

SDN Workshop

Contact: training@apnic.net



Introduction to SDN

SDN Workshop

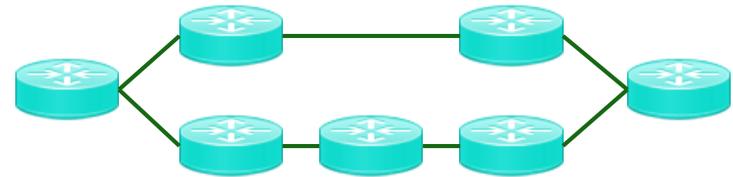


Overview

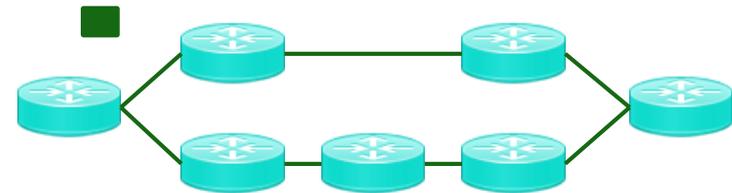
- Evolution of routers
- The Clean Slate project
- OpenFlow
- Emergence and evolution of SDN
- SDN architecture today
- Use cases
- Standards development
- Comparing and contrasting with NFV
- Guide to the workshop

Routers

- Two key roles:

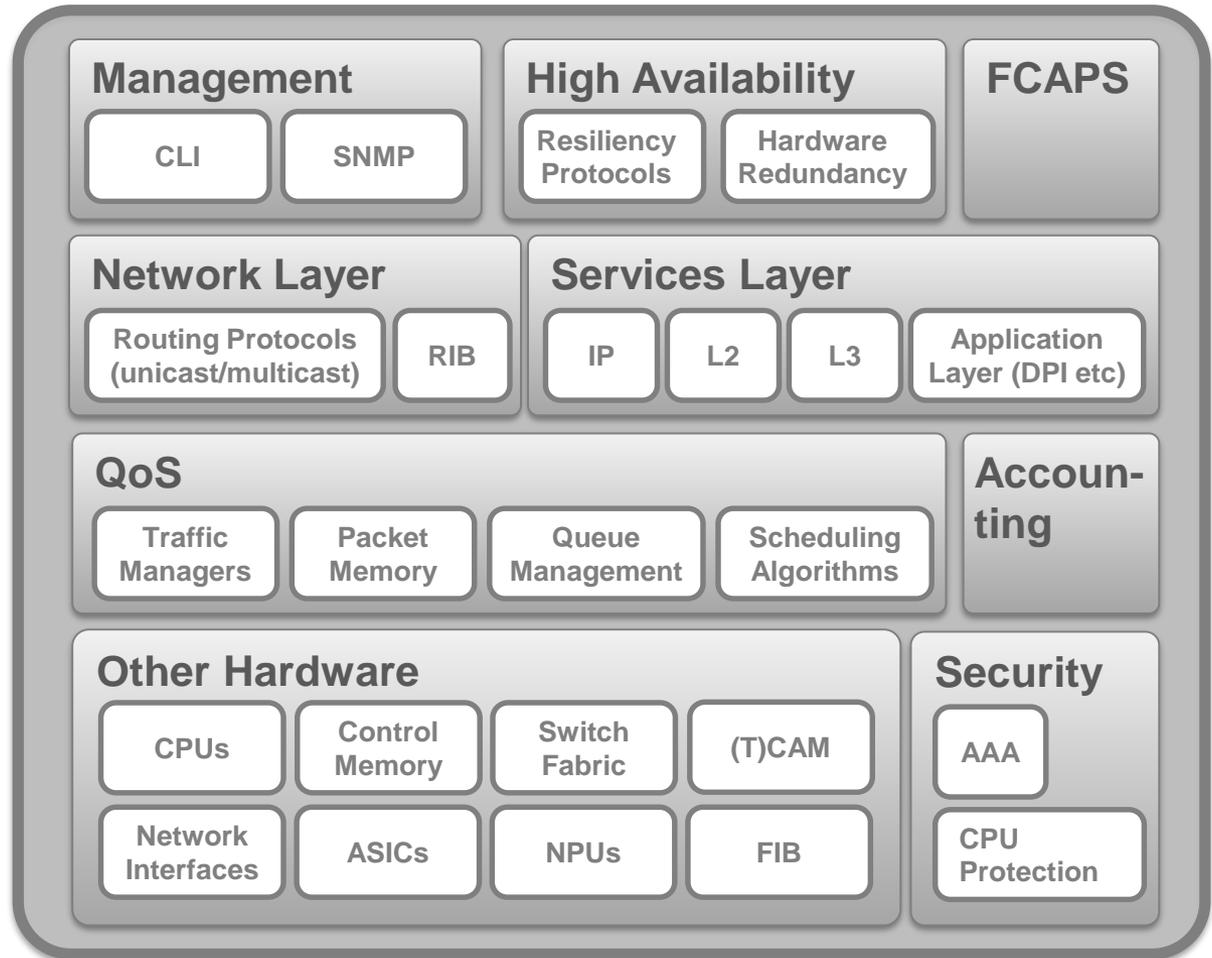


Determining Network Paths

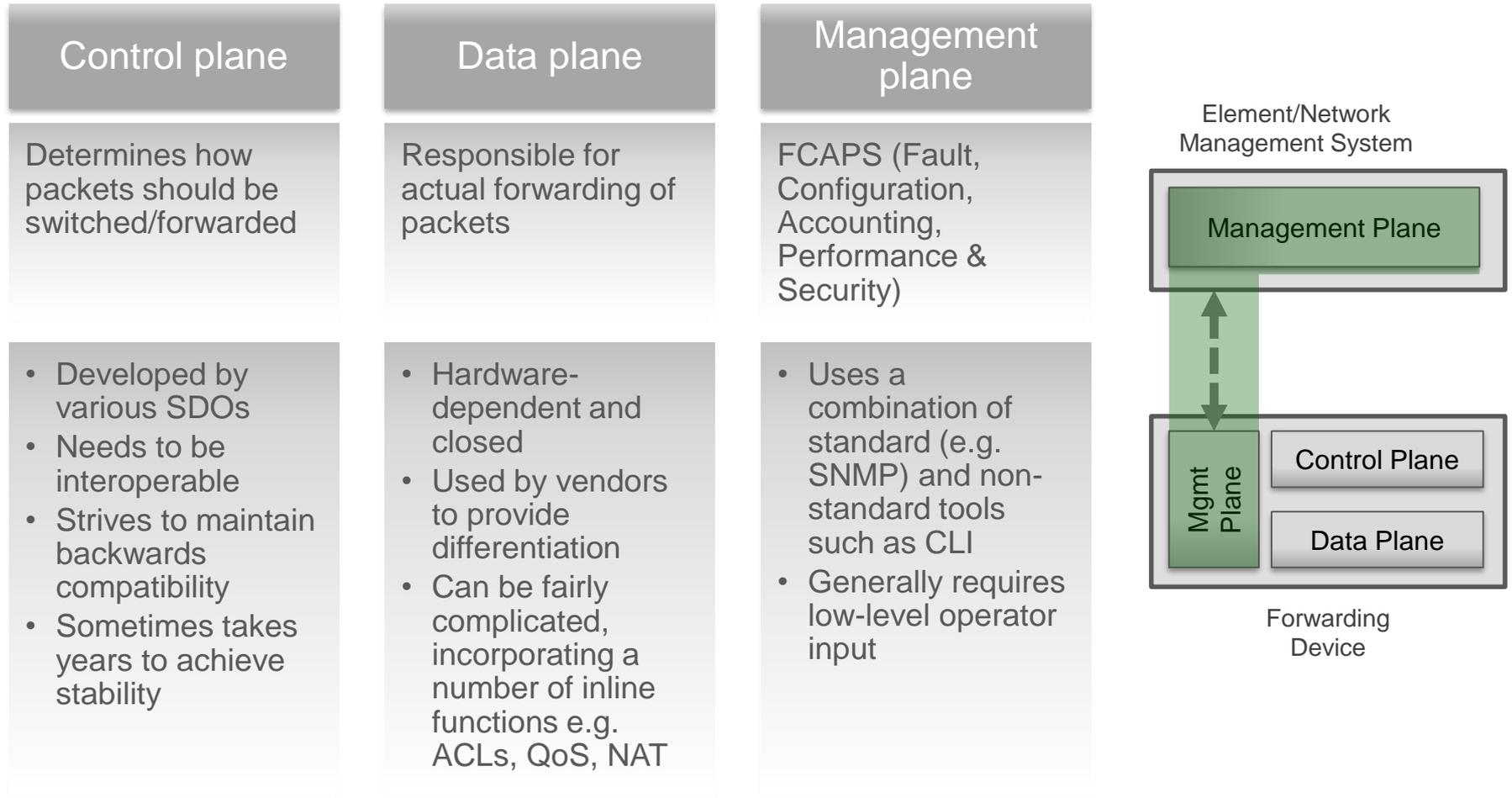


Packet forwarding

Today's router



Planes



How did we get here ?

