

TTL Procedures for SR-TE Paths in Label Switched Path Traceroute Mechanisms

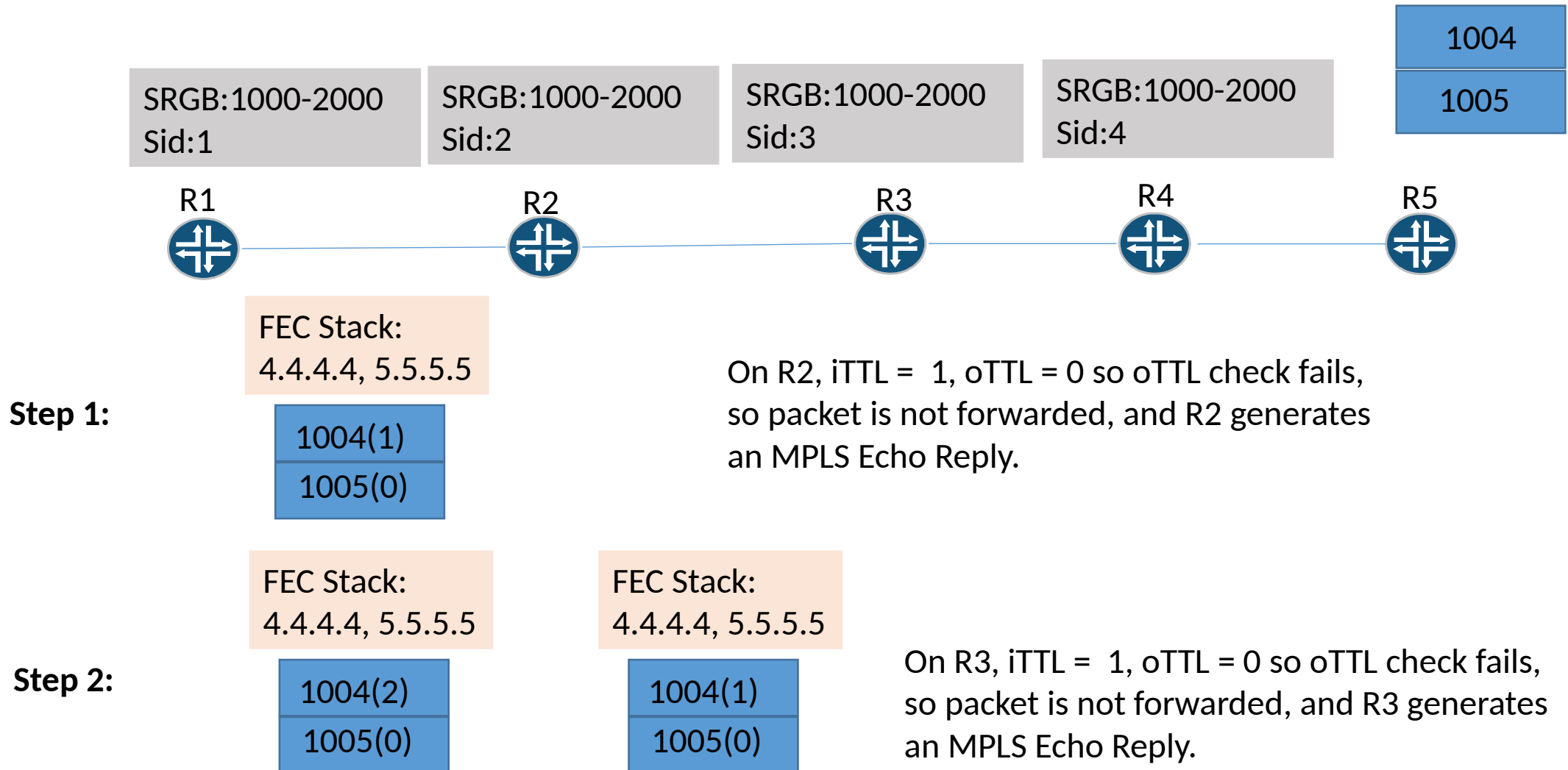
Kapil Arora

Shraddha Hegde

IETF-103

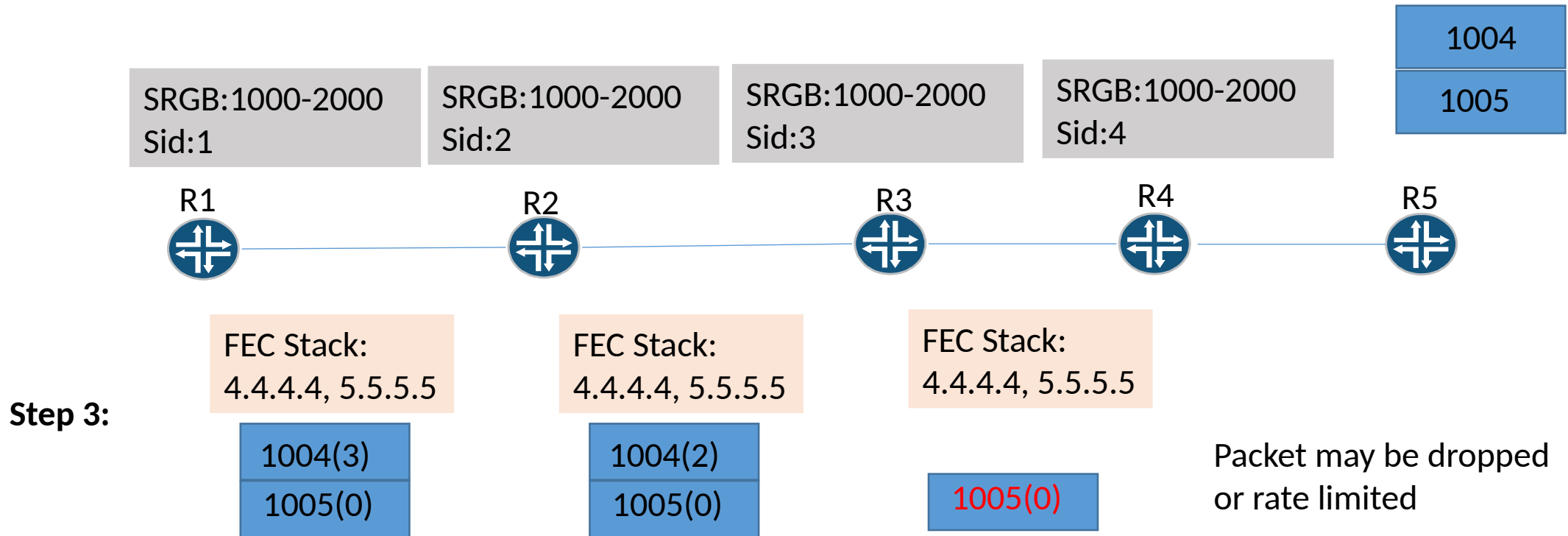
Problem

- Short pipe model with PHP on all nodes



