

# Advanced PCE

Adrian Farrel

Old Dog Consulting  
adrian@olddog.co.uk

Supported by the PACE Project

***[www.isocore.com/SDN-MPLS](http://www.isocore.com/SDN-MPLS)***



- Retrospective
  - Motivators
  - Architecture
  - Protocol operation
- Recent advances
  - Hierarchical PCE
  - Stateful PCE
  - Active PCE
- Forward-looking applications
  - SDN
  - Application-Based Network Operation
  - Applications to new technologies



# About the PACE Project

pace

Next Steps in **PA**th **C**omputation **E**lement Architectures  
From Software-Defined Concepts to Standards,  
Interoperability and Deployment

- PACE provides:
  - a portal to an open source PCE software and documentation repository
  - an open forum for exchange of ideas about current and future uses of PCE aimed to bring out radical new proposals and to consolidate existing research
  - an environment for innovation, research, and standardization through workshops, education, and communication.

This project has received funding from the European Union's Seventh Framework Programme (FP7) for research, technological development and demonstration under grant agreement no. 619712

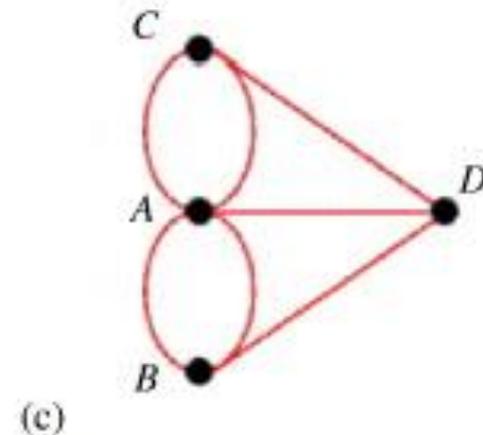
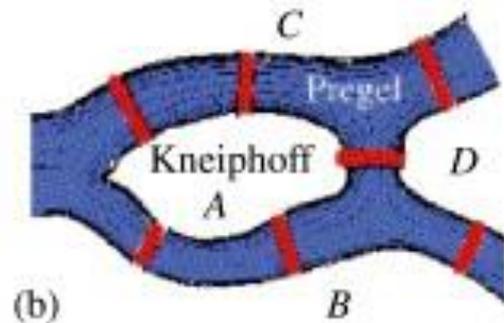
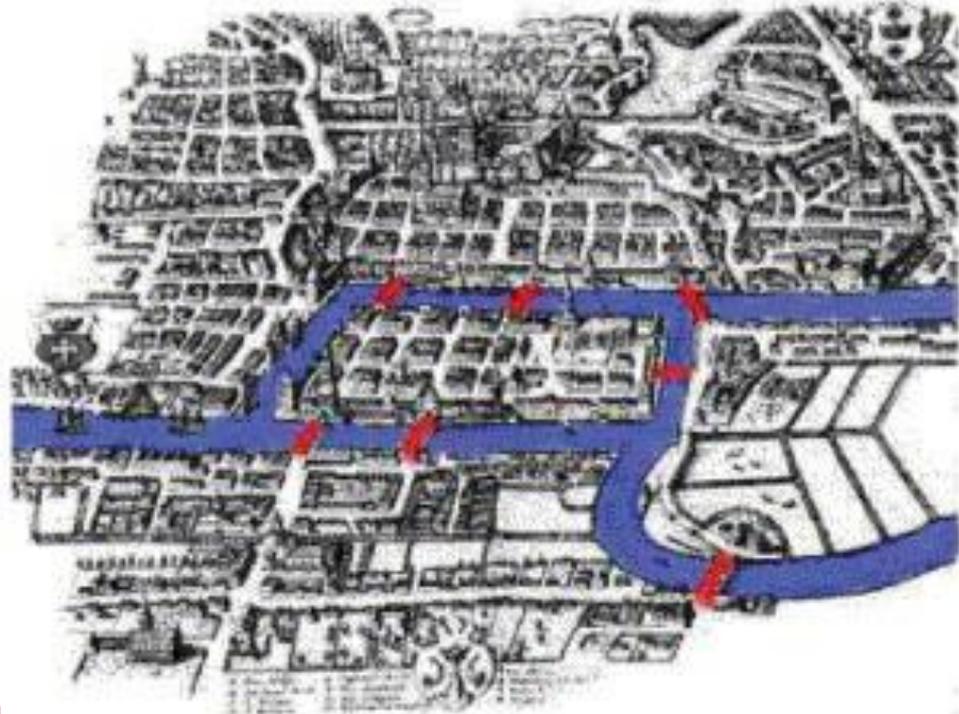
<http://www.ict-pace.net/> and @paceict on Twitter

# Retrospective



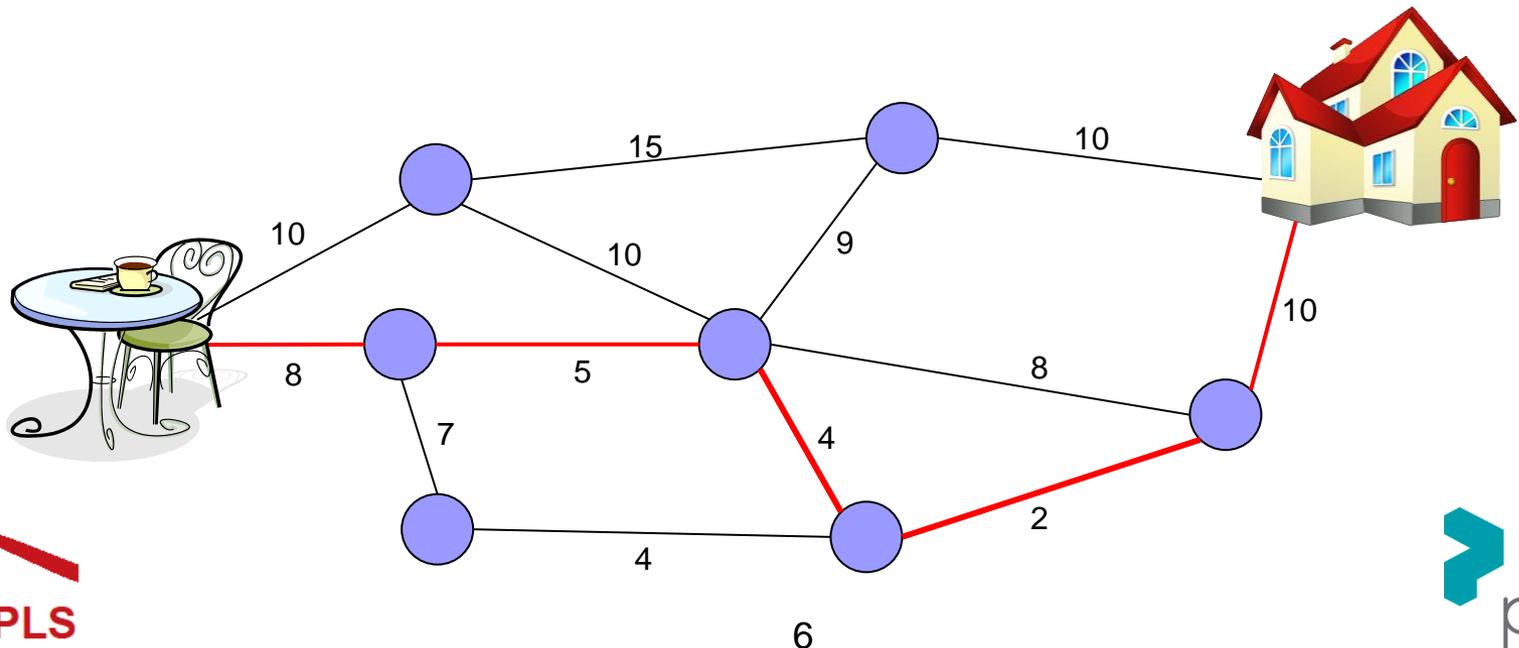
# Path Computation Starts with Graph Theory

