Network Functions in SR-MPLS environment

Łukasz Makowski
makowski@uva.nl

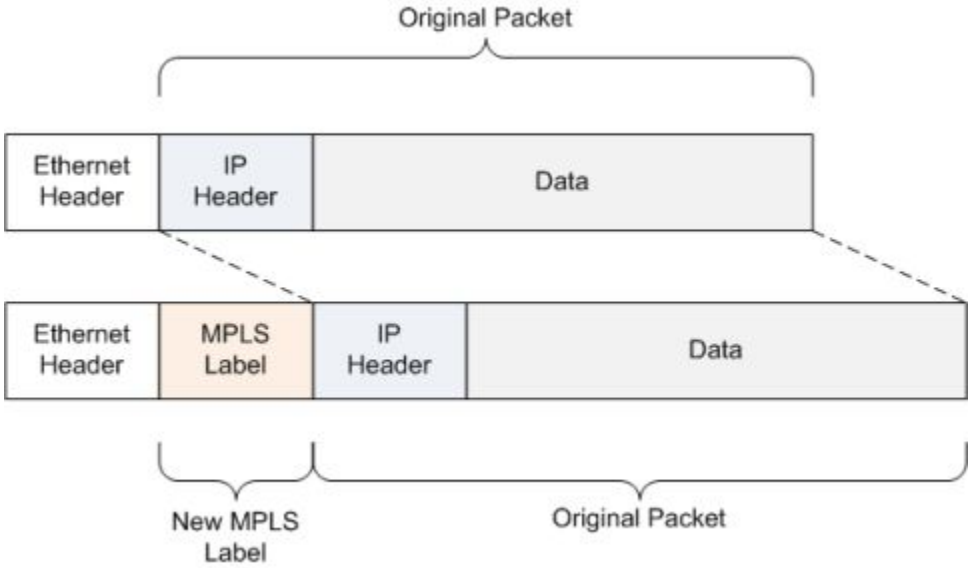
Paola Grosso
pgrosso@uva.nl

UNIVERSITEIT VAN AMSTERDAM

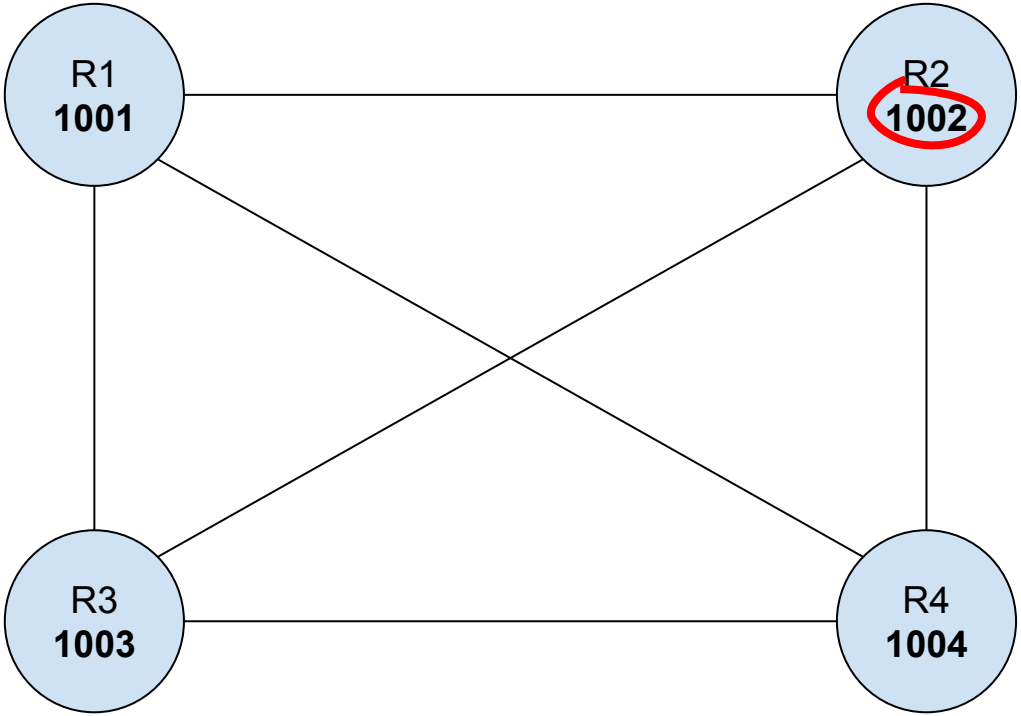
System and Network Engineering

SURF NET

Intro: MPLS

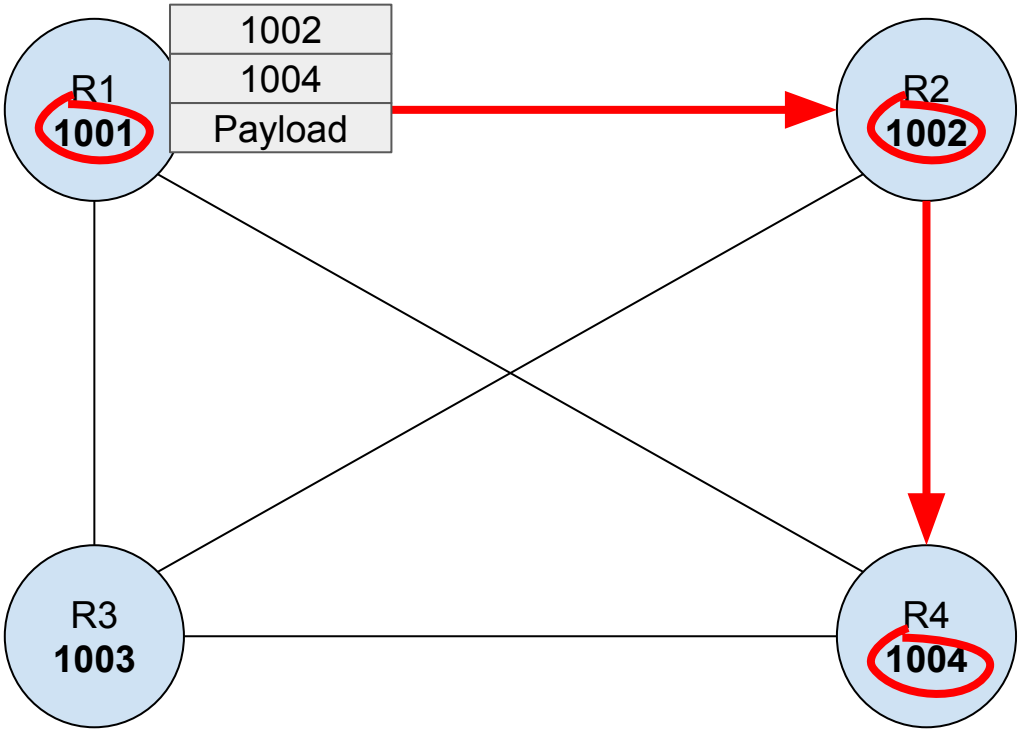

Original Packet
