Service Function Chaining: Virtualizing and Programming Your Data and Service Planes

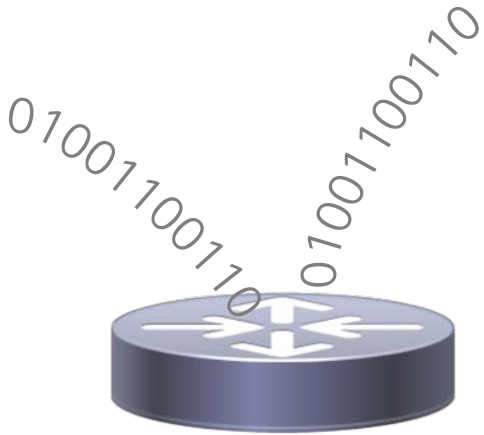
Joe Clarke
Distinguished Engineer, Services
April 18, 2018

Abstract

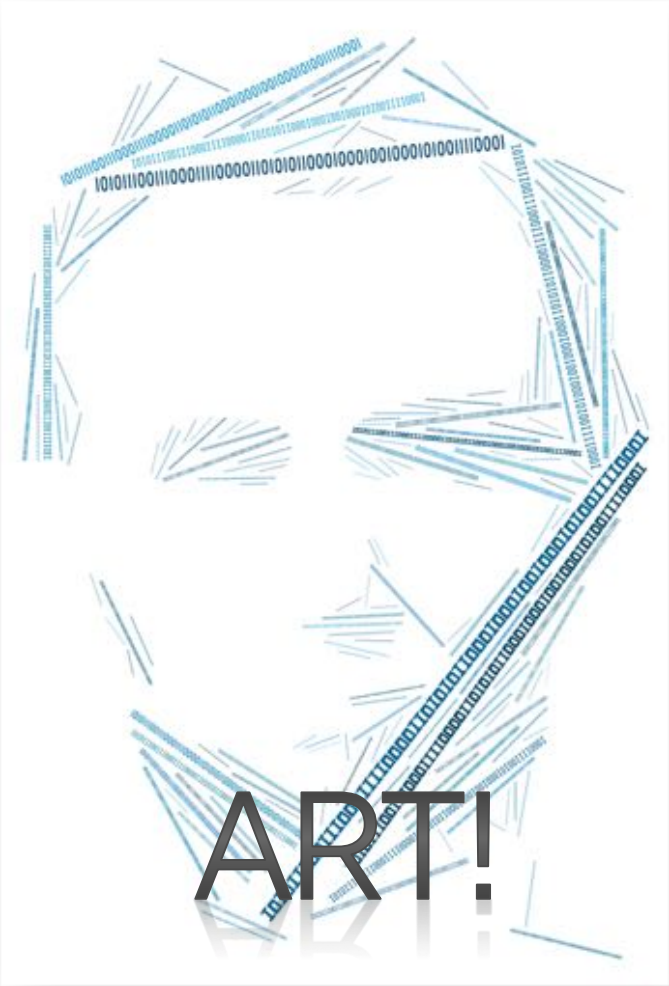
Software Defined Networking (SDN) combined with Network Function Virtualization (NFV) provide unprecedented programmatic access to drive networks in new and exciting ways. These capabilities lead to dynamic service creation, with Service Function Chaining (SFC) completing the programmable picture, creating a "whole stack" view. Service chaining realizes network virtualization, allowing network services to exchange metadata and be seamlessly deployed into the packet path independent of their physical location. Service chaining, and its underlying standards such as Network Service Header (NSH) offer a innovative way to "software define" packet processing. Complementing these technologies is the Open Source Vector Packet Processing (VPP) that enables one to add new data path services with tremendous performance.

This webinar will highlight Service Function Chaining and Network Service Header and describe how they enable Network Function Virtualization. Examples of NFV will be shown, as well as how SFC can be used to program ones network at the service and data planes.

Why Play With Your Traffic?



00110101010101



Why Play With Your Traffic

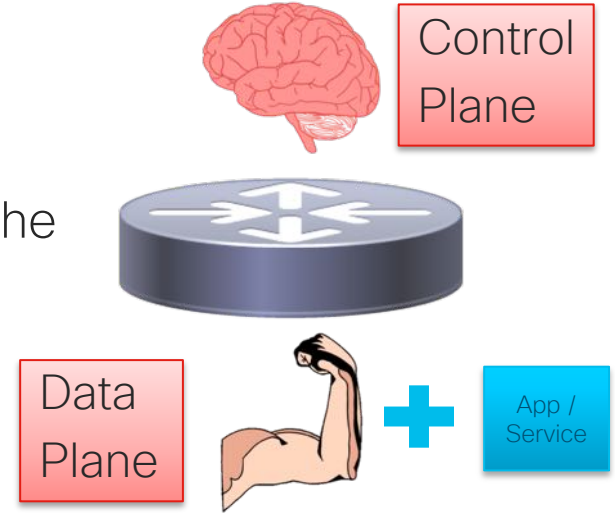
Software Defined Packet Control

Agenda

- Introduction to Data Path Programming
- Intro to Service Chaining (SFC), and Network Service Header (NSH)
- Examples
- Key Takeaways

Q&A

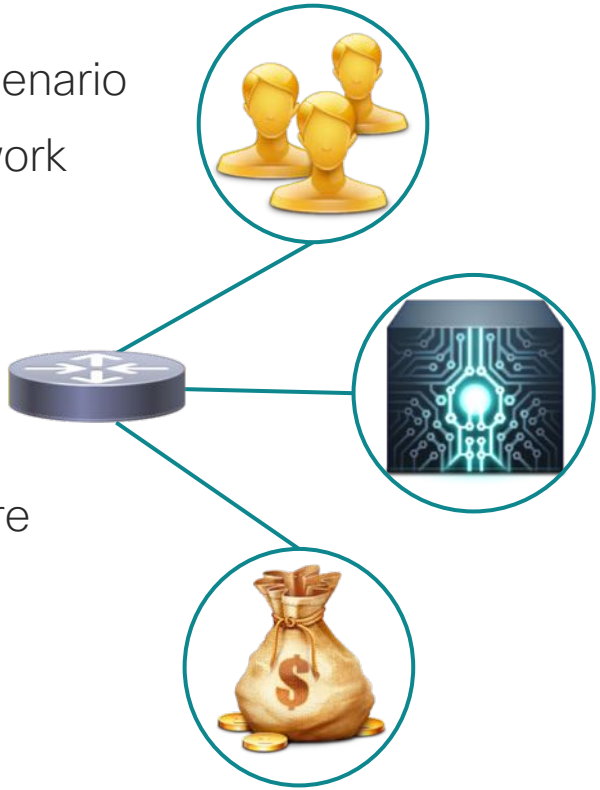
What Is Data Path Programming?