- In 1735 Leonhard Euler wanted to send a single OAM packet to test all of the fibres in the Königsberg metro area network
  - He was able to prove it couldn't be done



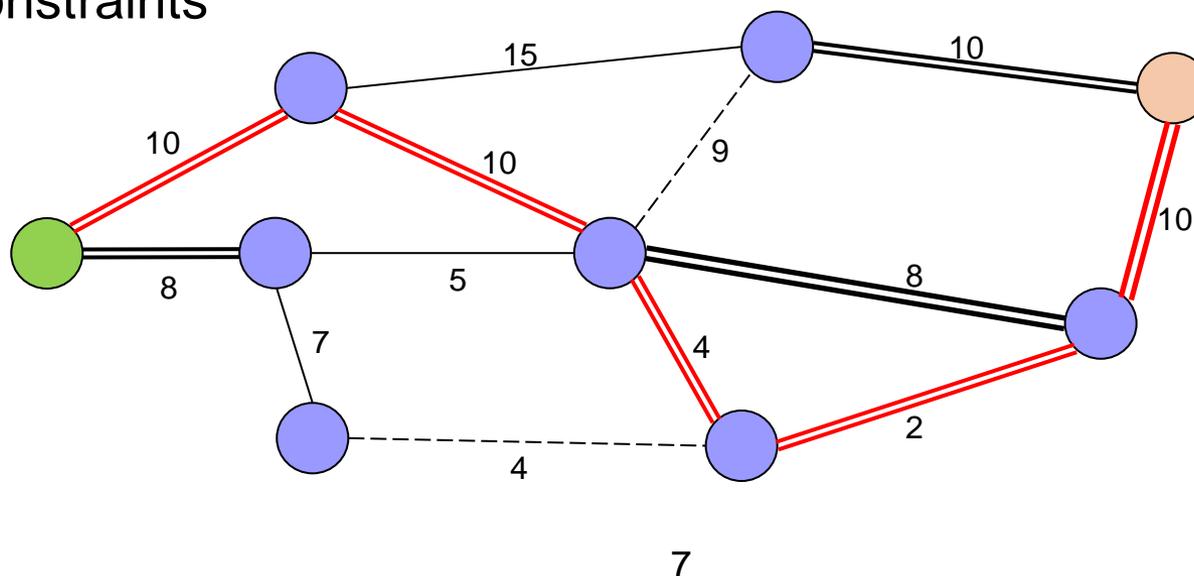
# Shortest Path First

- In 1956 Edsger Dijkstra wanted to find the shortest way home from the coffee house
  - Least hops quickly leads to per-hop metrics
  - Dijkstra's algorithm is embedded in OSPF and IS-IS so that all nodes in the network make the same forwarding assumptions
  - SPF is quick to compute even in very large networks



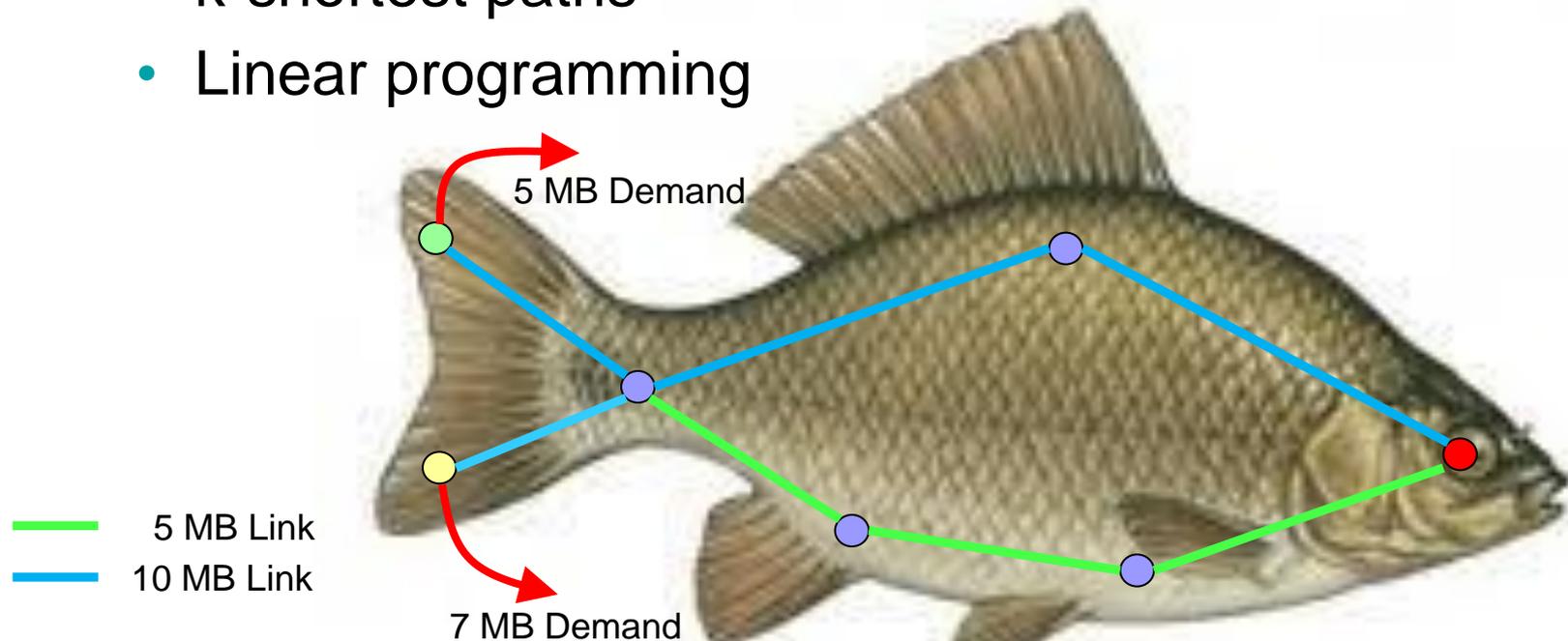
# Constraint-based Shortest Path First (CSPF)

- Traffic Engineering is fundamental to planned networking
  - Constraints may be per-hop (for example, bandwidth or lambda continuity)
    - Processing is simple pruning of the graph before or during SPF
  - Constraints may be cumulative (for example, delay)
    - Processing is just like SPF with multiple counters
  - CSPF is quick to compute even on complex networks with multiple constraints



# Multi-Path Problems

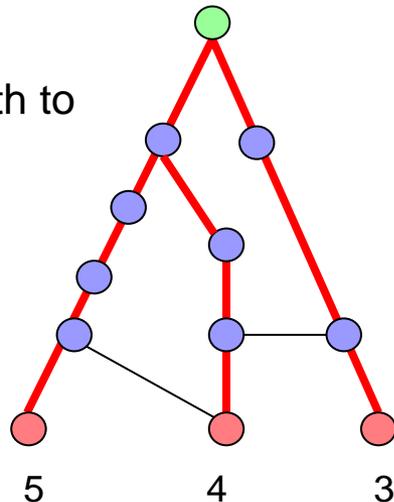
- The classic “fish” problem of ordered provisioning
- Diverse routes for protection paths
- More sophisticated algorithms
  - k-shortest paths
  - Linear programming



