Cisco Connect

Dubrovnik, Croatia, South East Europe
20-22 May, 2013

SDN

for Service Providers

Josef Ungerman
CSE, CCIE #6167

Contents

- Intro
- SDN in SP Backbones
- WAN Controller
- SP SDN Protocols
- Segment Routing and MPLSDN

Let’s Start with Some Definitions

What Is Software Defined Network (SDN)?

“…In the SDN architecture, the **control and data planes are decoupled,** network intelligence and state are logically **centralized**, and the underlying network infrastructure is **abstracted** from the applications…”

Note: SDN is not mandatory for network programmability nor automation

Source: www.opennetworking.org

What Is OpenFlow?

Open protocol that specifies **interactions between de-coupled control and data planes**

Note: OF is not mandatory for SDN

Note: North-bound Controller APIs are vendor-specific

What is OpenStack?

Opensource software for building public and private Clouds; includes Compute (Nova), Networking (Quantum) and Storage (Swift) services.

Note: Applicable to SDN and non-SDN networks

Source: www.openstack.org

What is Overlay Network?

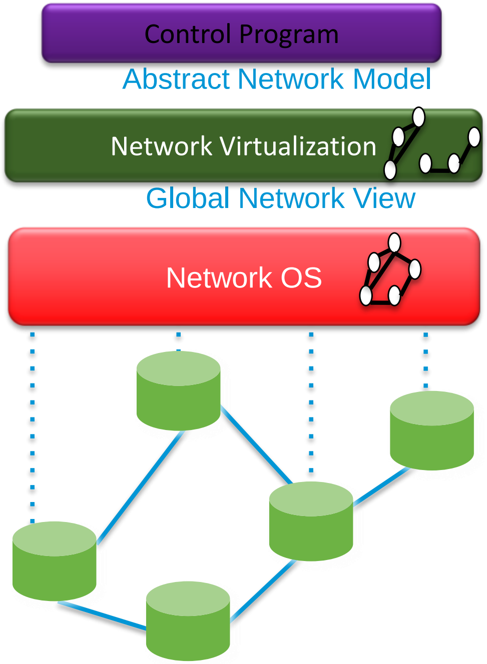
Overlay network is created on existing network infrastructure (physical and/or virtual) using a network protocol. Examples of overlay network protocol are: GRE, VPLS, OTV, LISP and VXLAN

Note: Applicable to SDN and non-SDN networks

SDN: Academic View

Professor Scott Shenker, UC Berkeley

- Abstractions do not eliminate complexity
- Move the complexity to the right place
- Control Program becomes a simple user interface
- Network Virtualization (aka network compiler) translates the request
- Network OS transmits to the network devices



http://www.youtube.com/watch?v=eXsCQdshMr4

Towards an Open Network Environment for SDN

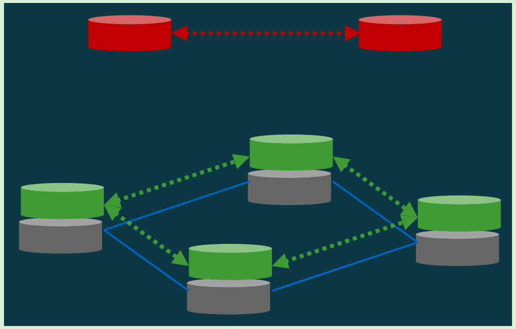
Implementation Perspective: Evolve the Control- and Management Plane Architecture

Traditional Control Plane Architecture

Evolved Control Plane Architecture (Examples)

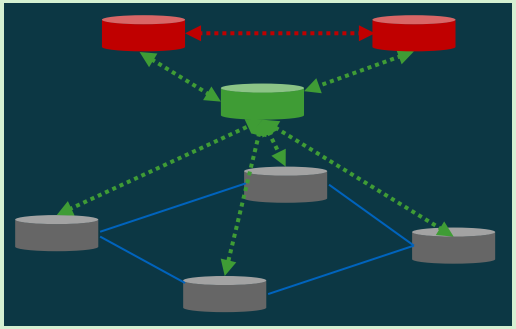
Distributed Control Plane

(selfish view of each node)



Centralized Control

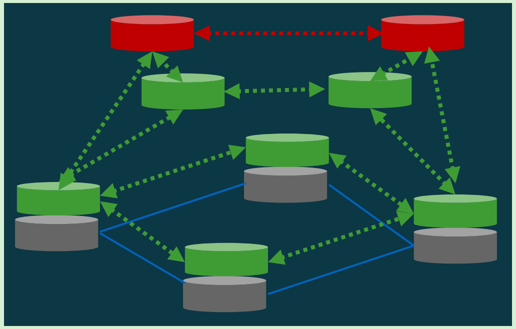
(Master of Puppets???)



…

Adding Intelligence

(what is the best for the network)



- Enable modularization and componentization of network management-, control- and data-plane functions, with associated open interfaces. This allows for optimized placement of these components (network devices, dedicated servers, application servers) and close interlock between applications and network functions.

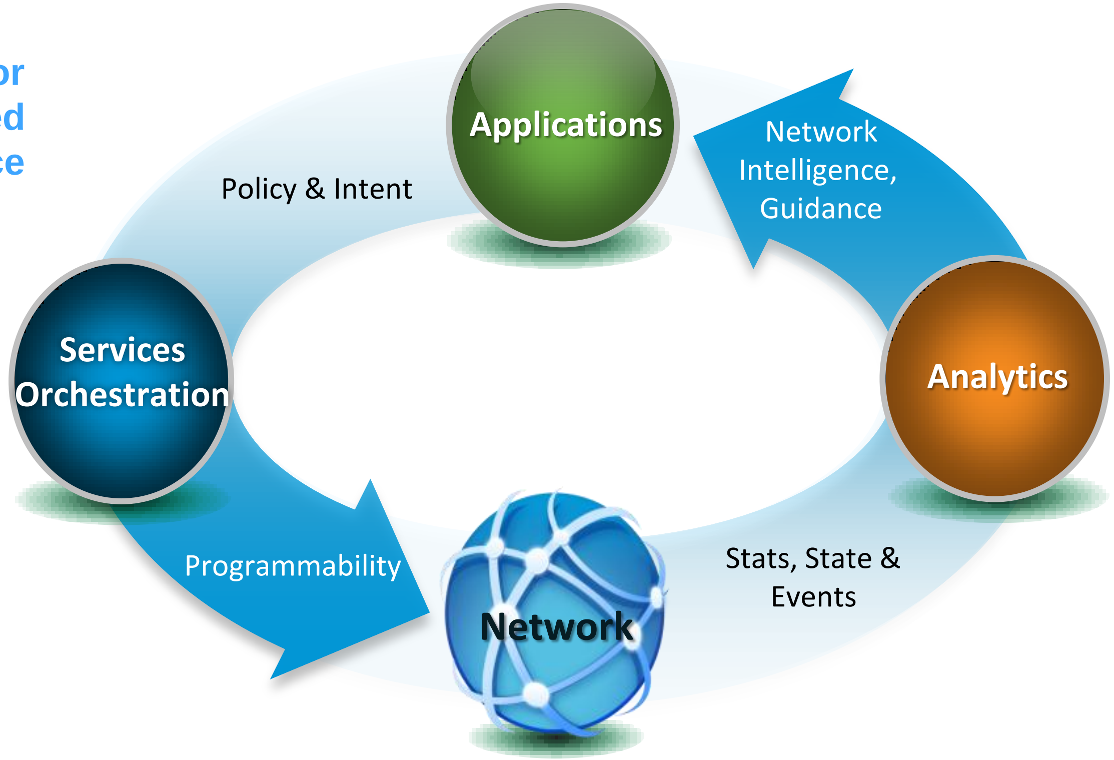
- Anticipated benefits include: Closely align the control plane with the needs of applications, enable componentization with associated APIs, improve performance and robustness, enhance and automate manageability, operations and improve consistency

🟩 Control/Network/Services-plane component(s) ⬛ Data-plane component(s) 🟥 Applications

Automation – Closed Loop

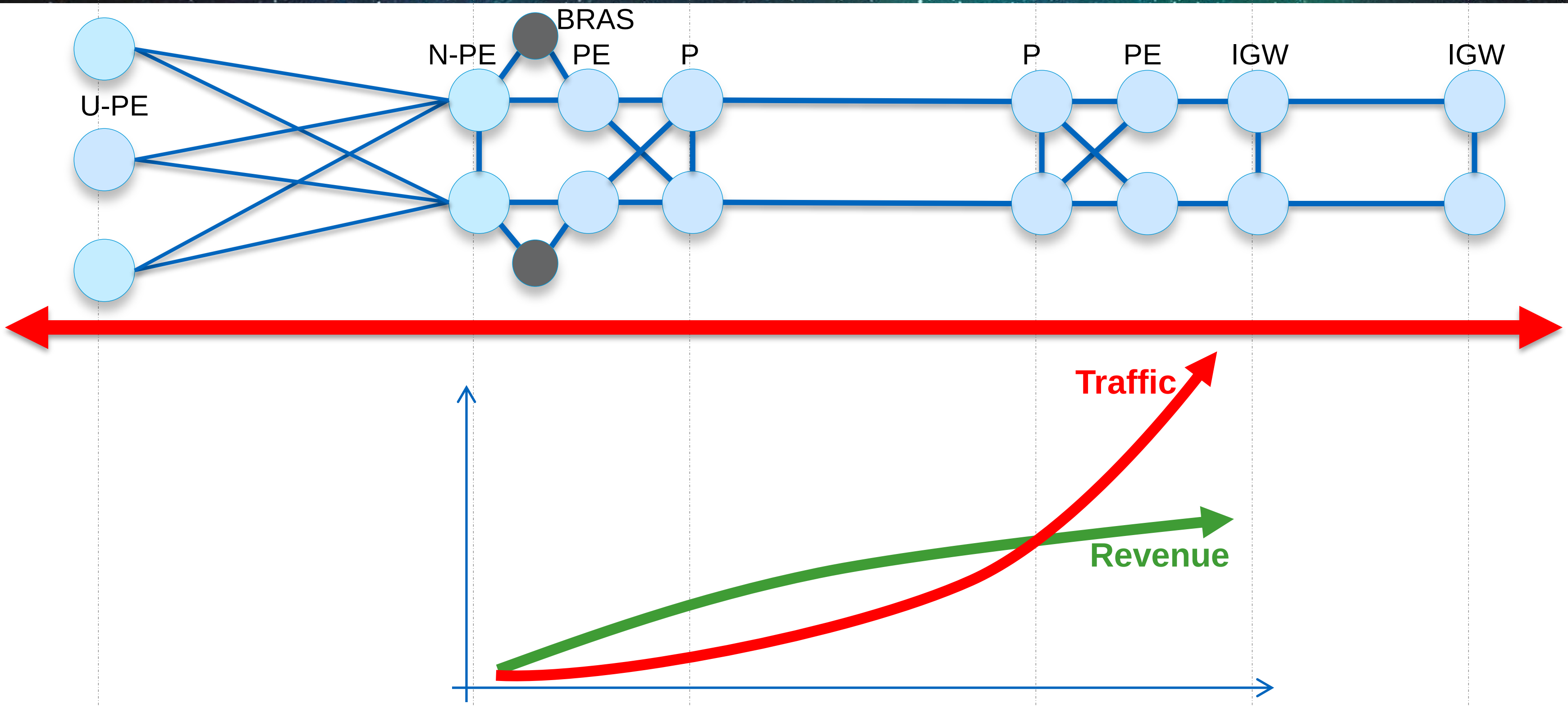
Program for Optimized Experience

Harvest Network Intelligence



Typical National SP Backbone

Legacy Architecture



Modern Architecture Analogy

Functionalism (leading architectural style in 1920-1970)

- Functionalism was introduced as new form that is able to move away from a pomp and ornamental aesthetics of the 19th century. Garishness and **unnecessary complexity** was elegantly replaced by pure geometry.
- It's typical for the functionalistic architecture to use **simple** shapes. It uses **new technology materials** – scarlet bricks, iron, concrete.

Villa Müller (Prague)

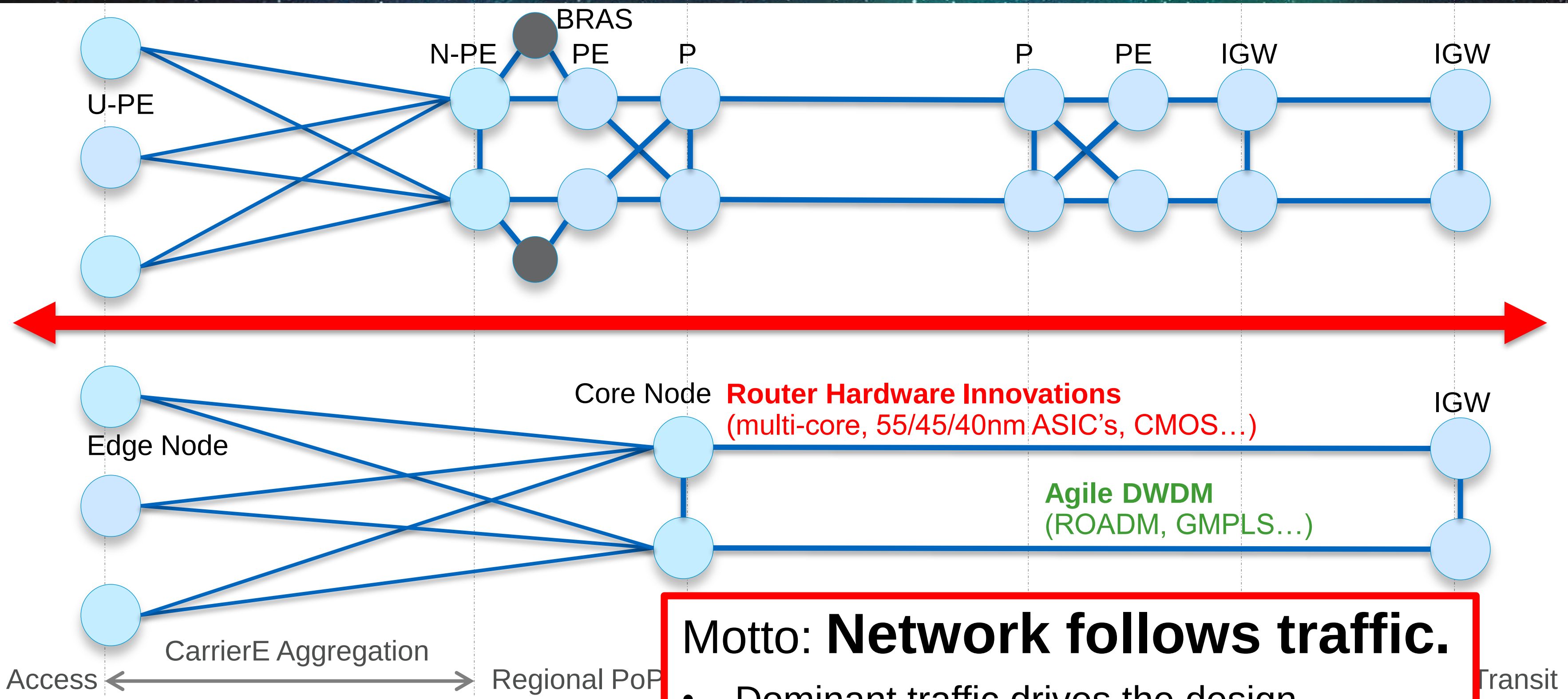
Villa Tugendhat (Brno)

Motto: **Form follows function.**

- Dominant functionality drives the design.

Typical National SP Backbone

Legacy Architecture → Modernism



Router Hardware Innovations
(multi-core, 55/45/40nm ASIC's, CMOS…)

Agile DWDM
(ROADM, GMPLS…)

Motto: **Network follows traffic.**

- Dominant traffic drives the design.