Ethernet Header
IP Header
Data
Ethernet Header
MPLS Label
IP Header
Data
New MPLS Label
Original Packet

Intro: SR-MPLS (1)



- Node-specific label (Node SID) known in the whole SR-domain
- The information about the labels and the topology is distributed by an IGP (OSPF/IS-IS)

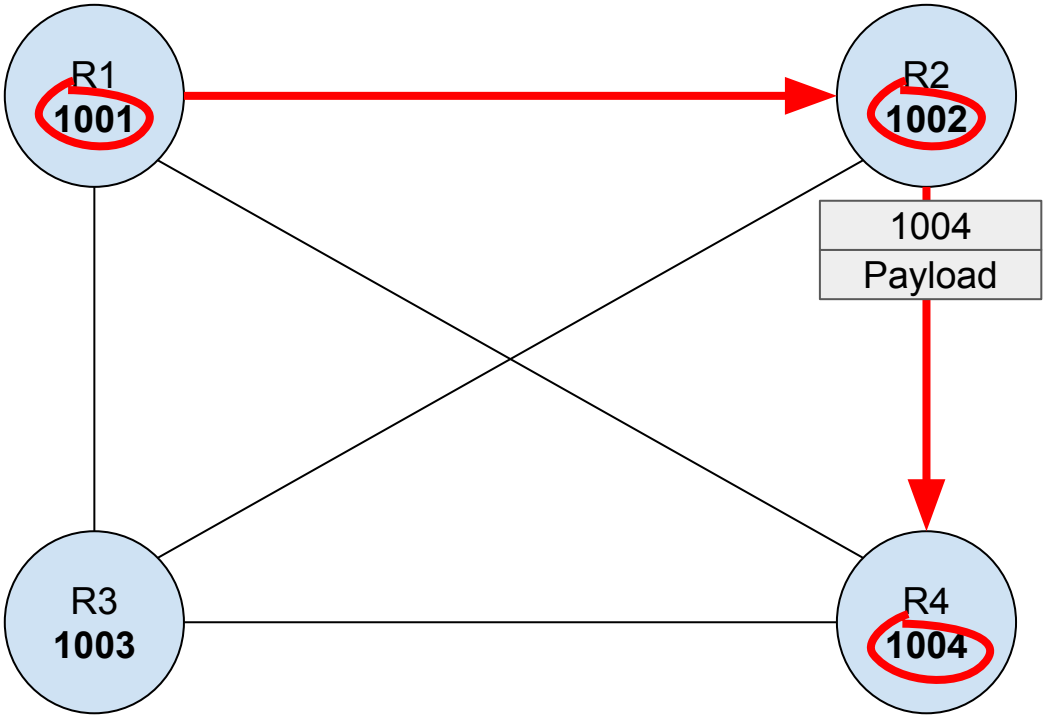
Intro: SR-MPLS (2)