Distribution of complexity

- 'End-to-end principle'
- Better scaling
- Survivability; spreading of risk

Backwards compatibility

- "Flag days" not realistic
- Short-term, incremental evolution of technology; no major overhaul in last 20 years

Unanticipated applications

- Networking is a victim of its own success
- New applications have been delivered on top of existing capabilities

Need for higher performance

- Tight coupling between different planes seen as critical for delivering higher performance

Clean Slate Project (1)

Two research questions:

With what we know today, if we were to start again with a clean slate, how would we design a global communications infrastructure

How should the Internet look in 15 years?

Mission: Re-invent the Internet

Clean Slate Project (2)

- One of the flagship projects was 'Internet Infrastructure: OpenFlow and Software Defined Networking'
- Seminal paper on OpenFlow...

OpenFlow: Enabling Innovation in Campus Networks

March 14, 2008

Nick McKeown
Stanford University

Tom Anderson
University of Washington

Hari Balakrishnan
MIT

Guru Parulkar
Stanford University

Larry Peterson
Princeton University

Jennifer Rexford
Princeton University

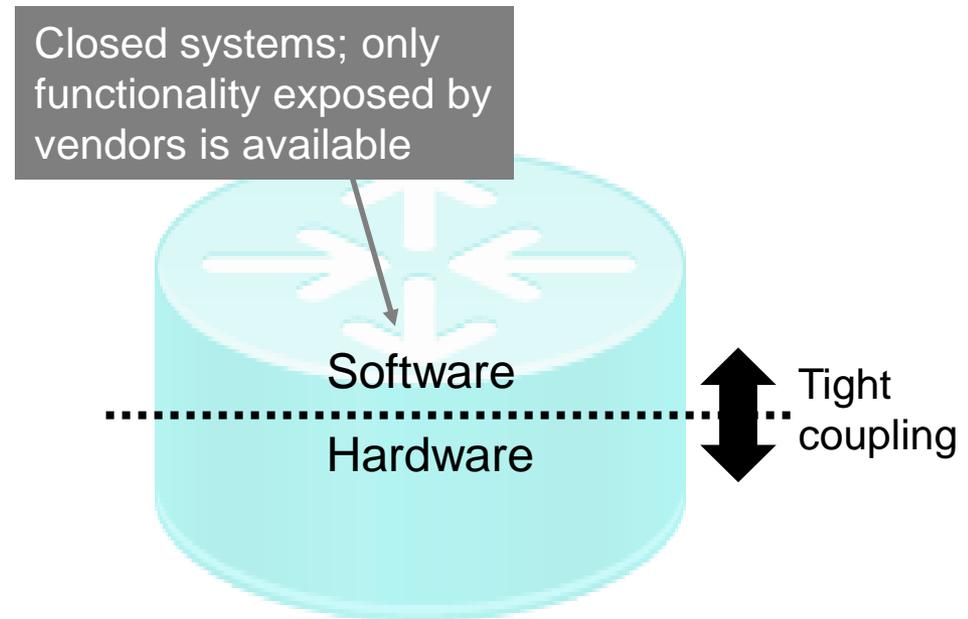
Scott Shenker
University of California,
Berkeley

Jonathan Turner
Washington University in
St. Louis

...kicked off the SDN movement and the data communications world would never be the same again

OpenFlow: The Problem

- Initial Problem:
 - A mechanism was required for researchers to run experimental network protocols.
 - Open software platforms did not provide the required performance and commercial solutions were too closed and inflexible.

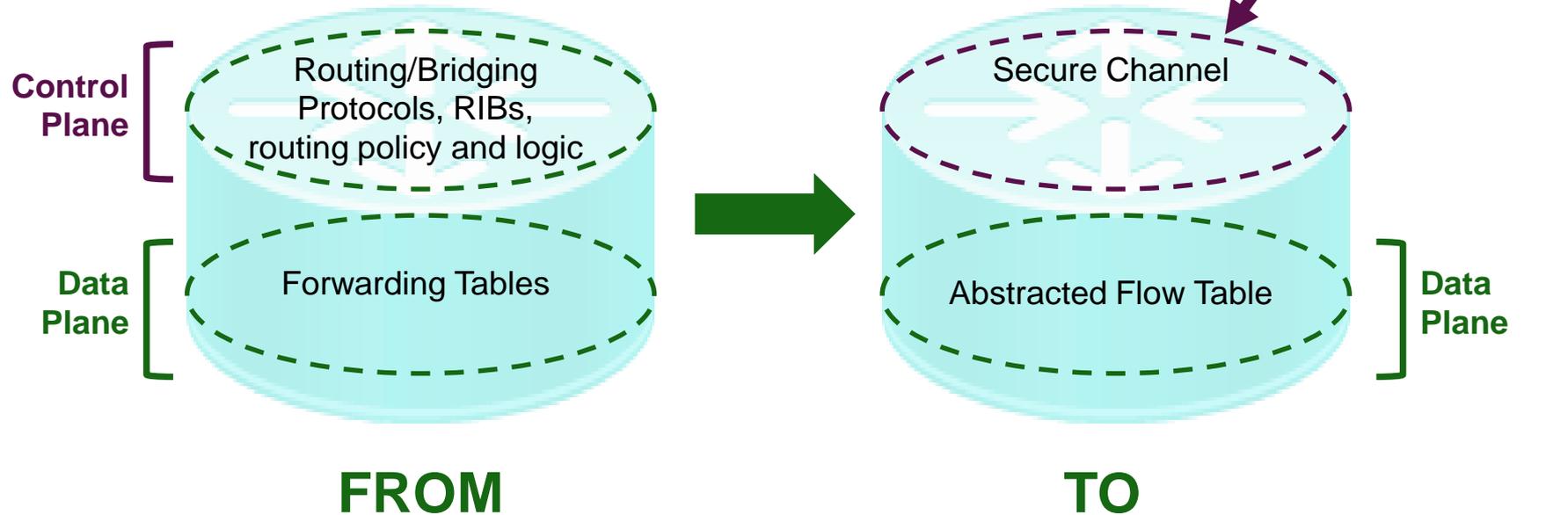


Challenge: how do we influence packet switching/forwarding behavior ?

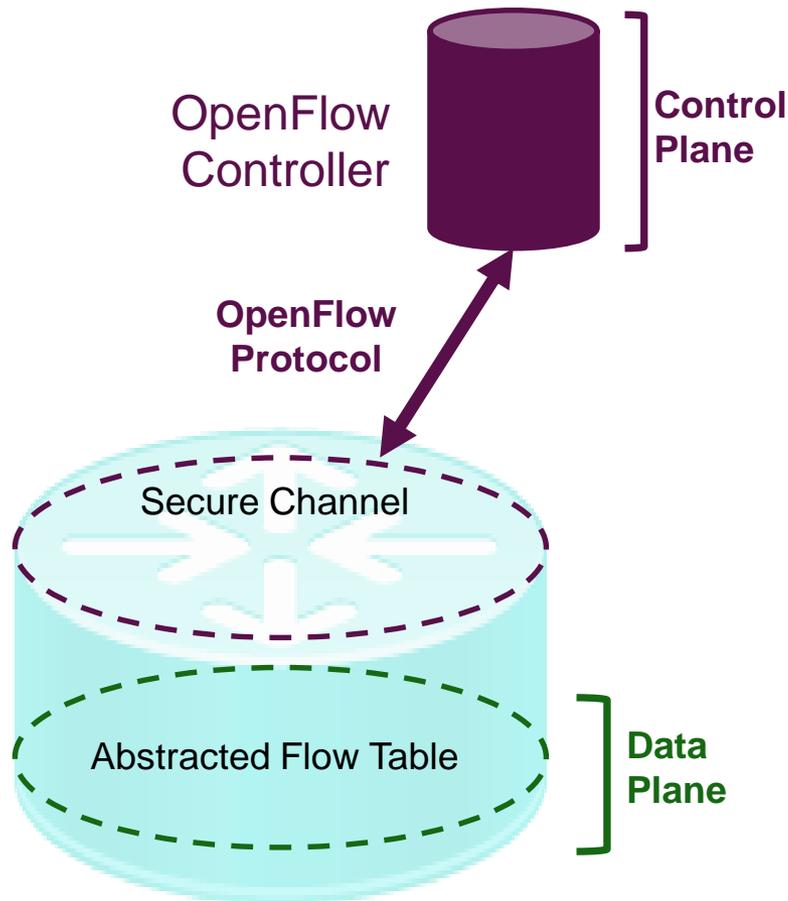
OpenFlow: The Solution (1)