Modern Architecture Analogy

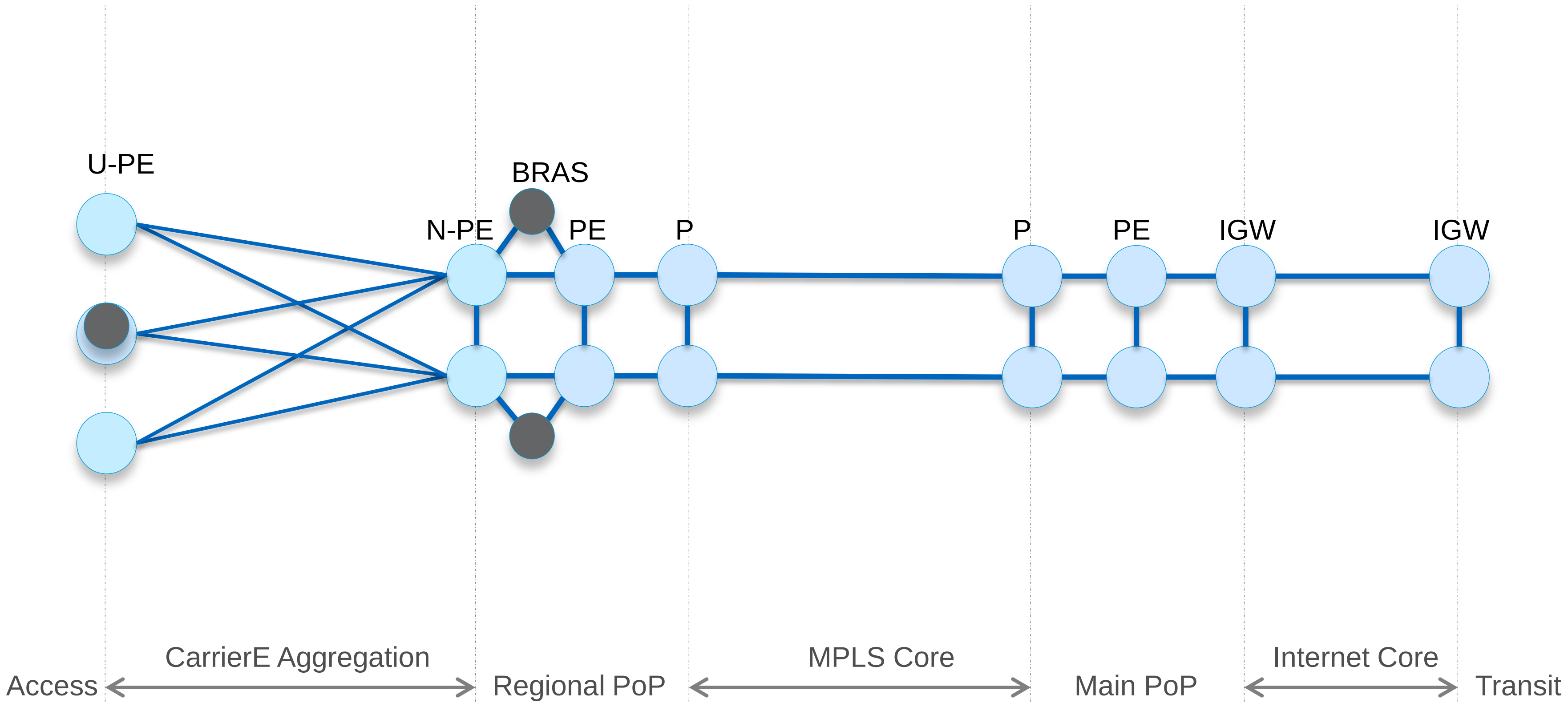
Art Deco

Functionalist structure with **features** applied in a **new and original way**.

SDN role in SP Backbones

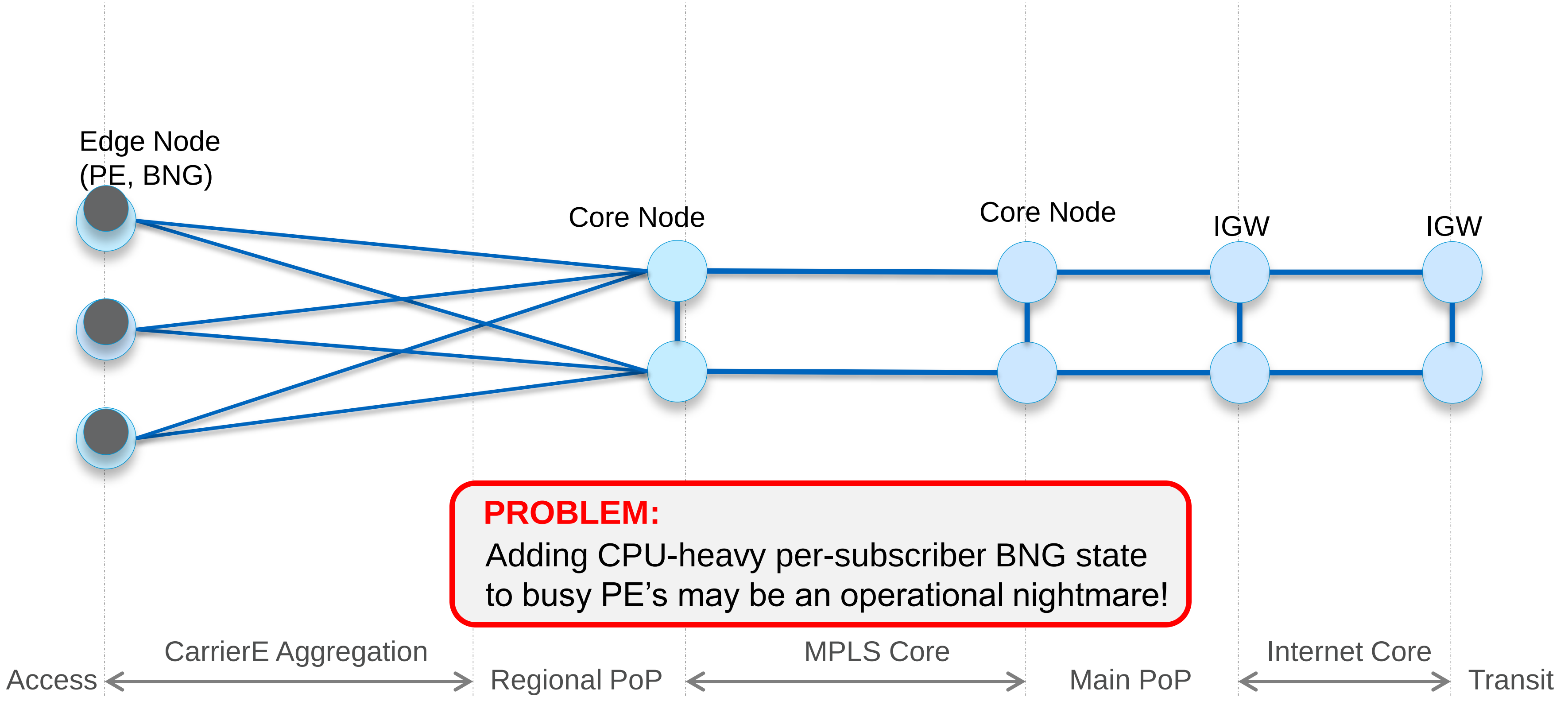
Typical National SP Network

Dominant Traffic's Path



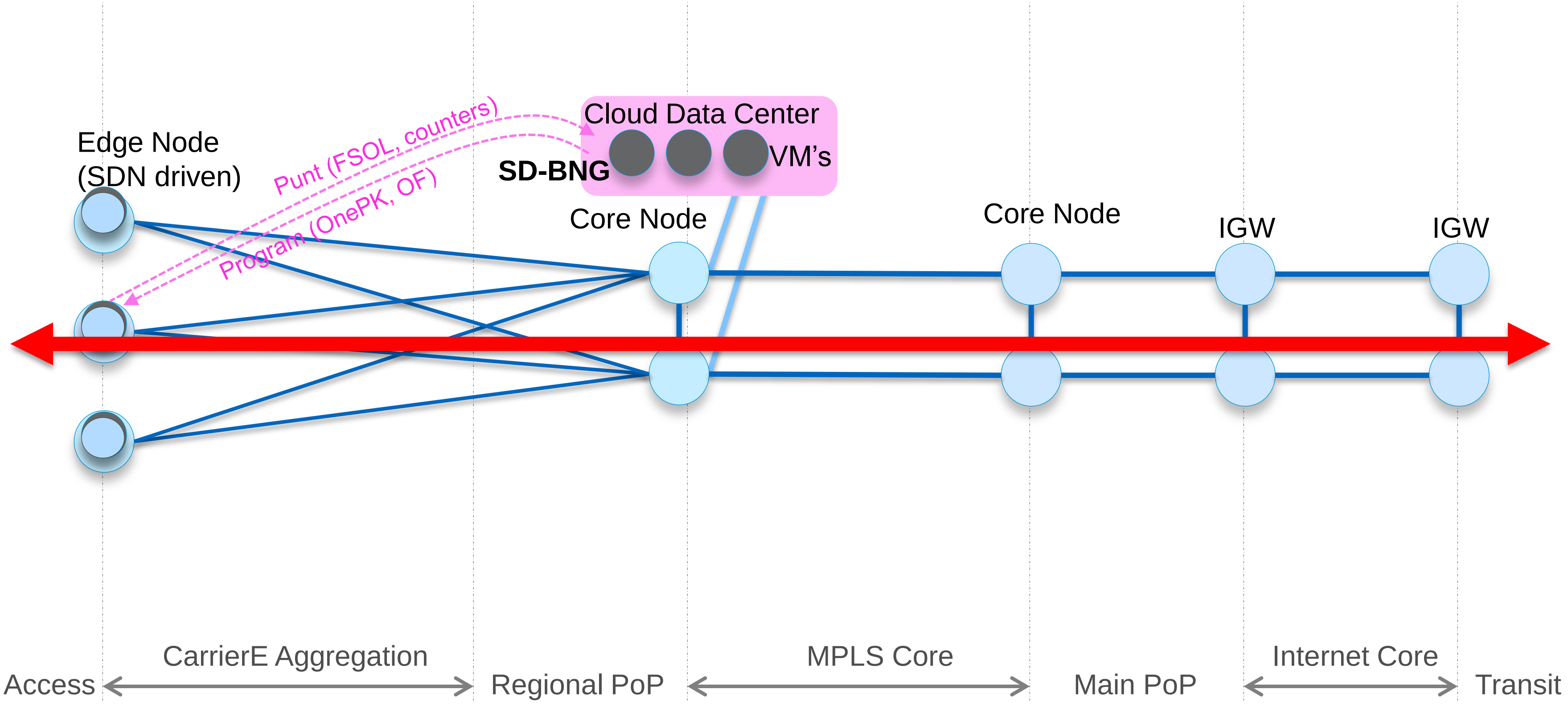
Typical National SP Network

Dominant Traffic's Path



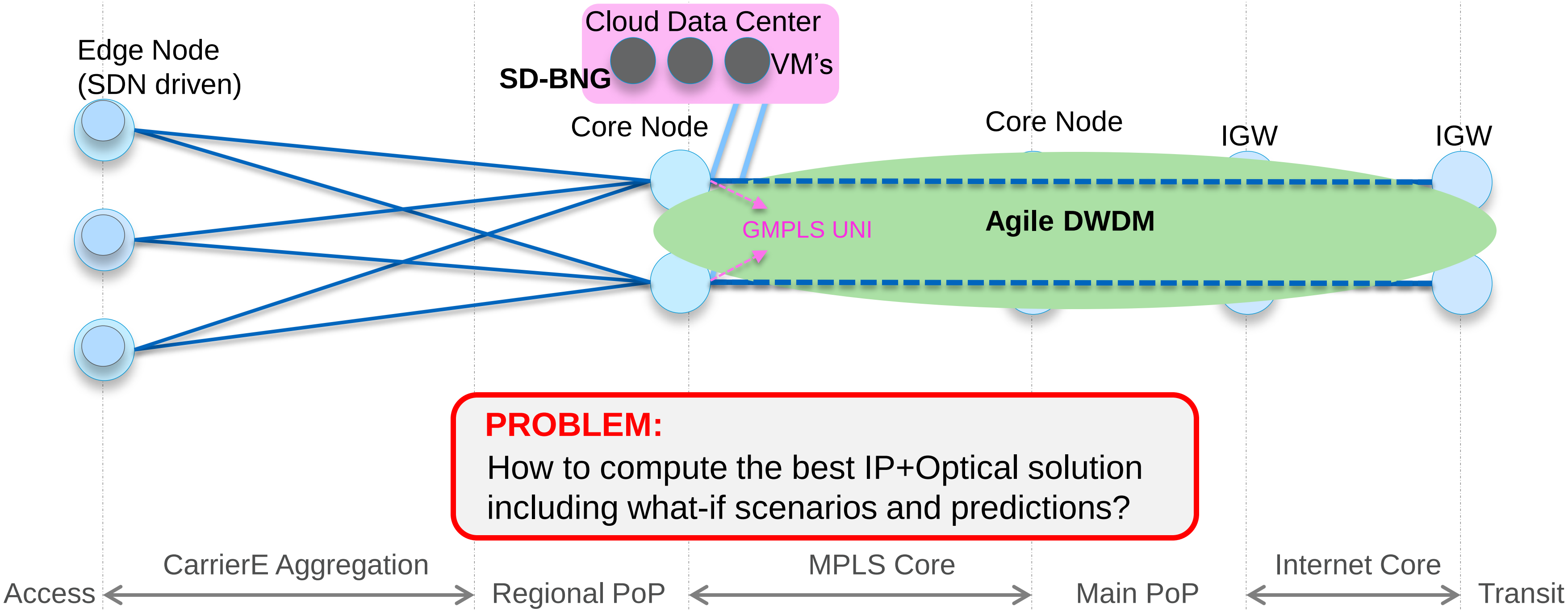
PROBLEM:
Adding CPU-heavy per-subscriber BNG state to busy PE's may be an operational nightmare!