Problem

- Short pipe model with PHP on all nodes

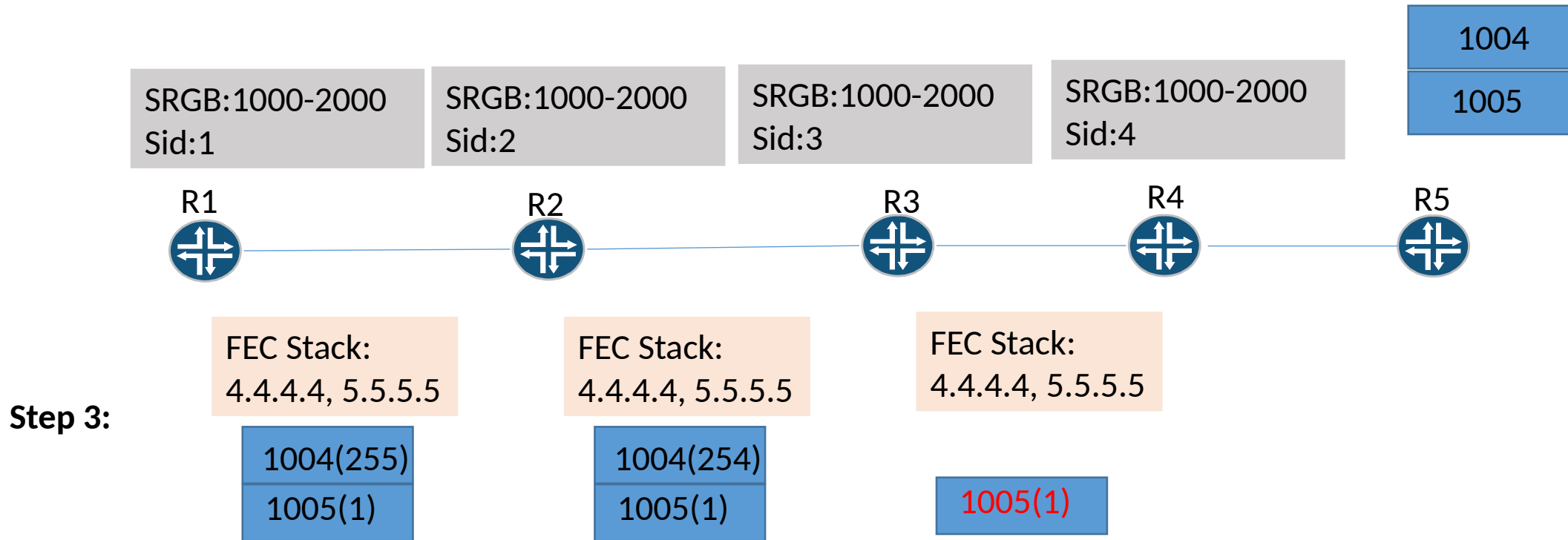


On R3, iTTL = 2,
oTTL = 1 so oTTL
check succeeds.

On R4, iTTL = 0, oTTL =
-1 so oTTL check fails.
R4 sends an MPLS
Echo Reply with stack
depth = 1.

