Security Level:

Building a Better Connected World,
ROADS to a Better Future

Transport Independent Service Function Chaining (SFC)

Jim Guichard
James.n.guichard@huawei.com

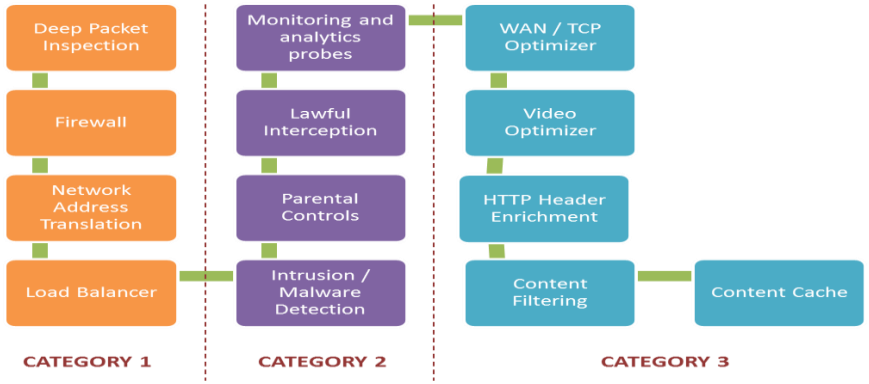
MPLS+SDN
+NFVWORLD
@PARIS2018

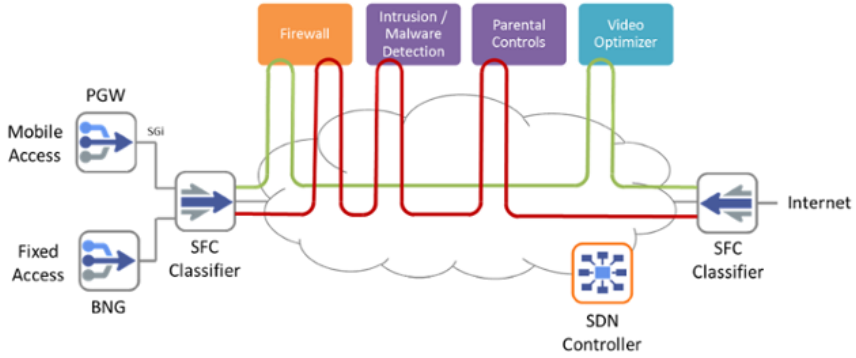
Agenda

- Emerging Requirements for an SFC Architecture
- An Overview of the SFC Architecture
- Transport Agnostic SFC – NSH-based Service Chaining
- Segment Routing Family Recap
- Leveraging SR for SFC
- Integration of NSH-based and SR-based Service Chaining
- Summary

Emerging Requirements for an SFC Architecture

- Distributed and virtualized network services driven by application requirements for performance, flexibility, programmability, and emerging new service types
  - Network services are driven down to be near to the users for optimal performance
  - Central offices are re-architected as data centers hosting virtualized services
  - Mobile, fixed, edge computing, and enterprise networks all require SFC to support the services users have come to require; and these services are end-to-end