Typical National SP Network
Dominant Traffic's Path



Typical National SP Network

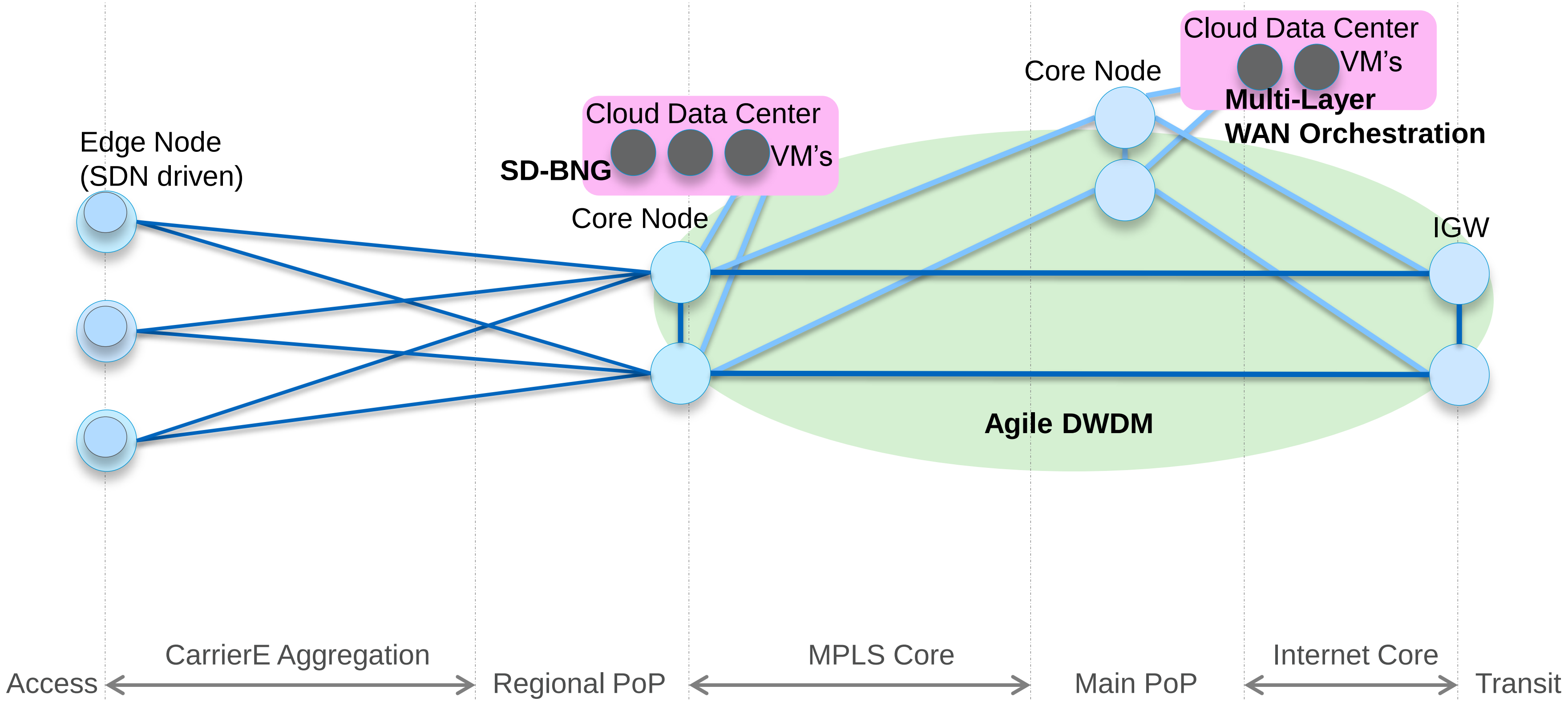
Dominant Traffic's Path



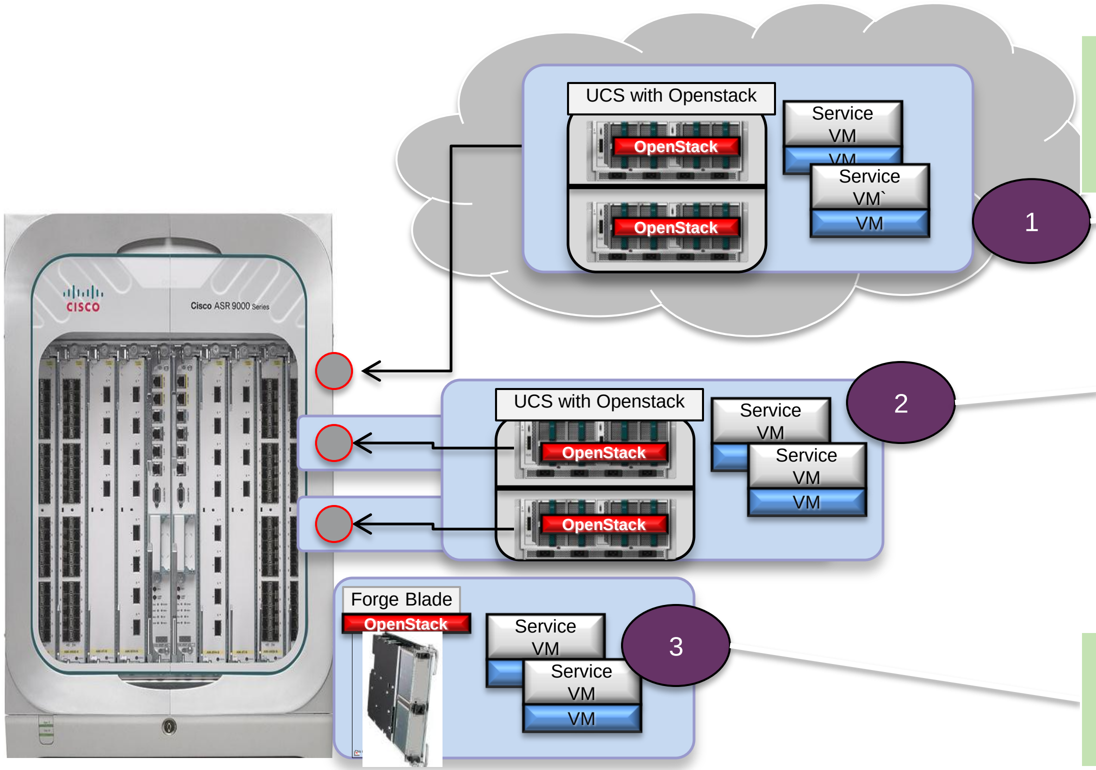
PROBLEM:
How to compute the best IP+Optical solution including what-if scenarios and predictions?

Typical National SP Network

Dominant Traffic's Path



Where to run?
Attaching Compute to the Network



Multiple UCS blades running OpenStack connected via a network to a single 10GB port on the ASR9K/CRS

- Minimizes number of data ports needed on the router
- Suitable for services that require lower network bandwidth

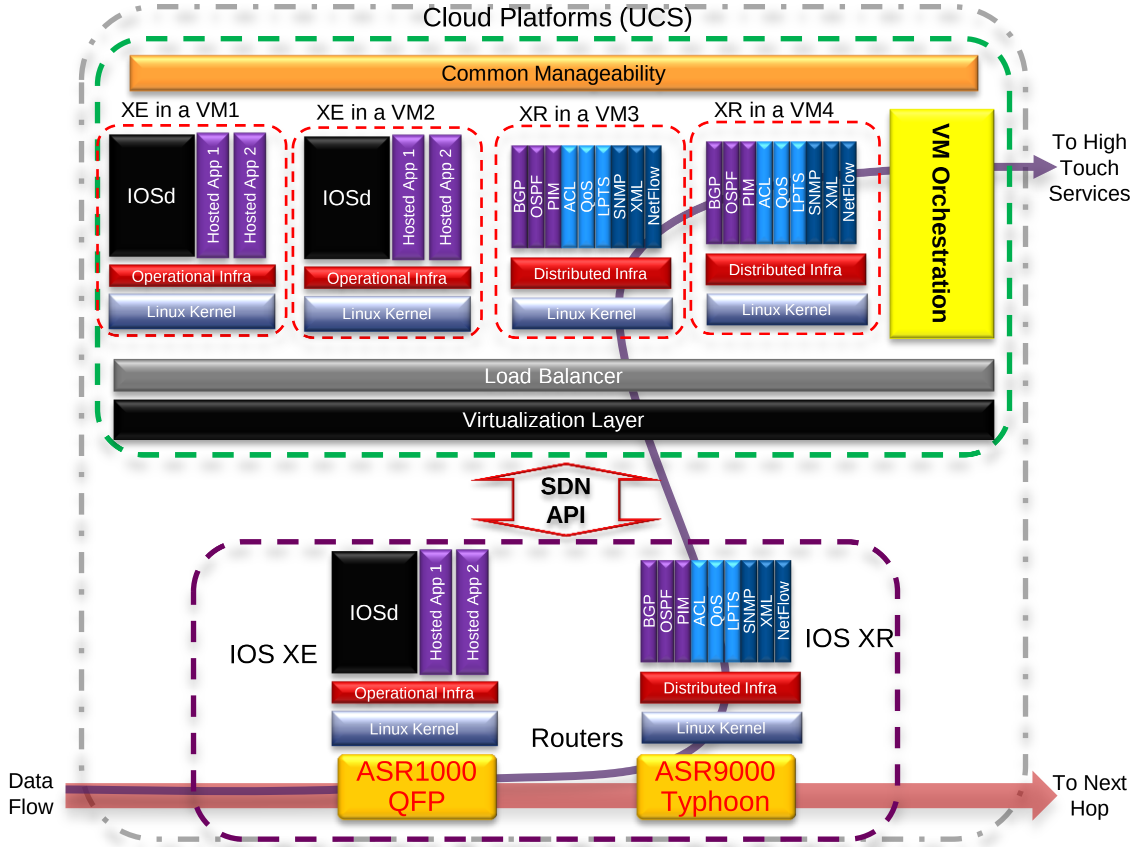
Each UCS blade (running OpenStack) directly connected to a 10G port on the ASR9K/CRS

- Requires a dedicated data port per UCS
- Suitable for higher services requiring higher network bandwidth and minimal latency

OpenStack running on the VSM blade (ASR9K)

- Takes up a service blade slot
- Suitable for small number of services that benefit from being connected to the ASR9K fabric

Where get the software?
IOS Virtualization

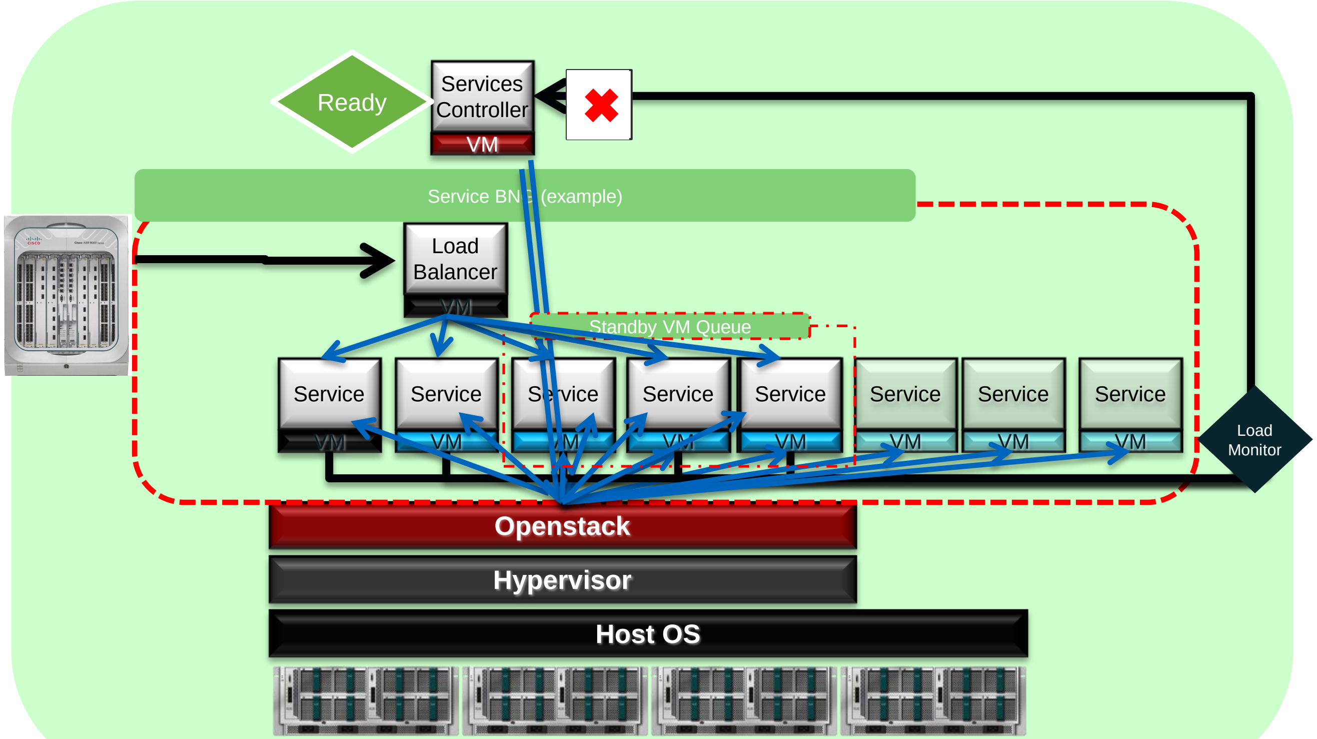


IOS XE: CSR1000v

IOS XR: VM's coming

OnePK API unifies them

How to scale?
Elastic Cloud resources

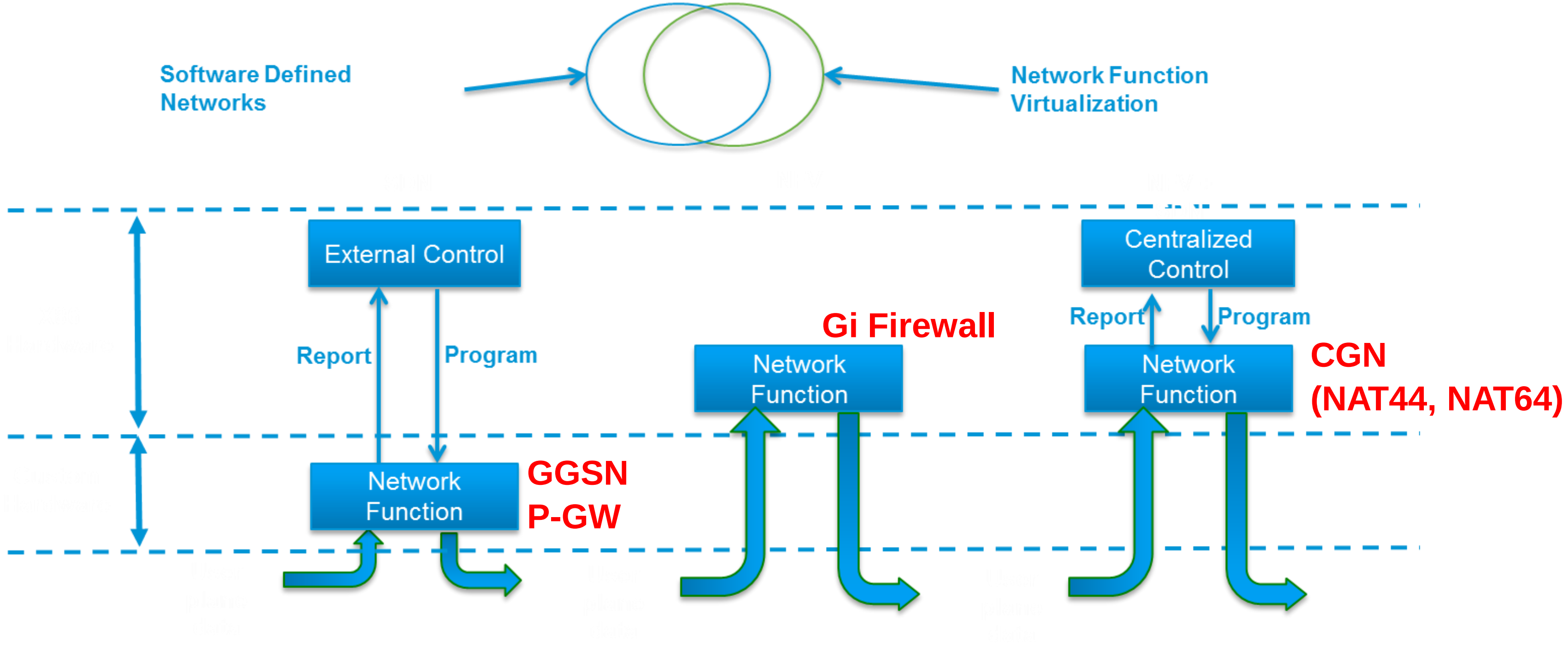


EXAMPLES:

- BNG scaling (10M’s of subscriber sessions, 10K’s sessions per second
- On-demand DDoS mitigation (Arbor) – Nx 10GE

NfV – Network Function Virtualization

Service Chaining

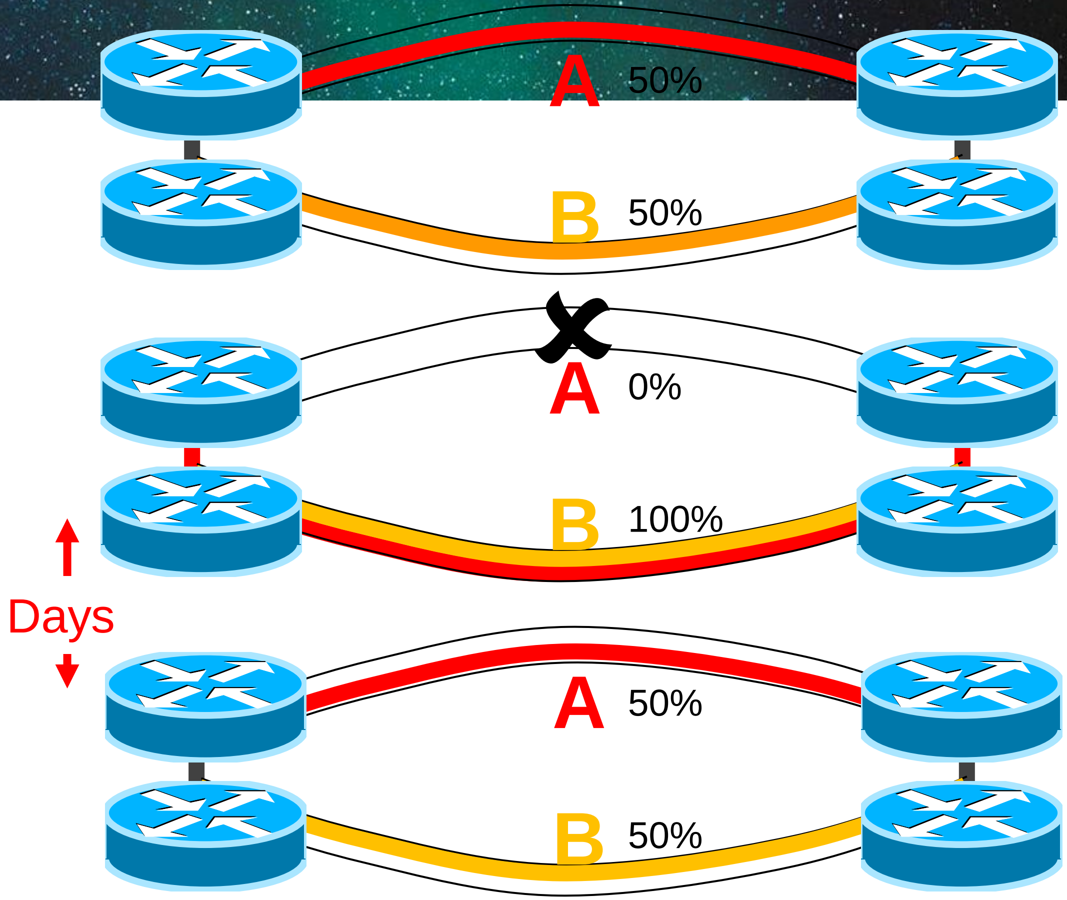


Delivering Network Functions on Commercial Compute Hardware
Leveraging cloud computing techniques for services flexibility and auto-scaling

100GE Core

Usual Network Redundancy

- Router Ports Peak Load 50% in none failure situation
- Traffic from A gets fast re-routed to B in case of failure
- Link on B utilized up to 100%
- Failure duration on A is not predictable, can be days !!!
- Failure restored → Traffic routed back to A