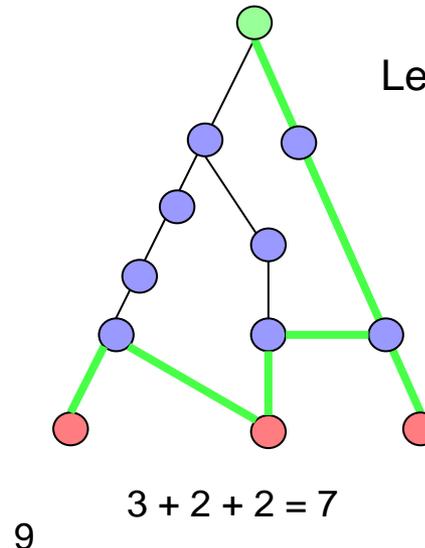
# Point-to-Multipoint

- Strikingly complex set of problems
- Different optimization criteria
  - Shortest path to each destination
  - Least cost tree
- Many sophisticated algorithms exist
- Fun to combine with multi-path problems

Shortest path to destination



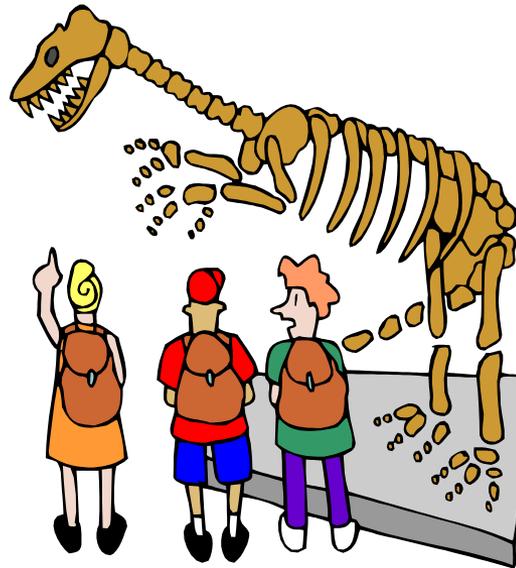
Least cost tree



# What Do We Know?

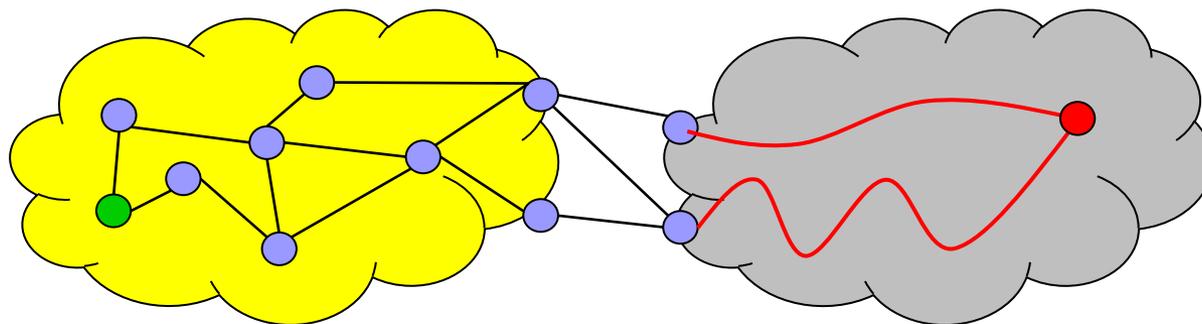
- There are lots of path computation problems in networking
- Many problems can be solved off-line
  - Service planning
  - Plenty of time
  - Even failure cases can be pre-planned
- Many problems take considerable computation resources
- Increasing demand for on-line or rapid response
  - Not all problems can be solved like this
  - Network nodes (routers and switches)
    - Do not have big CPUs
    - Do not have spare memory

# History Lesson



# So Why Was PCE Invented?

- All of these problems point to the use of dedicated path computation resources (i.e., servers)
- But PCE was invented for a completely different reason
- Aimed to solve a very specific problem
  - Find an MPLS-TE path to a virtual PoP
  - I can see in my domain, but not into my peer's
  - Which exit-point should I choose?



## Remind Me : What is PCE?

- *PCE: Path Computation Element. An entity (component, application, or network node) that is capable of computing a network path or route based on a network graph and applying computational constraints – RFC 4655*
  - This does not say it is a dedicated server
  - It can be embedded in a router
  - It can be embedded in **every** router
- For virtual PoP use case
  - PCE function in head-end LSR for local domain
  - PCE function in remote ASBR accessed through remote call

## And Remind Me : What is a Domain?

- *A domain is any collection of network elements within a common sphere of address management or path computation responsibility. Examples of domains include IGP areas, Autonomous Systems (ASes), and multiple ASes within a Service Provider network. Domains of path computation responsibility may also exist as sub-domains of areas or ASes. – RFC 4655*
- A PCE computes paths for a domain because a domain is what a PCE computes paths for
- For virtual PoP use case the domain was an AS

# Architecture

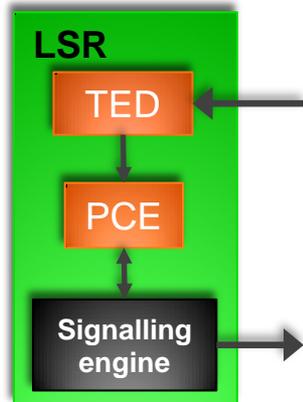


# Main Components

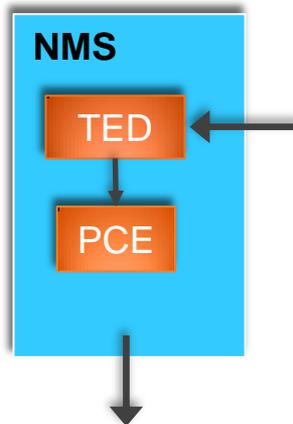
- The PCE architecture has two functional components
  - The PCE
    - The functional component that is able to perform complex path computations
  - The Path Computation Client (PCC)
    - Any client application or component requesting a path to be computed
- PCE depends on the Traffic Engineering Database (TED)
  - This is a collection of information about the nodes and links in the network

# Realisations of the Architecture

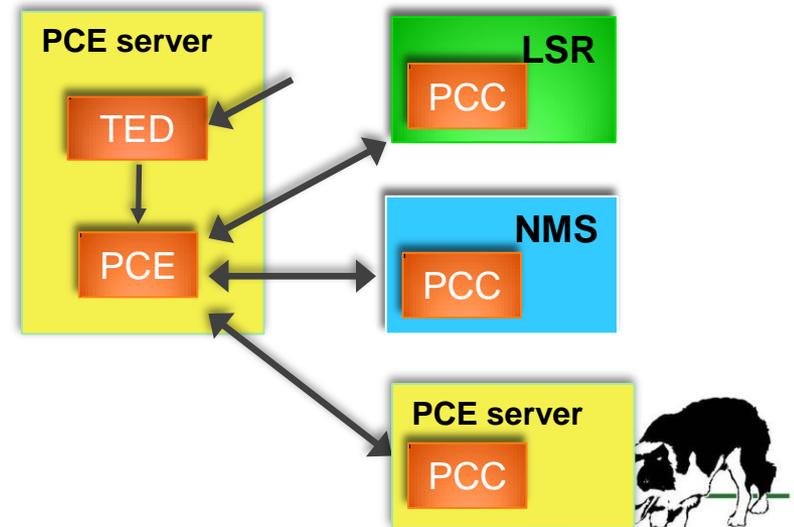
- Historically, head-end LSRs did path computation
  - They included a PCE component
- Historically, the NMS determined paths and instructed the network
  - It included a PCE component
- The PCE architecture recognises these and allows PCE to be externally visible perhaps on a dedicated server