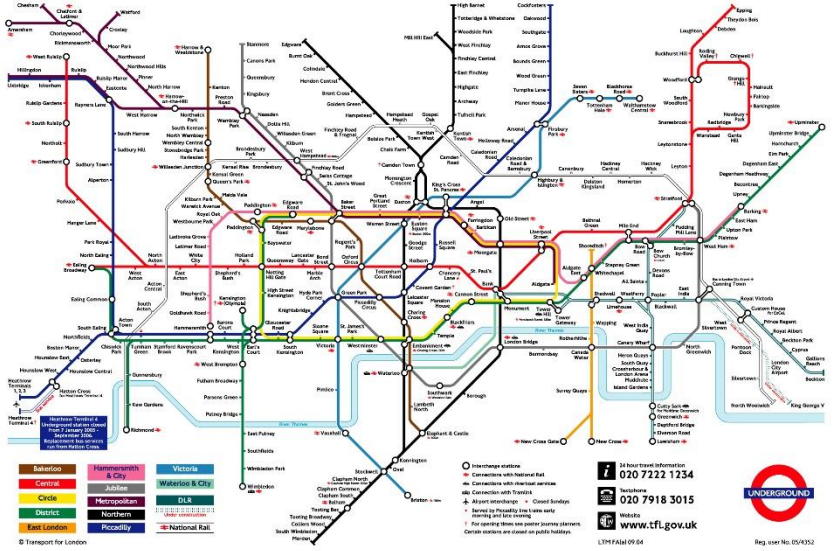


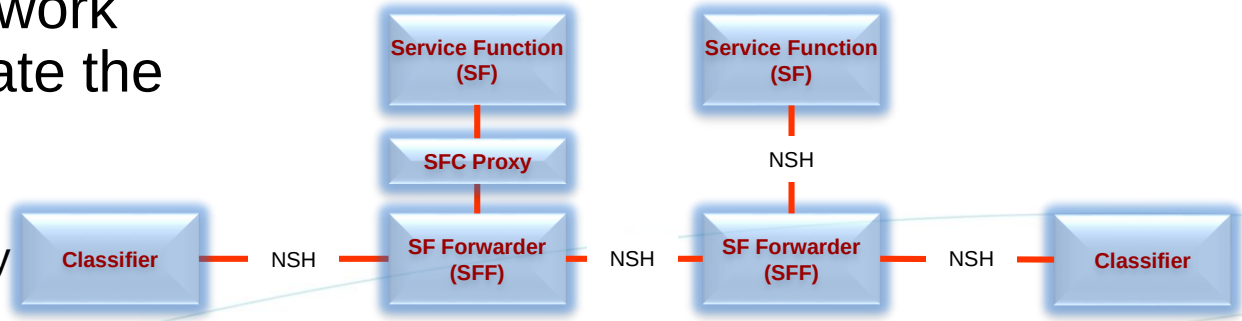


- Key point: service instantiation is end-to-end and must be flexible enough to deploy no matter the underlay/overlay – this trend will accelerate for the reasons stated above

An Overview of the SFC Architecture

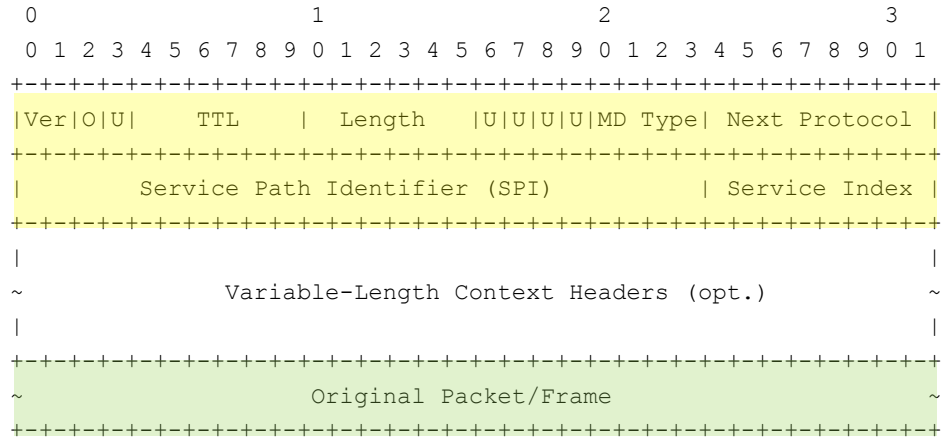
- SFC architecture defined in [RFC7665] – standards track RFC from the IETF
- Architecture consists of multiple network elements designed to create an end-to-end service plane
- Service plane has two major components:
  - Path information which is akin to a subway map: it tells the packets where to go without requiring per flow configuration
  - Metadata is information about the packets and can be used for policy
- Traffic is classified into a service chain and forwarded through SFFs and SFs
- New data plane encapsulation called Network Service Header (NSH) is used to instantiate the service plane
  - Intermediate nodes do not need to be NSH aware
  - Non-NSH enabled service supported through proxy



