

# Network Service Header (NSH)

draft-quinn-sfc-nsh

IETF 90

**P. Quinn**, et. al

*Cisco Systems*

P. Agarwal

R. Manur

*Broadcom*

Pankaj Garg

*Microsoft*

A. Chauhan

*Citrix*

U. Elzur

*Intel*

B. McConnell

*Rackspace*

C. Wright

*Red Hat Inc.*

# NSH Overview

- Describes a dataplane header used to carry information along a service path.
  - Identifier for service path selection
  - Opaque mandatory metadata fields
  - Optional TLVs
- Creates “service plane”
  - Transport independent (NSH in VXLAN, NSH in MPLS, NSH in UDP, etc.)
  - Service layer OAM

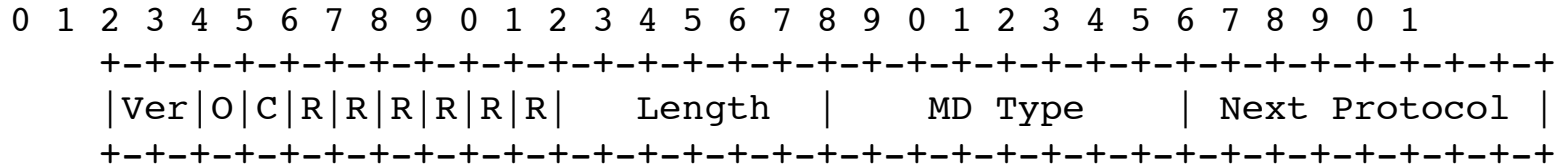
# Changes from -02

- New co-author
- Base header is first 4 bytes, includes type field
  - Encapsulated protocol type → 8 bit value
  - Explicit dataplane versioning
  - Critical TLV indicator
- 4 byte service path header
- Added optional metadata TLV (in addition to mandatory fixed context header)
  - TLV Class

# Implementation Update

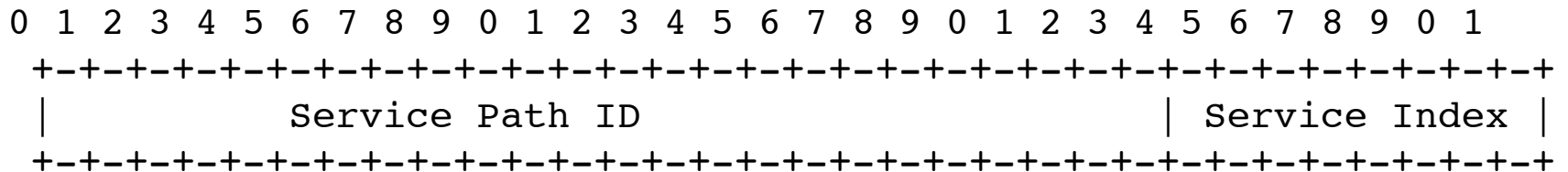
- Opensource implementations
  - OVS dataplane (with VXLAN)
  - OpenDaylight control plane (+ LISP)
- Several vendor specific implementations
- Early deployments underway

# Base Header



- 8 bit Next Protocol: support non-ET protocols + reclaim space
- MD type indicates format of header. NSH type = 0x1
- Critical TLV present