- Sending node can specify the list of hops which will be traversed by a sent packet

*This example assumes disabled MPLS PHP

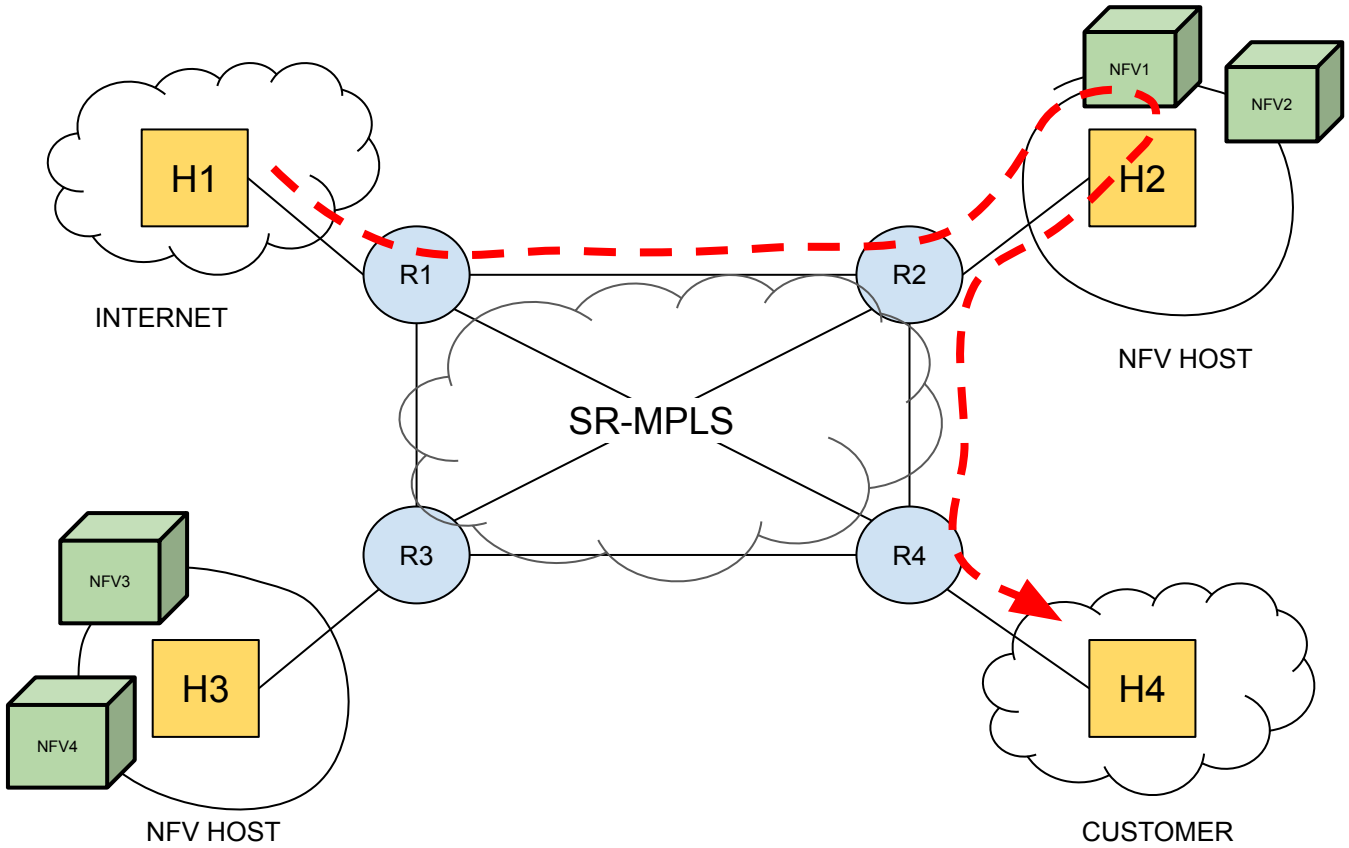
Intro: SR-MPLS (3)



- Sending node can specify the list of hops which will be traversed by a sent packet

*This example assumes disabled MPLS PHP

Per-customer NFV services



Research goals

Given the ISP SR-MPLS environment, how can per-customer services be provided?