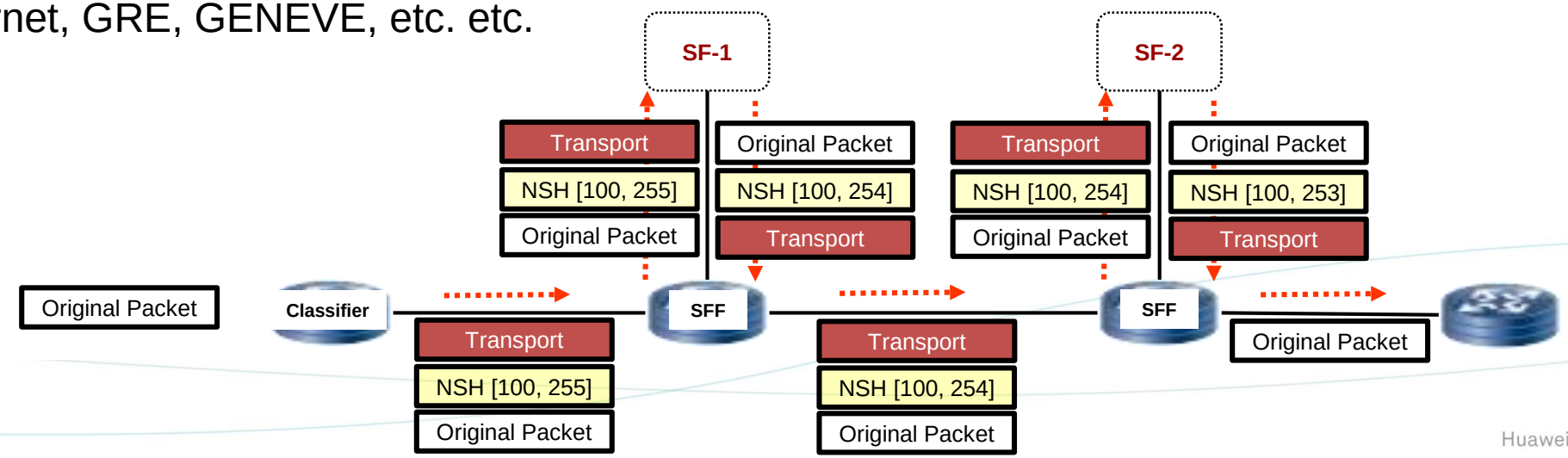


Transport Agnostic SFC – NSH-based Service Chaining

- Network Service Header (NSH) is a data plane header inserted between the original payload and the outer transport
- It's primary purpose is to provide a service plane which carries path identification
  - Service Path ID & Service Index
  - Fixed or variable length metadata
- It's independent of the transport
  - Can be carried in MPLS, SR, UDP/IP, Native Ethernet, GRE, GENEVE, etc. etc.



https://datatracker.ietf.org/doc/rfc8300/

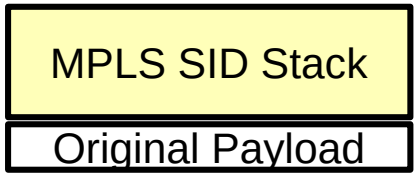


Is there an alternative to NSH-based Service Chaining?

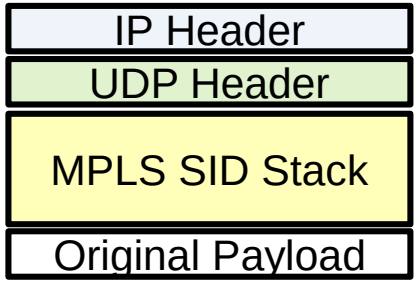
- Some alternative solutions have been proposed to use segment routing for service chaining
- Segment routing provides the ability to define a path using source routing techniques
  - With either an MPLS or IPv6 data plane
  - Segments of that path may be service functions and therefore in theory a service chain can be established
- Lets take a look at the details ..