Control Plane Protocols and algorithms to calculate forwarding paths

Data Plane Forwarding frames/packets based on paths calculated by control plane



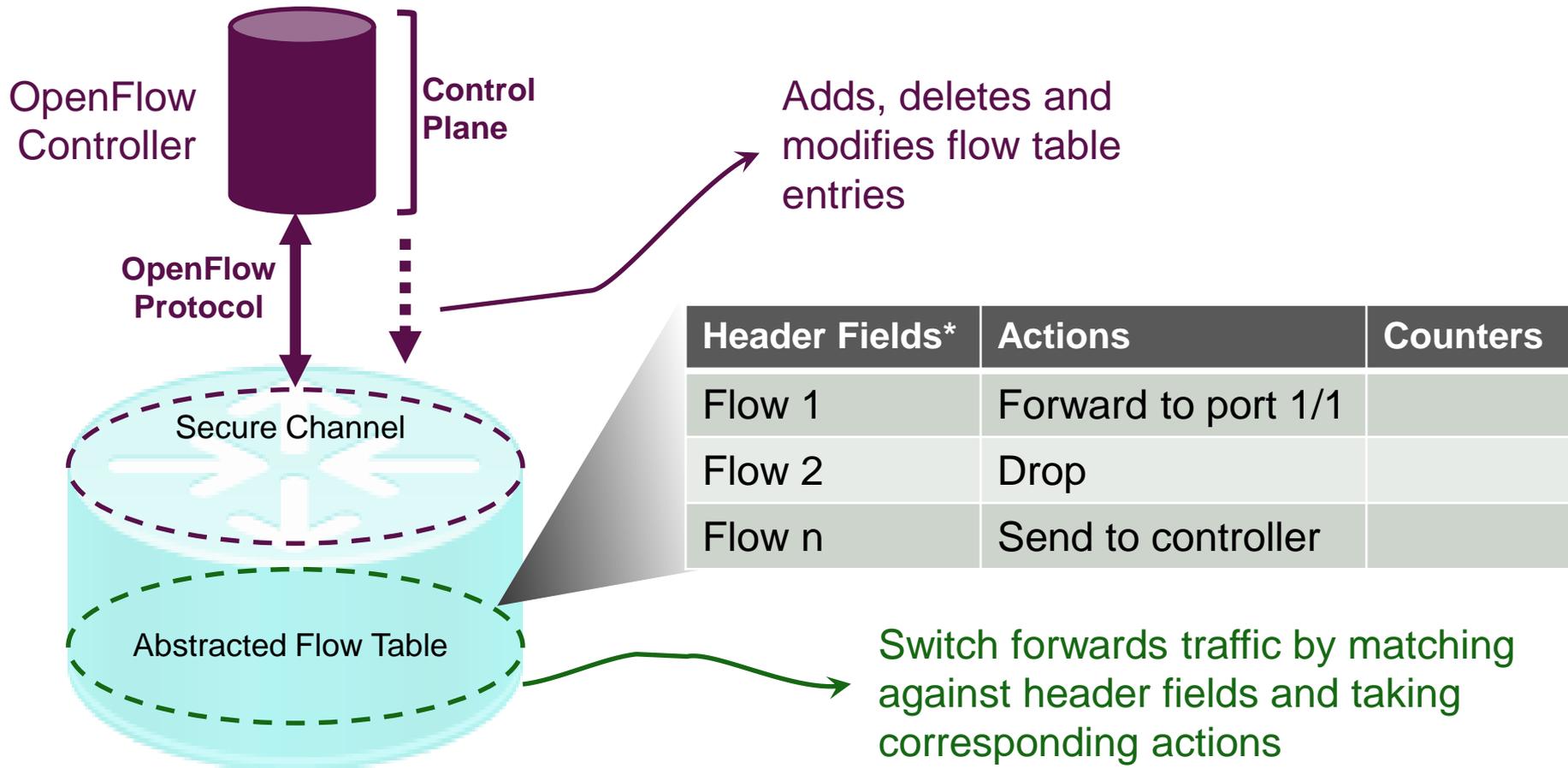
OpenFlow: The Solution (2)



The Solution:

- OpenFlow provided a compromise that provided a means of influencing switching/routing decisions without opening up network software.
- The control software would run on a controller; the outcomes of the calculations would be pushed down to the data plane running on the network element

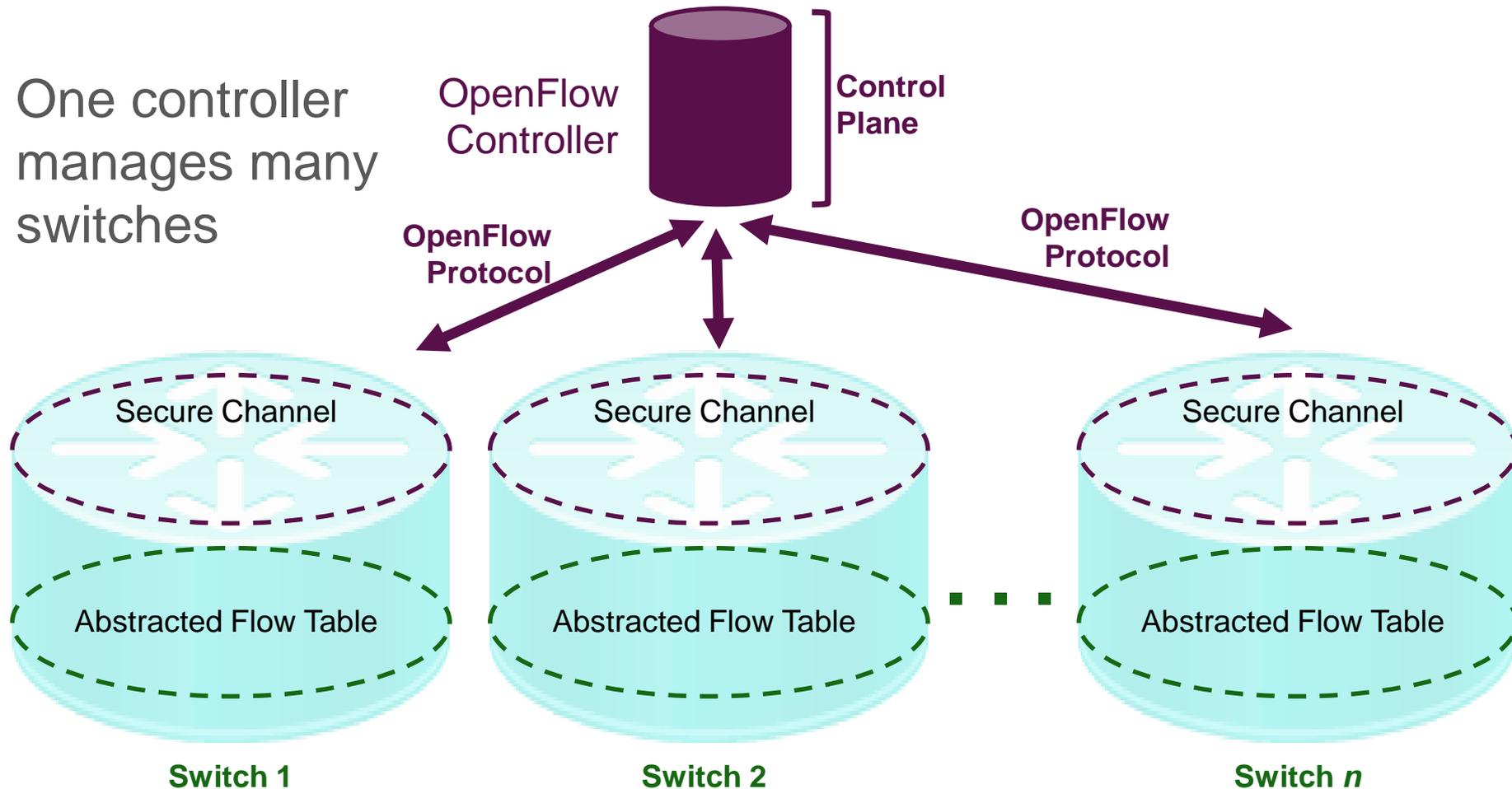
OpenFlow: How it works (1)



* Ingress Port, Ethernet SA, Ethernet DA, VLAN ID, VLAN PCP, IP SA, IP DA, IP Proto, IP ToS, Source L4 Port, Dest L2 Port etc....