Problem

- Short pipe model with PHP on all nodes

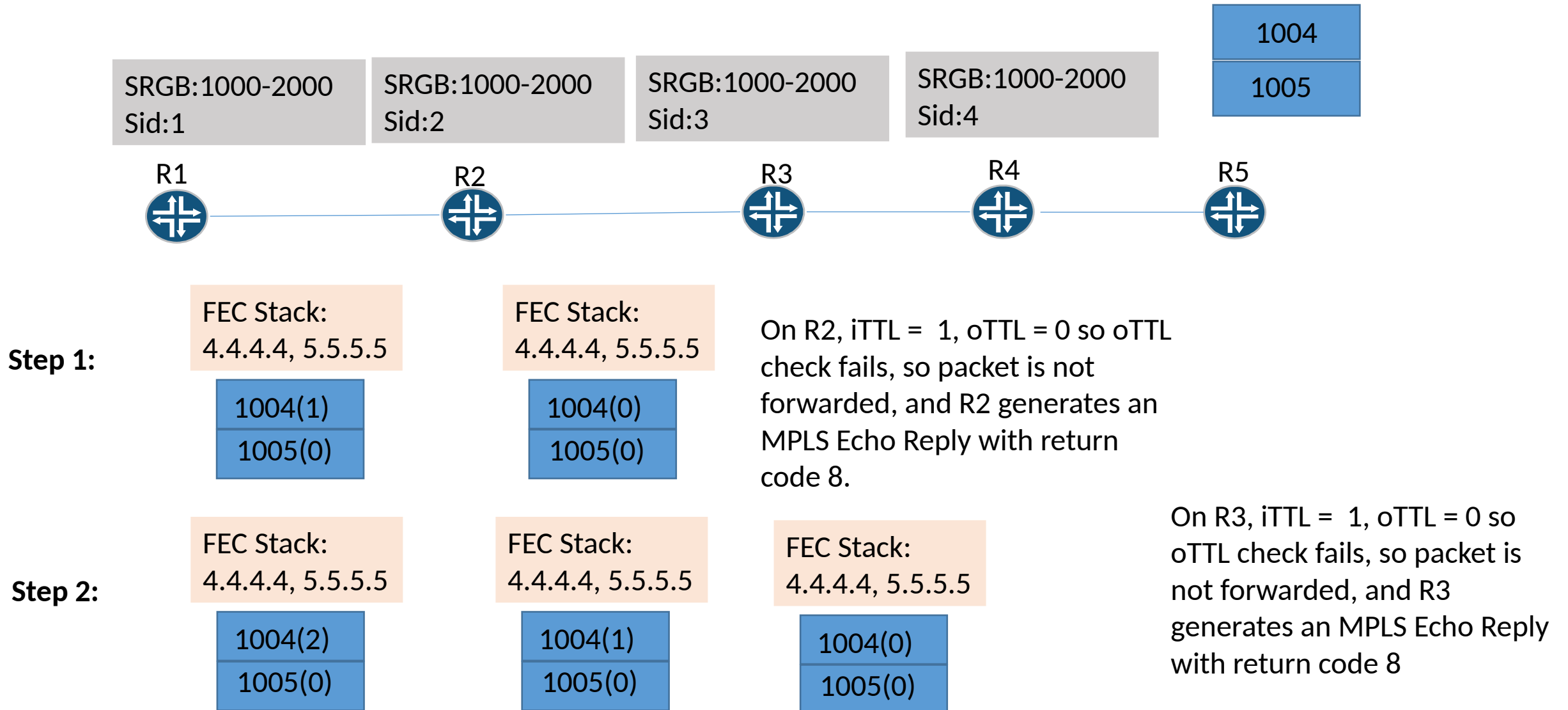


On R3, iTTL = 255, oTTL = 254 so oTTL check succeeds.

