgoing	

A VRB entry:

- Each VRB entry is a tuple of 4 elements:

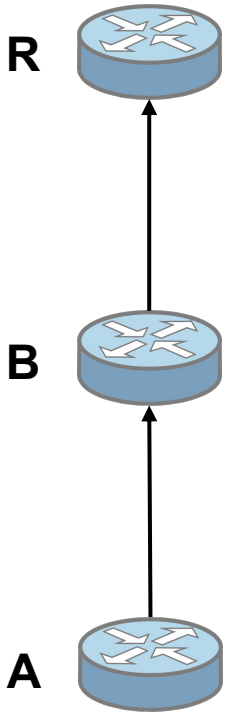


Node **B**'s VRB table.

incoming		outgoing	
L2 src			

A VRB entry:

- Each VRB entry is a tuple of 4 elements:
  - link-layer address of the previous hop.

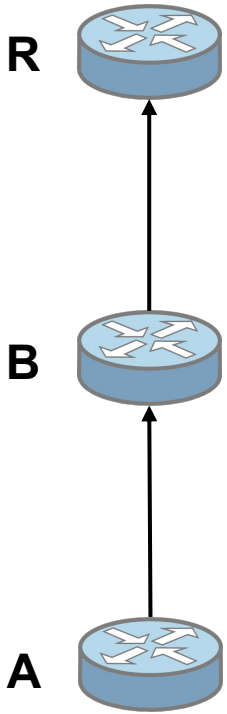


Node **B**'s VRB table.

incoming		outgoing	
L2 src	tag		

A VRB entry:

- Each VRB entry is a tuple of 4 elements:
  - link-layer address of the previous hop.
  - locally unique datagram_tag of the incoming fragment.

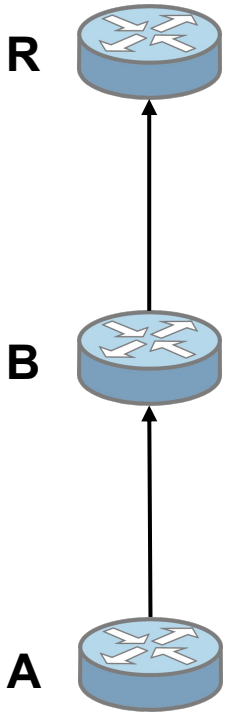


Node **B**'s VRB table.

incoming		outgoing	
L2 src	tag	L2 dest	

A VRB entry:

- Each VRB entry is a tuple of 4 elements:
  - link-layer address of the previous hop.
  - locally unique datagram_tag of the incoming fragment.
  - link-layer address of the next hop.

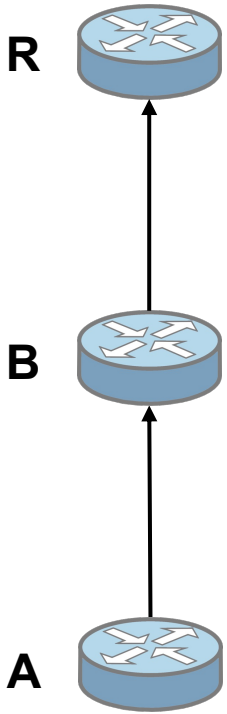


Node **B**'s VRB table.

incoming		outgoing	
L2 src	tag	L2 dest	tag

A VRB entry:

- Each VRB entry is a tuple of 4 elements:
  - link-layer address of the previous hop.
  - locally unique datagram_tag of the incoming fragment.
  - link-layer address of the next hop.
  - locally unique datagram_tag for the outgoing fragment.

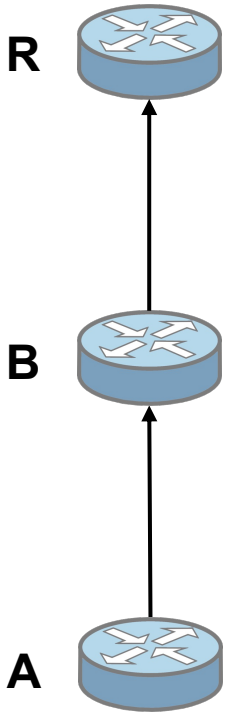


Node **B**'s VRB table.

incoming		outgoing	
L2 src	tag	L2 dest	tag

A VRB entry:

- Each VRB entry is a tuple of 4 elements:
  - link-layer address of the previous hop.
  - locally unique datagram_tag of the incoming fragment.
  - link-layer address of the next hop.
  - locally unique datagram_tag for the outgoing fragment.
- Each VRB entry requires 20 bytes.