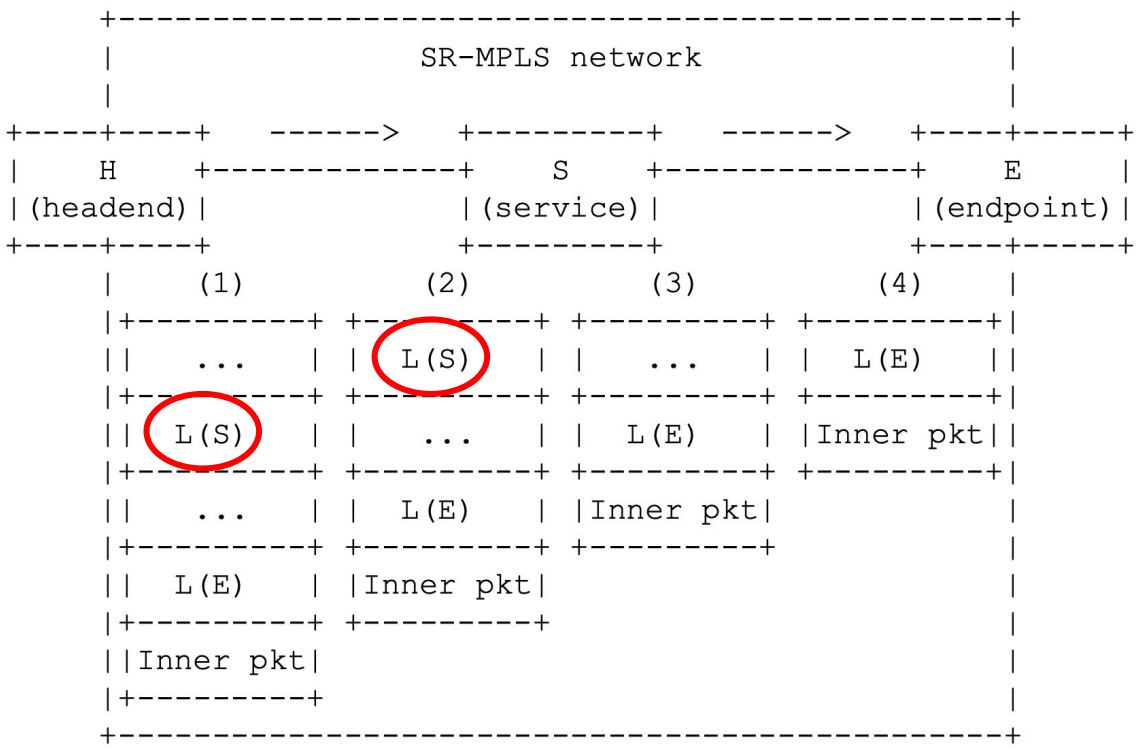
- Customer in charge of their own traffic
- Sinking-in selected traffic to a chosen service (NFV)
- Chaining the services together (NFV1 -> NFV2)

Services in SR network (1)

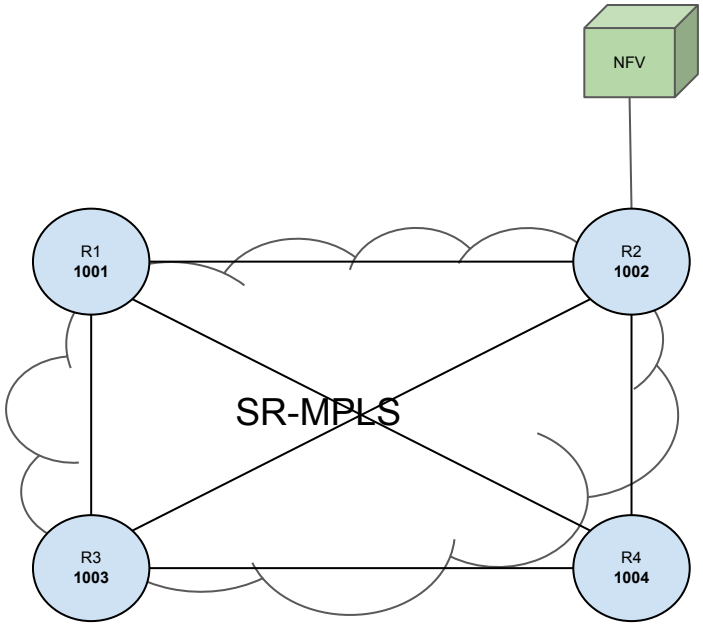
“Service Programming with Segment Routing” RFC draft (draft-xuclad-spring-sr-service-programming-02)