PCE co-located in the LSR



PCE in the NMS



PCE in a dedicated server

# Path Computation Models

Two models are defined

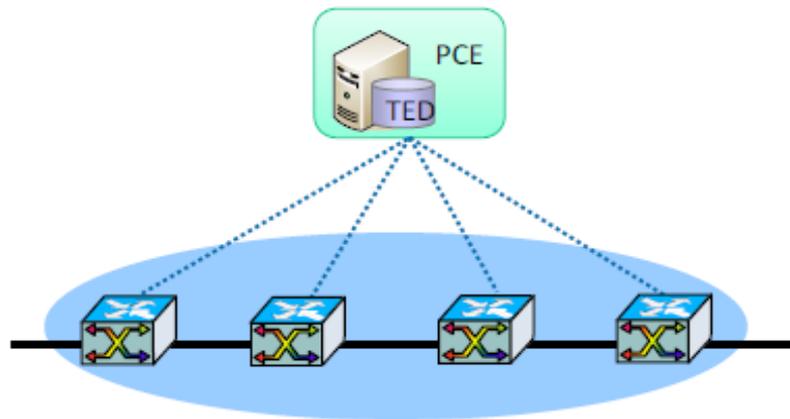
## 1 Centralized path computation

- all path computations for a given domain are performed by a single, centralized PCE
  - dedicated server (external PCE node)
  - designated router (a composite PCE node)
- all PCCs in the domain send their path computation requests to the central PCE
- to avoid single point of failure issues, backup PCE(s) may be deployed

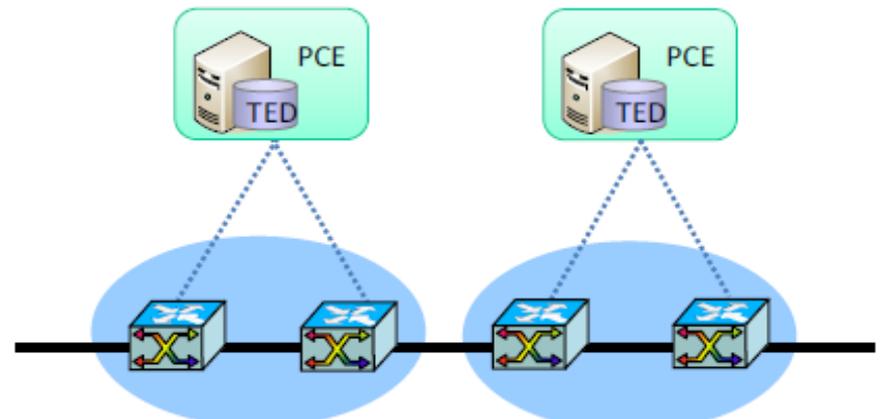
## 2 Distributed path computation

- multiple PCEs are deployed in a given domain
- computation of paths is shared among those PCEs
- single PCE path computation
  - A single PCE computes entire paths in a single IGP area without collaboration of other PCEs
  - e.g. at the ingress LSR/composite PCE node , or at an external PCE
- multiple PCE path computation
  - more than one PCE cooperate in the computation of a single path
  - E.g. loose hop expansion performed by transit LSRs/composite PCE nodes, or computations in multi-domain scenarios

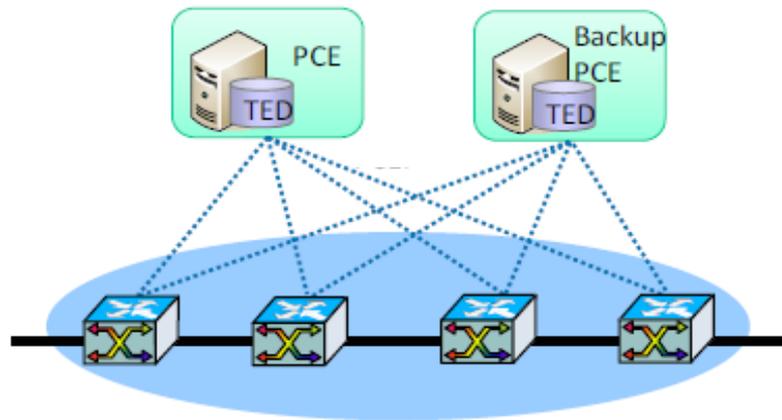
# Path Computation Models



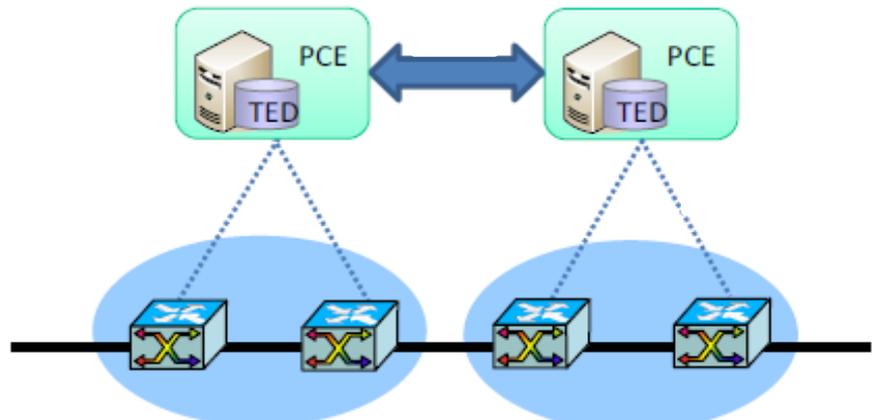
Centralized Path Computation



Distributed path computation without collaboration



Centralized Path Computation with backup



Distributed path computation with collaboration

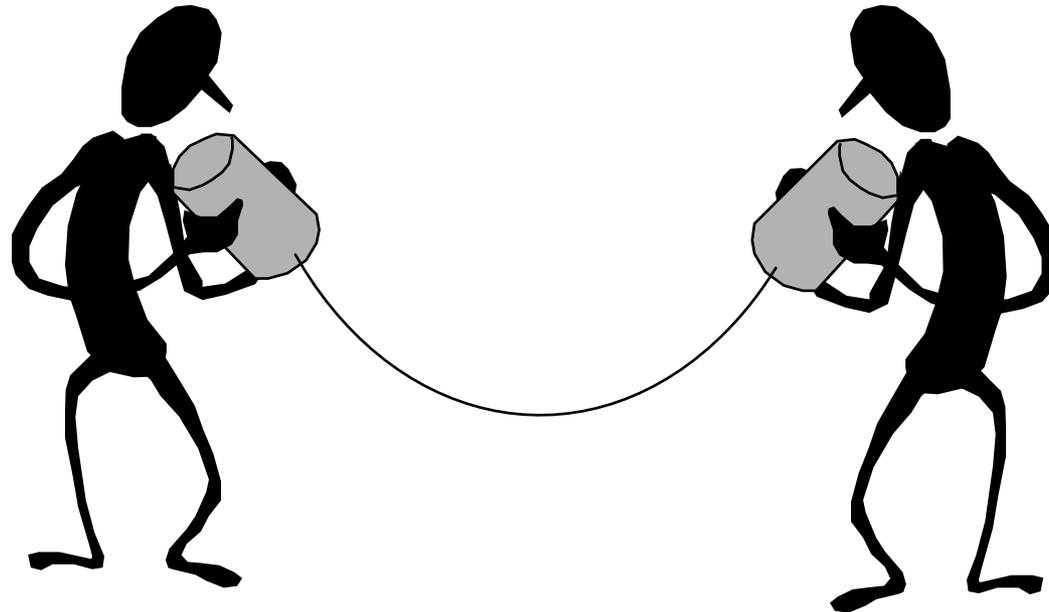
# Reminder : Improvements Brought by the PCE Architecture

- Dedicated CPU to enable more complex path computations
  - multiples constraints (e.g. optical impairments)
  - point to multi-point, over single or multiple domains
  - coordinated requests, global optimization
- Optimized tunnel placement in multi-area/AS/domain networks
  - in a scalable manner
  - while keeping domain topology confidentiality
- Optimized tunnel placement in multi-layer networks
  - e.g. MPLS over GMPLS-controlled WDM
  - topology usually advertised per layer (“overlay” model)
    - respect operational constraints
    - addresses the scalability of the “peer” model