- Device programmability has typically focused on the control plane of the device
  - CLI
  - SNMP
  - Embedded Event Manager
  - NETCONF
- Data path programmability provides fine-grain control over the packets passing through the network device
- Can come in various forms
  - Instructing hardware how to forward packets (e.g., OpenFlow)
  - Attaching applications to the packet path of the device
  - Operations, Administration, and Maintenance

Why Is This Data Path Control Interesting?

- Vendors do not always design for every use case and scenario
- Data path programming fills the gap by opening the network to all levels of customization
- Make the network more relevant to your business
  - Create specific optimizations
  - Deliver unique, customized traffic behavior
  - Build new network services
- Enable new network architectures using existing hardware
  - Cloud applications
  - Network Function Virtualization
  - Network visualizations and analytics



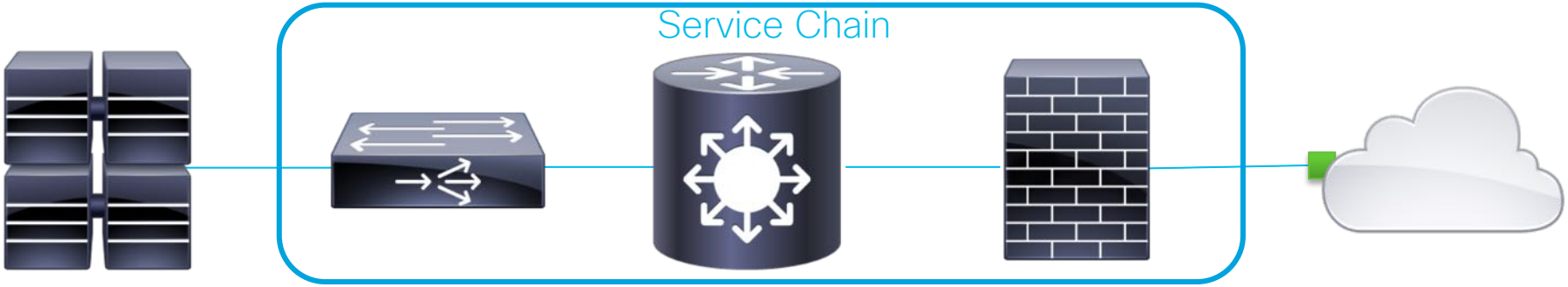
Service Function Chaining (SFC) & Network Service Header (NSH)

What Is A Service?

- An application?
  - Web portal for managing your {internet, bank, wireless} account
  - A CRM system
  - A productivity suite
- A collection of resources?
  - Database
  - Compute
  - Storage
  - Bandwidth
- A transit packet processor?
  - Firewall
  - Load balancer
  - NAT gateway
  - Video Transcoder

Chaining Services Together

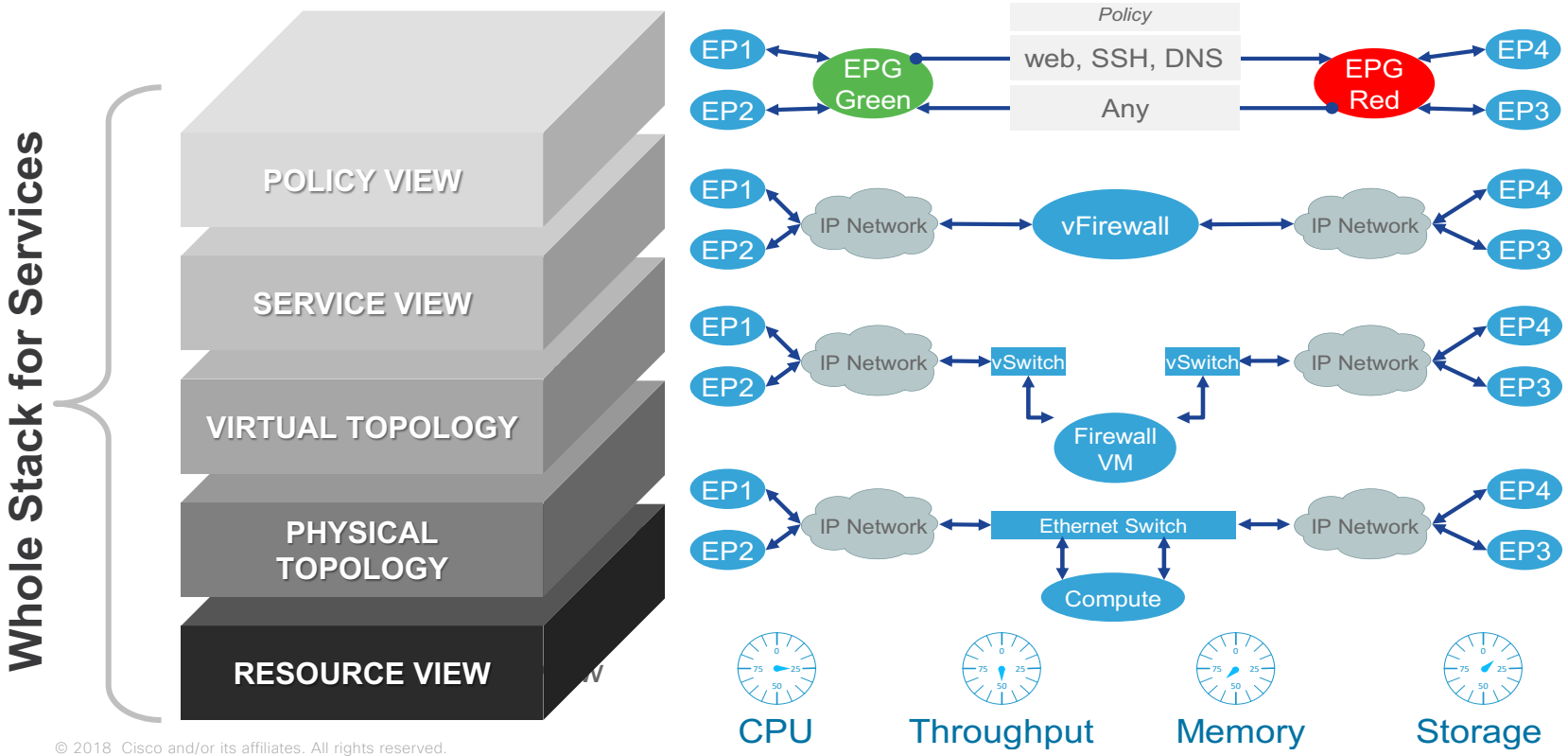
- A Service Chain is an ordered graph of specific network service functions through which packets must flow
- E.g.: Firewall → NAT → Load Balancer