On R4, iTTL = 1, oTTL = 0 so oTTL check fails again

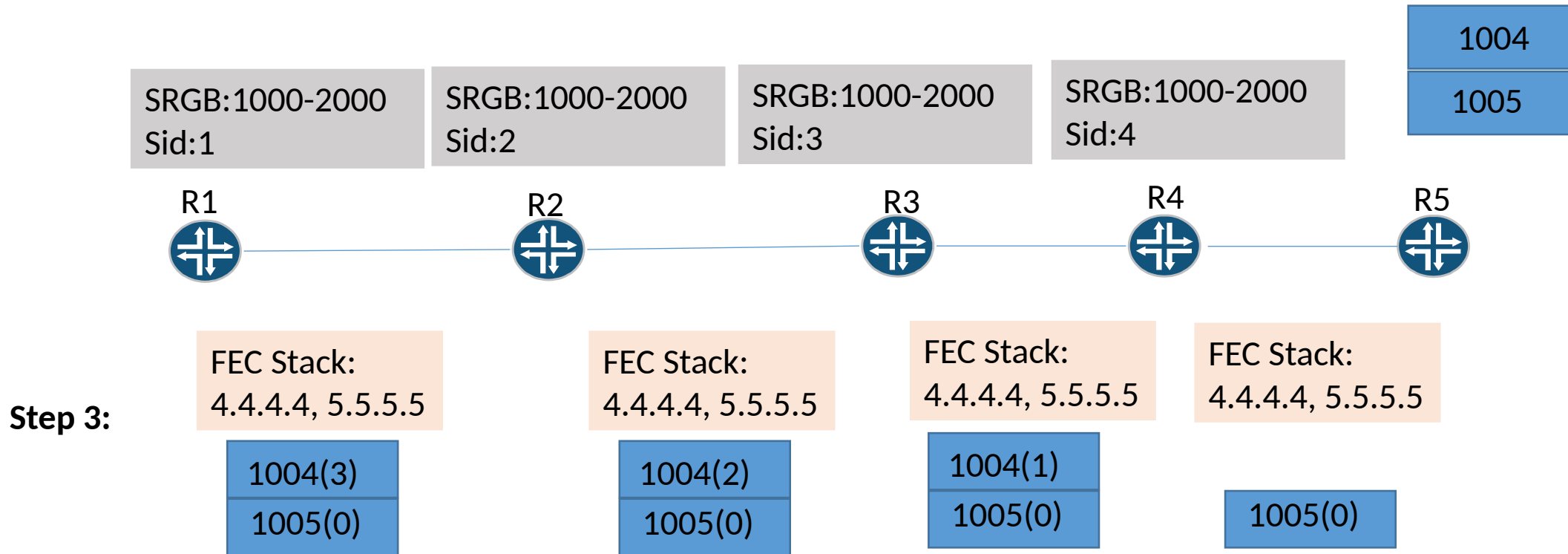
Problem

- Uniform model with PHP on all nodes



Problem

- Uniform pipe model with PHP on all nodes

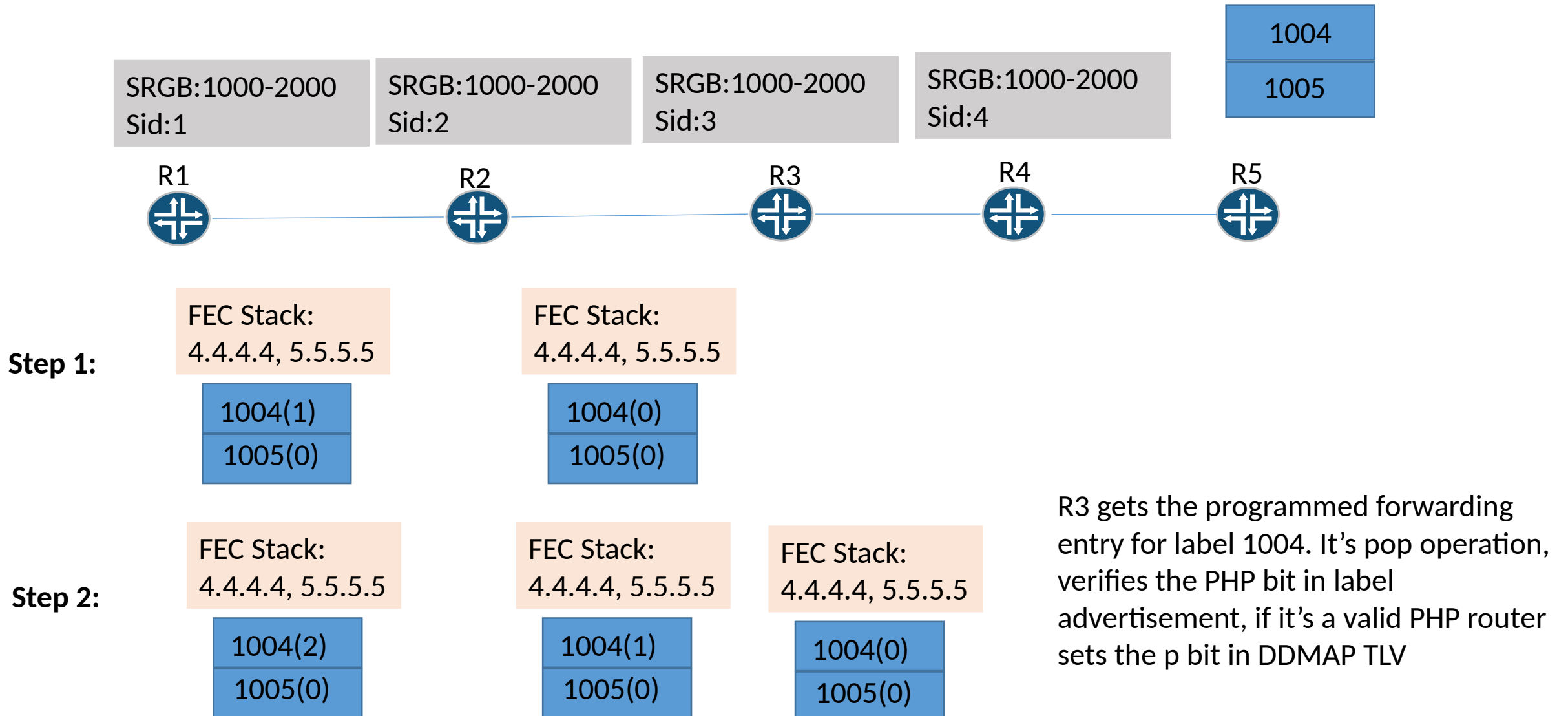


TTL not copied from top label as it is greater. Problem is same as in short-pipe mode

MPLS TTL copy mechanisms should not copy the outer Tunnel TTL to inner Tunnel if the inner Tunnel TTL is lower