OpenFlow: How it works (2)

One controller manages many switches



OpenFlow: Today

- Initially synonymous with SDN
- Today, OpenFlow is relegated to being just a part of the greater SDN architecture, with other protocols competing in the same space
- It is, however, responsible for the most radical paradigm shift in IP in recent times.

OpenFlow: Implications

- Two primary implications:

The control plane (processes to determine how traffic is handled) is physically **decoupled** from the data plane (forwards traffic according to decisions passed down by the control plane).

The control plane is **consolidated and centralised**: a single software control plane controls multiple data planes (previously a 1:1 correspondence).

Aside: challenges of data/control plane separation

Scalability

The control element now needs to be scaled to support a very large number of forwarding elements

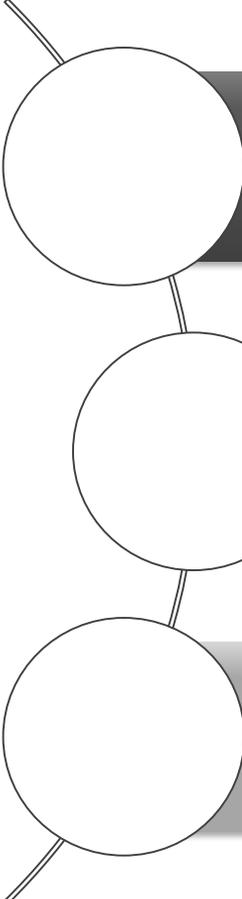
Reliability

The controller can NOT be a single point of failure (SPOF)

Consistency

When multiple controllers are used for redundancy consistency has to be assured across multiple replicas

The Birth of SDN



The separation of control and data plane was not an objective in itself but was a consequence of the compromise approach taken by OpenFlow

It heralded a new era of programmability that has been vastly enhanced with new architectures and capabilities

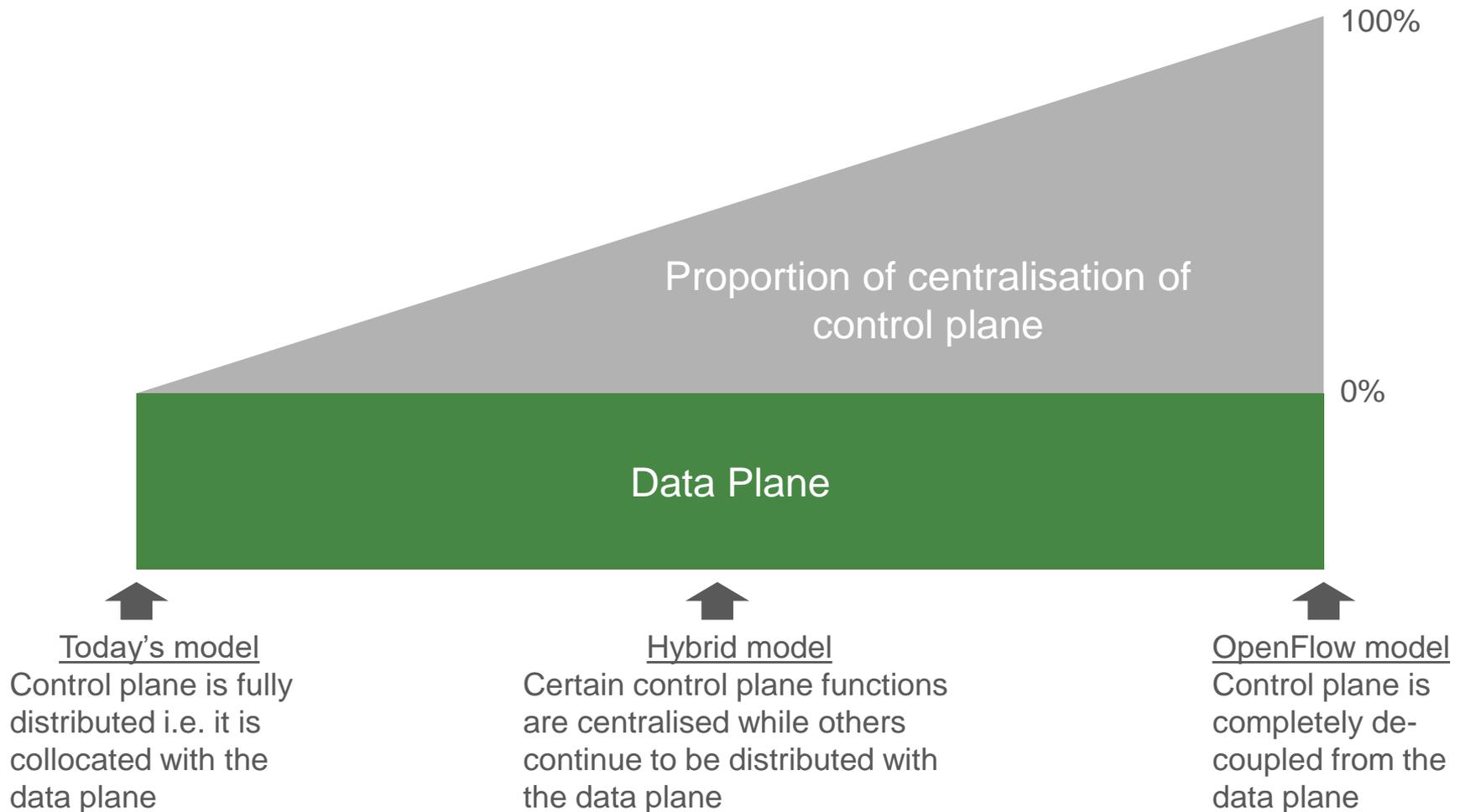
The term 'SDN' itself was coined in an article about the OpenFlow project at Stanford

(<http://www2.technologyreview.com/news/412194/tr10-software-defined-networking/>)

Emergence and evolution of SDN

- OpenFlow was a starting point...
 - Ushered in an era of programmability
 - But a complete decoupling of the control plane and data plane was not practical:
 - We would have had to solve all the problems the industry had spent decades solving and refining: resiliency, scalability, convergence, redundancy etc
- SDN architecture today
 - Hybrid approach where some elements of the control plane remain distributed while others are centralised.
 - Many different architectural models
 - All of them aspire to achieve the goals of agility and network programmability

Hybrid model of SDN



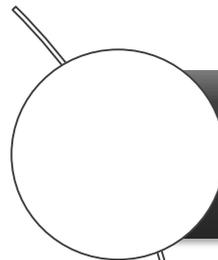
Defining SDN

ONF: The physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices.

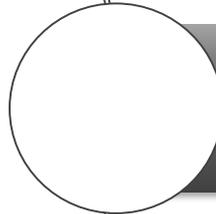
This definition is too narrow...

SDN is ...