Service Chaining Today

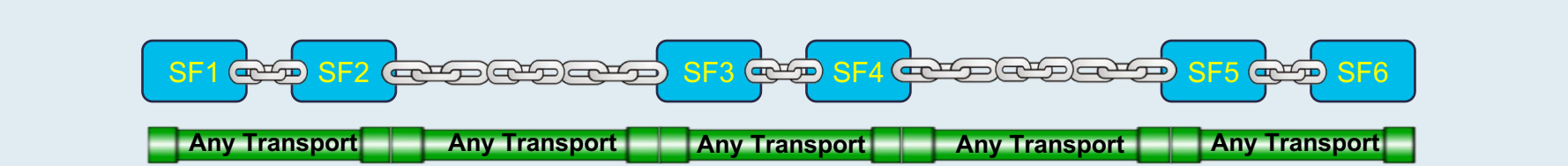
- Services are built using rudimentary service chaining techniques; accomplished via hop-by-hop switching/routing changes
  - Dependencies: Leading to a dependencies between the network & service topologies
  - Very complex: VLAN-stitching, Policy Based Routing (PBR), Routing tricks, etc.
  - Static: no dynamic, horizontal or vertical scaling, and requires network changes
- Service functions are deployed based on physical network topology and physical network elements
  - Changes to service functions or ordering within a service chain require network changes
    - Example: Firewalls typically require “in” & “out” layer-2 segments; adding new FW means adding new layer-2 segments to the network topology
  - Inhibits optimal use of service resources; limits scale, capacity & redundancy

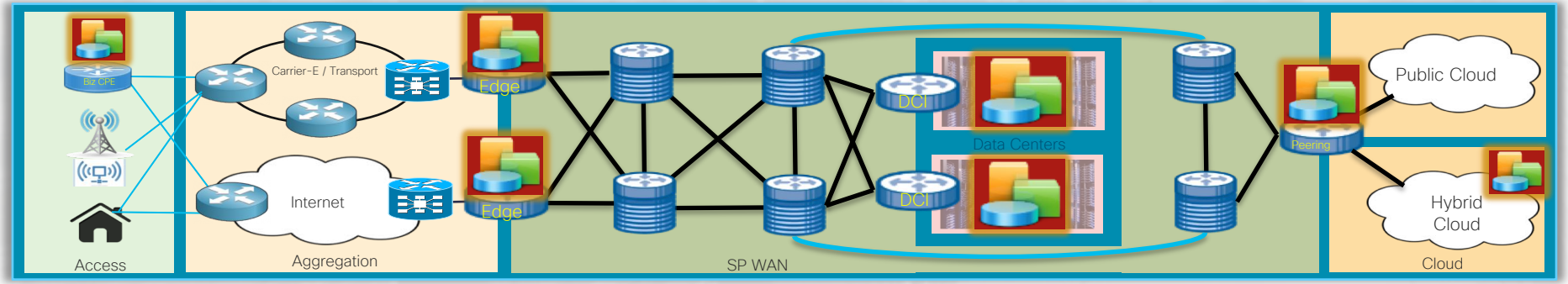
SFC – Creating a Services-layer Ecosystem Explosion



The “Big Idea”

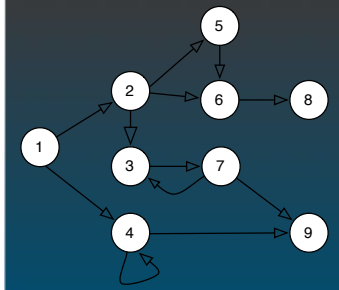
“With a common Service Plane, service policies and context are end-to-end, and can leverage any transport.”





The “Big Idea”

“With a common Service Plane, service policies and context are end-to-end, and can leverage any transport.”



Service Graphs

Service creation through graphs vs. linear Chains

- Support for all graph topologies
- Collection of SFs form a graph

Rich Metadata

Policy distribution b/w Service Functions through metadata

- Shares network and service context
- Ingress (e.g., VRF), classification/DPI, user/service context (e.g., Subscriber) consumed by services
- Includes application-ID