Node **B**'s VRB table.

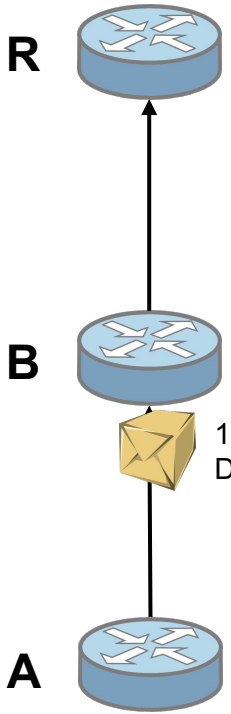
incoming		outgoing	
L2 src	tag	L2 dest	tag

At the reception of the 1st fragment:

2 3 4 5 6 7 8
# # # # # # #

Outgoing **fragmentation** buffer

1st fragment
Datagram_tag = **23**

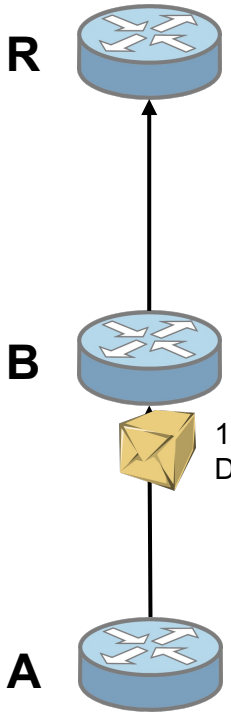


Node **B**'s VRB table.

incoming		outgoing	
L2 src	tag	L2 dest	tag
A	23		

At the reception of the 1st fragment:

- Record the *link-layer address* and *datagram_tag* of the **previous hop**.



1st fragment
Datagram_tag = 23

Node **B**'s VRB table.

incoming		outgoing	
L2 src	tag	L2 dest	tag
A	23	R	

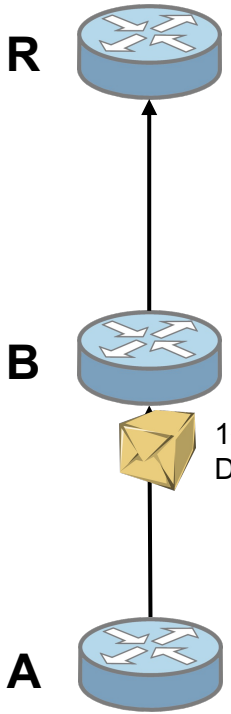
2 3 4 5 6 7 8
# # # # # # #

Outgoing **fragmentation** buffer

At the reception of the 1st fragment:

- Record the *link-layer address* and *datagram_tag* of the **previous hop**.
- Determine the *link-layer address* of the **next hop**.

Institut Mines-Télécom



Node **B**'s VRB table.

incoming		outgoing	
L2 src	tag	L2 dest	tag
A	23	R	47

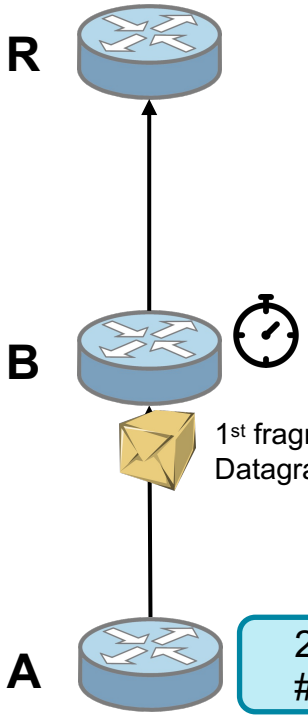
2 3 4 5 6 7 8
# # # # # # #

Outgoing **fragmentation** buffer

At the reception of the 1st fragment:

- Record the *link-layer address* and *datagram_tag* of the **previous hop**.
- Determine the *link-layer address* of the **next hop**.
- Pick a **new unique** *datagram_tag* for the **next hop**.