# Reminder : New Doors Opened by the PCE Architecture

- Allows finer routing customization on a per tunnel basis
  - inclusion/exclusion of network resource
  - metric to use in the routing algorithm
  - metric bound for acceptable responses
  - objective function to specify optimization criteria
- Standardizes the protocol to access a path selection engine
  - higher influence on parameters to support
    - for vendors' equipment capable of acting as PCE
  - enables the use of standalone path computation devices
    - could be finely tuned/implemented by operators
    - a first step of MPLS/GMPLS in Software-Defined Networks

# Protocol



# PCEP – Path Computation Element (Communication) Protocol

- Why?
  - If the PCE is divorced from the PCC we need a standard protocol
- What?
  - Needs to be able to carry requests for path computation and the responses
  - Needs to offer basic reliability and security
- How?
  - Operate over TCP
  - Simple messages
  - Contents modelled on RSVP-TE

# PCEP Steps: Overall view

- Session establishment
- Path Computation Request
- PCE Responses
  - Path Computation Reply
  - Notification Message
  - Error Message
- Finishing the PCEP session
  - Session duration

# PCEP Messages

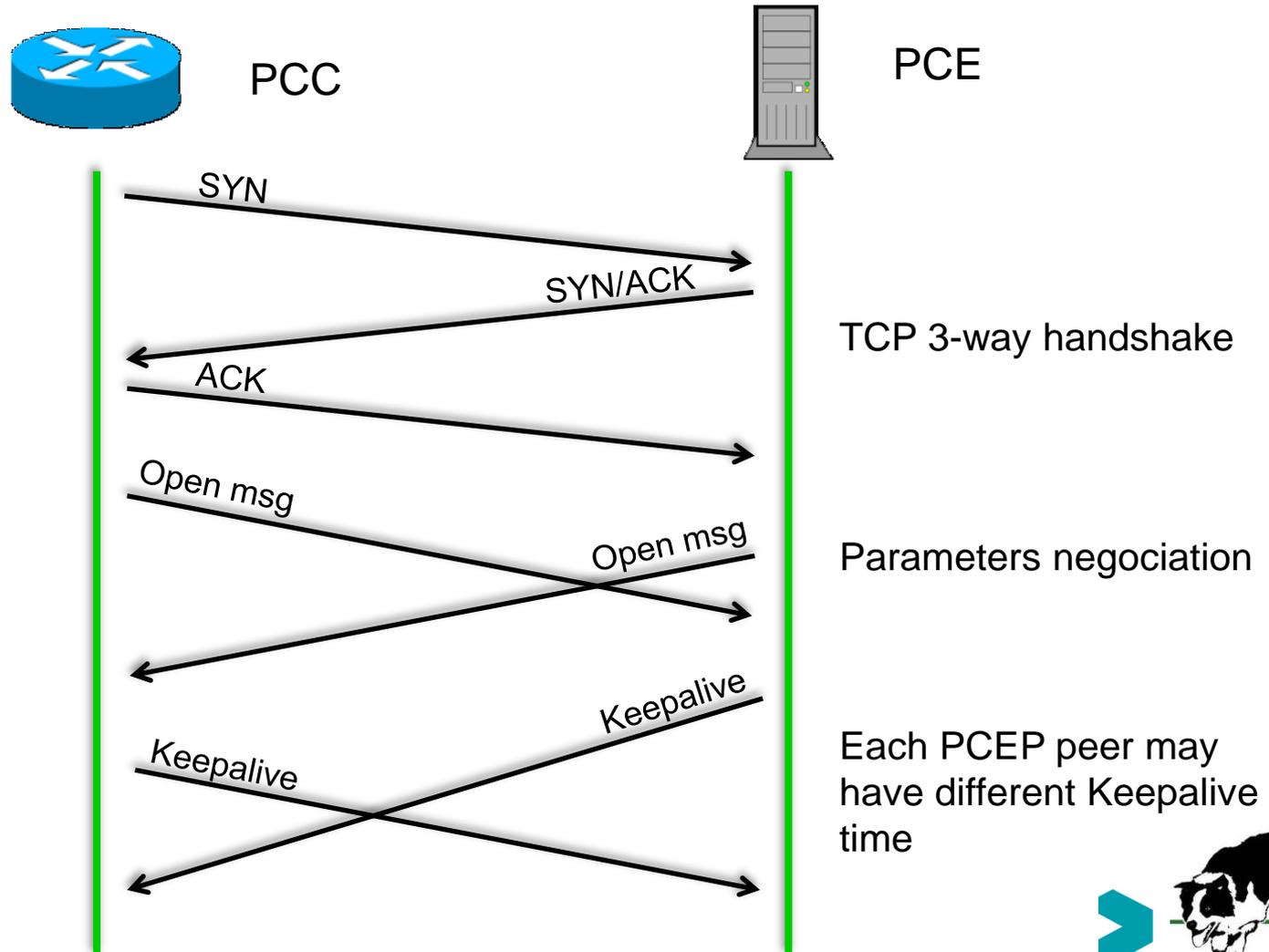
- Each PCEP message follows a common format
- All messages (excluding the Keepalive message) carry information organized as **objects**

Message	Purpose
Open	Start a PCEP session
Keepalive	Maintain a PCEP session active
PCReq	Path Computation Request sent from the PCC to the PCE
PCRep	Path Computation Reply sent from the PCE to the PCC
PCNtf	Notification message used to inform the occurrence of predetermined events
PCErr	Error message
Close	End of a PCEP session

# PCEP Steps: Session establishment

- The first step is the establishment of a TCP connection between the PCC and PCE. The initialization is triggered by the PCC which needs to reach some other network node.
- The second step is the establishment of a PCEP session over the TCP connection exchanging Open messages, which includes various session parameters like Keepalive message frequency, Deadtimer, and others.
- Each time the Keepalive timer expires, a Keepalive message is sent and the timer is restarted. Nevertheless, other messages may also restart this timer, avoiding unnecessary traffic of Keepalive messages. The Deadtimer is also restarted when a message is received.

# PCEP Steps: Session establishment



# PCEP Steps: Path Computation Request

- A PCC uses a PCReq message to request a path computation to the PCE. It must include at least the source and destination.
- It may include path constraints like requested bandwidth, points of failure that should be excluded from the path being calculated, etc.
- PCC may also assign the request priority and if load-balancing is allowed, including maximum number of paths and minimum path bandwidth in a path group.

# PCEP Steps: PCE reaction

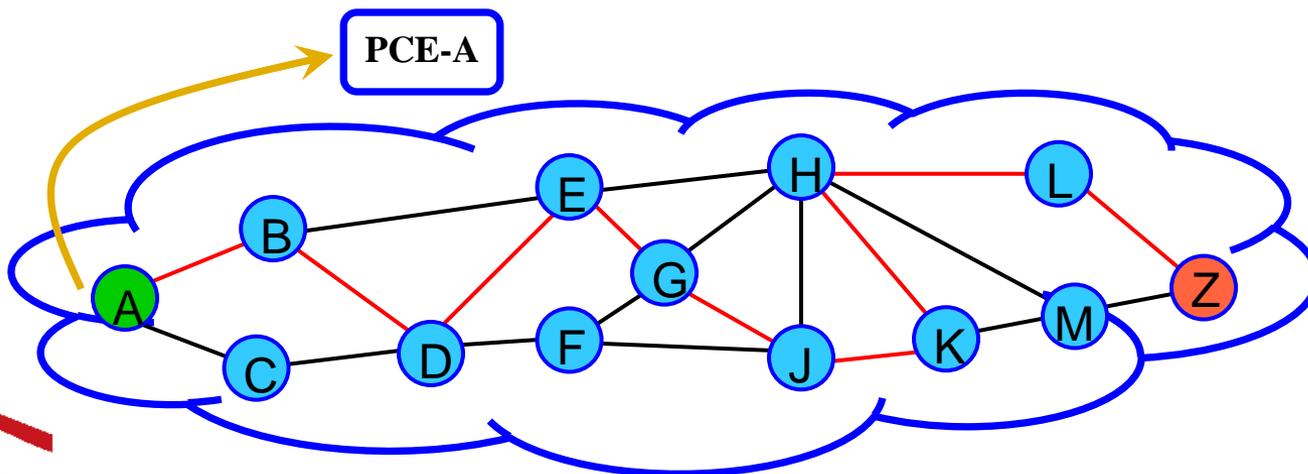
- Once PCE receives a PCEReq from the PCC it can issue any of the following messages:
  - PCE Reply
  - PCE Notification Message
  - PCE Error Message
- PCE can also do nothing
  - In that case the PCC may issue a Notification Message to the PCE cancelling the previous request

## PCEP Steps: Path Computation Reply

- A PCE uses a PCRep message to send the path computation result to the PCC. It may carry one or more computed paths if the path computation succeeded, or a negative reply if a path could not be determined.
- When sending a negative reply, the PCE may specify the reasons why the path could not be determined and include advice about which constraints could be relaxed to be more likely to succeed in a future request.