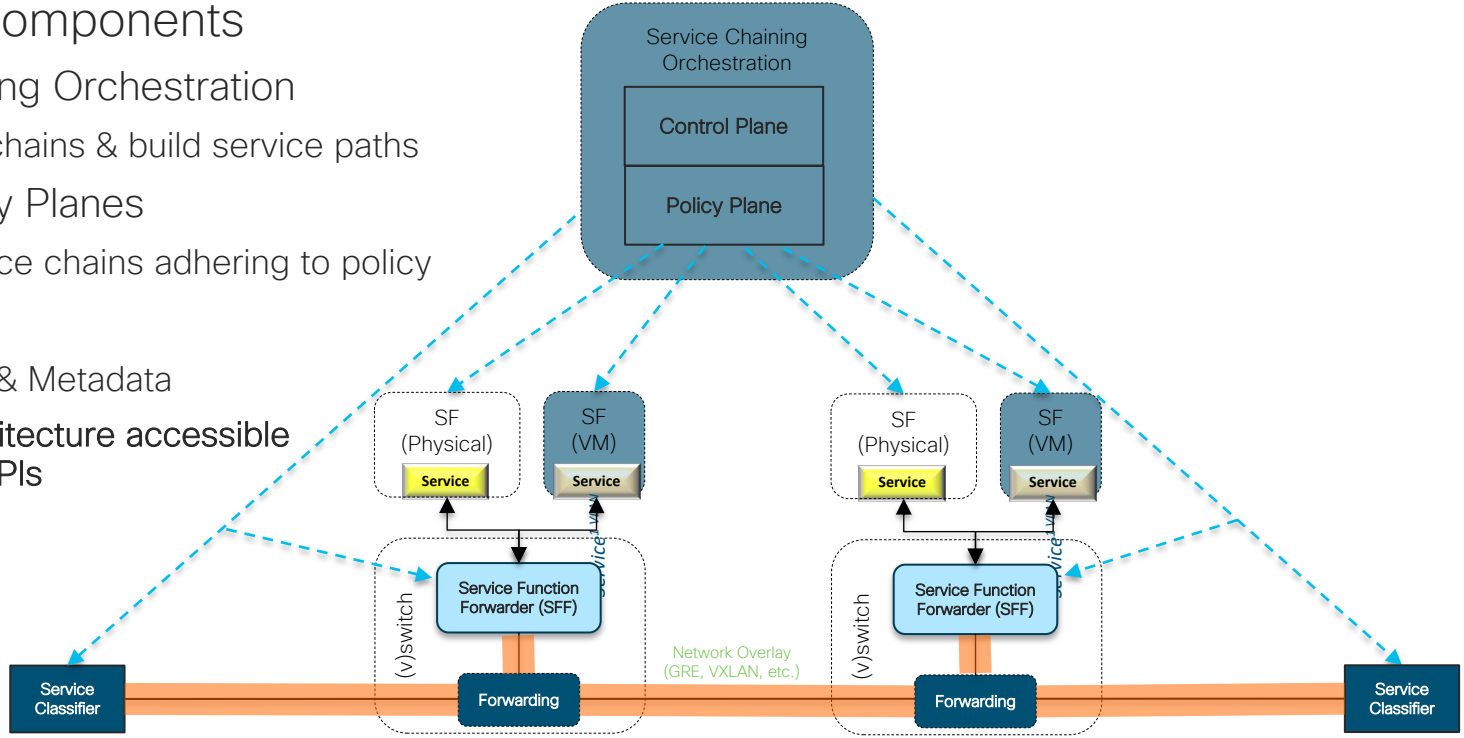
End-to-End Service Path and Policy

NSH

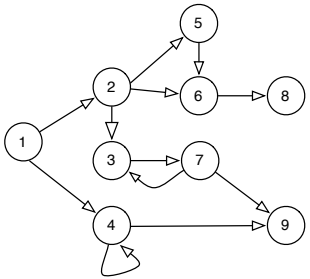
- Single FIB disposition based on Service Path Index (SPI) Field
- Tunnel encapsulation choices; (VXLAN-GPE, GRE, MPLS, etc.)
- NETCONF, YANG, GBP
- OAM and Service Innovation

Service Function Chaining Architecture
High-level Component Structure

- Architecture components
  - Service Chaining Orchestration
    - Define service chains & build service paths
  - Control / Policy Planes
    - Instantiate service chains adhering to policy
  - Data Plane
    - Traffic steering & Metadata
    - Data plane architecture accessible through open APIs



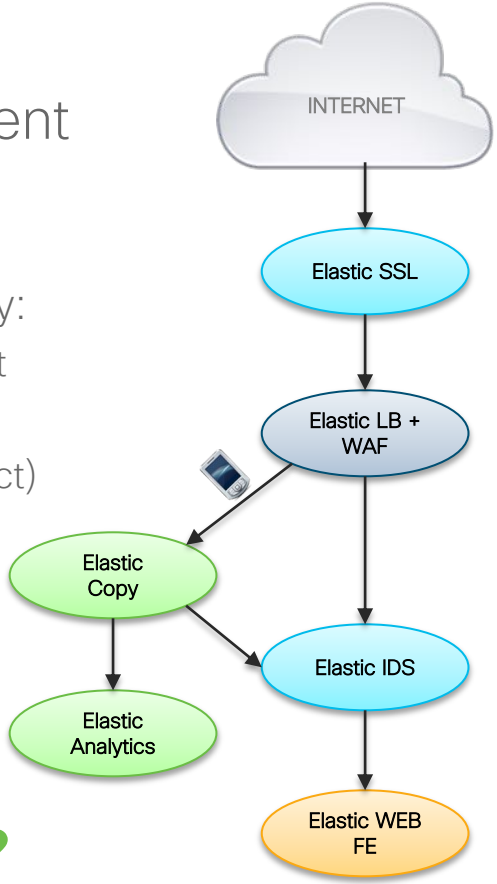
Service Function Chaining
Terminology



- Service Classifier
  - Determines which traffic requires service and forms the logical start of a service path.
- Service Function (SF)
  - Component used to provide some level of processing to received packets.
- Service Function Forwarder (SFF)
  - Responsible for delivering traffic received from the network to one or more connected service functions according to information carried in the network service header as well as handling traffic coming back from the SF.
- Service Path
  - A service path is the actual forwarding path used to realize a service chain
  - Think of service chain as an abstract “intent for packets”; the service path is the actual instantiation of the chain in the network.
  - Service Chain: Firewall → NAT → Load Balancer; Service Path: svc-fw1 → svc-nat44 → svc-iosslb

Evolving Service Chaining
Example: Business Policy Drives Service Deployment

- A service is rendered based on a business policy like …
- To all traffic between the Internet & web front end servers apply:
  - De/Encryption with highest throughput / low latency and least $$ cost
  - Copy all “mobile” only transactions to a Big Data analytics system
  - Perform the copy at most optimal point ($$ cost & least latency impact)
  - Send all traffic through a load balancer/web application firewall and IDS
- Additionally, deploy this policy with other features like:
  - Service functions are both virtual and physical **and** vendor neutral
  - Compute & service elasticity; compute mobility



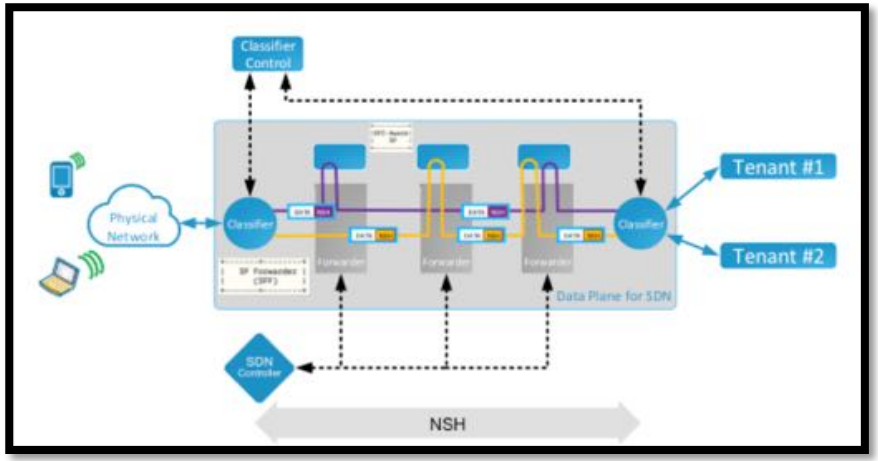
But how do we steer the traffic through this path?