Institut Mines-Télécom



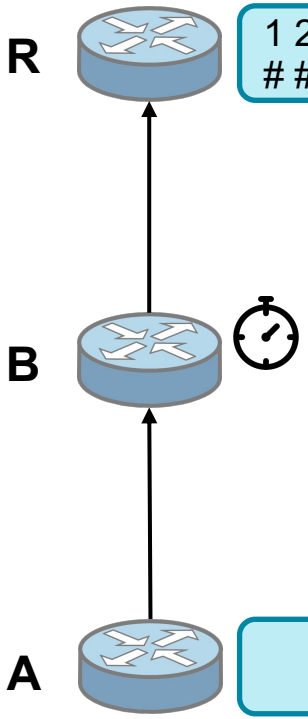
Node **B**'s VRB table.

incoming		outgoing	
L2 src	tag	L2 dest	tag
A	23	R	47

Outgoing **fragmentation** buffer

At the reception of the 1st fragment:

- Record the *link-layer address* and *datagram_tag* of the **previous hop**.
- Determine the *link-layer address* of the **next hop**.
- Pick a **new unique** *datagram_tag* for the **next hop**.
- Set a timer to allow discarding a **partially** reassembled packet after some timeout.



1 2 3
# # #

Incoming **reassembly** buffer

Node **B**'s VRB table.

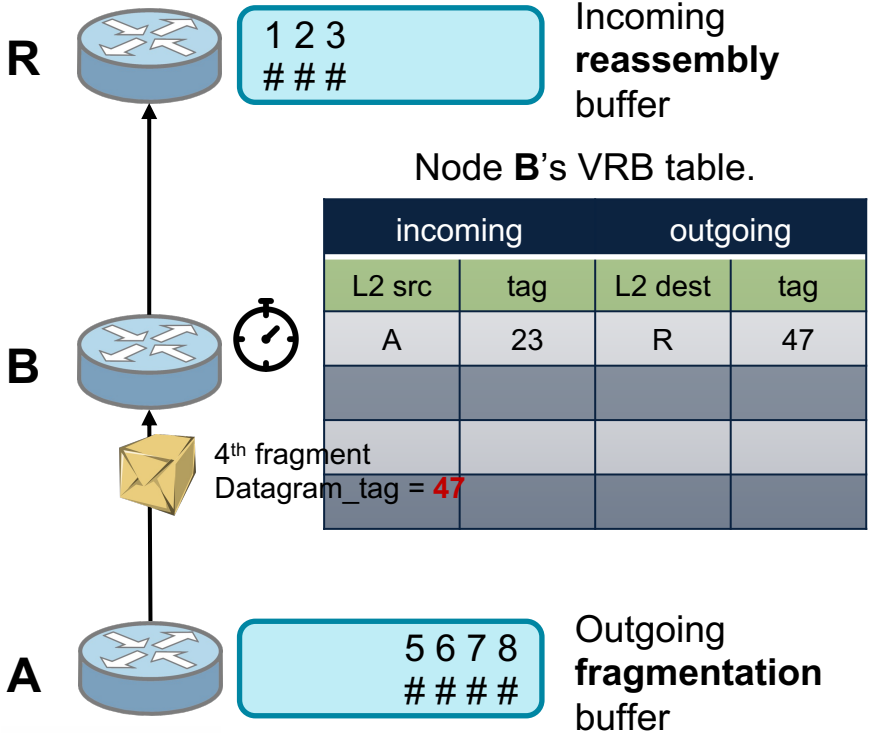
incoming		outgoing	
L2 src	tag	L2 dest	tag
A	23	R	47