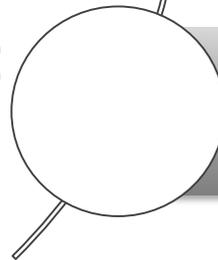
A new approach to networking that provides greater **network agility** and **flexibility** by:



Automation through enhanced **programmability** and **open** interfaces



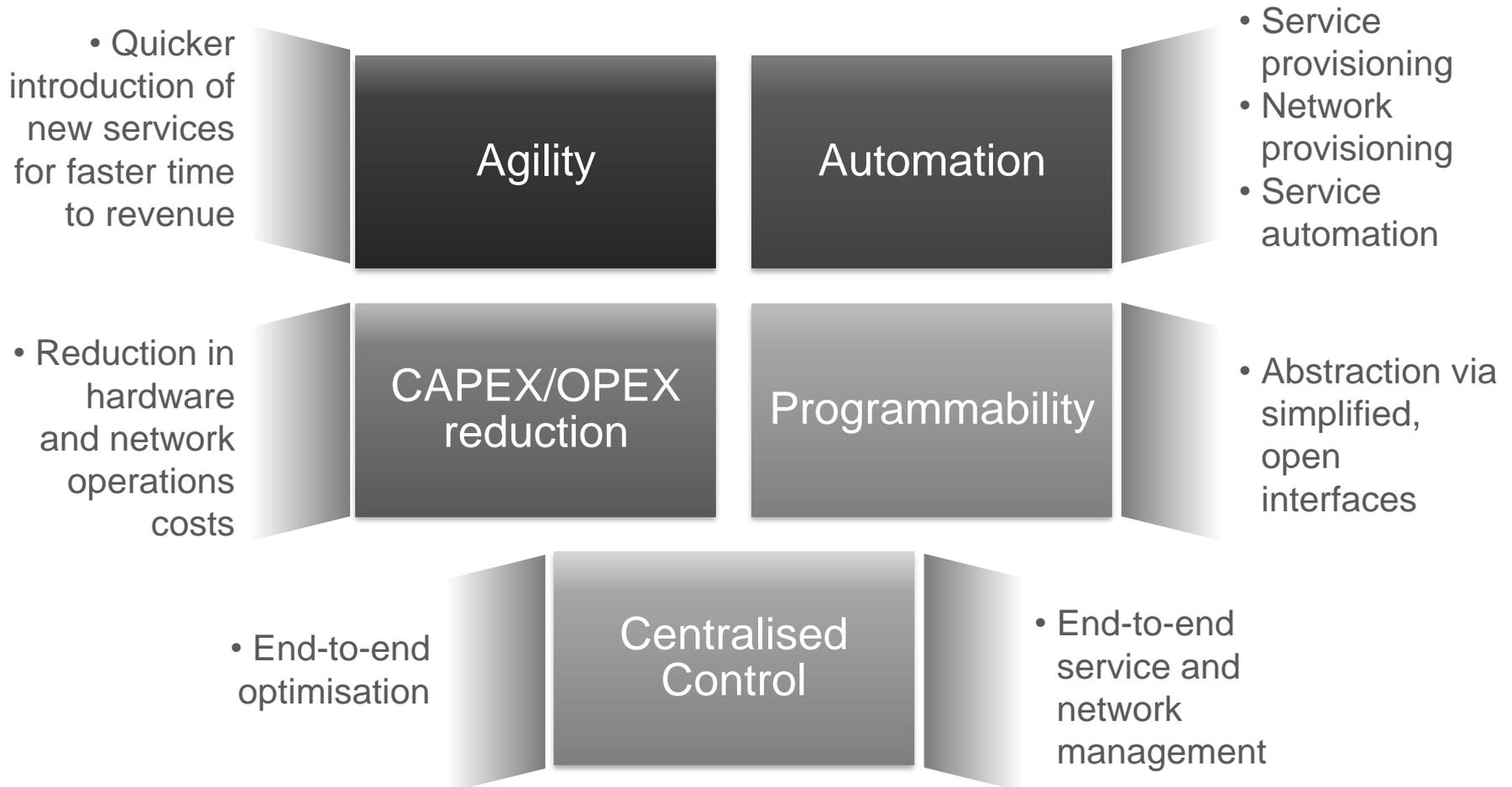
Dis-aggregation and **abstraction**



Centralisation of network control with real-time network **visibility**

As much a marketing term as a technical one

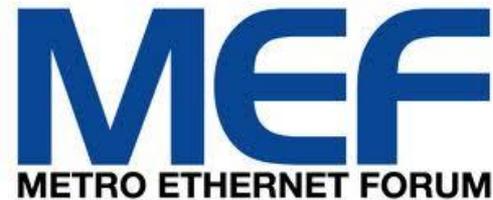
Objectives and benefits of SDN



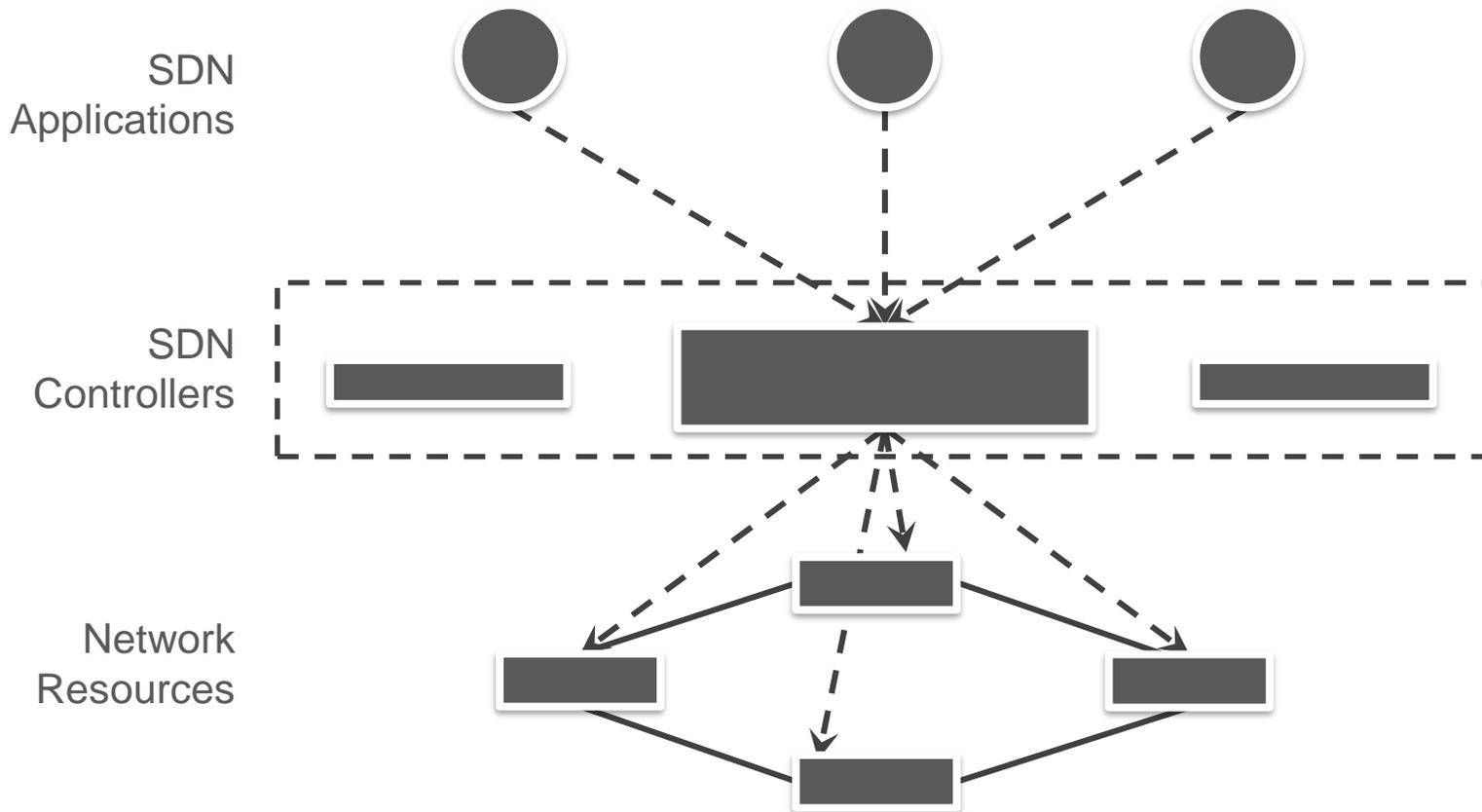
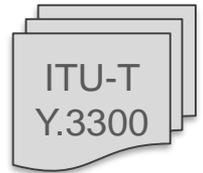
SDN SDOs