# A Simple Example

- Lots of potential paths from A to Z
- Find me the least hop path (ABEHLZ)
- Find me a path using only red links (ABDEGJKHLZ)
- Find me two link disjoint paths, one using red links
  - (ABDEGJKHLZ) (ACDFGHMZ)



# PCEP Steps:

## Notification message and Error message

- The PCNtf message may be sent by the PCE or by the PCC:
  - PCE may notify the PCC that it is overloaded, that some requests will not be satisfied, etc. The PCC may decide to redirect requests to other PCE.
  - PCC may notify some particular events to the PCE, like cancellation of pending requests.
- PCErr message must be sent when a protocol error condition is met, like:
  - when a message without a mandatory object is received.
  - unexpected message
  - policy violation

# PCEP Steps: Finishing a PCEP session

- Any PCEP peer may decide to finish the session, send a Close message and close the TCP connection.
- When receiving a Close message:
  - PCC clears all pending requests sent to PCE
  - PCE clears all pending requests received from PCC
- A PCEP session is immediately finished if a TCP connection failure occurs.

# PCEP Objects

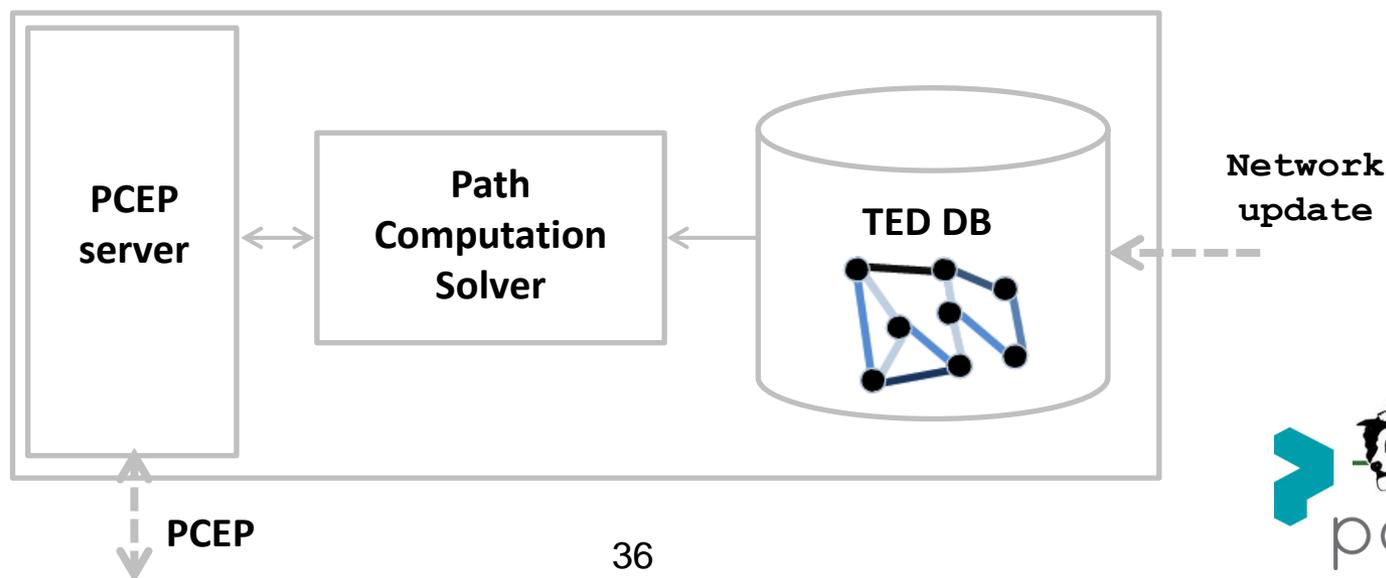
Object	Purpose	Message
Open	Session capabilities negotiation	Open, PCErr
RP	Request Parameters for path computation	PCReq, PCRep, PCNtf, PCErr
No-Path	Indication of failed path computation	PCRep
End-Points	Source and destination of a path	PCReq
Bandwidth	Bandwidth for a path	PCReq, PCRep
Metric	Desired or resultant metric of a path	PCReq, PCRep
ERO	The computed path	PCRep
RRO	An existing path for re-optimisation	PCReq
LSPA	LSP Attributes for a computed path	PCReq, PCRep
IRO	Path to include in the computation	PCReq, PCRep
SVEC	Synchronisation Vector to coordinate paths	PCReq
Notification	Information about request and PCE state	PCNtf
PCEP-Error	Error conditions	PCErr
Load-Balancing	Request set of paths summed to bandwidth	PCReq
Close	Reason for closing the session	Close

# The Traffic Engineering Database



# The Traffic Engineering Database (TED)

- Traffic Engineering Database (TED) is an essential internal component of a PCE
  - Provides snapshot of the controlled network and its resources
  - PCE algorithms resort to TED as primary information source input
- The TED contains
  - Topology of the controlled network (Nodes, Links, relationships)
- The available resources and attributes
  - Available link bandwidth and metrics (e.g., costs)



# A Little More About the TED

- Where does the TED come from?
- PCE can listen to the IGP
  - Passive listener
- TE Export from the network
  - BGP-LS
    - Can apply filters before exporting data
  - YANG
    - Simpler to represent topology
    - Still needs work for standardisation
- May be enhanced by additional information
  - From management systems according to policy
  - From inventory systems

# Metrics

- Metrics apply to links
  - Just as normal for IGP-TE
  - An intrinsic part of the TED
- The base specification defines three metrics
  - IGP metric
  - TE metric
  - Hop count
- Other metrics defined for other networks
  - Delay, optical, ...
- Used in PCRep to indicate quality of computed path
- Used in PCReq in conjunction with Objective Function

# Objective Functions

- Path computation is subject to a set of one or more specific optimization criteria
  - Called objective functions (OFs)
- Applied to single paths
  - Minimum Cost Path (MCP)
  - Minimum Load Path (MLP)
  - Maximum residual Bandwidth Path (MBP)
- Applied to sets of paths
  - Minimize aggregate Bandwidth Consumption (MBC)
  - Minimize the Load of the most loaded Link (MLL)
  - Minimize the Cumulative Cost of a set of paths (MCC)
- New OFs can be defined to facilitate new computations

# Protocol Bits'n'Pieces : Explicit Route Exclusions

- What does it mean a route exclusion in path computation?
  - Sometimes, a Path Computation Client (PCC) needs to specify abstract nodes, resources, and Shared Risk Link Groups (SRLGs) that are to be explicitly excluded from the path computation.
- When do we need to specify a route exclusion?
  - In inter-domain Label Switched Paths (LSPs), disjoint paths may be computed by cooperation between PCEs (computing separated segments)
  - When a network operator wants a path to avoid specified nodes for administrative reasons.
- Protocol extensions on a PCReq
  - Include Route Object (IRO)
    - Exclude Route Subobject
  - Exclude Route Object (XRO)