- Conceptualizes the idea of running the services in SR network
- SR-aware and SR-unaware services
- SR-MPLS data plane

Services in SR network (2)

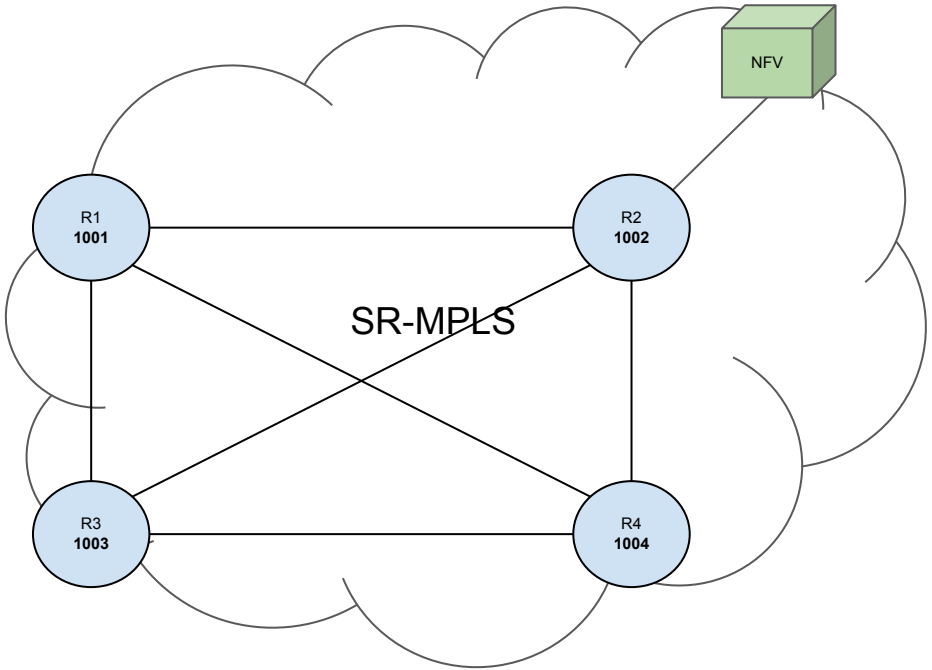


SR-unaware NFV



- NFV is just another endpoint placed behind a PE router
  - Does not use MPLS
  - Does not have a notion of MPLS topology

SR-aware NFV



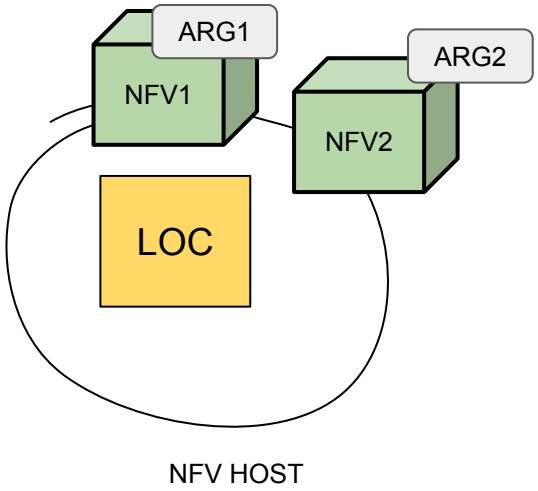
- NFV participates in SR-MPLS topology
  - MPLS
  - SR signaling (OSPF/IS-IS)

“Network Programming” concept (1)

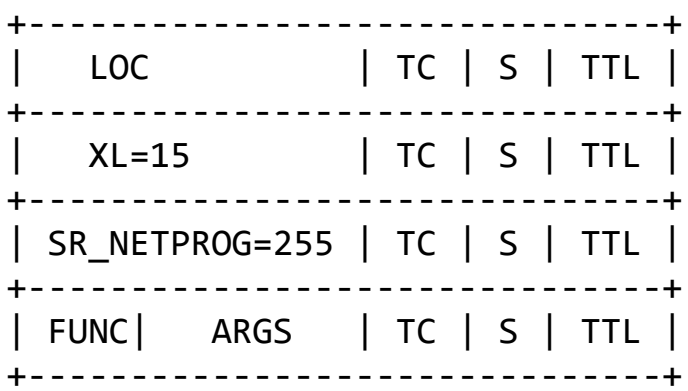
SRv6 Network Programming (draft-ietf-spring-srv6-network-programming-00) proposes the data-plane scheme for Network Programming using IPv6 protocol:

- Locator (LOC), variable-length n most significant bits. Should be routable.
- Function (FUNC), variable-length (32-bit is suggested)
- Function arguments (ARGS), variable-length, optional parameter providing an extra input to a specific function.

“Network Programming” concept (2)



“Network Programming” with MPLS



There was no NetProg dataplane representation defined for SR-MPLS

Our approach:

- LOC: 20-bits in the top label
- Define “Extended Special-Purpose” MPLS Label label (RFC 7274) for NetProg
  - FUNC: 4-bits
  - ARGS: 16-bits

Our work

1. Prototypes of SR-aware services
    a. Firewall
    b. Mirror
2. Dynamic SR-proxy
3. Created the virtual environment allowing the experimentation

Packet manipulation with eBPF

All prototyping done in eBPF

- eBPF has a capability of delivering higher packet processing rates
- Required putting extra effort (e.g. implementing own MPLS stack)

Firewall (1)

Inspired with “SERA: SEgment Routing Aware Firewall for Service Function Chaining scenarios” paper
http://netgroup.uniroma2.it/Stefano_Salsano/papers/18-ifip-sera-firewall-sfc.pdf

- SR-aware
- 5-tuple match
- Actions:
  - BASIC: drop/accept
  - MPLS: push MPLS header
  - NetProg: pushing LOC, FUNC, ARGS headers

Firewall (2)

Examples:

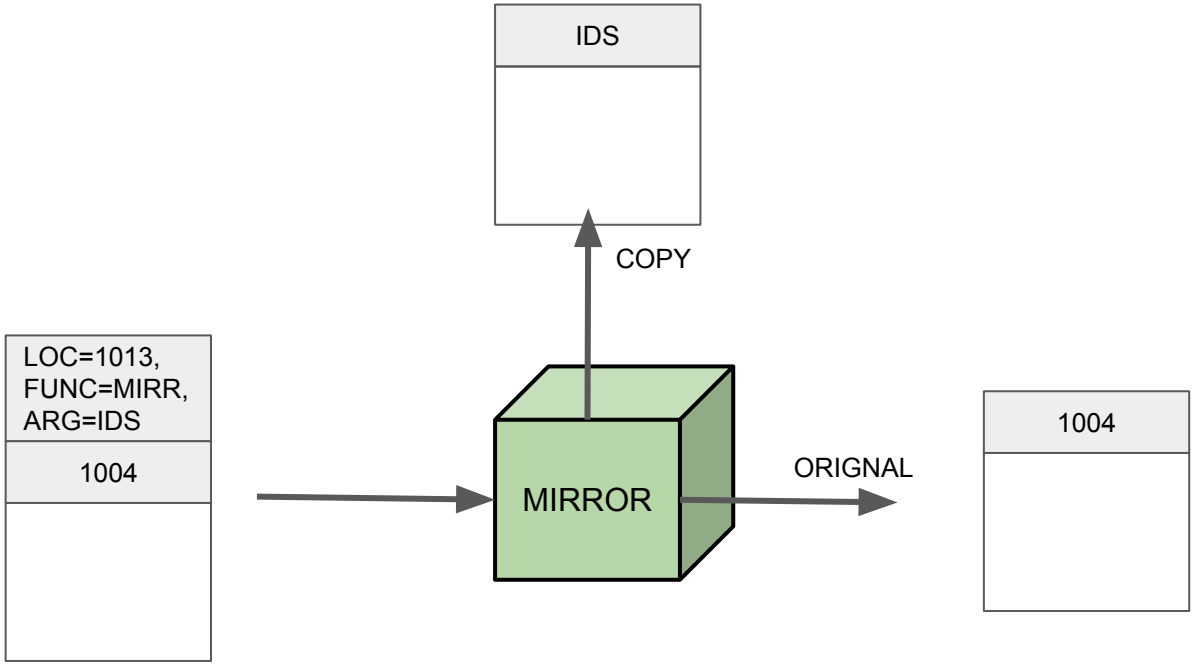
- match TCP dport 80, action: accept
- match TCP dport 8080, action: push 1013 label
- match UDP dport 666, action:
    - push LOC=1013, FUNC=MIRROR, ARGS=IDS NetProg labels