OPEN NETWORKING
FOUNDATION



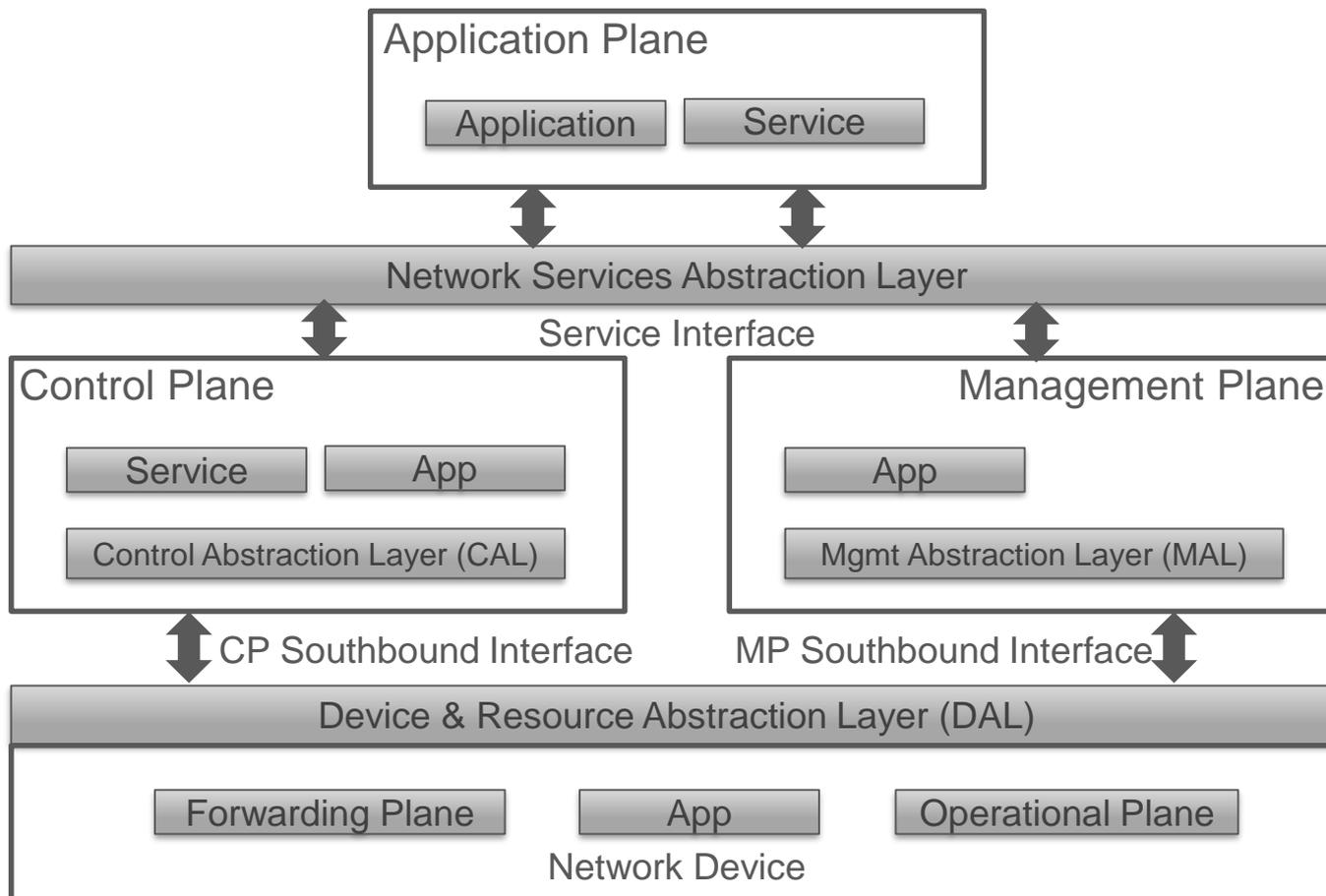
SDN architectural framework (1)



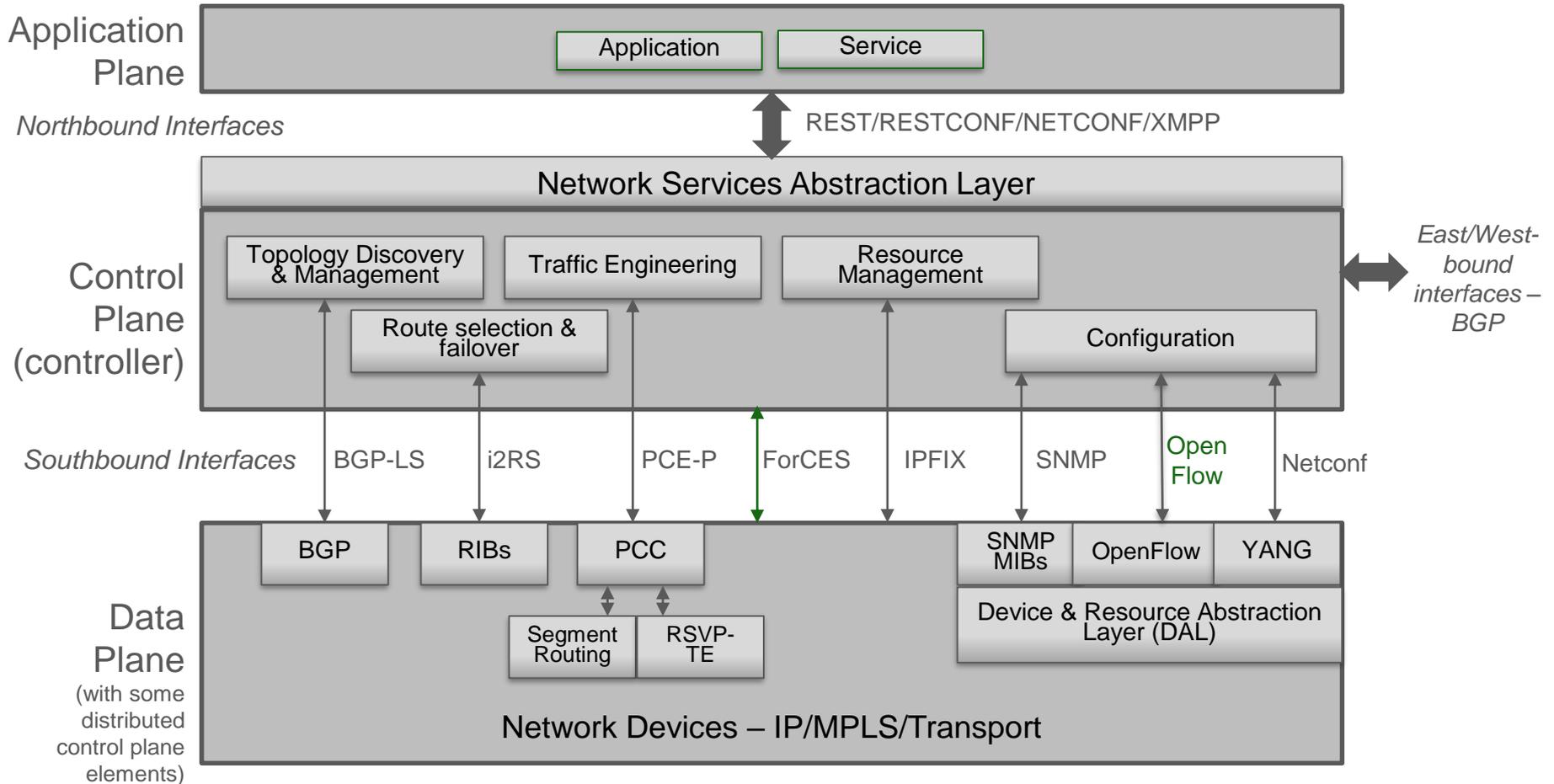


SDN architectural framework (2)

I E T F®

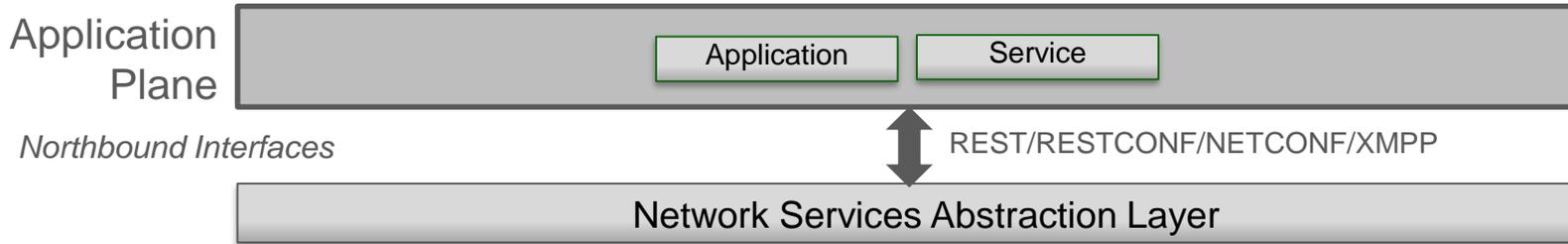


SDN architectural framework (3)