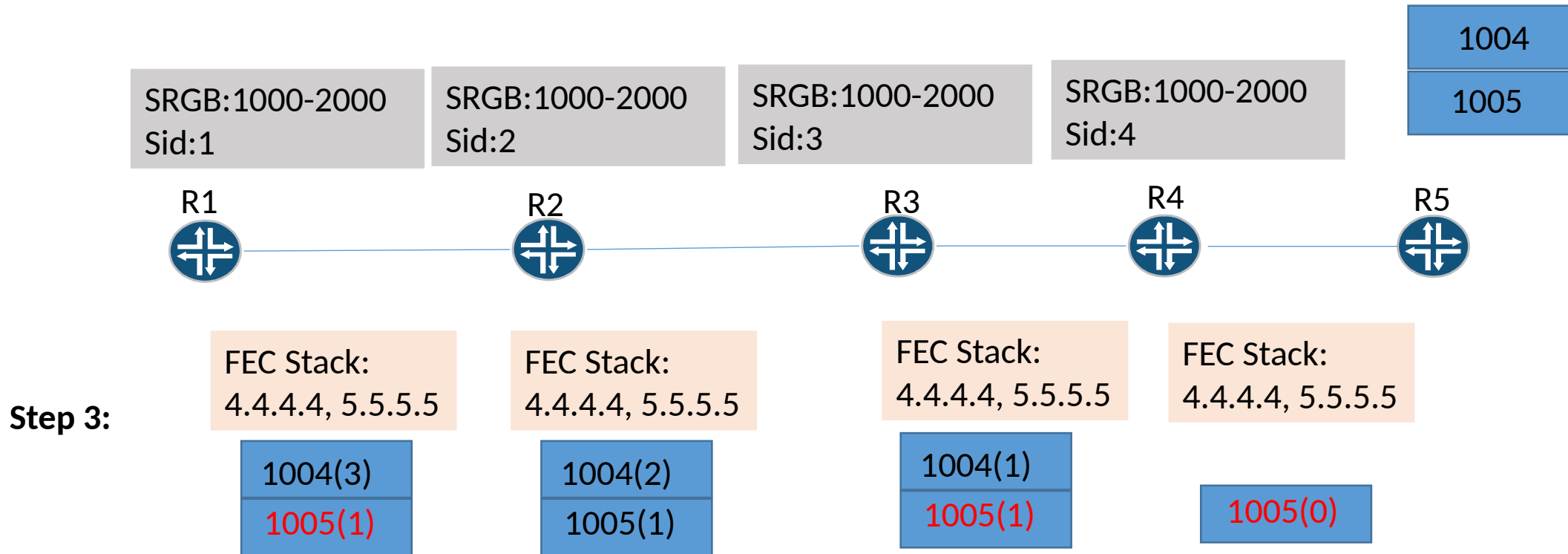
Solution

- Short pipe model with PHP on all nodes



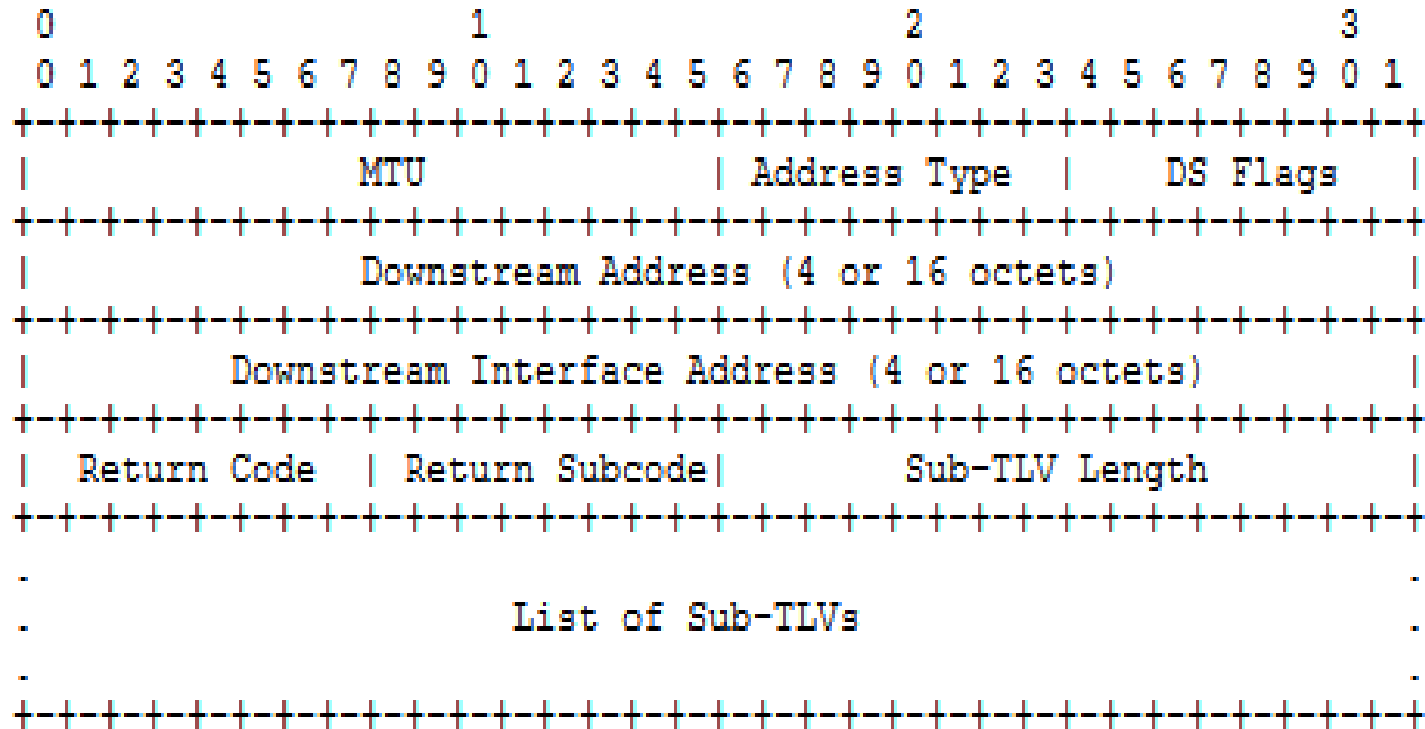
Solution

- Short pipe model with PHP on all nodes



When R3 sends DDMAP with p-bit set, the head-end node increments the TTL of the top label as well as next inner label.

New Flag in DDMAP TLV



Bit Number	Name	Reference
7	N: Treat as a Non-IP Packet	[RFC8029]
6	I: Interface and Label Stack Object Request	[RFC8029]
5	E: ELI/EL push indicator	[RFC8012]
4	L: Label-based load balance indicator	[RFC8012]
3	P: Penultimate Hop router	
2-0	Unassigned	

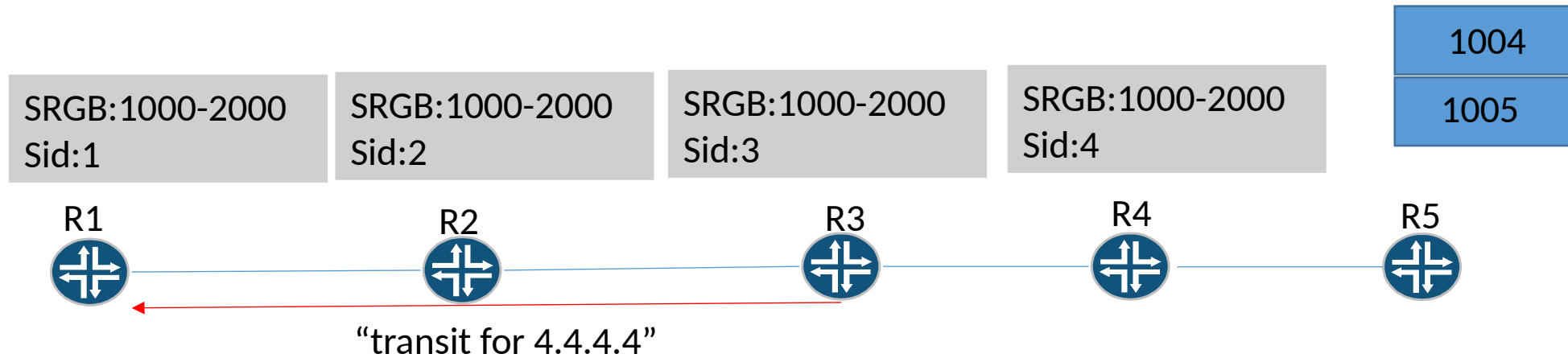
Alternate Approaches Evaluated

- PHP router sending egress return code
- Setting inner labels to 1
- Setting inner labels to 255
- Setting inner label to 2 on reaching egress

PHP Router sending “egress” return code

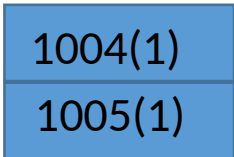
- > The Tunnel is not being traced till egress.
- > Can this mechanism effectively detect mis-programmed entries?
- > What if R2 had programmed a pop operation for 1004?
 - > What if R4 had programmed a swap operation?
 - > When the label-stack [1004,1005] was instantiated at the head-end, the intention was to hit the router R4 and go to R5.
Wouldn't it make sense to return “egress” return code from R4 instead of R3?

Setting Inner Labels to 1



Step 1:

FEC Stack:
4.4.4.4, 5.5.5.5

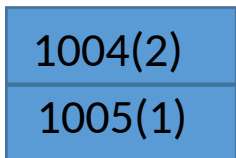


On R2, iTTL = 1, oTTL = 0 so oTTL check fails, so packet is not forwarded, and R2 generates an MPLS Echo Reply.

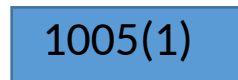
Assume R2 is mis-programmed and pops the labels 1004. since egress is not reached, ingress would not increase the TTL of 1005. The FEC validation passes and also the incoming label validation passes

Step 2:

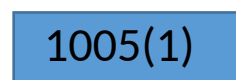
FEC Stack:
4.4.4.4, 5.5.5.5



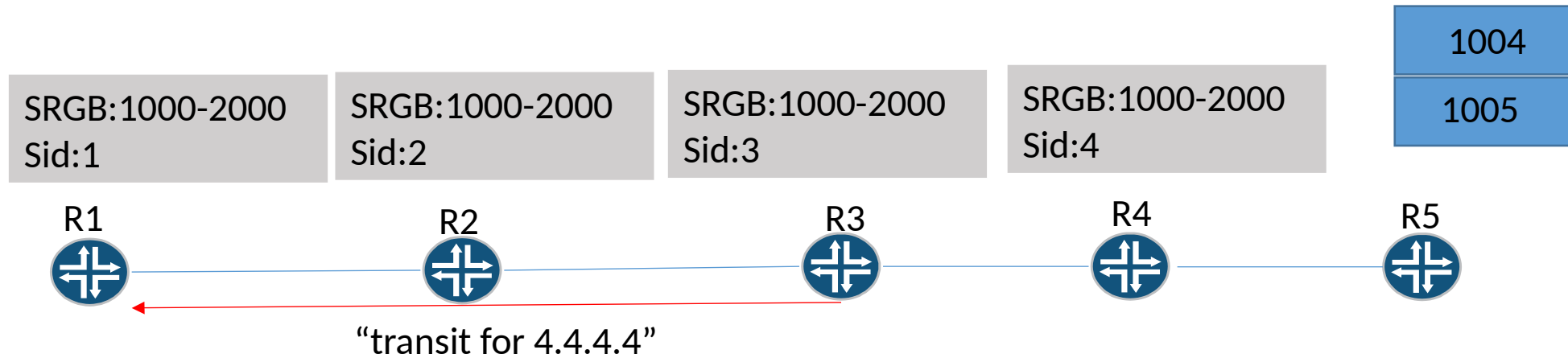
FEC Stack:
4.4.4.4, 5.5.5.5



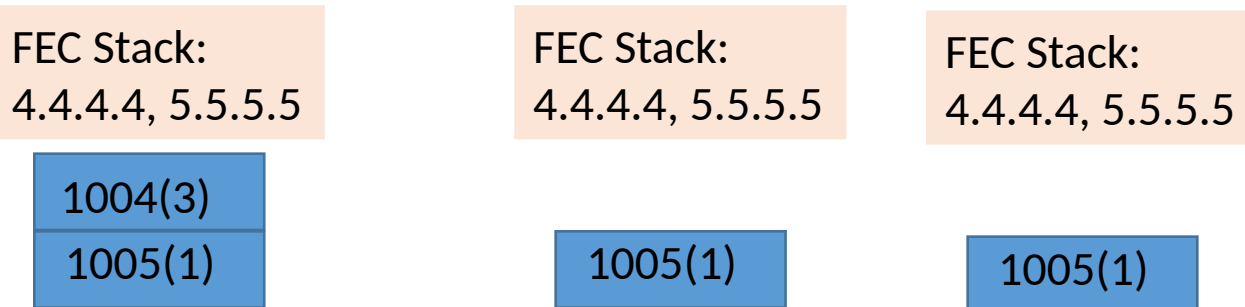
FEC Stack:
4.4.4.4, 5.5.5.5



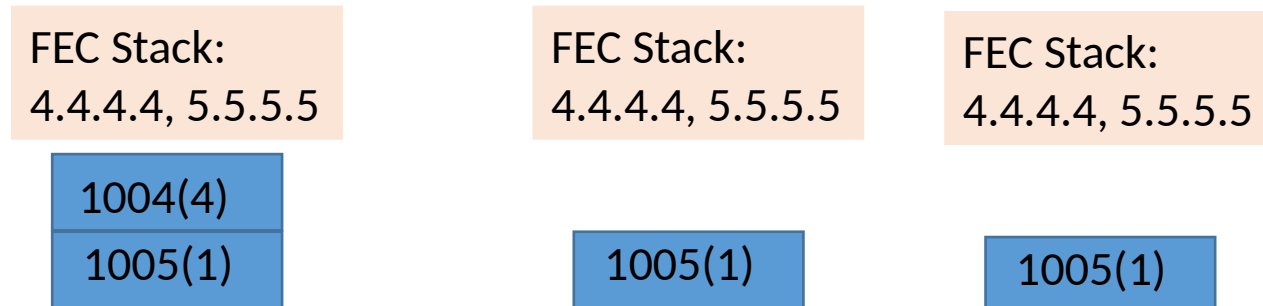
Setting Inner Labels to 1



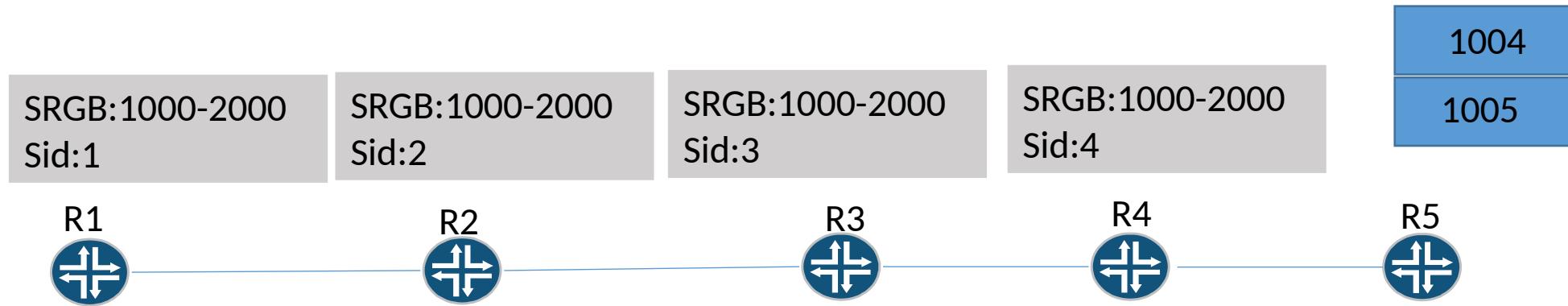
Step 3:



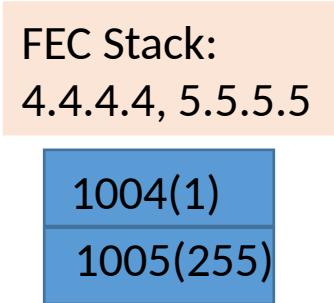
Step 4:



Setting Inner Labels to 255



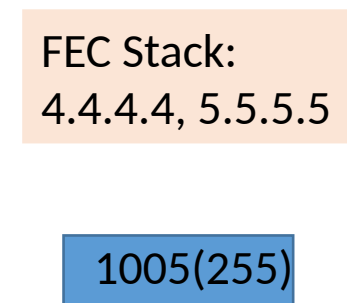
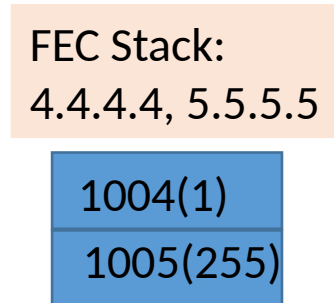
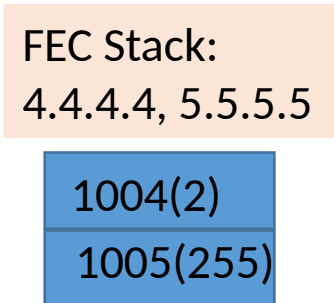
Step 1:



On R2, iTTL = 1, oTTL = 0 so oTTL check fails, so packet is not forwarded, and R2 generates an MPLS Echo Reply.

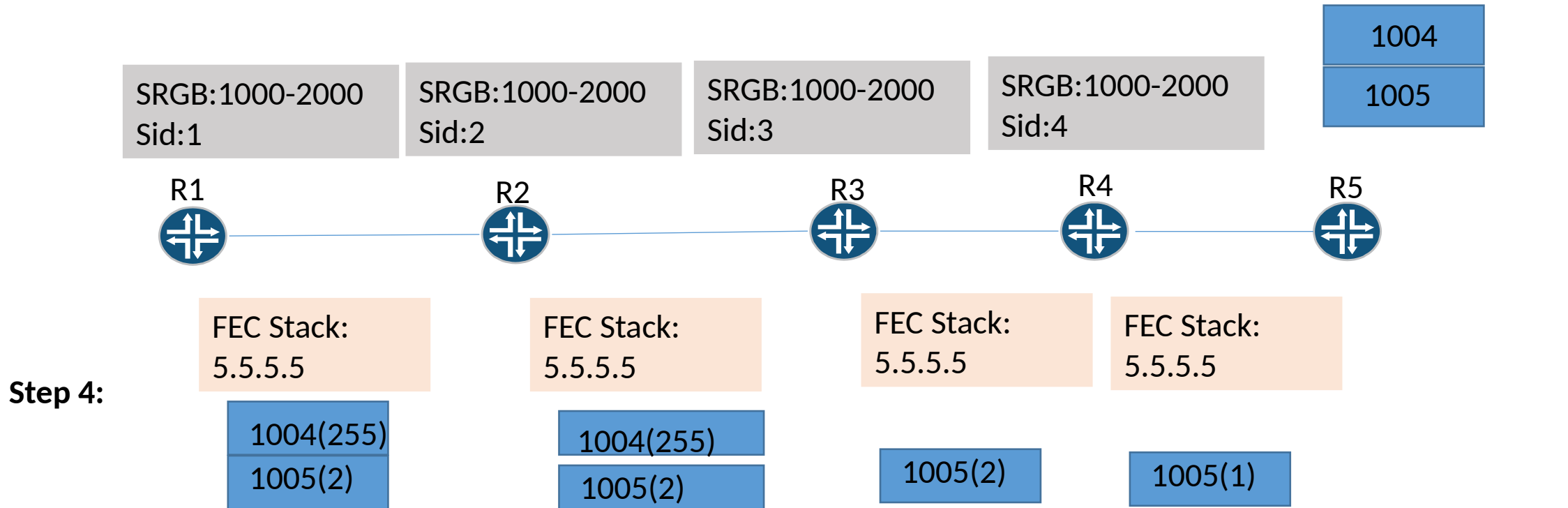
In short pipe mode, since TTL is not getting copied to bottom label, setting inner label to 255 does not work as the TTL will not expire on next router.

Step 2:



Setting inner label to 2

- Short pipe model with PHP on all nodes



On R4, iTTL = 2,
oTTL = 1 so oTTL
check succeeds.

On R5, iTTL = 1,
oTTL = 0 so oTTL
check fails. R5 sends
an MPLS Echo Reply
with stack depth = 1.

This has the same problem as setting all inner labels to 1 as the misprogrammed router could lead to looping of OAM procedure.

Next steps

- Request Feedback from WG

Questions & comments