Mirror (1)

- SR-aware
- No configuration/stateless, behaviour is inferred from NetProg headers
- Mirrors received packets, the copy is sent to another function (specified as ARGS of received packet)

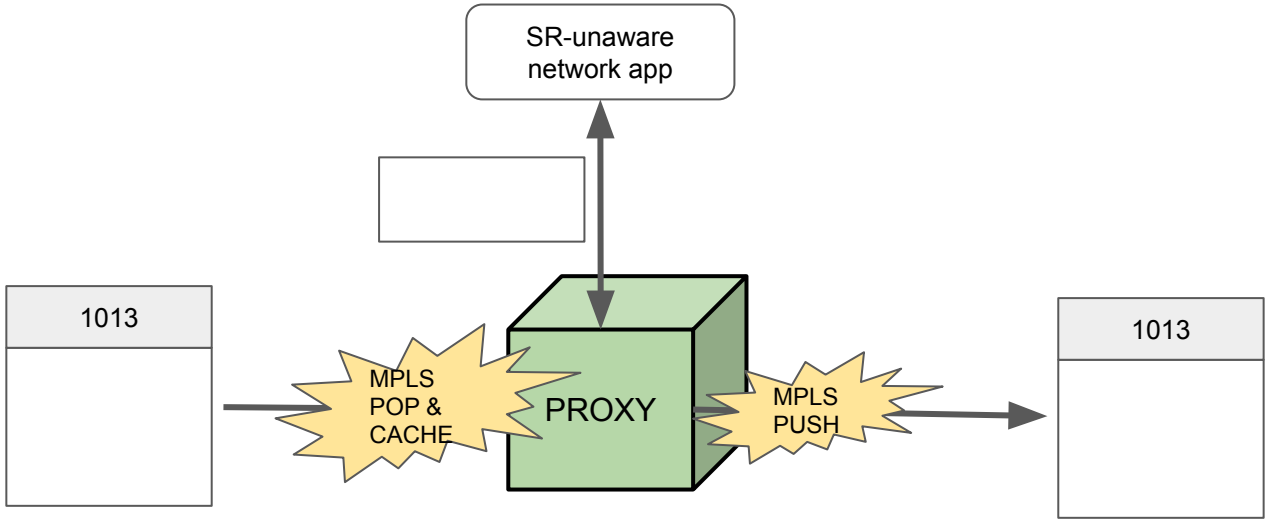
Mirror (2)