Note: designations of north-bound and south-bound are relative to the control plane (“controller”)

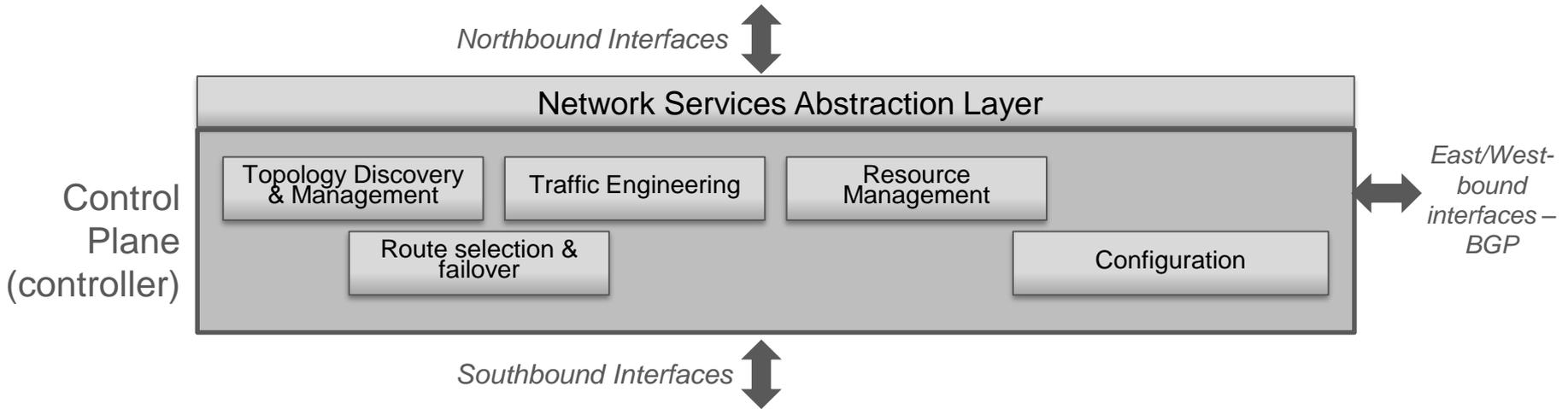
Elements of SDN architecture (1)



- Application Plane
 - “Consumers” of the network
 - Traffic optimisation applications
 - OSS systems
 - End-customer self-service portals
 - Etc.

- Northbound interfaces
 - Abstraction of network services towards applications and services
- Network Services Abstraction Layer:
 - Normalises network and service constructs via an open API or interfaces - YANG models, NETCONF, RESTCONF

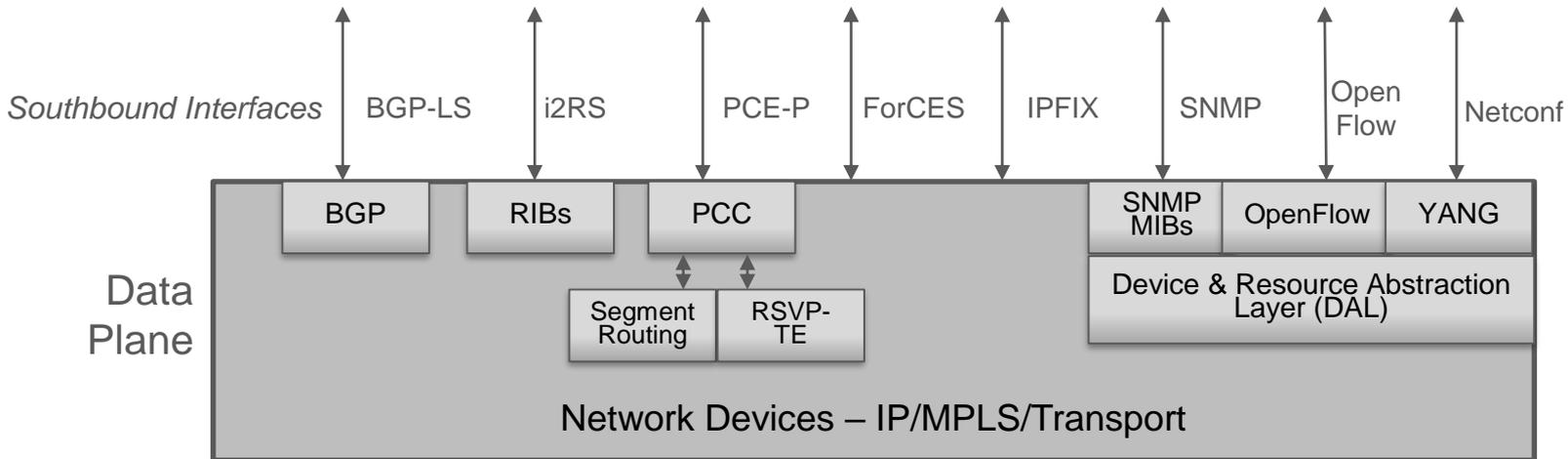
Elements of SDN architecture (2)



- Control Plane layer
 - “The Controller”; the brains of the operation
 - Translates high-level instructions from north-bound interfaces and converts them to instructions for the resource layer

- Collection of key functions:
 - Topology discovery
 - Traffic engineering
 - Resource management
 - Route selection and failover
 - Service configuration
 - Mediation
- Southbound interfaces

Elements of SDN architecture (3)



- Southbound interfaces
 - Myriad interfaces, plug-ins, and protocols, including OpenFlow
 - Device-specific details abstracted from higher layers of the controller

- Data Plane
 - Traditional and newer generation dataplanes, physical and virtual
 - Augmented by SDN-friendly protocols such as Segment Routing

Key SDN use cases

Data Centre network automation

- Most widely-deployed and mature solution
- Automation of network connectivity via overlay networks
- Multi-tenancy

SD-WAN

- Extension of DC automation concepts
- Site connectivity via overlay networking

Service Automation & provisioning

- Direct customer access via portals
- Bandwidth on demand
- Bandwidth calendaring

Network optimisation

- Link and path optimisation based on real-time network state
- Running networks "hotter"

Open source projects

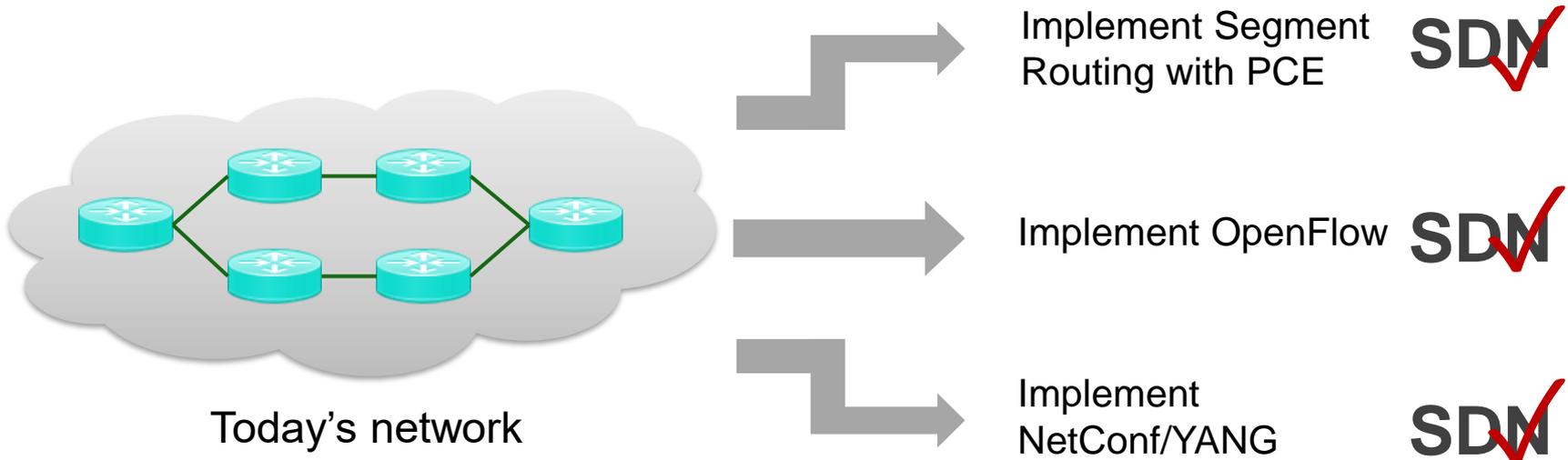


Evolution, not revolution

- Despite the hype, SDN is an evolution of current networks and technologies
- There is no one protocol that defines SDN; it is a new architectural framework for developing data networks
- Protocols and technologies that enable the principles of SDN, such as:
 - centralising control plane
 - abstracting networks and topologies
 - enhancing programmability via standard interfaces,are considered to be part of the SDN framework of technologies
- The introduction of any of these technologies can be considered to be SDN-enabling the network

Enabling SDN

There is no one protocol that defines SDN...