Segment Routing – a quick recap

SR-MPLS Family

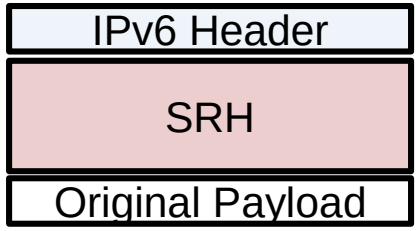


- MPLS encapsulation
- MPLS label stack carries the network instructions (SIDs)



- IPv4 or IPv6 encapsulation
- MPLS label stack carries the network instructions (SIDs)

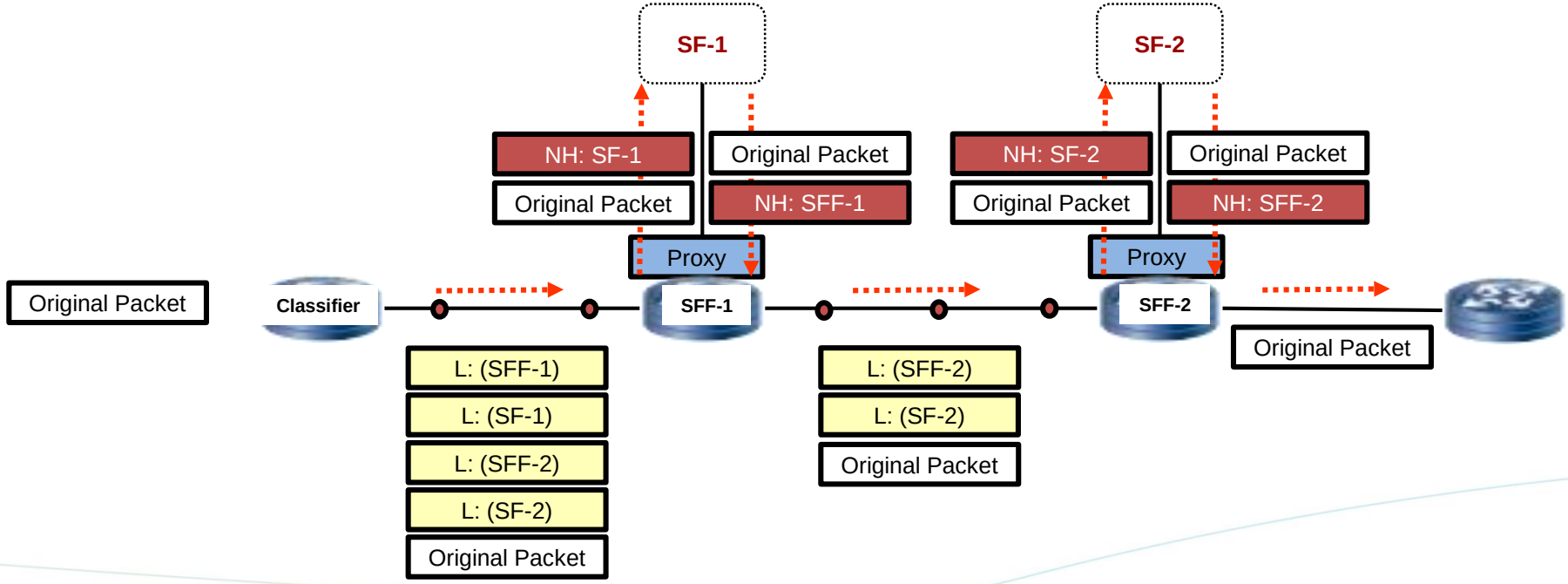
SRv6



- IPv6 encapsulation
- Segment Routing Header (SRH) carries the network instructions (SIDs)