100GE Core

With IP+Optical

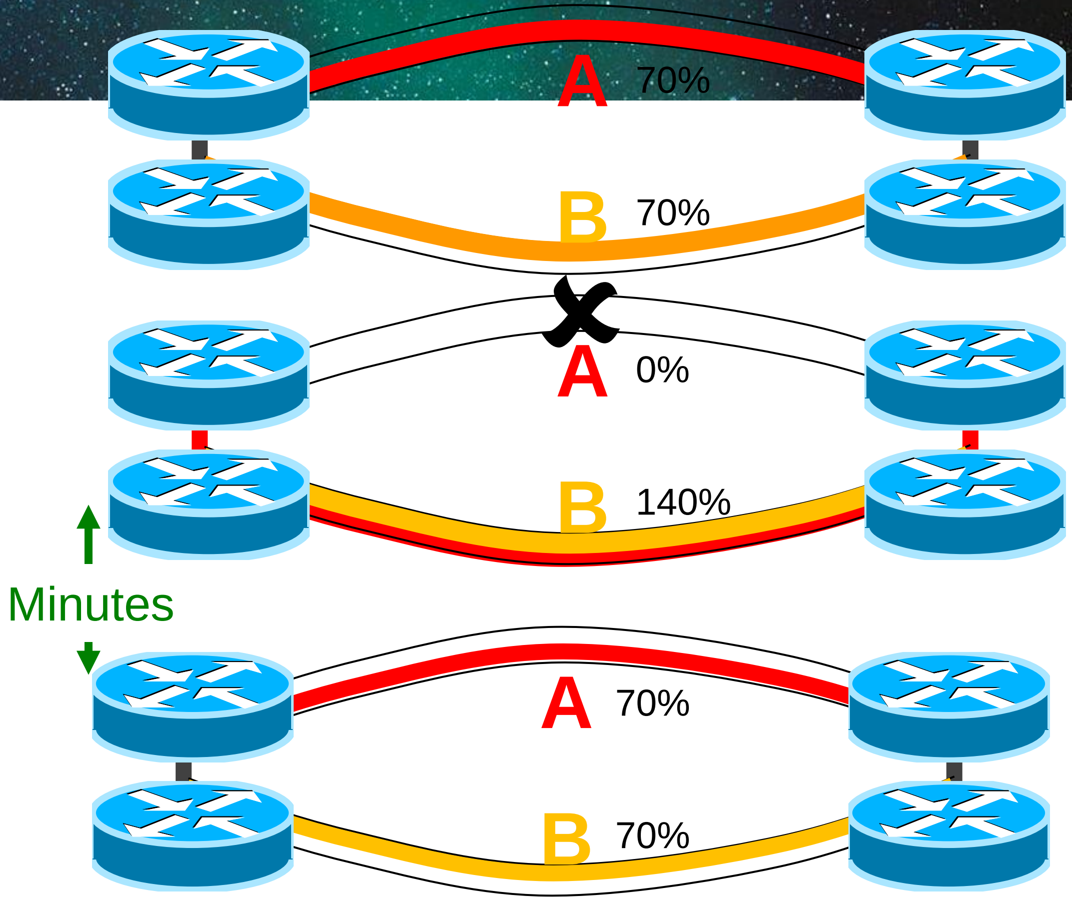
- Router Ports Peak Load
  70% in none failure situation

- Fast Re-Route (L3) of A Traffic
  to B in case of failure

- Link on B utilized up to 140%
  → No drop of priority Traffic;
  Only BE Traffic dropped
  during Peak Hours

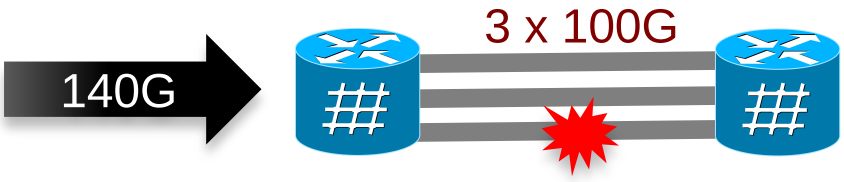
- A gets optically restored to A‘ using
  same Router interfaces in
  minutes → ROADM based

- Failure restored → A‘ lambda will be
  reverted to path A once trunk is repaired

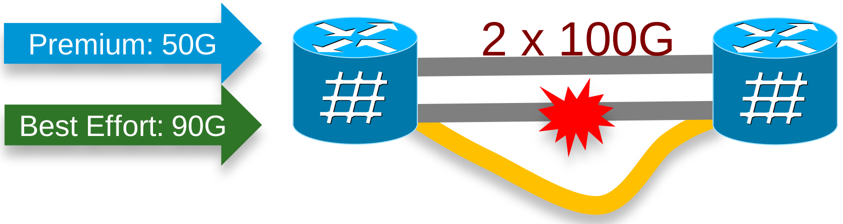


Multi-Layer Optical Restoration

Better IP Interface Utilization!



Worst-case (stable):
140G on 200G
Avg IP util: 140/300= 47%



Restore in ~30 seconds

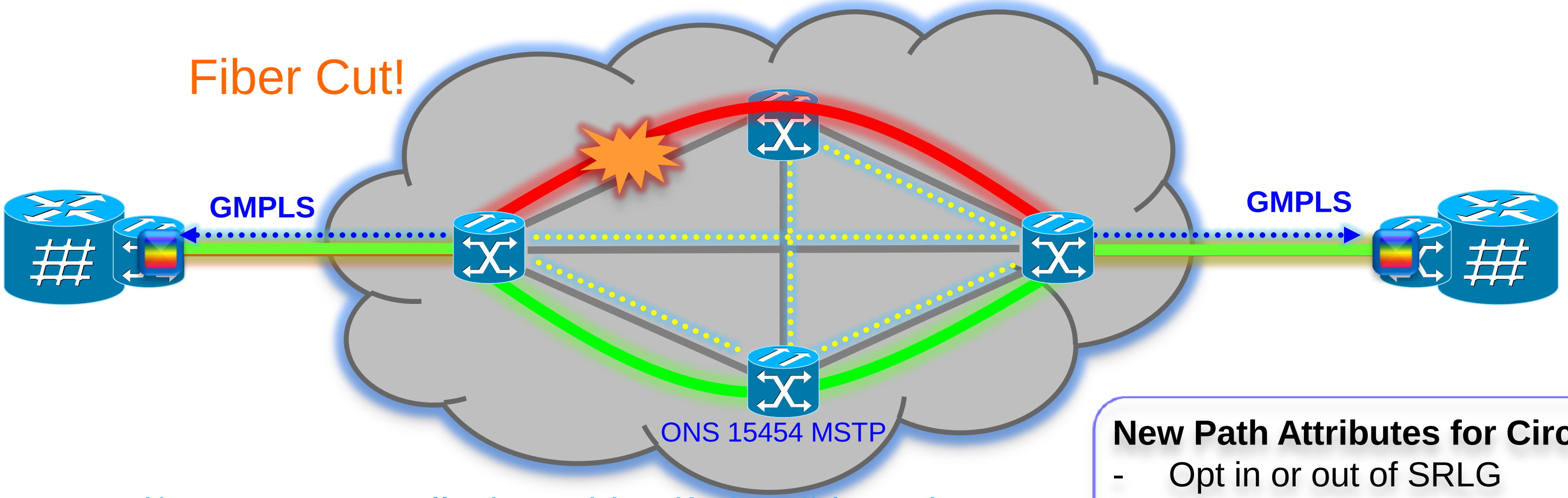
Worst-case (transient):
140G on 100G
Oversubscription, BE loss

Worst-case stable:
140G on 200G
Avg IP util: 140/200= 70%

Study based on major SP: 26% Fewer Interfaces

Multi-Layer Optical Restoration

Leverage Embedded Intelligence



New Path Attributes for Circuit:

- Opt in or out of SRLG
- Follow/avoid another Circuit
- Use path with latency bounds

WSON and/or WAN Controller/PCE identify feasible paths

Paths verified for circuit against requested attributes

ROADM instructs client via UNI to re-tune its wavelength

Colorless, Omni-Directional ROADM switches to the best path

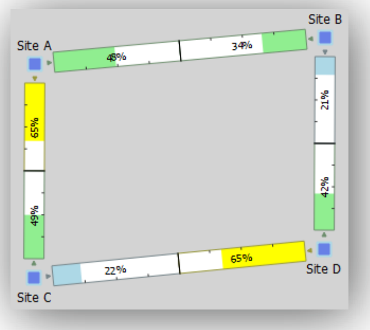
Service is brought back up with the **same Client and Optical interfaces**, zero touches

More Resilient-Fewer Router Interfaces & DWDM Wavelengths–50% Savings

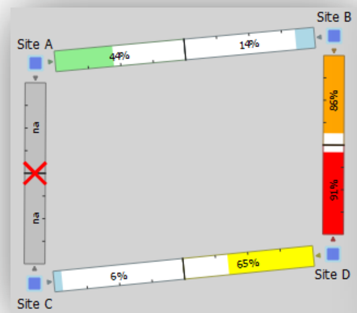
WAN Controller

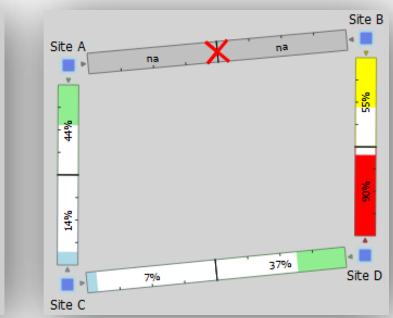
Predictive Analysis – Assessing Risk (what-if)

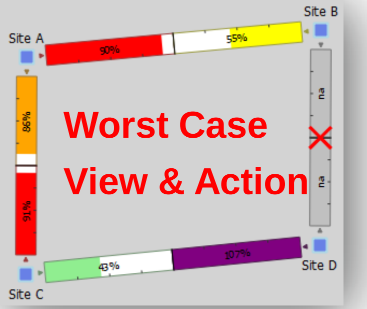
- By simulating failures, you can examine
  - Where traffic will go (and what impact this traffic will have)
- By simulating failures over a set of objects, you can examine risk network-wide. This includes
  - The impact a failure will have
  - The worst-utilization an interface will have

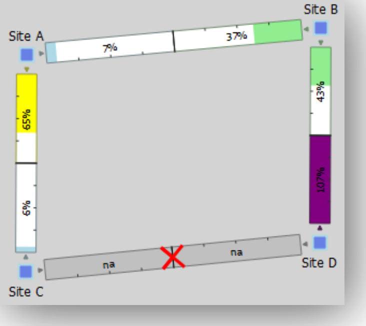


- Example – Examine a set of *circuit* failures (one-by-one)





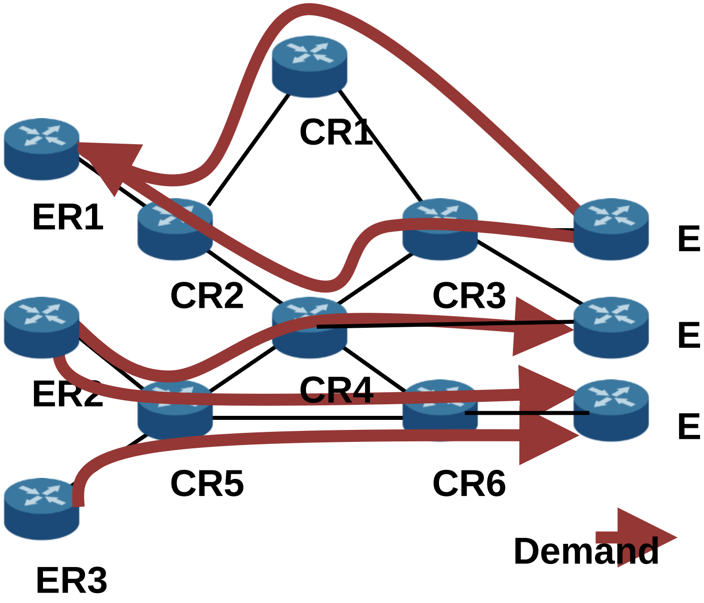




Traffic Matrix

Ref: Best Practices in Network Planning and Traffic Engineering

- Traffic demands define the amount of data transmitted between each pair of network nodes
  - Typically per Class
  - Typically peak traffic or a very high percentile
  - Measured, anticipated, or estimated/deduced
- A network's traffic matrix is list of demands
- The traffic matrix has two functions
  - Indicate why a network’s traffic distribution looks the way it looks
  - Help predict what would happen in the network if something were to change (topo/traffic)



http://www.nanog.org/meetings/nanog52/abstracts.php?pt=MTc2NyZuYW5vZzUy&nm=nanog52&printvs=1

Measuring a Traffic Matrix

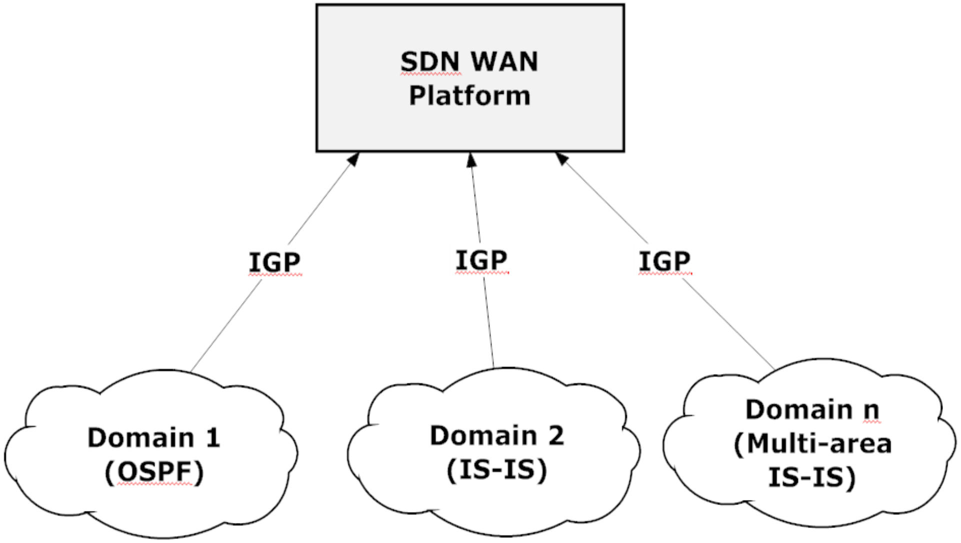
- LDP
  - Internal matrix only, Not per class
  - O(N^2) measurements + Inconsistencies in vendor implementations
- RSVP-TE
  - Internal matrix only, Not per-class
  - O(N^2) measurements + Requires a full mesh of TE tunnels
- Netflow v9
  - BGP NextHop Aggregation scheme provides almost direct measurement of the Traffic Matrix
  - CoS ready (finally, per class!)
  - Sampled information (possible inaccuracies, SNMP time mismatch)

SP SDN Protocols

Data Collection – Link State Database (LSDB)

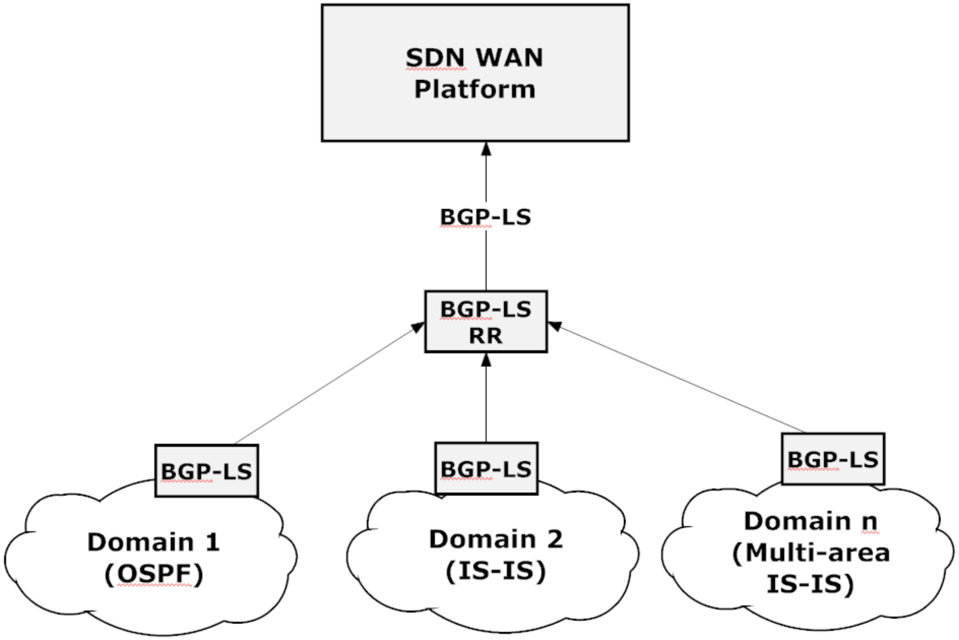
- ISIS or OSPF
- Links, nodes and attributes
- Synchronized by flooding