# Point-to-Multipoint (P2MP)

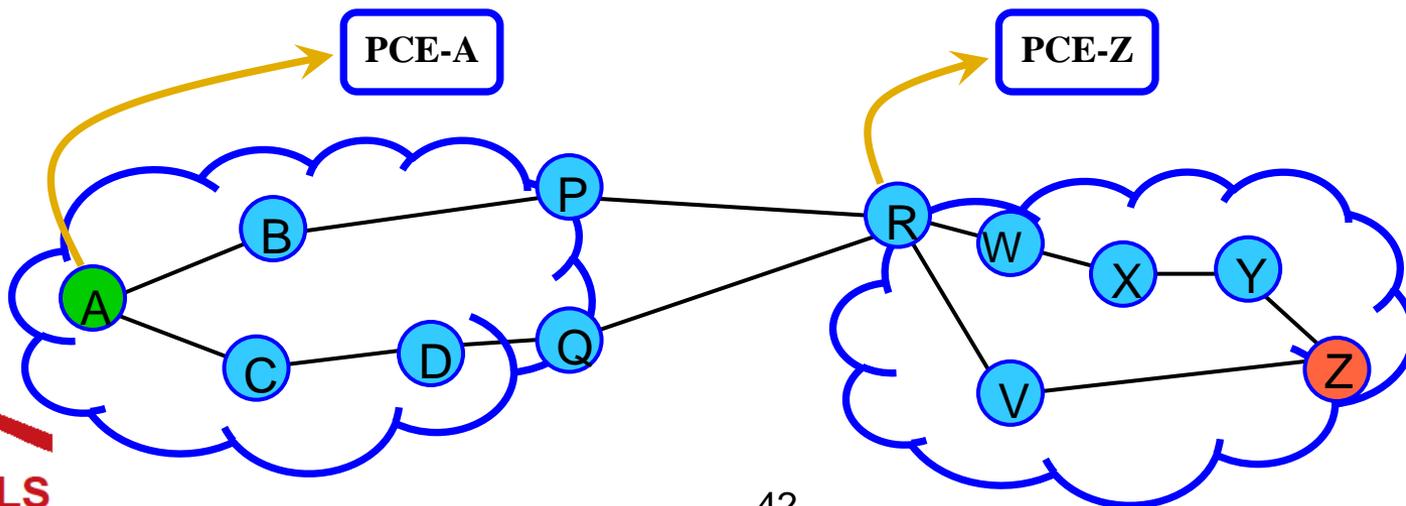
- Not supported in the original protocol spec
  - “Walk before you can run”
- P2MP is one of the more challenging computations
  - Simple solutions are an overlay of multiple P2P LSP
    - This is not optimal
  - P2MP is computationally expensive (hard problems)
    - Prime candidate to offload to a dedicated PCE
- Protocol extensions
  - Specify the set of destinations
  - Specify the branch points
  - Specify the objective functions
    - Least cost to the destinations
    - Least cost tree

Diverse paths

Distinct source nodes

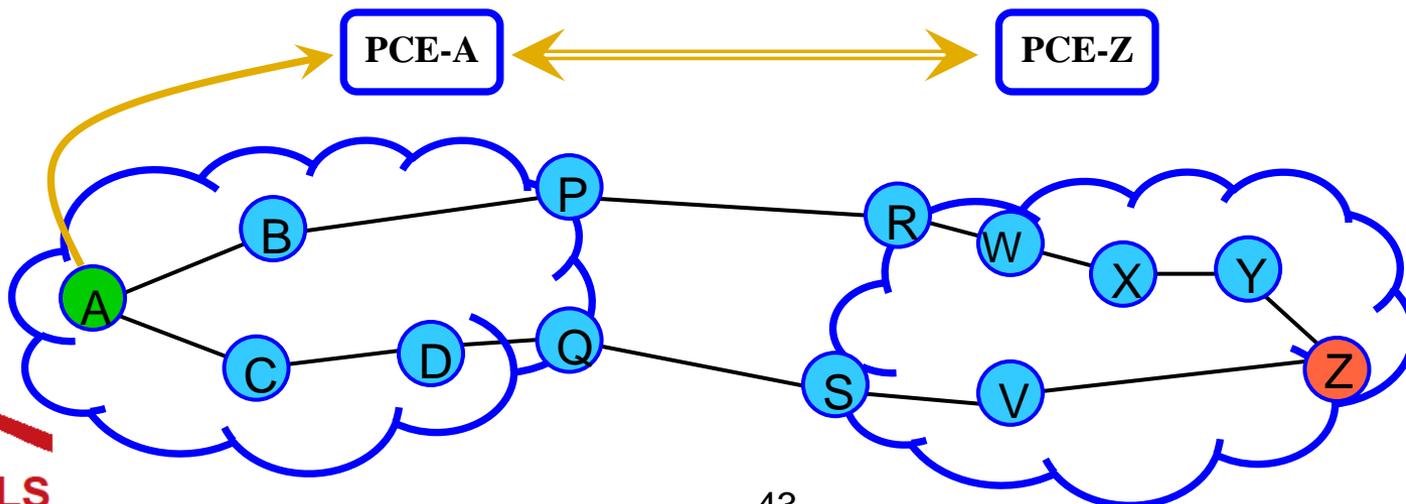
# Per-Domain Path Computation

- Ingress asks PCE-A for a path towards Z
  - PCE-A returns ABP and signalling starts
- Signalling reaches R
- Border node R asks PCE-Z for a path towards Z
  - PCE-Z returns RVZ and signalling continues
- This is what we're used to with PCE embedded in ASBR



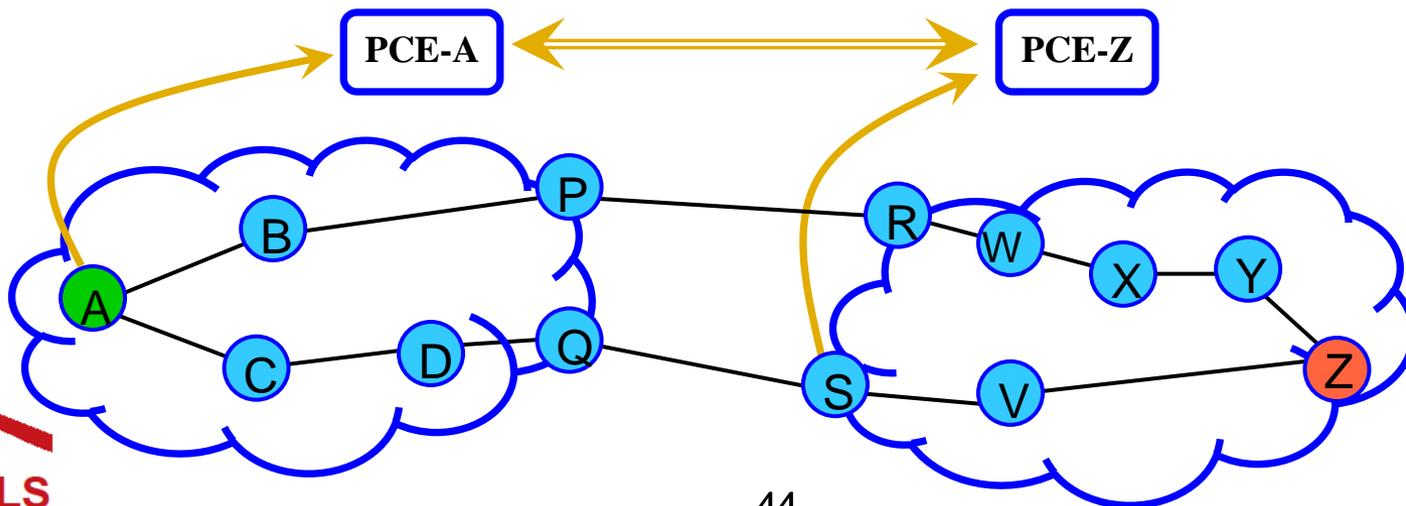
# Backward Recursive Path Computation (BRPC)