5 6 7 8
# # # #

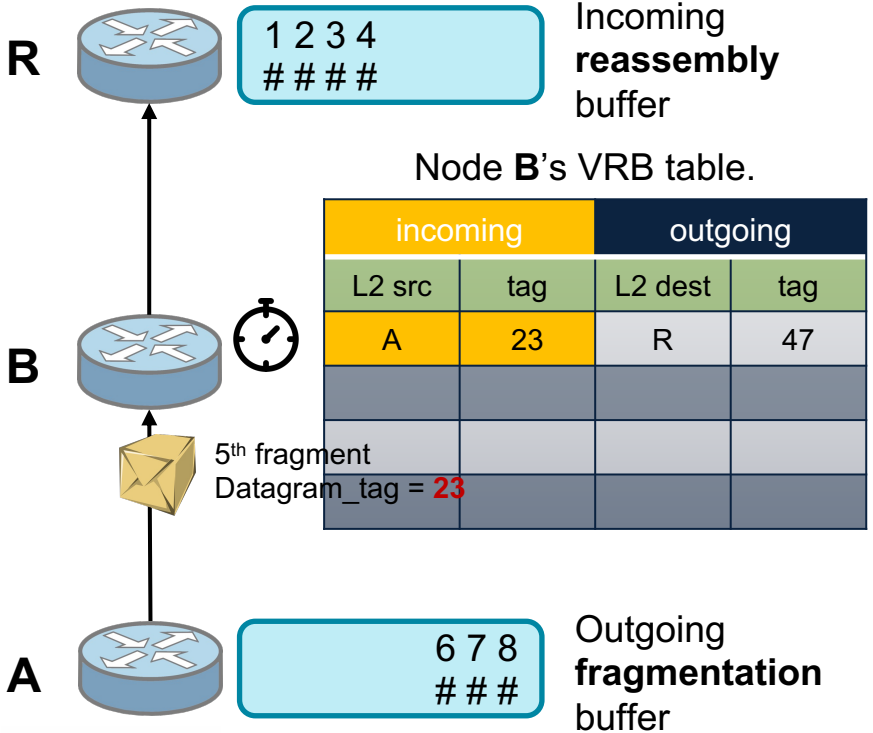
Outgoing **fragmentation** buffer

4th fragment
Datagram_tag = **23**

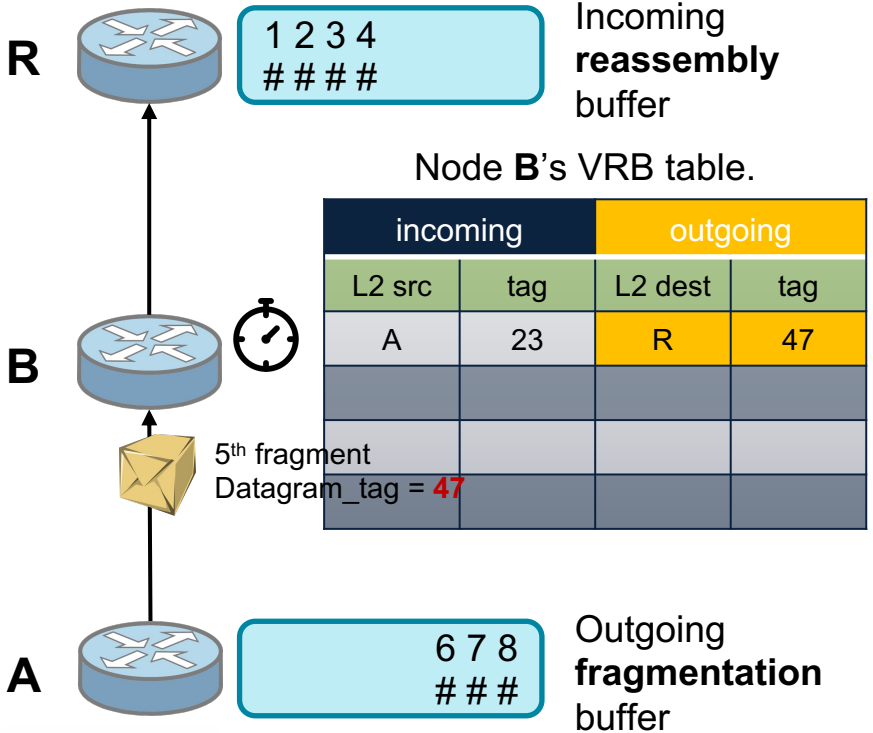
RFC 8930


R
1 2 3
# # #
Incoming **reassembly** buffer
Node **B**'s VRB table.
incoming
outgoing
L2 src
tag
L2 dest
tag
A
23
R
47
B
4th fragment
Datagram_tag = 47
A
5 6 7 8
# # # #
Outgoing **fragmentation** buffer