- IGP Listener Challenges
  - Operators typically do not like to expose their IGP to external entities
  - O(# of domains) cost and complexity
  - Raw LSDB feed – no way to abstract or control what is released outside of the domain



Data Collection – use BGP to collect LSDB

- BGP Link-State (BGP-LS)
- Redistribute IGP LSDB into per-domain BGP speaker
- Advantages
  - Single upstream topology feed (BGP)
  - IGP isolated from external entities
  - Leverage well-known BGP security, transport and policy knobs
  - Enables operator control
- draft-ietf-idr-ls-distribution



Data Collection – BGP-LS

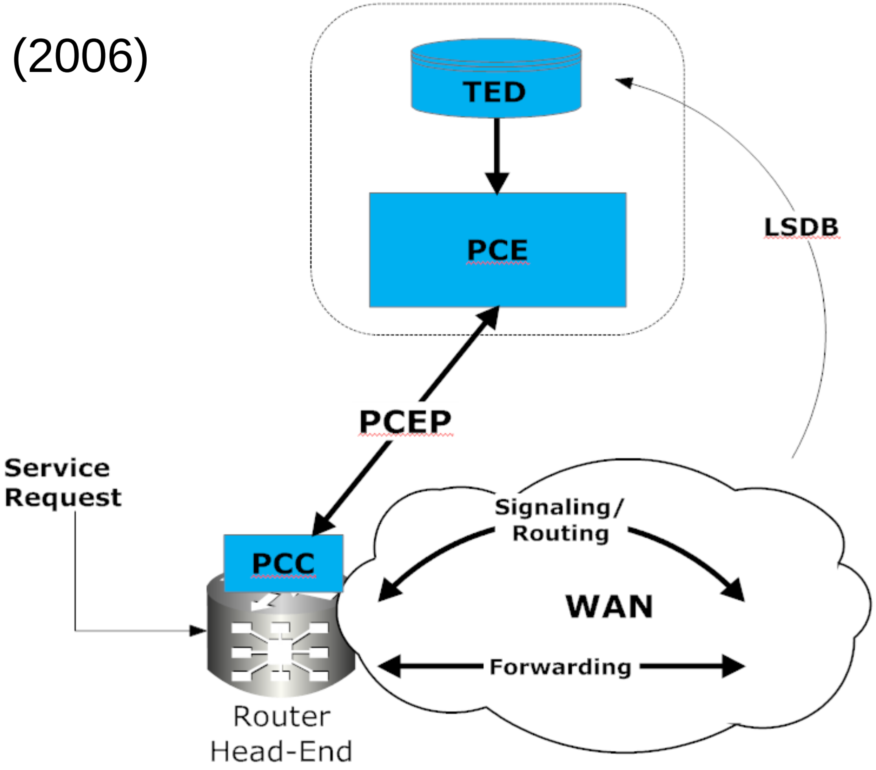
- Allows over-the-top topology export, scale via RR/RS
- BGP policy mechanisms can be used to control the redistribution and advertisement topology data
- IGP LSDB can contain more information than just cost
  - Link delay, Delay variation, Packet loss, Residual bandwidth, Available bandwidth (extensions to ISIS/OSPF – new TLV's)

- BGP speakers express their BGP-LS support in capabilities
- LSDB carried in BGP Messages using:
  - MP_REACH_NLRI, MP_UNREACH_NLRI, Link-State Attribute
- Link State
  - LS NLRI: link, node or prefix (IPv4/IPv6)
  - LS Attribute: Describes a topology element

Network Programming –
PCE (Path Computation Element) basics

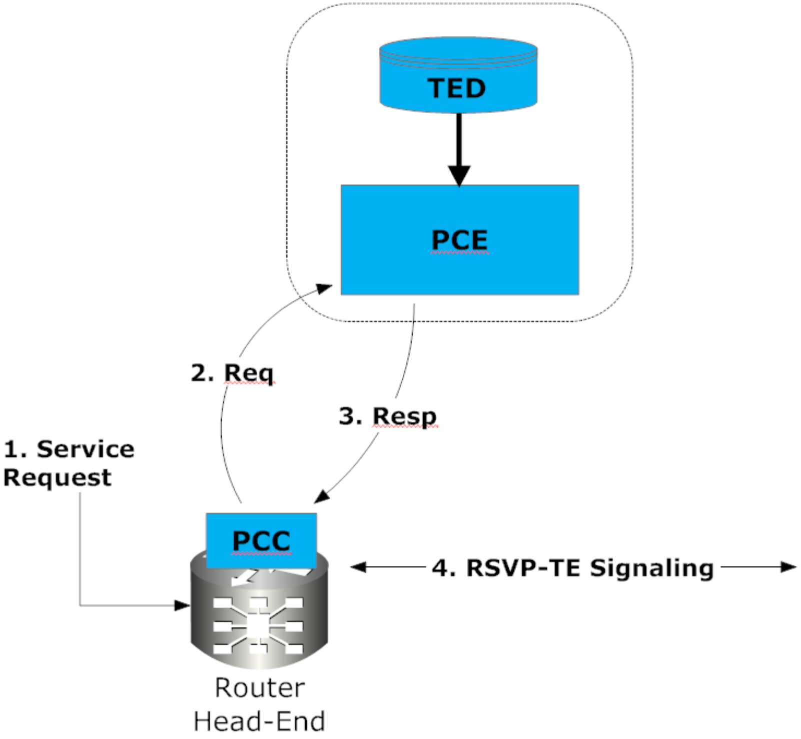
- Centralized Computation Model for MPLS (2006)
  - Computes Paths
  - Originally for Inter-AS TE (explicit paths)

- PCE Server (PCS)
- Path Computation Client (PCC)
  - Agent on router(s) that interact with PCE Server
- PCE Protocol (PCEP)
  - Protocol that runs between PCC on router and PCE server
- Traffic Engineering Database (TED)
  - Contains topology and resource information (LSDB etc.)



Classic (Stateless) PCE Workflow

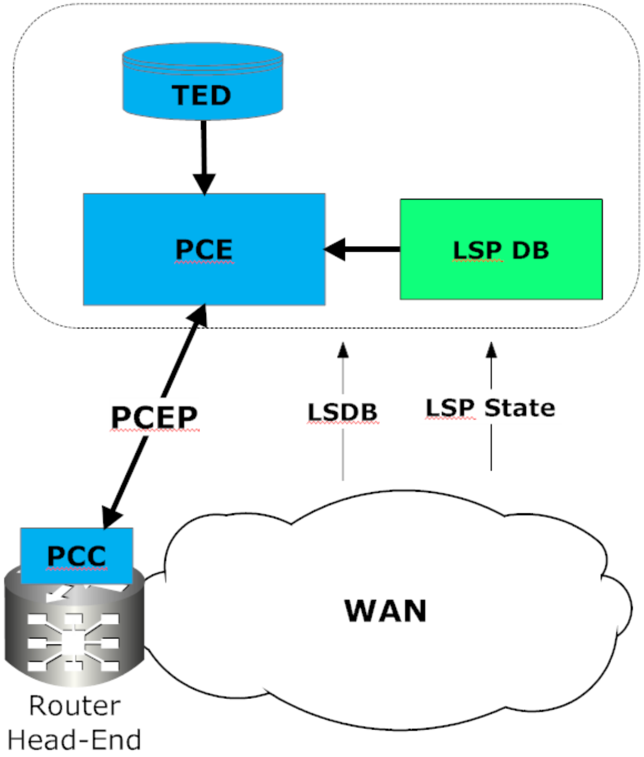
- Basic request/response interaction between the PCC and PCE
- PCE will only compute and convey path computation results in response to request generated by PCC
  - Uses response info to then signal TE tunnel setup thru network
- Note: this is NOT your general SDN notion where application drives controller to program (push) state into the network



- Stateless vs Stateful PCE (RFC4655)
  - Stateless – Just independent transactions, does not remember computed LSPs
  - Stateful – Topology, resource, LSP state is synced to PCE

Stateful PCE

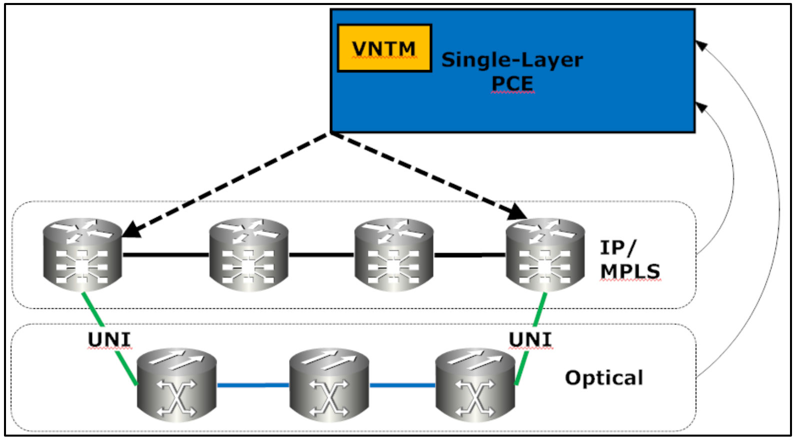
- LSP Database
  - Contains info/status on active LSPs communicated by PCCs in LSP state reports messages
- Passive Stateful PCE
  - References LSP DB for path computations
- Active Stateful PCE
  - References LSP DB for path computations
  - Programs LSP state in network
- Delegation
  - PCC delegates LSP control responsibility to PCE



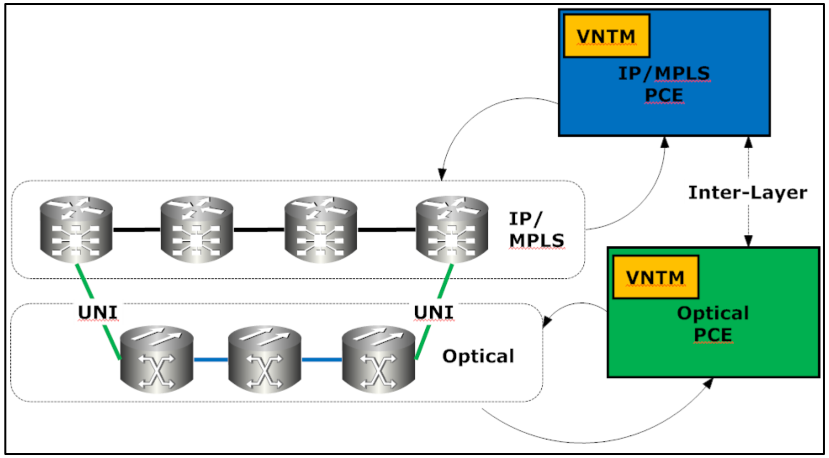
Multi-Layer IP/Optical PCE

- RFC5623: Virtual Network Topology Manager (VNTM)
  - Abstracts and presents virtual network topology to next layer up; inter-layer path control
  - Example: GMPLS optical path is presented as a virtual link to the IP/MPLS topology

- Single-Layer PCE
  - Visibility into L3 and optical topologies
  - Programs L3 and L3 UNI to optical



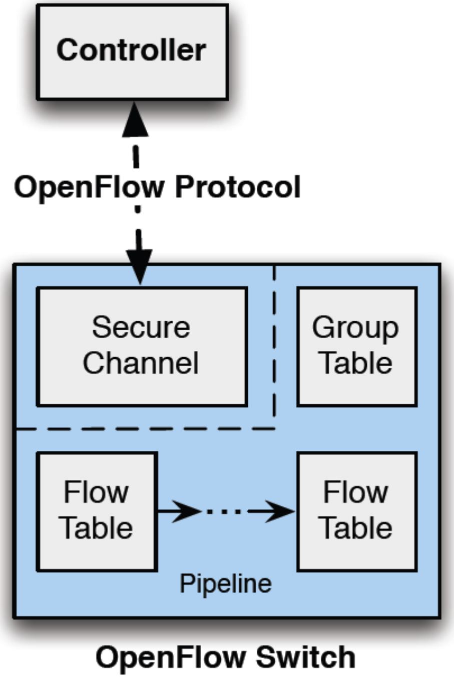
- Separate PCE
  - Operates on each layer
  - Optional inter-layer PCE communications



What about Openflow (OF)?

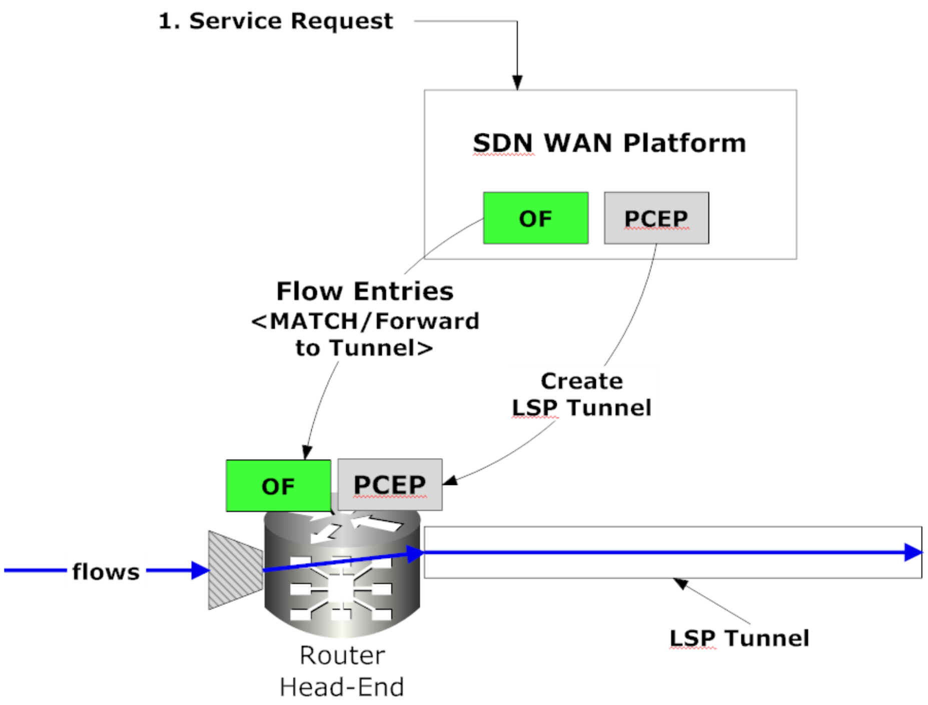
- Original SDN “southbound” protocol operating between the Controller and agent on a switch (Data Center/Cloud Research community)
- Facilitates separation of control and data planes
- App on top of controller uses Openflow protocol to program flow table entries on the Openflow switch
- www.opennetworking.org

- Openflow and SP Network:
  - Not for Core or IP+Optical (no per-flow state there)
  - May be used at the Edge (PBR-level granularity)



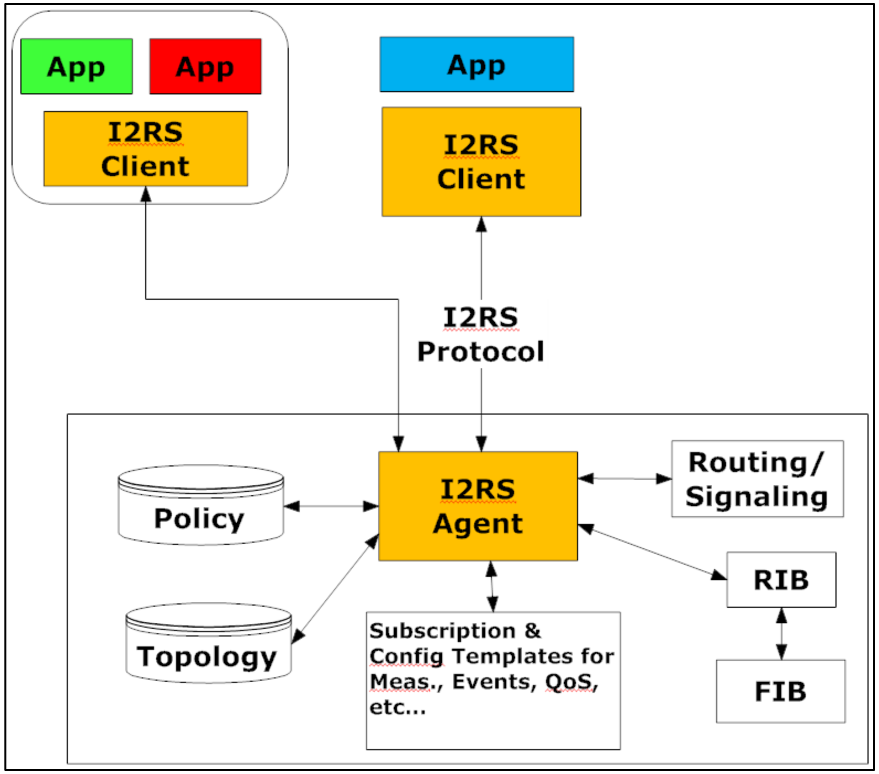
Example: SDN WAN and Openflow for Traffic Steering