Network Service Header 101

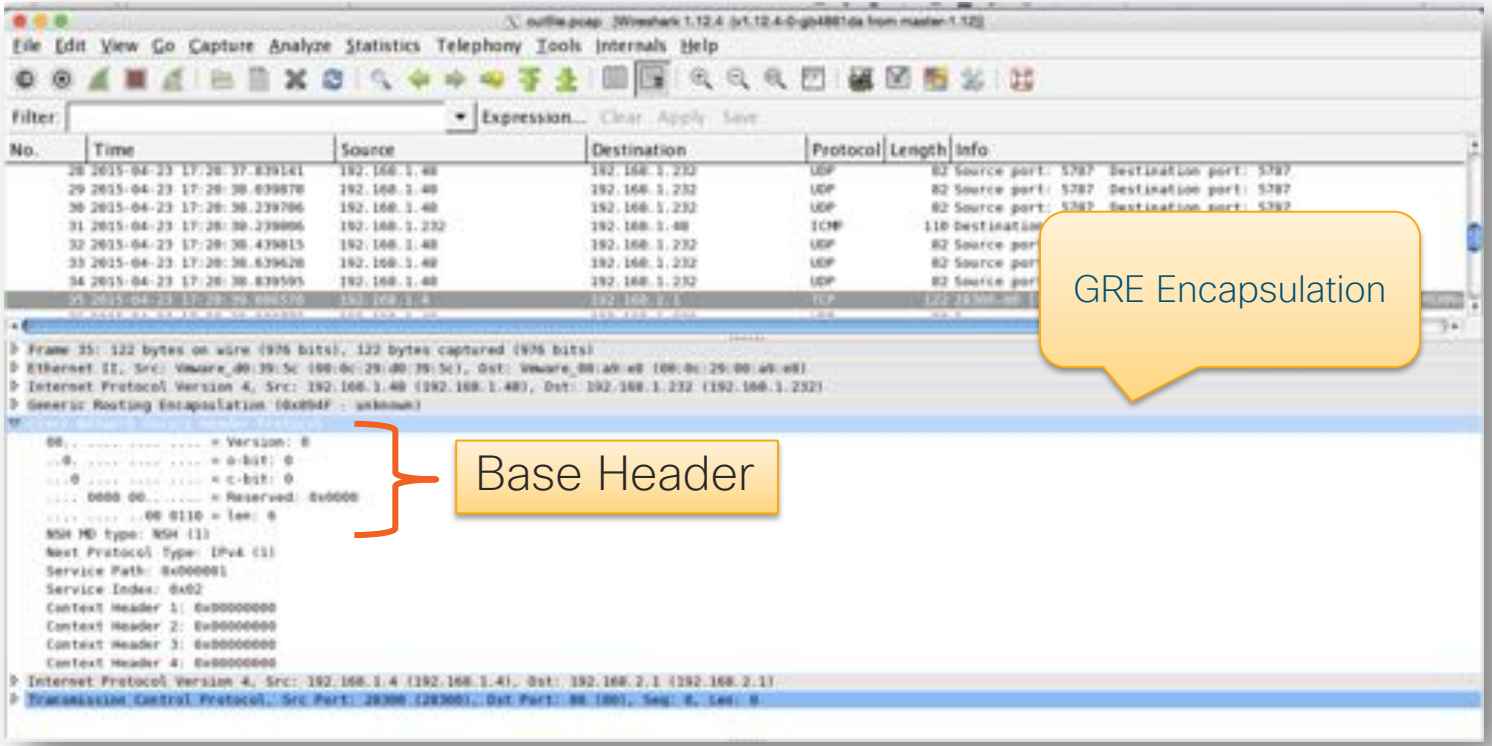
Network Service Header is a data-plane protocol that represents a service path in the network and provides a common service plane fully orchestrated top to bottom

- Two major components: path information and metadata
  - Path information 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
- NSH is added to packet via a classifier
- NSH is carried along the chain to services
  - Intermediate nodes do not need to be NSH aware
  - Non-NSH enabled services are supported