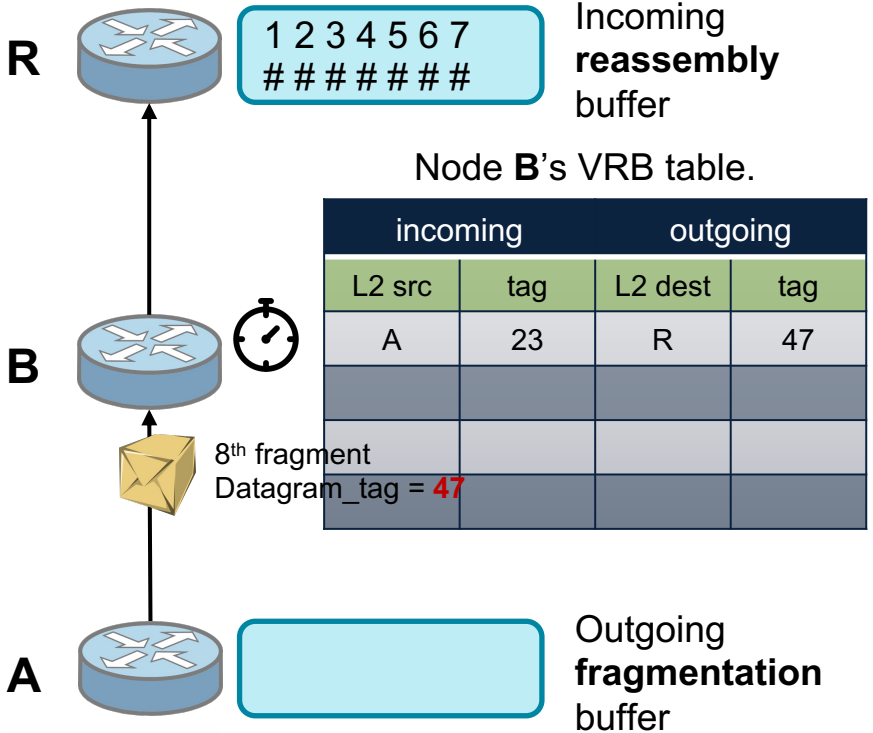
RFC 8930


R
1 2 3 4
# # # #
Incoming **reassembly** buffer
Node **B**'s VRB table.
incoming
outgoing
L2 src
tag
L2 dest
tag
A
23
R
47
B
5th fragment
Datagram_tag = 23
A
6 7 8
# # #
Outgoing **fragmentation** buffer

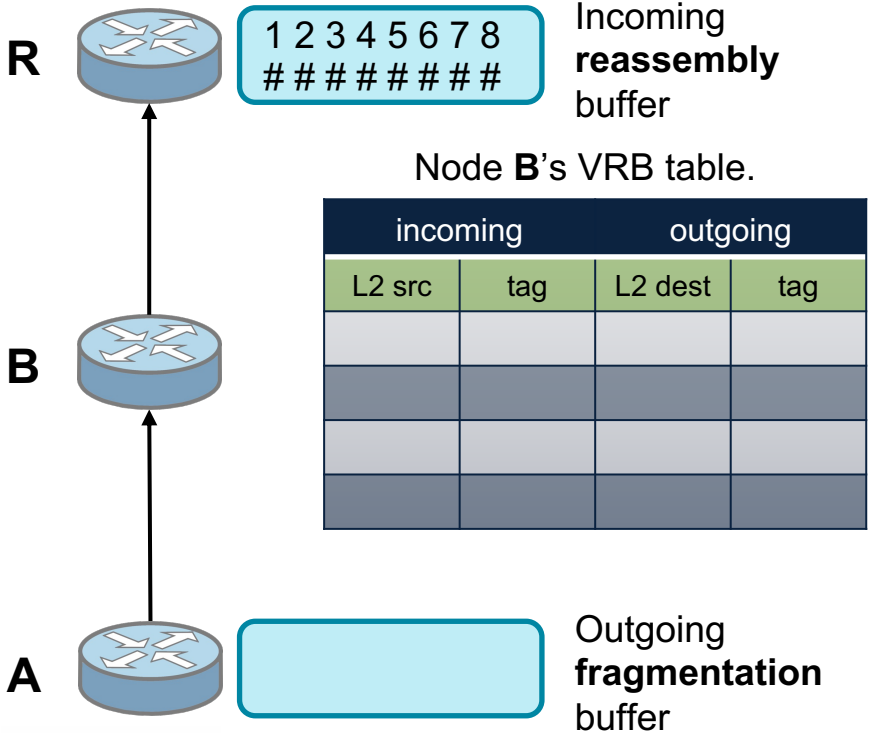
RFC 8930


R
1 2 3 4
# # # #
Incoming **reassembly** buffer
Node **B**'s VRB table.
incoming
outgoing
L2 src
tag
L2 dest
tag
A
23
R
47
B
5th fragment
Datagram_tag = 47
A
6 7 8
# # #
Outgoing **fragmentation** buffer