SR-MPLS/MPLS Service Chaining

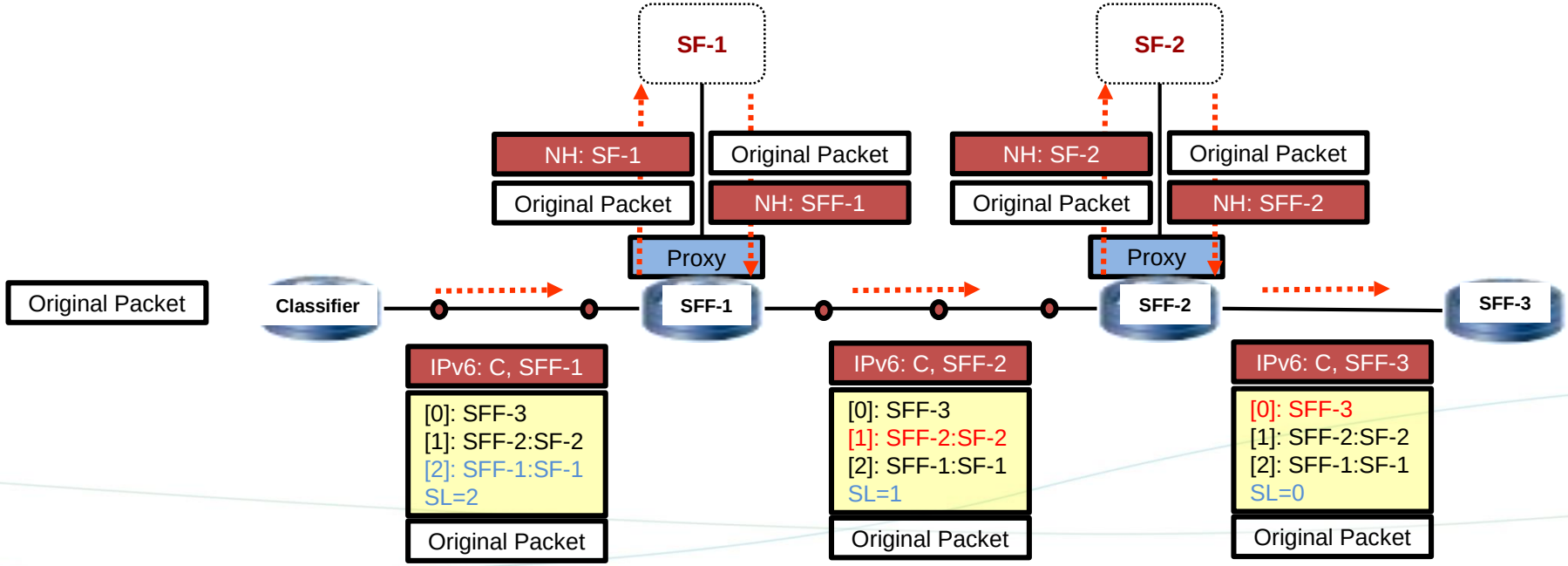
- SR-MPLS provides a list of SIDs by way of an MPLS label stack
- Each SFF and SF is allocated an MPLS label value at the SFF
  - Theoretically an SF could provide direct MPLS-SR functionality but without that an SR proxy is required at the SFF
- MPLS label stack stripped at the SFF and reapplied based upon inbound interface



SRv6 Service Chaining

- SRv6 provides a SID list by way of an SRH (Segment Routing Header) and each SFF/SF is allocated a SID
- An SF can be either SR-aware or SR-unaware; SR-unaware requires an SR proxy at the SFF
  - The SR proxy strips (and caches) the SRH when sending packets to an SF and puts the SRH back when sending packets between SFFs
  - The SR proxy needs to correlate a packet to an SRH on return from an SF (based upon interface ID)

https://tools.ietf.org/html/draft-xuclad-spring-sr-service-chaining-00



Can we combine these technologies to satisfy market demands?