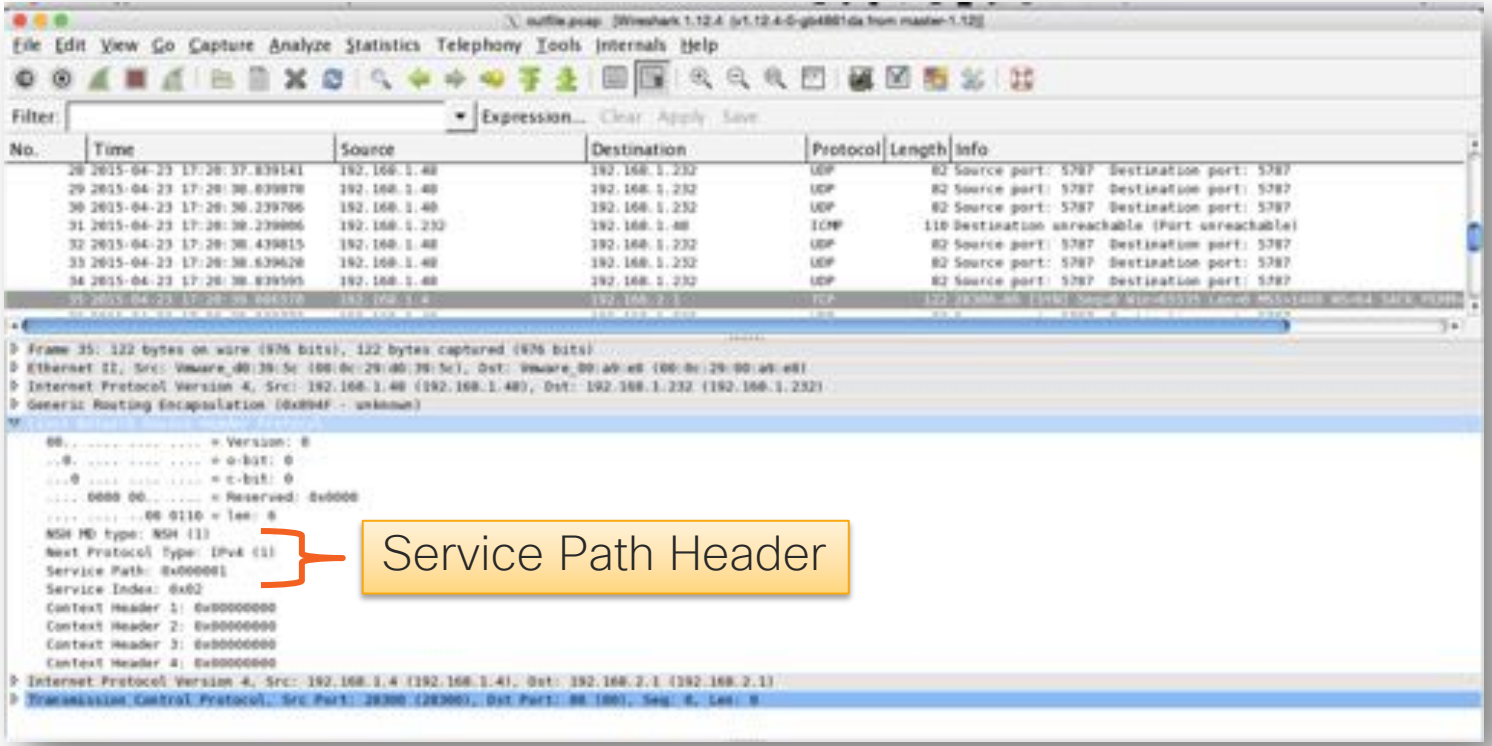
IETF proposed standard



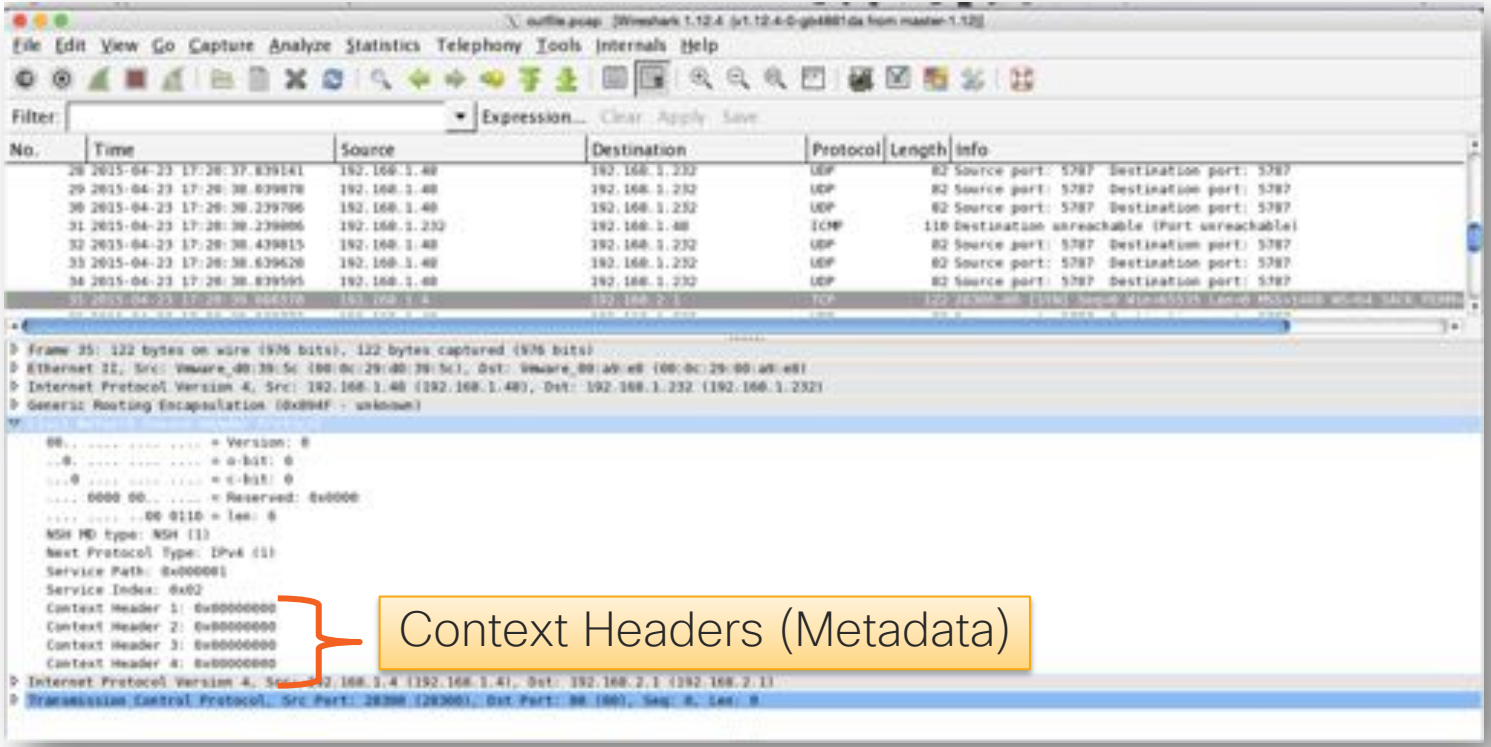
The NSH From A CSR1Kv



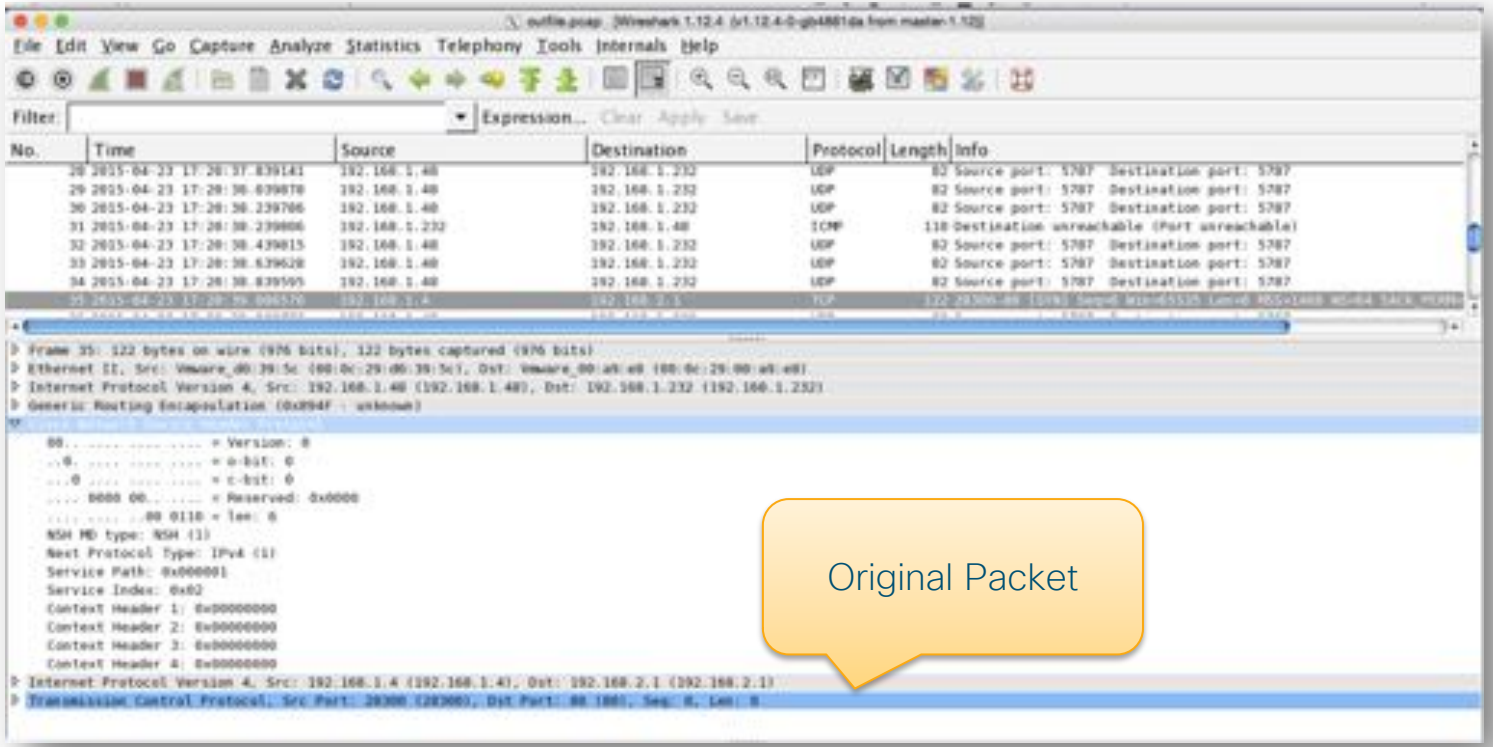
The NSH From A CSR1Kv



The NSH From A CSR1Kv



The NSH From A CSR1Kv

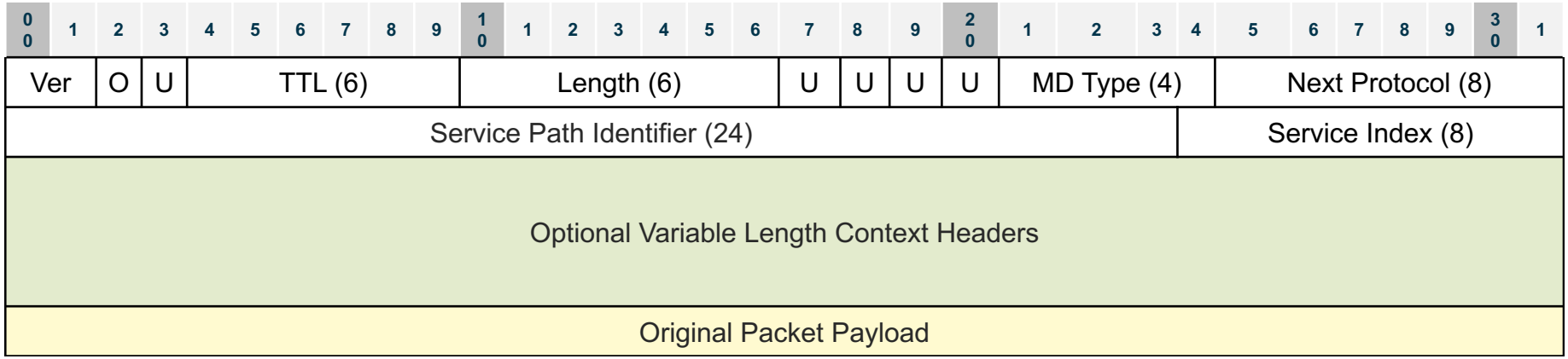


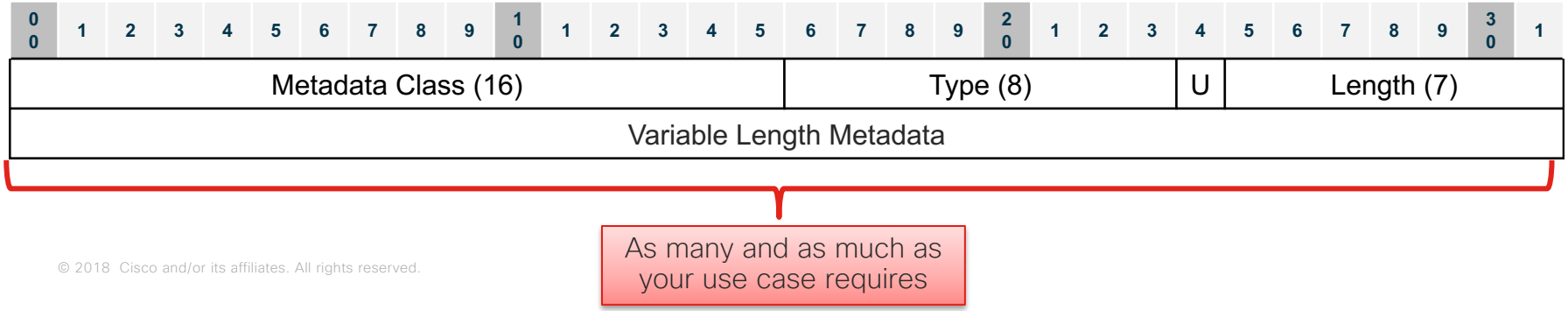
But Sometimes More Is More

- The 16-byte context header block really requires one to think about what is absolutely needed to share between nodes and services
- But sometimes you need more
  - Mobility use cases want to transport IMEI and user info
  - Data Center use cases want to transport tenant and application information
- NSH allows for this with Type 2 variable length headers

Network Service Header (NSH)

MD-Type 2 Format





Service Function Chaining – Ecosystem

- Dedicated Service Function Chaining (SFC) Working Group in the Internet Engineering Task Force (IETF)
- Several vendors support this idea of Service Function Chaining with Network Service Header
  - Close to becoming a standard
- Hear ye, hear ye! Read all about it!
  - https://tools.ietf.org/html/rfc7498 (Problem Statement)
  - https://tools.ietf.org/html/rfc7665 (SFC Architecture)
  - https://tools.ietf.org/html/rfc8300 (NSH)

NSH Implementation in Open Source

- *Vibrant open source community*