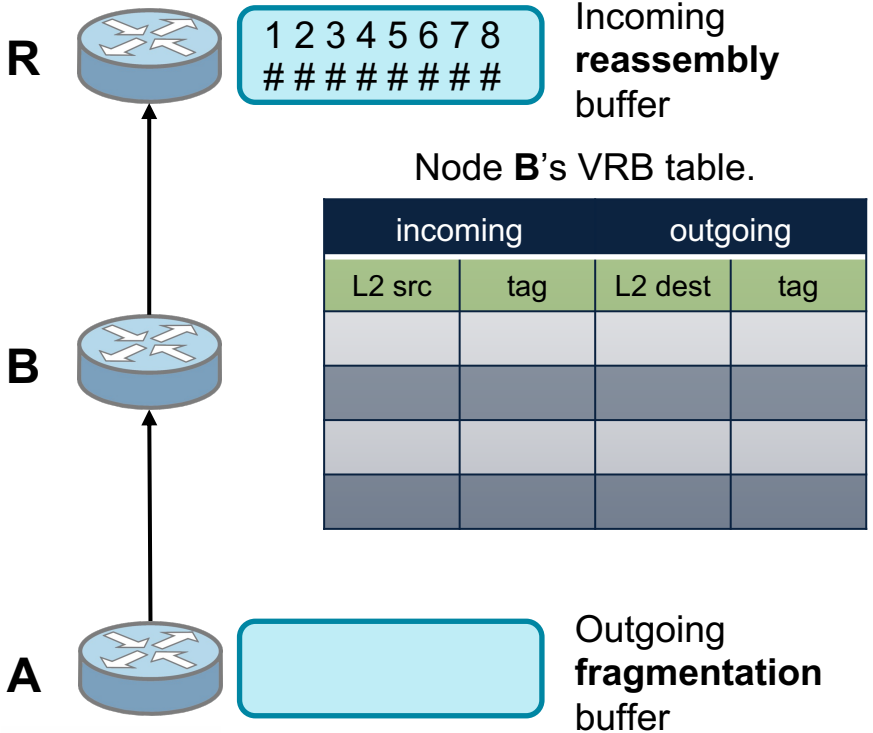
RFC 8930


R
1 2 3 4 5 6 7
# # # # # # #
Incoming **reassembly** buffer
Node **B**'s VRB table.
incoming
outgoing
L2 src
tag
L2 dest
tag
A
23
R
47
B
8th fragment
Datagram_tag = 47
A
Outgoing **fragmentation** buffer

RFC 8930


R
1 2 3 4 5 6 7 8
# # # # # # # #
Incoming **reassembly** buffer
B
Node **B**'s VRB table.
incoming
outgoing
L2 src
tag
L2 dest
tag
A
Outgoing **fragmentation** buffer

RFC 8930


R
1 2 3 4 5 6 7 8
# # # # # # # #
Incoming **reassembly** buffer
Node **B**'s VRB table.
incoming
outgoing
L2 src
tag
L2 dest
tag
B
A
Outgoing **fragmentation** buffer

RFC 8930

- The **end-to-end latency** should be greatly reduced.

RFC 8930

RFC 8930

- The **end-to-end latency** should be greatly reduced.
- The **end-to-end network reliability** should be improved.

RFC 8930

- The **end-to-end latency** should be greatly reduced.
- The **end-to-end network reliability** should be improved.
- The **datagram_tag issue** is solved.

RFC 8930

- There is **no Fragment Recovery** built-in.

RFC 8930

- There is **no Fragment Recovery** built-in.
  - It introduces unnecessary traffic in the network.

RFC 8930

- There is **no Fragment Recovery** built-in.
  - It introduces unnecessary traffic in the network.
- 6LFF **does not support per-fragment routing**.

RFC 8930

- There is **no Fragment Recovery** built-in.
  - It introduces unnecessary traffic in the network.
- 6LFF **does not support per-fragment routing**.
- The **size of the VRB table** necessarily **remains finite**.