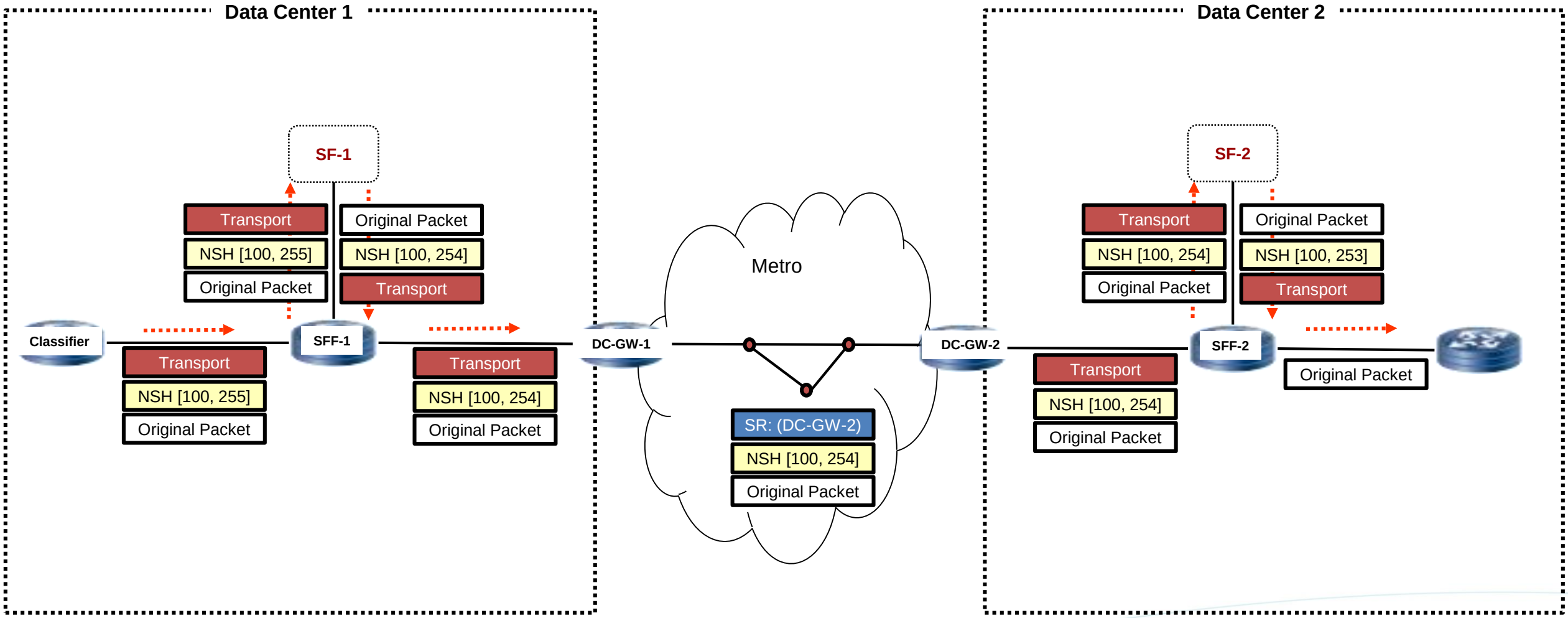
… lets take a closer look

Leverage SR to realize SFC

- SR-MPLS service chaining has some obvious missing pieces and drawbacks
- SRv6 service chaining to be fully useful requires SRv6-aware SFs – this will take time
- Therefore can we leverage both technologies to address immediate market needs? Answer: yes we can!
- NSH-based SFC with SR-based transport tunnel:
  - in this scenario segment routing provides a transport tunnel between SFFs of an NSH-based SFC
- SR-based SFC with integrated NSH service plane:
  - in this scenario each service hop of the SFC is represented as a segment of the SR segment-list. SR is responsible for steering traffic through the necessary SFFs as part of the segment routing path and NSH is responsible for maintaining the service plane, and holding the SFC instance context and associated metadata