- Example: Mirror the packet and send it to an IDS

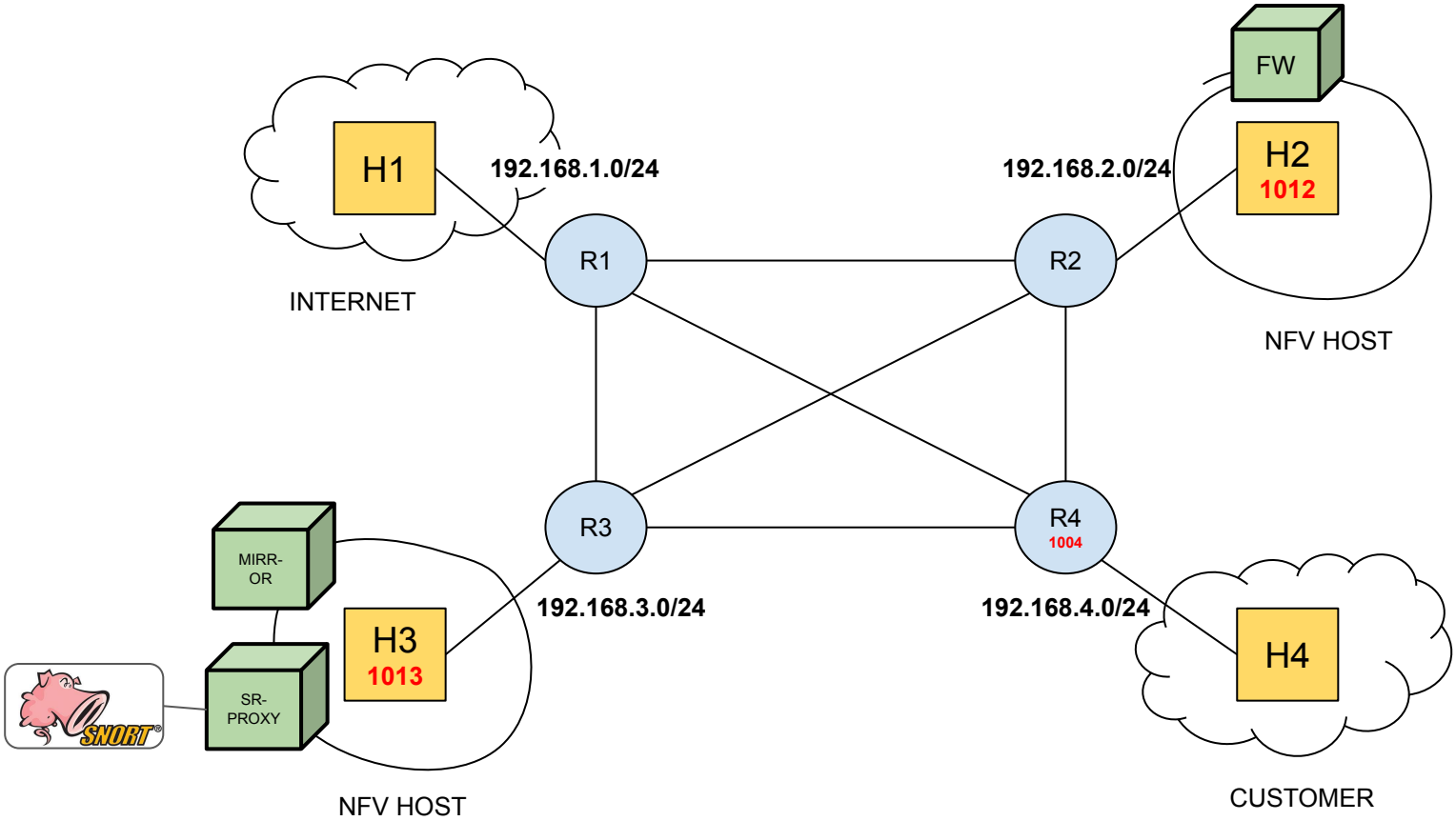


SR-proxy

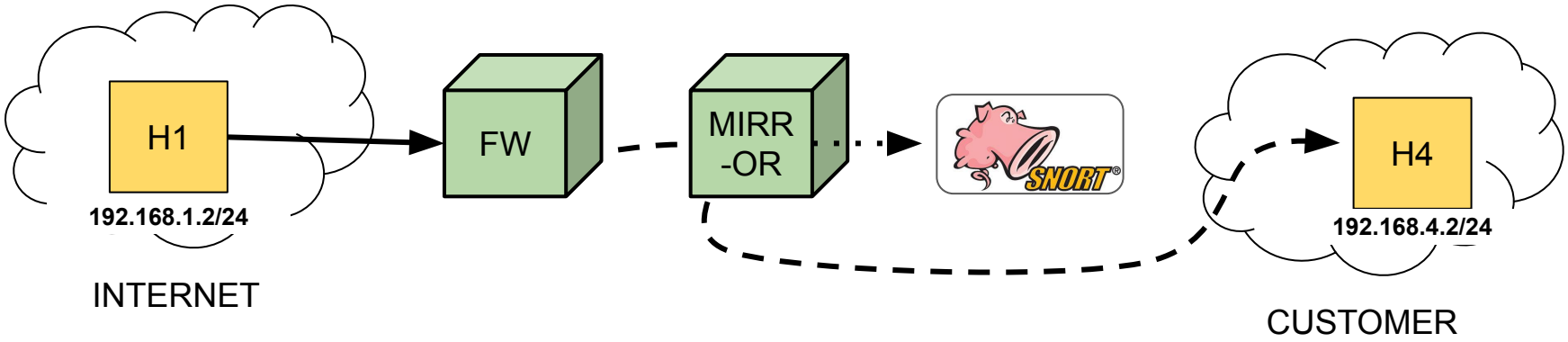
- Required to enable the use of SR-unaware services
- Implemented MPLS dynamic SR-proxy



DEMO environment



DEMO scenario



Future work

- Evaluate load-balancing and HA scenarios for NFVs
- Performance testing
- Use a network controller to deploy SR policies

Resources

https://bitbucket.org/uva-sne/ron19_sr

(Not public, need to be a member of uva-sne on bitbucket)

Name	Size	Last commit	Message
docs		2019-11-04	Improve docs
lab → vqfx10k-vagrant [89d5e884a41b]	40 B		
nfv [eb047e5934eb]	40 B		
.gitignore	39 B	2019-10-21	add gitignore
.gitmodules	181 B	2019-10-16	Initial commit