RFC 8930

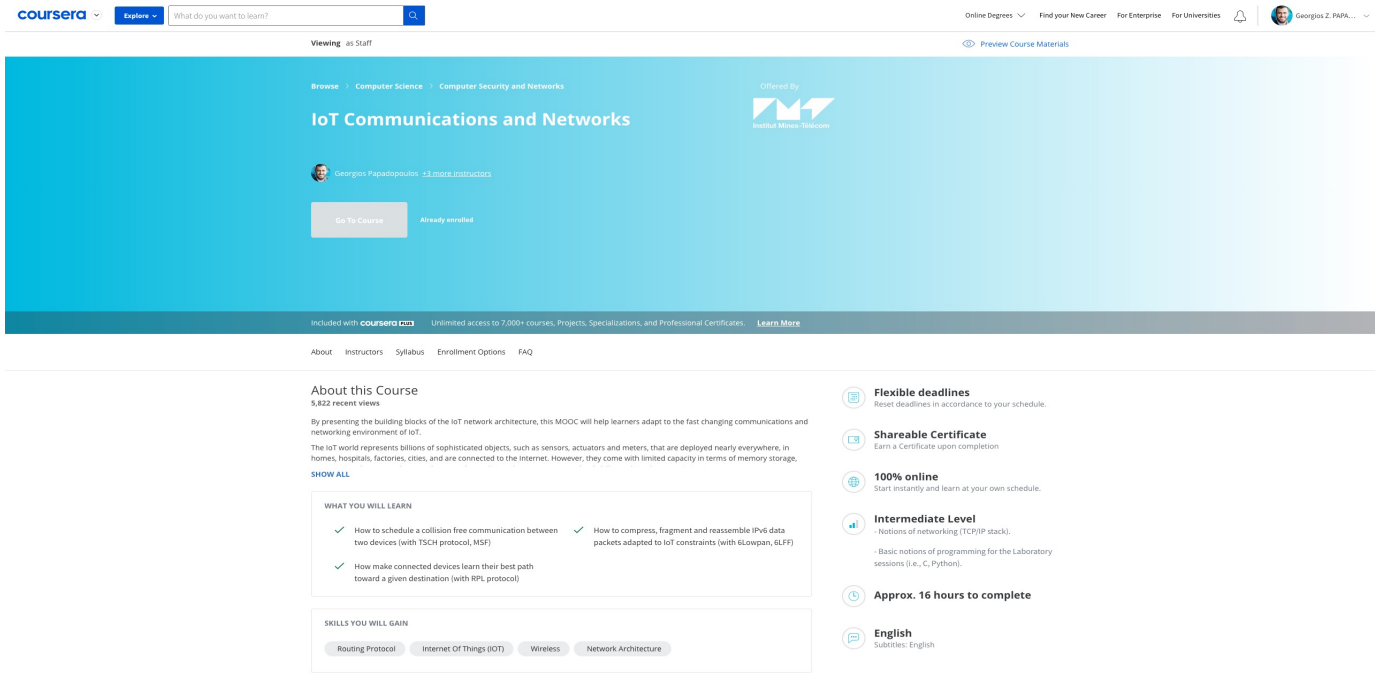
Institut Mines-Télécom

6LoWPAN: Fragmentation & Reassembly, Frame Delivery Modes, & Fragment Forwarding

Georgios Z. PAPADOPOULOS, Associate Professor, IMT Atlantique
georgios.papadopoulos@imt-atlantique.fr
www.georgiospapadopoulos.com
www.youtube.com/c/gzpapadopoulos

Miscellaneous

Institut Mines-Télécom



Institut Mines-Télécom

- Outline of the MOOC:
  - Week 1: Welcome & MAC Methods (i.e., TSCH)
  - Week 2: 6TiSCH
  - Week 3: IPv6 & 6LoWPAN
  - Week 4: RPL

- Educational Team:
  - Georgios Z. Papadopoulos
  - Nicolas Montavont
  - Géraldine Texier
  - Remous-Aris Koutsiamanis

- Interviews from the industrial community:
  - Pascal Thubert, Cisco Systems
  - Thomas Watteyne, Analog Devices, Falco
  - Rémi Dubaele, ENEDIS