- Use Openflow to program classifiers on WAN Edge
- Flow entries something like:
  - MATCH/Forward-into-LSP Tunnel
- Useful for services and applications requiring Traffic Steering of specific flows into a programmed WAN resource

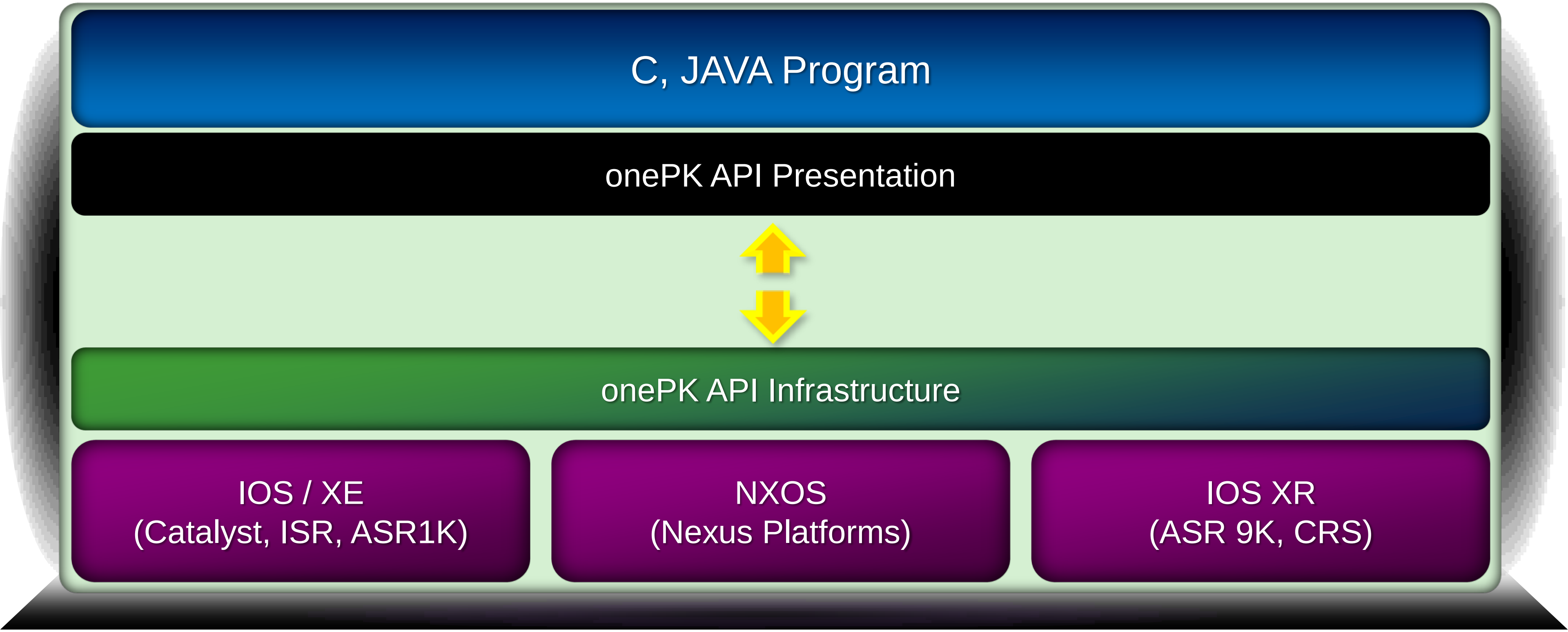


Enter I2RS

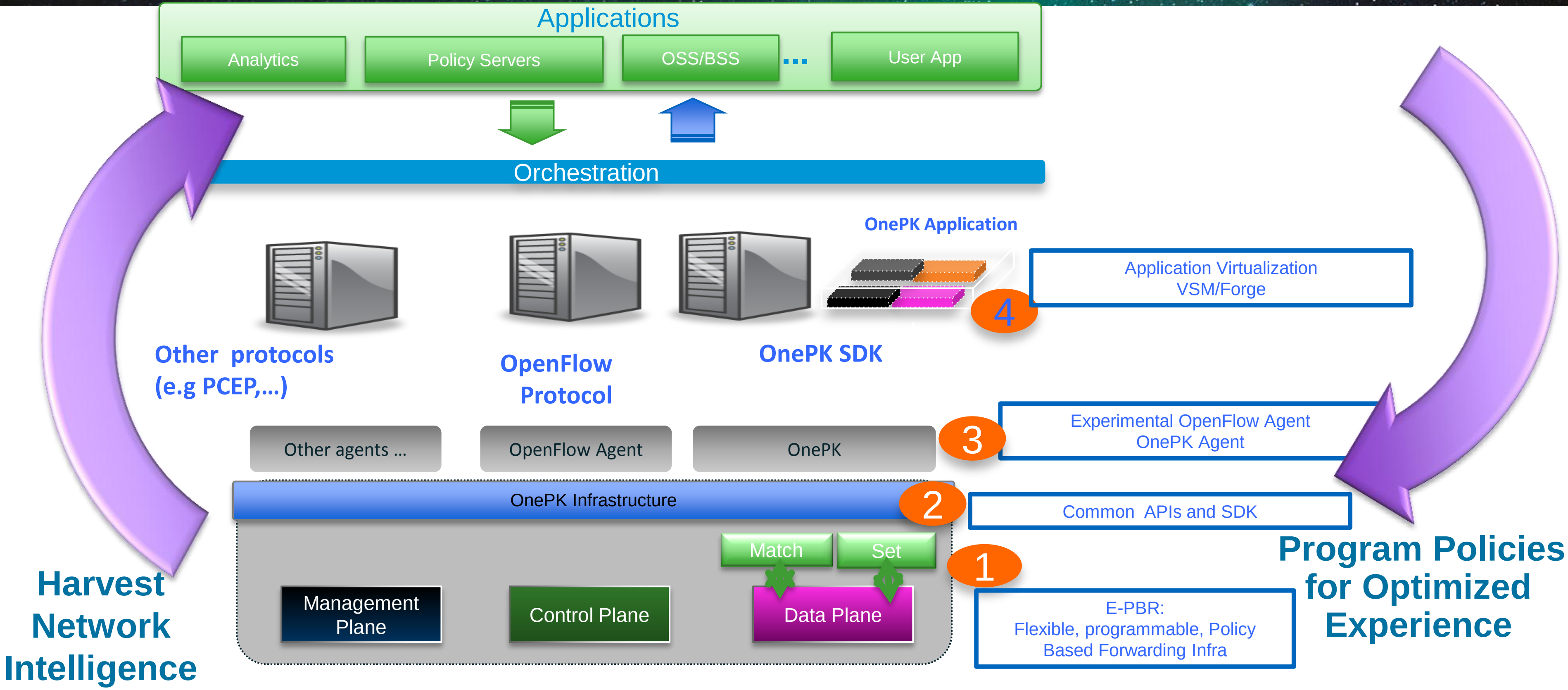
- Interface to the Routing System
- Framework for a common, standard interface enabling programmatic access to information maintained inside a router
  - e.g. RIB, interface, stats, policy
- Key aspects are:
  - Interface must be fast, async, bidirectional
  - Access to state/information/events not normally available for configurable via existing methods
  - Focus on YANG as the data model language (RFC602, used in Netconf), draft-rfernando-i2rs-yang-mods
- http://datatracker.ietf.org/wg/i2rs/



Cisco ONE (Open Networking Environment) onePK Architecture



Example: OnePK on ASR9000



Applications

Analytics

Policy Servers

OSS/BSS

...

User App

Orchestration

OnePK Application

Other protocols (e.g PCEP,...)

OpenFlow Protocol

OnePK SDK

Other agents ...

OpenFlow Agent

OnePK

OnePK Infrastructure

Match

Set

Management Plane

Control Plane

Data Plane

4 Application Virtualization VSM/Forge

3 Experimental OpenFlow Agent OnePK Agent

2 Common APIs and SDK

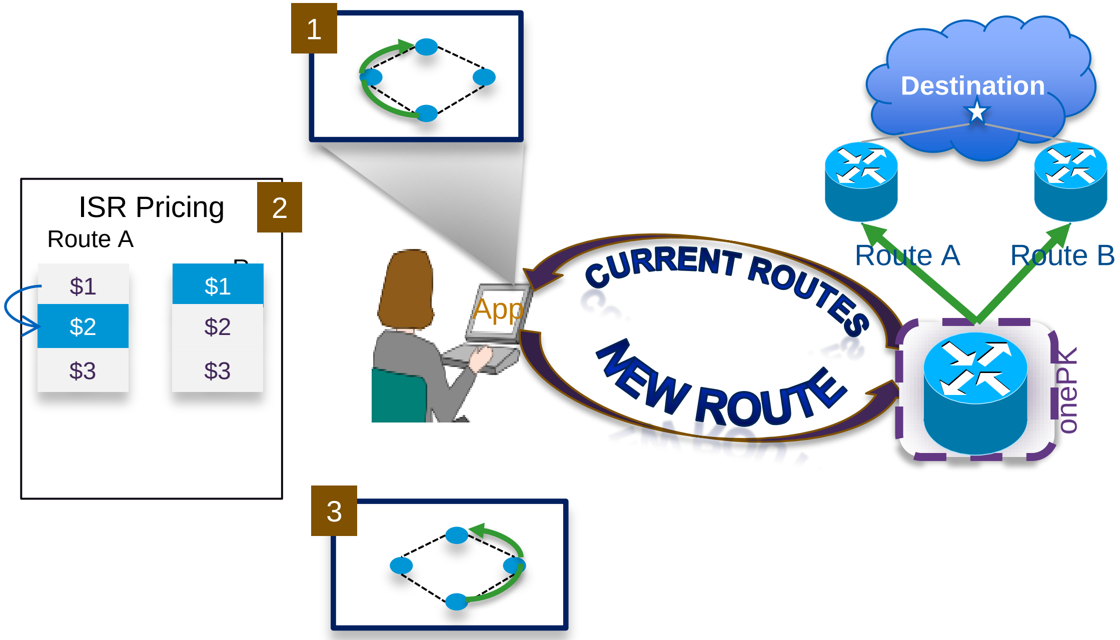
1 E-PBR: Flexible, programmable, Policy Based Forwarding Infra

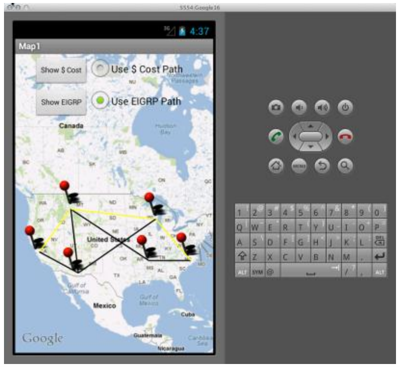
Harvest Network Intelligence

Program Policies for Optimized Experience

Classical SDN Use Case: Custom Routing

Example: Data Center Traffic Forwarding Based on a Custom Algorithm



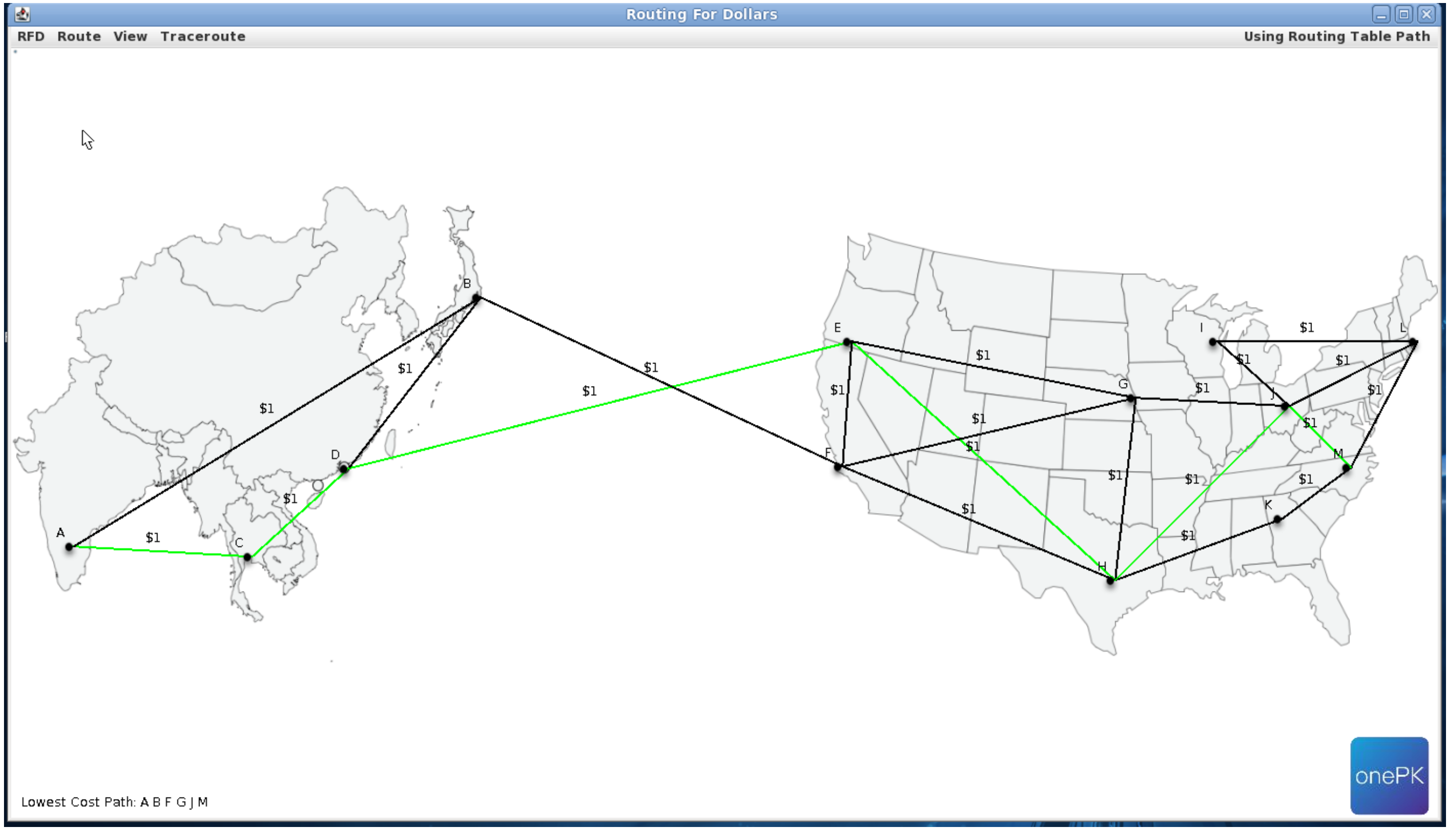


- Lowest Cost
- Lowest Delay (with IP SLA real-time information)

Unique Data Forwarding Algorithm Highly Optimized for the Network Operator’s Application

Custom Routing

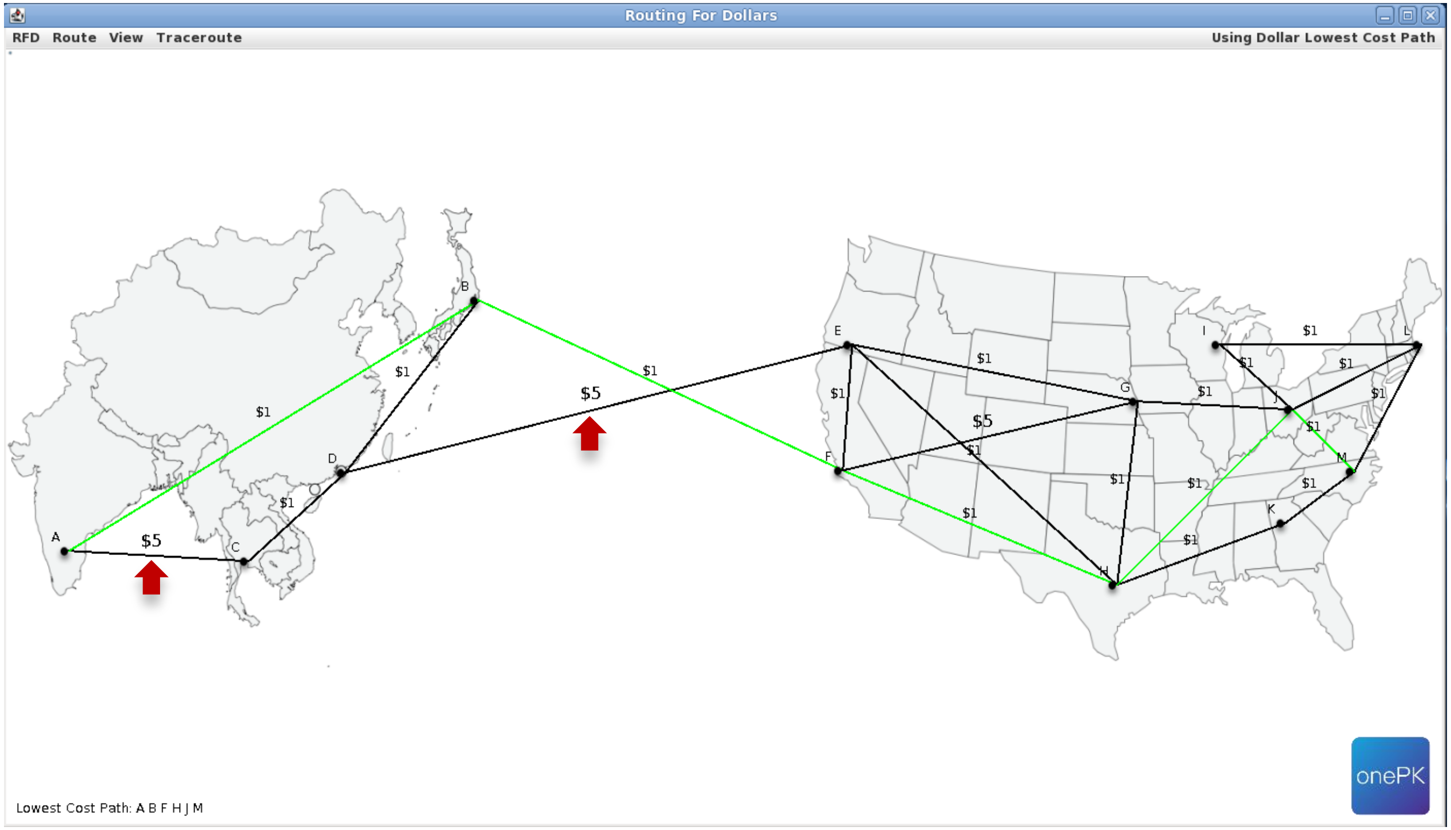
Initial Setup: Default routing using IGP (shortest path)