# Service Path Header



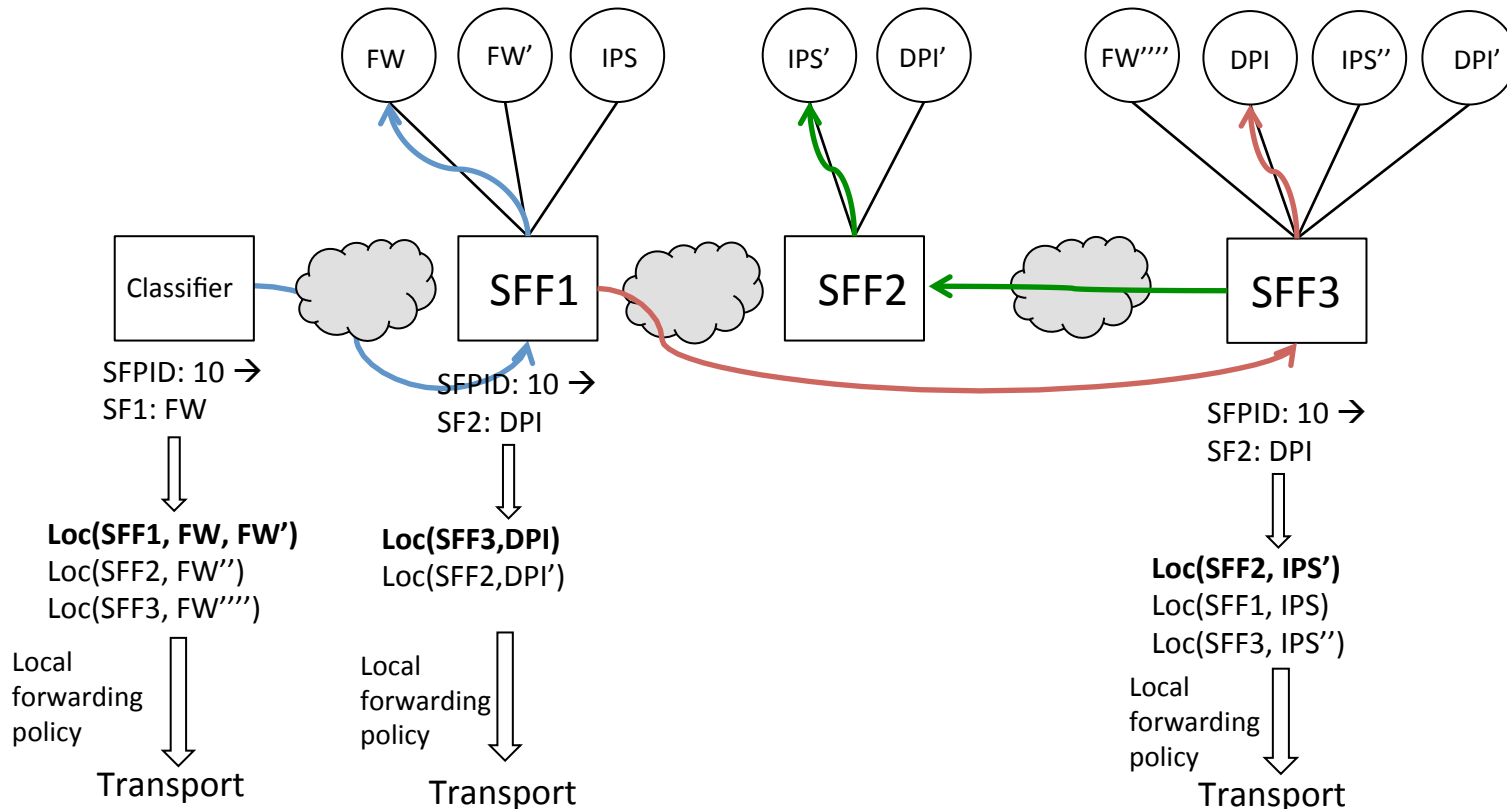
- Represents the rendering of the chain policy
- Simple identifier: does **not** imply a static, explicit path
  - Resolved locally
- Can be changed: branching within a service graph
  - Re-classification (and therefore policy) decision
- Index conveys node within the graph