  - Data Plane in Linux kernel
    - NSH (along with VXLAN-GPE) support for Lightweight Tunneling
  - Data Plane in OVS
    - Classifier and SFF controlled via OVSDB and Openflow protocols
    - Decoupled transport and NSH encap/decap
  - Data Plane in FD.io (in progress)
    - NSH encap/decap in VPP
    - NSH-aware SF
    - Control plane agent, Honeycomb
    - Classifier, SFF and Proxy to be supported in new NSH SFC (https://wiki.fd.io/view/Project_Proposals/NSH_SFC) sub-project

Implementation Update: Open Source

- Control Plane in OpenDaylight
  - Vibrant project, with new features ever release
  - Integration with ODL Group Based Policy (as a classifier controller) and OVSDB NetVirt (as a classifier controller)
  - Standalone classifier control for Telco use case
  - Pipeline Coexistence: allowing GBP, Netvirt, and SFC to all Coexist on the same OpenFlow switch
  - Refactor of Openflow renderer and YANG models for better stability
- Control Plane in Openstack
  - Networking-sfc in Mitaka release, with backend driver directly control NSH based traffic steering of OvS
- Integration with OPNFV
  - Service Function Abstract Data Types, allowing for better integration with OPNFV.
  - ODL Beryllium (inc. SFC) in Brahmaputra release

Where Is SFC+NSH Supported At Cisco?