Lowest Cost Path: A B F G J M

Custom Routing

Routing for Dollars: Application driven routes installed in network

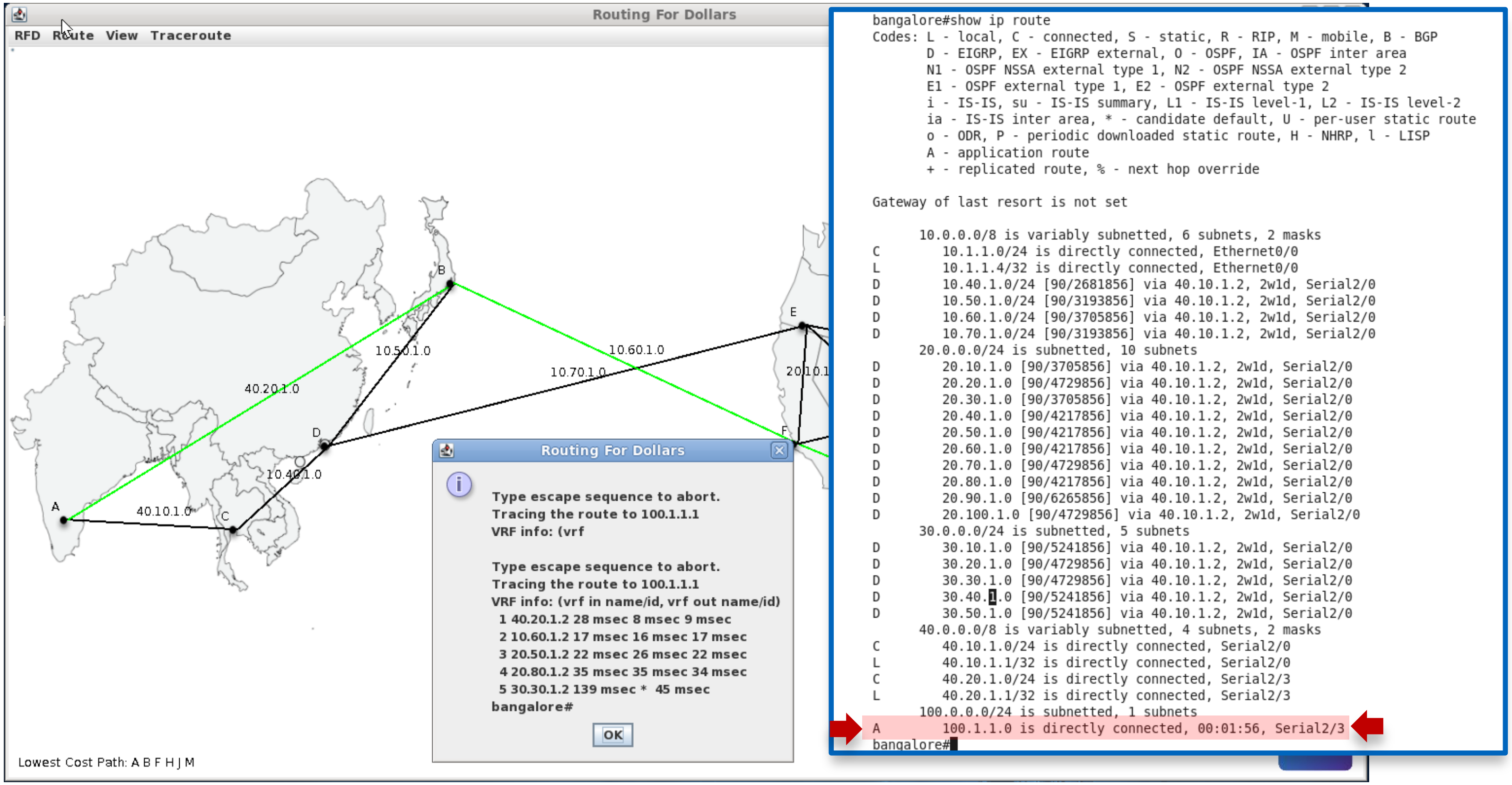


Using Dollar Lowest Cost Path

Lowest Cost Path: A B F H J M

Custom Routing

Tracing the application installed route – using the developer and element services



```
Type escape sequence to abort.
Tracing the route to 100.1.1.1
VRF info: (vrf

Type escape sequence to abort.
Tracing the route to 100.1.1.1
VRF info: (vrf in name/id, vrf out name/id)
  1 40.20.1.2 28 msec 8 msec 9 msec
  2 10.60.1.2 17 msec 16 msec 17 msec
  3 20.50.1.2 22 msec 26 msec 22 msec
  4 20.80.1.2 35 msec 35 msec 34 msec
  5 30.30.1.2 139 msec *  45 msec
bangalore#
```

```
bangalore#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       A - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C        10.1.1.0/24 is directly connected, Ethernet0/0
L        10.1.1.4/32 is directly connected, Ethernet0/0
D        10.40.1.0/24 [90/2681856] via 40.10.1.2, 2w1d, Serial2/0
D        10.50.1.0/24 [90/3193856] via 40.10.1.2, 2w1d, Serial2/0
D        10.60.1.0/24 [90/3705856] via 40.10.1.2, 2w1d, Serial2/0
D        10.70.1.0/24 [90/3193856] via 40.10.1.2, 2w1d, Serial2/0
      20.0.0.0/24 is subnetted, 10 subnets
D        20.10.1.0 [90/3705856] via 40.10.1.2, 2w1d, Serial2/0
D        20.20.1.0 [90/4729856] via 40.10.1.2, 2w1d, Serial2/0
D        20.30.1.0 [90/3705856] via 40.10.1.2, 2w1d, Serial2/0
D        20.40.1.0 [90/4217856] via 40.10.1.2, 2w1d, Serial2/0
D        20.50.1.0 [90/4217856] via 40.10.1.2, 2w1d, Serial2/0
D        20.60.1.0 [90/4217856] via 40.10.1.2, 2w1d, Serial2/0
D        20.70.1.0 [90/4729856] via 40.10.1.2, 2w1d, Serial2/0
D        20.80.1.0 [90/4217856] via 40.10.1.2, 2w1d, Serial2/0
D        20.90.1.0 [90/6265856] via 40.10.1.2, 2w1d, Serial2/0
D        20.100.1.0 [90/4729856] via 40.10.1.2, 2w1d, Serial2/0
      30.0.0.0/24 is subnetted, 5 subnets
D        30.10.1.0 [90/5241856] via 40.10.1.2, 2w1d, Serial2/0
D        30.20.1.0 [90/4729856] via 40.10.1.2, 2w1d, Serial2/0
D        30.30.1.0 [90/4729856] via 40.10.1.2, 2w1d, Serial2/0
D        30.40.1.0 [90/5241856] via 40.10.1.2, 2w1d, Serial2/0
D        30.50.1.0 [90/5241856] via 40.10.1.2, 2w1d, Serial2/0
      40.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        40.10.1.0/24 is directly connected, Serial2/0
L        40.10.1.1/32 is directly connected, Serial2/0
C        40.20.1.0/24 is directly connected, Serial2/3
L        40.20.1.1/32 is directly connected, Serial2/3
      100.0.0.0/24 is subnetted, 1 subnets
A        100.1.1.0 is directly connected, 00:01:56, Serial2/3
bangalore#
```

Custom Routing: Statistics

- Code Metrics
  - Total lines of code: 4700 (JAVA)
  - 40% SWING GUI
  - 20% Dijkstra’s algorithm, lowest cost path determination
  - 25% Housekeeping: Node and link database
  - 15% Calls to onePK infrastructure + error checking

Framework makes it easy to modify code and change business logic.

- Code increase to add “Latency based routing” on top of “Routing for Dollars”
  - 100 lines of code

Modular java code makes it easy to deploy on multiple clients.

- Modular code base written in Java has allowed us to port this to mobility client.

Network Optimization and Segment Routing

Network Optimization

❑Network Engineering

- Manipulating your network to suit your traffic
  - Typically based on Link utilization (Intf MIB), sometimes Class utilization (QoS MIB)

❑Traffic Engineering

- Manipulating your traffic to suit your network
  - More complex inputs – Traffic Matrix

❑MPLS TE is an unsuccessful technology → IETF looks into alternatives

- 85% MPLS networks - no TE
- 7% - tactical TE, no bw resv, few tunnels, static route
- 8% - strategic RE, bw reservations → case for WanO (eg. google, global-xings, linx)

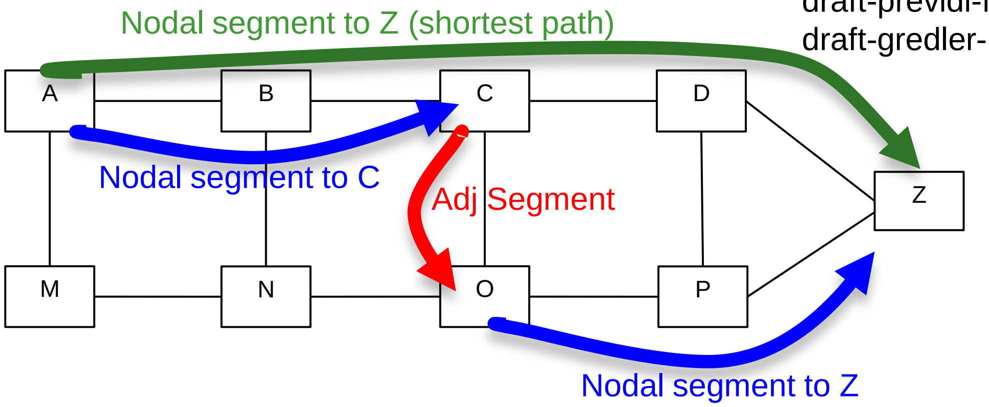
❑post-Moore era may bring the real need for TE → get ready!

MPLS Segment Routing Overview

draft-previdi-filsfils-isis-segment-routing
draft-gredler-rtgwg-igp-label-advertisement

Nodal segment : a shortest-path to the related node

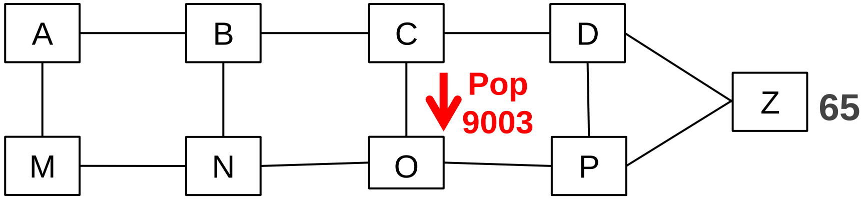
Adjacency segment: one-hop through the related adjacency



- Emergence of Stateless MPLS
- Simplification – label distribution via IGP; no need for LDP and RSVP
- Scale – less state for routers to maintain to maintain
- Combined with SDN WAN Platform controller for path computation and programming
- Backward compatible with existing networks

The state is no longer in the network, it’s in the packet.

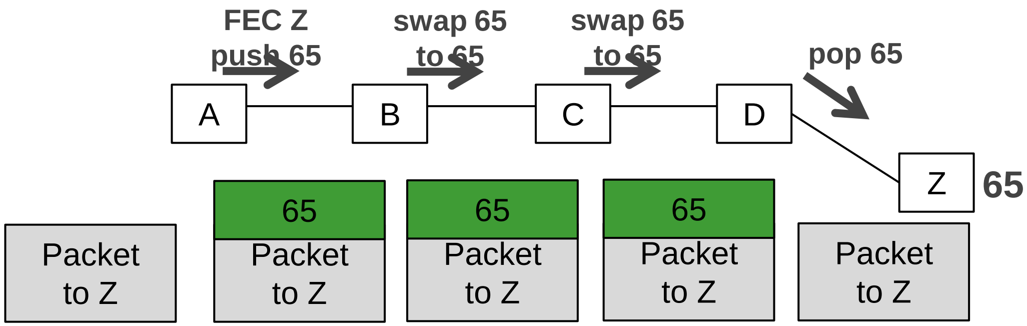
Adjacency Segment



A packet injected at node C with label 9003 is forced through datalink CO

- C allocates a local label
- C advertises the adjacency label in ISIS
  - simple sub-TLV extension
- C is the only node to install the adjacency segment in the MPLS dataplane
- We can construct an explicit-path from adjacency segments (labels), but this is not the point

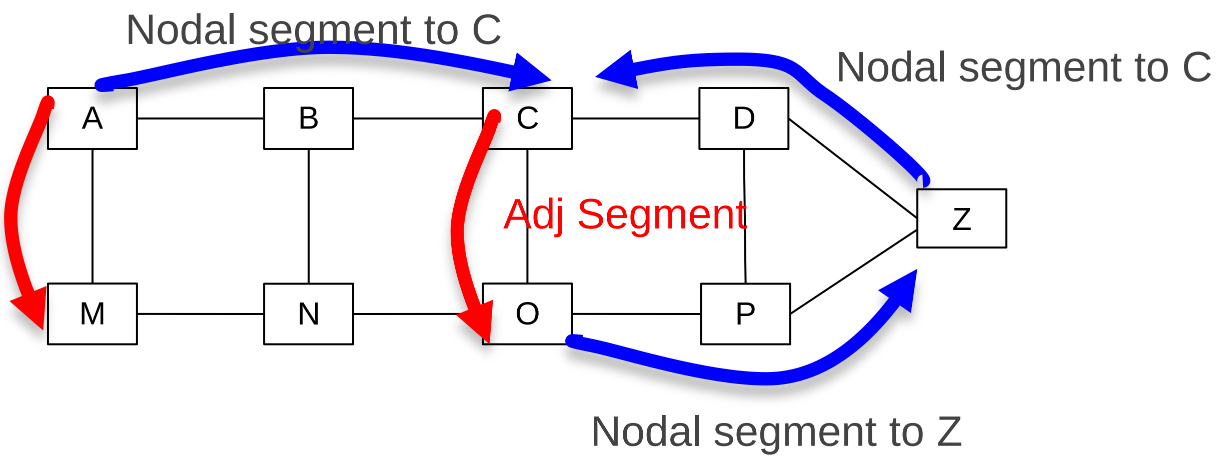
Node Segment



A packet injected anywhere with top label 65 will reach Z via shortest-path

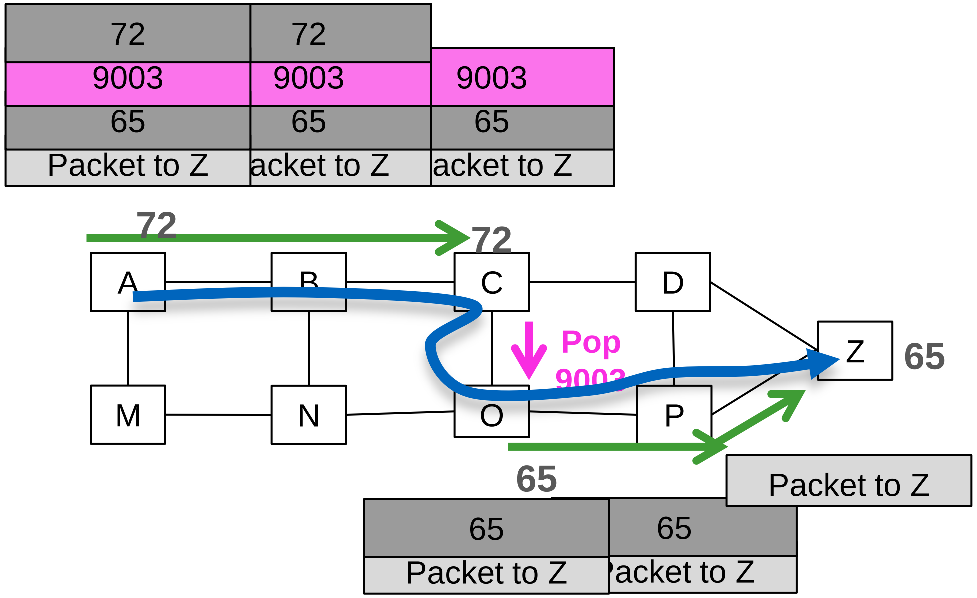
- Z advertises its node segment
  - simple ISIS sub-TLV extension
- All remote nodes install the node segment to Z in the MPLS dataplane
  - only 1 label per node in IGP domain (insignificant: < 1% of label space)
- Node SR Range (eg. global MPLS labels)
  - a range of labels allocated to the SR control-plane, e.g. [64, 5000]
- Each node gets one unique label from SR Range