# Chain and Paths

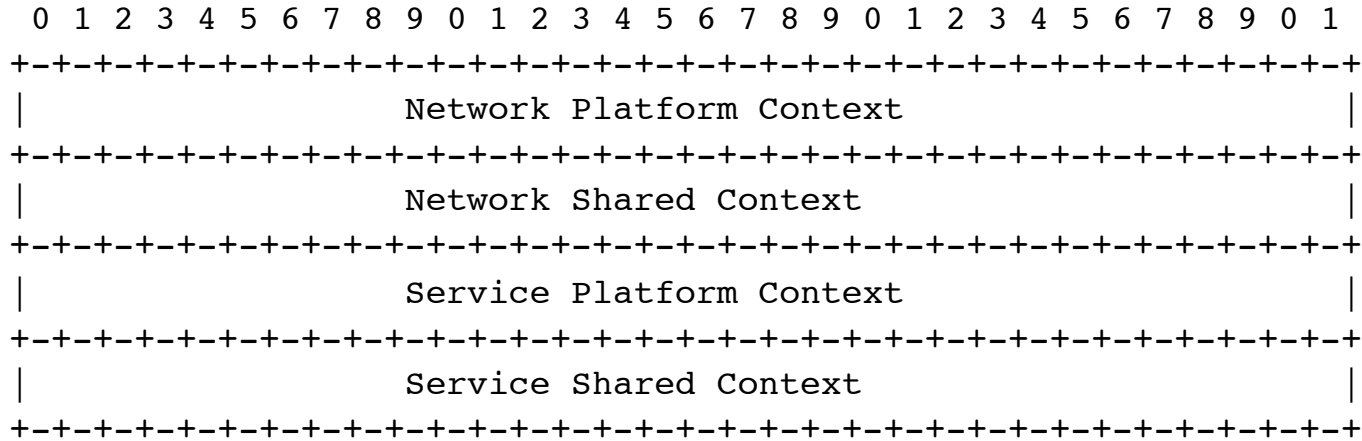
(no load distribution)

Chain1: Firewall → DPI → IPS

Chain1 is rendered as SFPID = 10



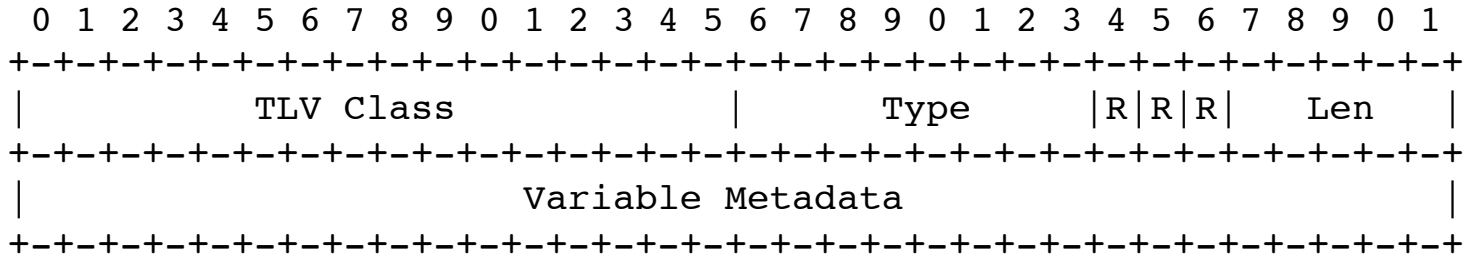
# Mandatory Context Headers



- Based on initial deployments: many use cases satisfied with fixed size context headers
- Hardware friendly: easy to parse and skip at high speed
- Opaque, significance allocated via control plane



# Optional TLV



- TLV Class: describes the scope of the type field
- Type: type of metadata carried, includes critical indication

# Next Steps

- Continue development
  - Opensource and vendors
- Continued deployments
- Ask for adoption as a working group document:  
SFC encapsulation

`“Generic SFC Encapsulation: This document will describe a  
single service-level data plane encapsulation format that:  
- indicates the sequence of service functions that make up the  
Service Function Chain  
- specifies the Service Function Path,  
- communicates context information between nodes that  
implement  
service functions and Service Function Chains...”`