- Almost all major Cisco platforms have a roadmap
  - ASR1K, ENCS, ISR4K, CSR1Kv, ASR9K, Nexus 7K, Nexus 9K
  - IOS-XE 16.x support exists today for ISR, ASR, and CSR
- Services coming along, including 3rd parties
  - NBAR classification
  - Server Load Balancing
  - Firewalls
- Orchestration integration
  - APIC-EM, DNA Center, OpenDaylight

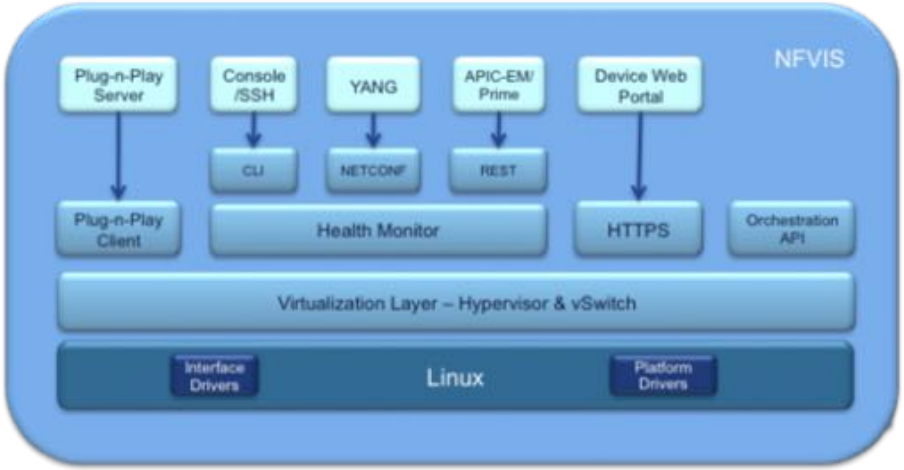
Why Is This Cool?

- SFC and NSH enable a more software-defined landscape by allowing service deployment to be fluid and independent of the network topology
- This is key in order to move from a purely physical network into network function virtualization (NfV) and a Full Stack view

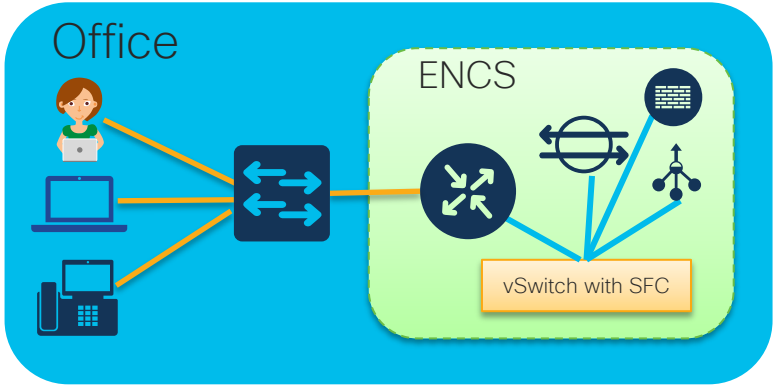
Today	Tomorrow
Physical appliances	Virtual & physical appliances
Static services	Dynamic services
Separate domains: physical vs. virtual	Seamless physical & virtual interoperability
Hop-by-hop service deployment	Chain of “service functions”
Underlay networks	Dynamic overlay networks
Topological dependent service insertion	Insertion based on resources & scheduling
No shared context	Rich metadata
Policy based on VLANs	Policy based on metadata

Examples

Cisco NFVIS

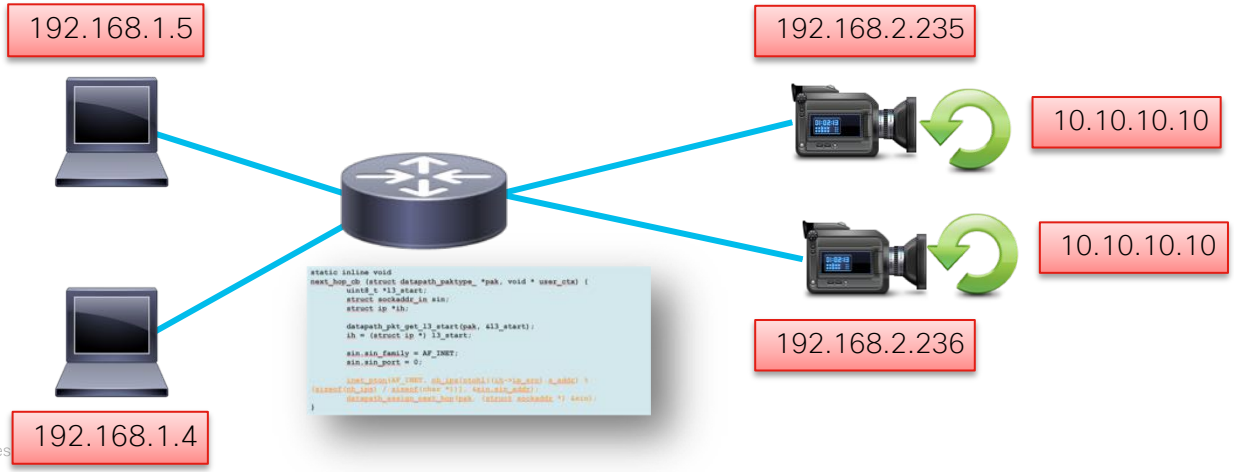


- NFV Infrastructure Software enables virtualized network services in the branch
- Traffic within the appliance is forwarded through services with SFC



Programming The Data Plane With LibESF

- Cisco provides a C library called libESF that uses SFC with MD-Type 2 NSH to expose data plane control in a programmatic way
- With libESF customized applications (i.e., your own service functions can be written)
- Build a simple load balancer using a few lines of C
- Demo video at https://cisco.box.com/v/lb-demo



Key Takeaways

- Data Path Programming provides a way to control the traffic flowing through network elements
- Service Function Chaining provides a way of defining a path of packet “actors” in a very flexible way
- Network Service Header adds steering information and metadata in order to direct traffic through service paths
- SFC, together with NSH enable Network Function Virtualization and provide a way to customize and extend data and service path operations

CISCO