ISIS/OSPF automatically installs segments



- Simple extension
- Excellent Scale: a node installs N+A FIB entries
  - N node segments and A adjacency segments

Combining Segments

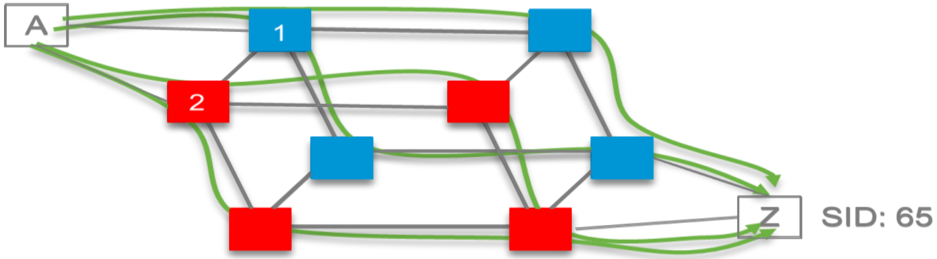


- Source Routing
- Any explicit path can be expressed: ABCOPZ

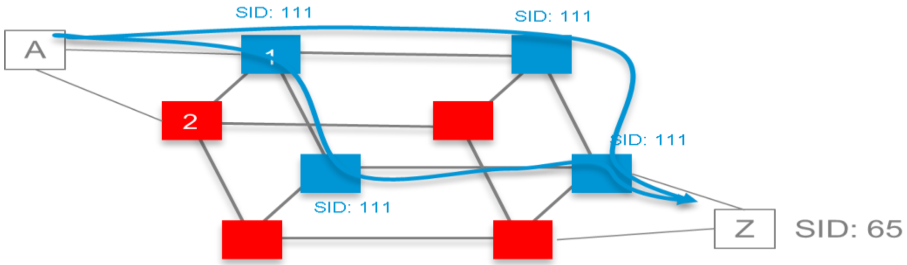
Simple Disjointness with Segment Routing

- SR avoids state in the core
- SR avoids enumerating RSVP-TE tunnels for each ECMP paths

- A sends traffic with [65]
  - Classic ECMP

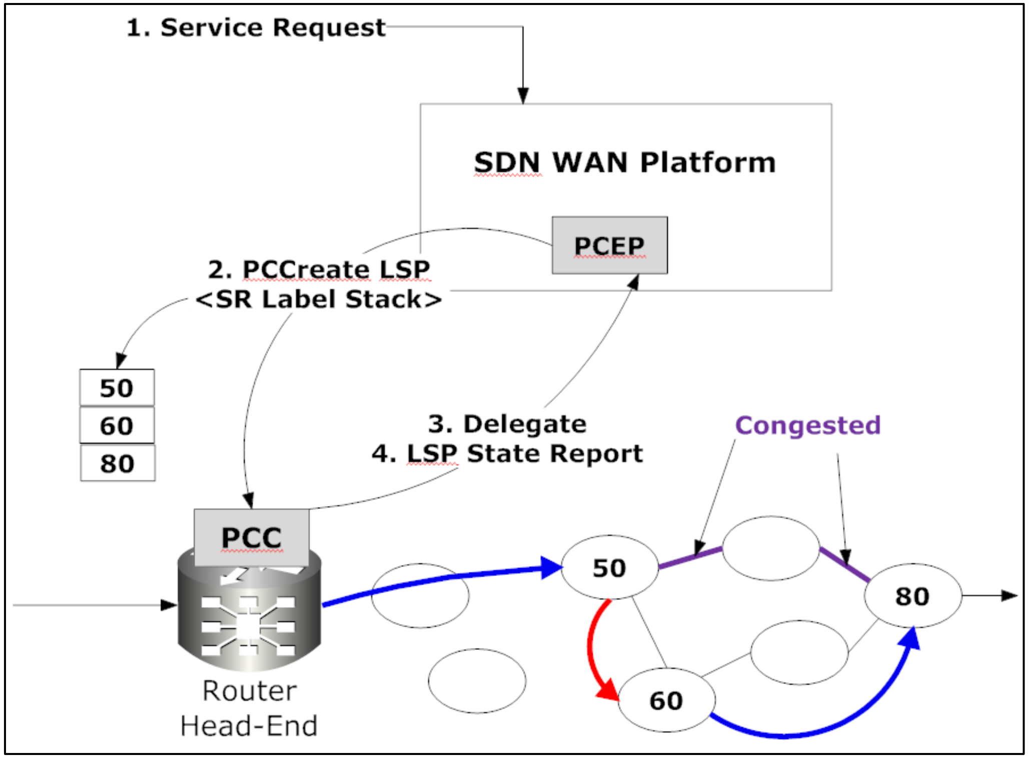


- A sends traffic with [111, 65]
  - Packet gets attracted in blue plane and then uses classic ECMP



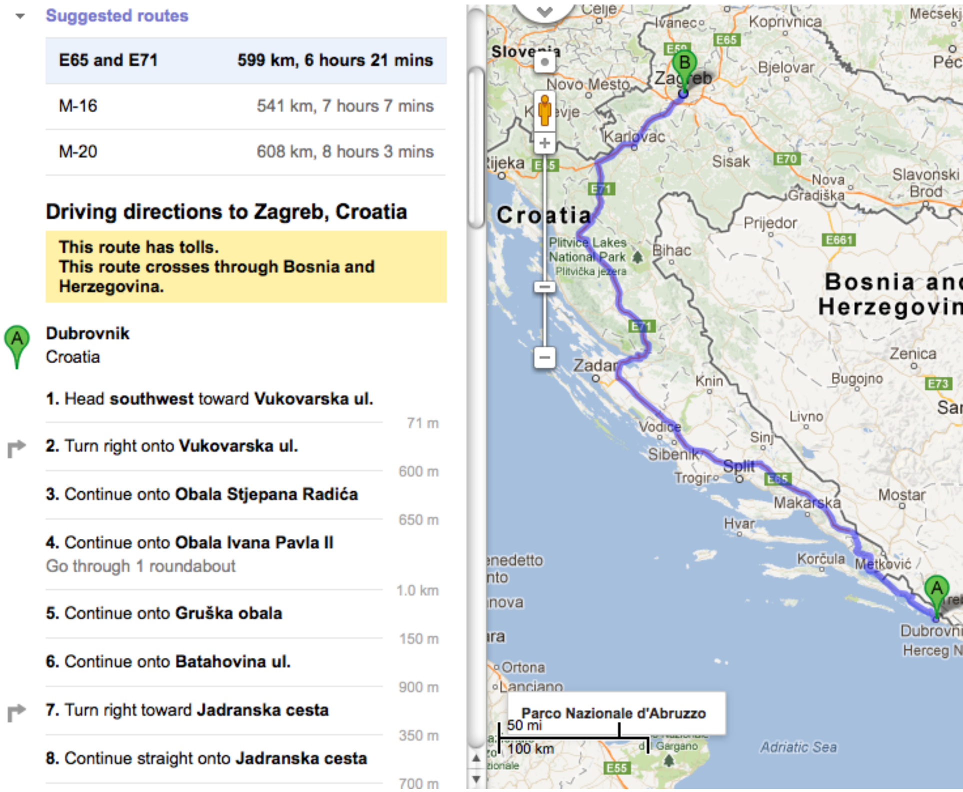
Stateful PCE Programming of Explicit SR Paths

- PCE knows topology and node/adj segment IDs via BGP-LS
- Computes path that avoids congested links (based on service request constraints)
- PCEP extensions needed to program SR path (label stack) in router
  - SR path (label stack prepended to each packet)
- No RSVP-TE signaling needed



Solves MPLS Operator Challenges

- Simplicity
  - less protocols to operate & troubleshoot
  - no LDP sessions between routers
  - deliver automated FRR for any topology
- Scale
  - avoid millions of labels in LDP database
  - avoid millions of TE LSP’s in the network
  - avoid millions of tunnels to configure
- Simple to deploy and operate
  - coexistence, incremental deploymet
  - MPLS: segment = label (push, pop, swap)
  - Same behavior – ECMP, PHP, LFA…



Suggested routes

E65 and E71	**599 km, 6 hours 21 mins**
M-16	541 km, 7 hours 7 mins
M-20	608 km, 8 hours 3 mins

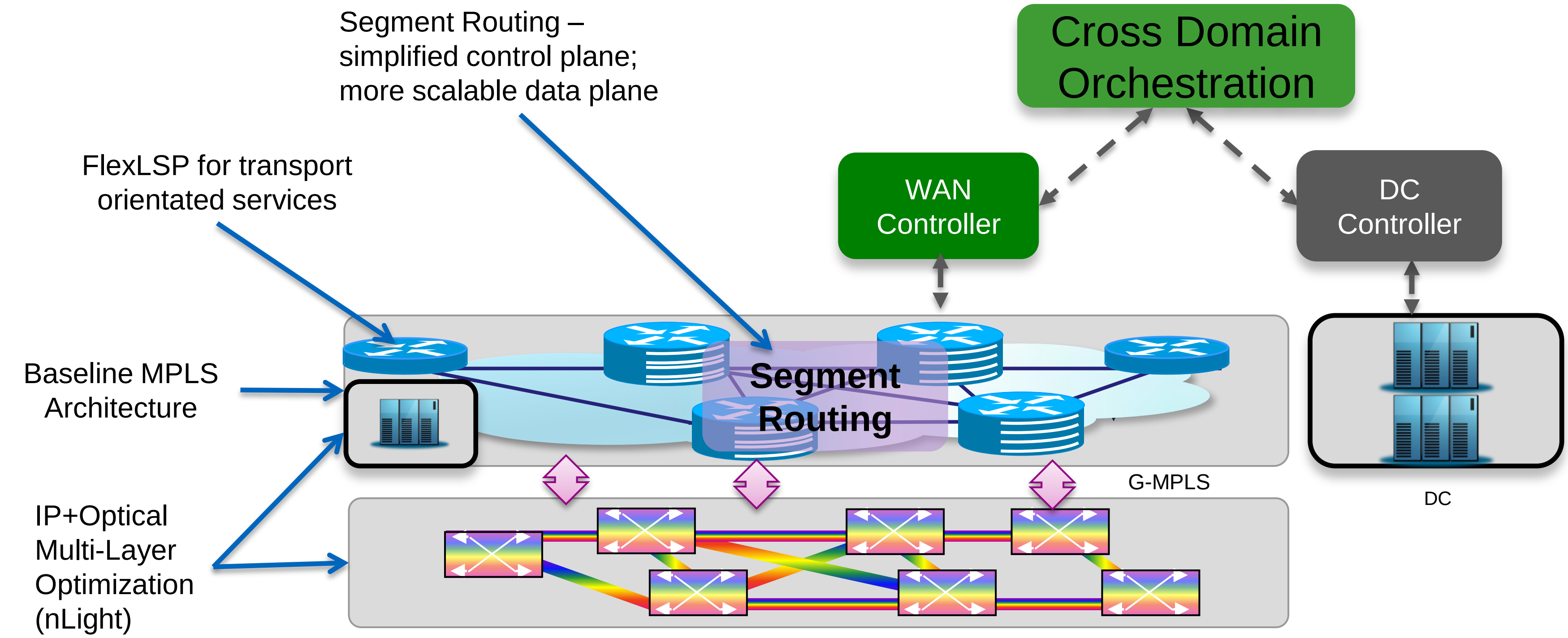
Driving directions to Zagreb, Croatia

This route has tolls.
This route crosses through Bosnia and Herzegovina.

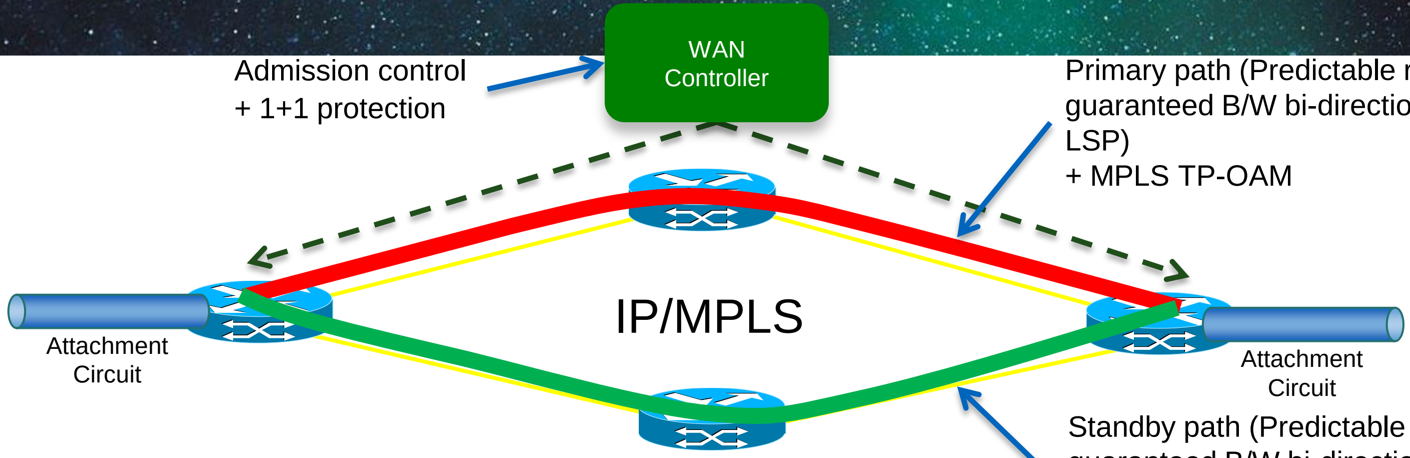
Dubrovnik
Croatia

1. Head **southwest** toward **Vukovarska ul.** — 71 m
2. Turn right onto **Vukovarska ul.** — 600 m
3. Continue onto **Obala Stjepana Radića** — 650 m
4. Continue onto **Obala Ivana Pavla II**
   Go through 1 roundabout — 1.0 km
5. Continue onto **Gruška obala** — 150 m
6. Continue onto **Batahovina ul.** — 900 m
7. Turn right toward **Jadranska cesta** — 350 m
8. Continue straight onto **Jadranska cesta** — 700 m

Evolution of MPLS → "MPLSDN"


Segment Routing –
simplified control plane;
more scalable data plane
Cross Domain Orchestration
FlexLSP for transport orientated services
WAN Controller
DC Controller
Baseline MPLS Architecture
Segment Routing
G-MPLS
DC
IP+Optical Multi-Layer Optimization (nLight)

FlexLSP: Simple, Orchestrated, Unified Transport



Admission control
+ 1+1 protection

Primary path (Predictable route, guaranteed B/W bi-directional associated LSP)
+ MPLS TP-OAM

Standby path (Predictable route, guaranteed B/W bi-directional associated LSP)
+ MPLS TP-OAM

- IP/MPLS provides excellent support for connectionless services
- FlexLSP brings transport orientated services to IP/MPLS environments
- Bi-directional transport orientated tunnels supporting pseudo-wires
  - Predictable route, guaranteed B/W bi-directional associated LSP
  - MPLS-TP OAM monitoring LSP status and driving protection
- Programmatic VPN services enabling NfV

Benefit: 20-60% saving for transport services with FlexLSP vs. OTN

Summary

- SDN in SP Backbones
  - Simplification, Automation, Multi-Layer WAN Optimization
- WAN Controller
  - Add “network-wide” intelligence to “selfish” routers
- SP SDN Protocols
  - PCEP, Openflow, I2RS, BGP-LS… and OnePK API
- Segment Routing and MPLSDN
  - MPLS evolution to simplicity and scale

CISCO™