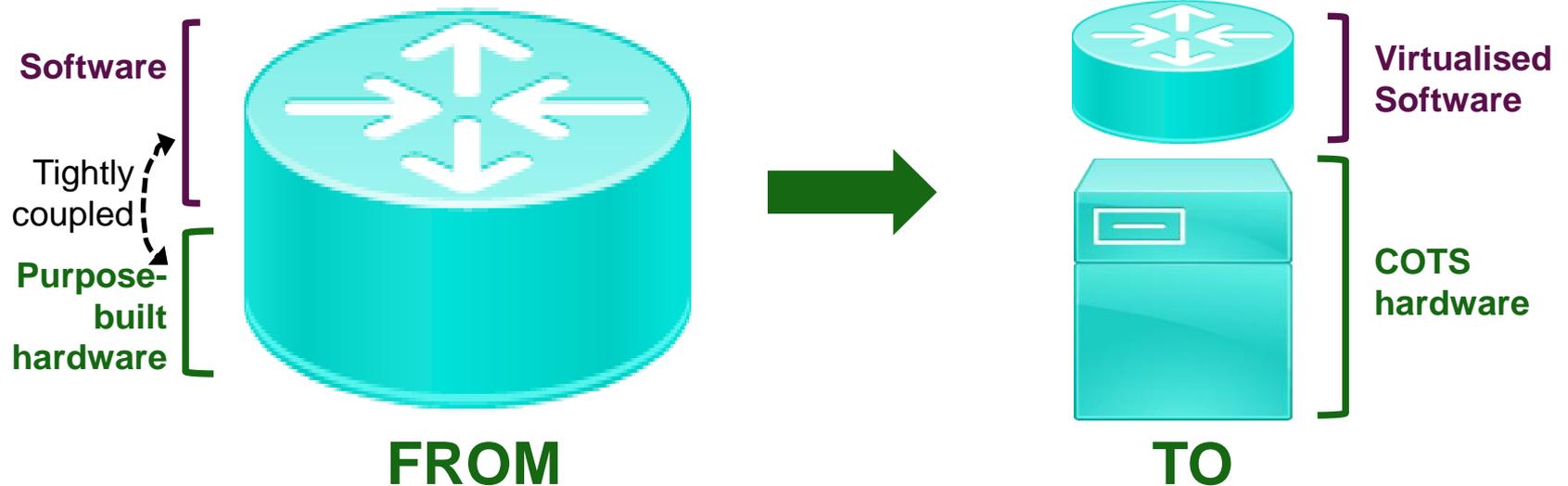


... all of these qualify as having implemented SDN in the network

Comparing and contrasting with NFV

SDN: decouples elements of the control plane from the data plane

NFV: decouples network software from closed, proprietary hardware systems



Guide to the workshop

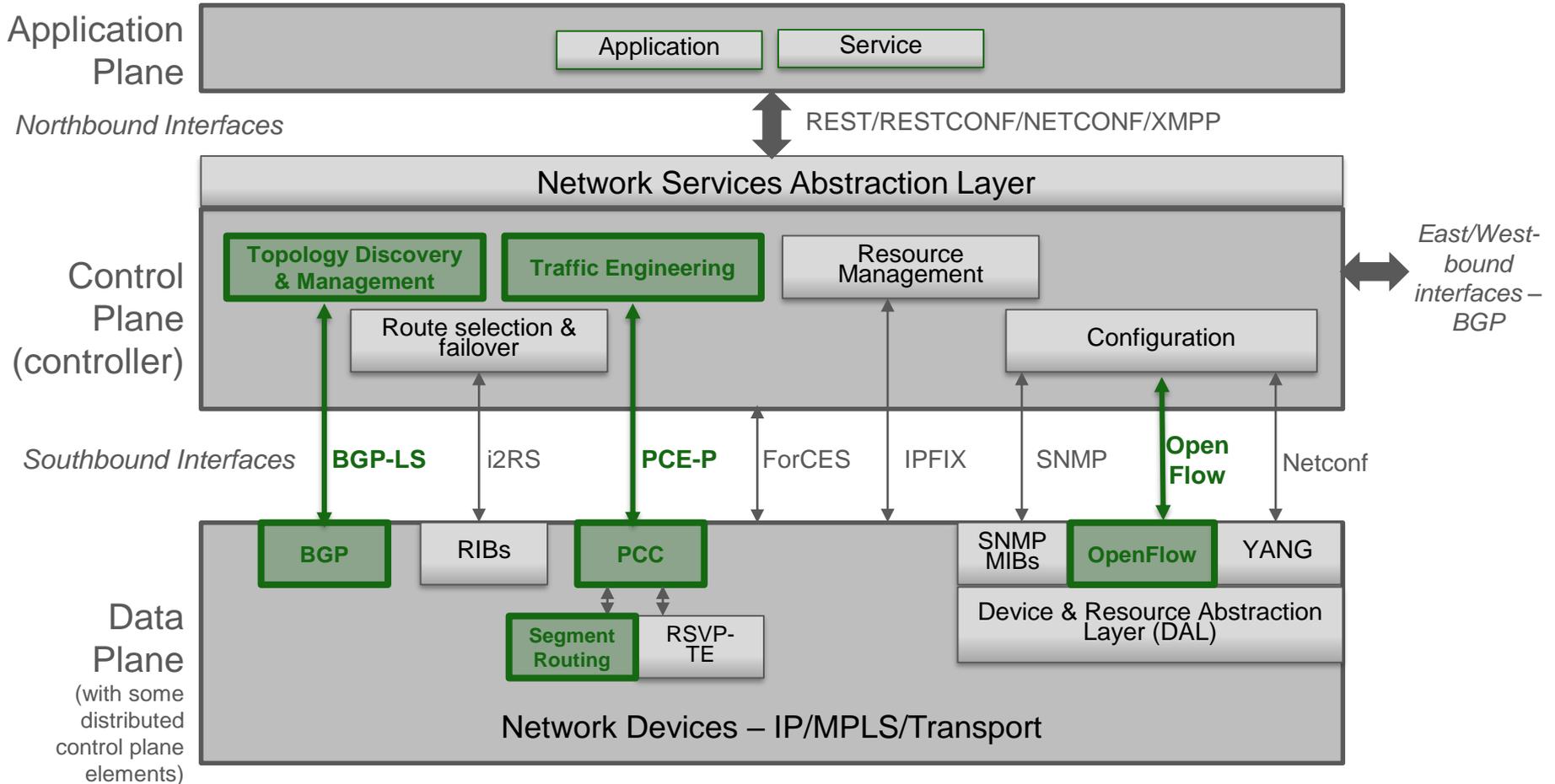
SDN Workshop



Scope

- As you have seen, SDN covers a vast range of technologies
- In this workshop, we will focus on:
 1. OpenFlow: the foundational technology for SDN, and;
 2. The move from distributed path computation to centralised path computation. Technologies of focus will include:
 - Segment routing
 - BGP-LS
 - Path Computation Element (PCE)

Areas of coverage



Note: designations of north-bound and south-bound are relative to the control plane (“controller”)

Objectives

- At the end of the workshop, you should be able to articulate:
 - the motivation behind SDN
 - the reasons for the development of OpenFlow
 - the currently accepted architectural framework for SDN
- In addition, you should have a detailed understanding of:
 - the OpenFlow protocol and the improvements introduced by various versions
 - segment routing and the problems it solves
 - acquiring network topology information using BGP-LS
 - how a path computation element (PCE) can be used to enhance path calculations

Thank You !

End of session

APNIC

Issue Date: [Date]

Revision: [xx]