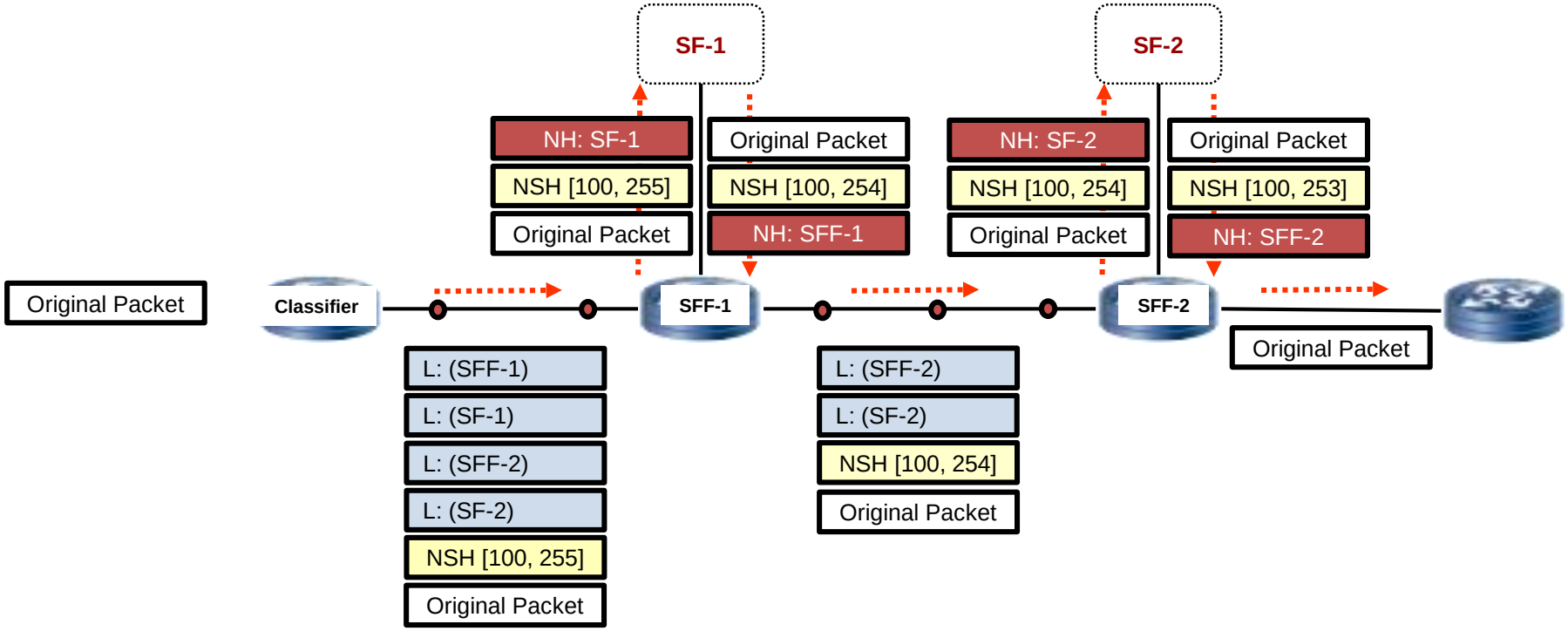
https://datatracker.ietf.org/doc/draft-guichard-sfc-nsh-sr

NSH-based SFC with SR as the Transport Tunnel



Totally decouple service & transport layers; leverage SR for traffic steering capabilities

SR-based SFC with Integrated NSH Service Plane



Still transport agnostic, keep all the merits of NSH and SR but maintain less state at SFFs

Conclusion

- NSH-based service chaining and segment routing are complimentary technologies
- SR based SFC has several options, each has its own pros and cons
- NSH is designed to be transport agnostic, NSH based SFC is more and more accepted by the industry
- Segment Routing(SR) provides an efficient way for steering traffic without requiring intermediate nodes to maintain per-flow state
- By combining SR and NSH, a transport-independent SFC can be realized
- A solution that keeps merits of both NSH and SR is attractive

Thank You.