- Using normal per-domain mechanisms
  - PCE-A selects ABP (cost 2 in domain A)
  - Resulting path is ABPRWXYZ (cost = 7)
- In BRPC PCE-A consults PCE-Z
  - PCE-A acts as a PCC to PCE-Z
  - Two paths are returned {RWXYZ, 4} and {SVZ, 2}
  - PCE-A can now make a better choice
    - ACDQSVZ (cost = 6)



# Path Keys and Confidentiality

- Domain Z might not want to expose the details of its internal topology
- BRPC is enhanced with Path Keys
- PCE-Z returns {R, key1, 4} and {S, key2, 2}
- PCE-A selects ACDQSkey1 (cost = 6)
- Signalling reaches S
  - S consults PCE-Z for the path SVZ



# Support for GMPLS

- GMPLS is just Traffic Engineered LSPs
  - Initial effort focused on basic path computation
- Application to transport networks requires
  - Diverse protection paths
  - Special metrics and objective functions
    - Wavelength continuity
    - Path length and regeneration
- Technology-specific constraints and objective functions
  - Timeslots
  - Optical constraints
    - Even simple constraints are complex
    - Vendor-specific constraints
- Multi-domain is very applicable to transport networks
- Client-server relationships between networks

# Optical Constraints

- These are really non-trivial!
- They are the province of individual vendors
  - But they are still open for use in a PCE

## Link OSNR

$$\text{Amplifier\_span\_OSNR}_i = p_i^{\text{out}} / NF \cdot h \nu_s \cdot G_i \cdot B_0 [9]$$

$$\text{Link\_OSNR}_k = 1 / \left( \sum_{i=1..k} 1 / \text{amplifier\_span\_OSNR}_i \right)$$

## Link PMD

$$\text{Link\_PMD}_k = \sqrt{\left( \sum_{i=1..k} (\text{amplifier\_PMD}_i)^2 + \sum_{i=1..k} (\text{fiber\_PMD}_i \cdot \sqrt{\text{span\_length}_i})^2 \right)}$$

## Link Nonlinearity

$$\text{Link\_NLPS}_k = \sum_{i=1..k} \frac{2\pi}{\lambda} \left( \int \left( \frac{n_2^{\text{line}} \cdot P(z)}{A_{\text{eff}}^{\text{line}}} \right) dz + \int \left( \frac{n_2^{\text{DCF}} \cdot P(z)}{A_{\text{eff}}^{\text{DCF}}} \right) dz \right)_{\text{link}-i}$$

# Global Concurrent Optimisation (GCO)

- Sometimes paths have to be computed together
  - Remember the fish!
- Even more important during re-optimisation
  - Moving one LSP onto a new path might block moving others
- Need to batch multiple computation requests
  - May be more than fits in one message
- Protocol extensions in RFC 5557
  - New network-wide metrics
    - Maximum link utilisation
    - Overbooking factor
  - New objective functions
    - Minimize aggregate Bandwidth Consumption (MBC)
    - Minimize the load of the Most Loaded Link (MLL)
    - Minimize Cumulative Cost of a set of paths (MCC)

# Moving Forward



# Inter-Layer PCE and Virtual Network Topology Manager (VNTM)

- PCE is a “logical component”
  - PCE limited to specific function
    - Path computation only
    - PCE cannot make provisioning requests
- VNTM is a “logical component”
  - Responsible for:
    - Applying policy
    - Preparing lower layer trails for use by higher layer
    - Using PCE to compute lower layer paths
    - Invoking provisioning in lower layer
    - Causing lower layer LSPs to be advertised as TE links in higher layer
    - Interacting with lower layer management and policy
    - Interacting with higher layer PCE, management, and policy.

# VNT Manager Interactions with PCE

- VNT Manager acts on triggers
- Uses PCE to determine paths in lower layer
- Uses management systems to provision LSPs
- Causes new TE links to be advertised

2. I can't find a path

1. Compute a path

3. I failed to compute a path

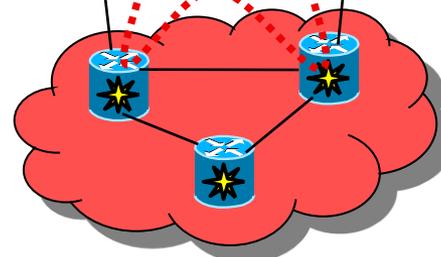
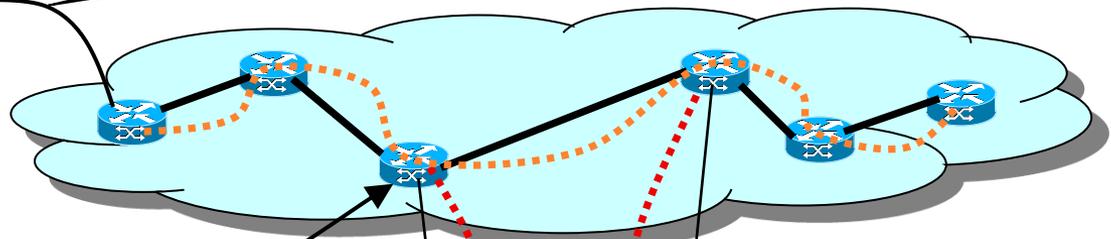
4. Compute a path

5. Provision an LSP and make a TE link

PCE

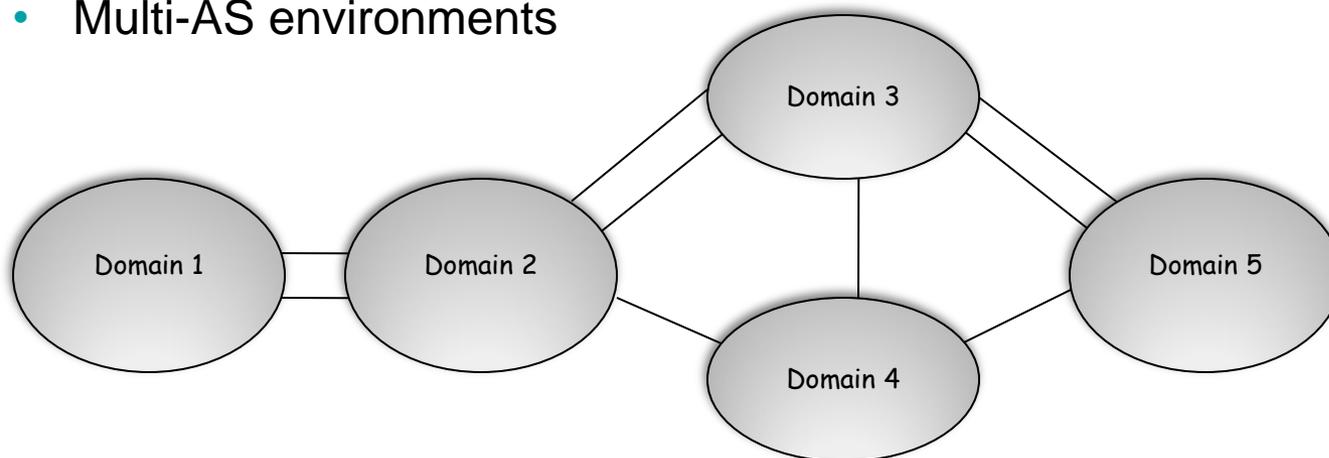
VNTM

PCE



# Domain Meshes

- A mesh of domains
  - Optical networks are constructed from multiple vendor-specific sub-domains
  - Multi-AS environments



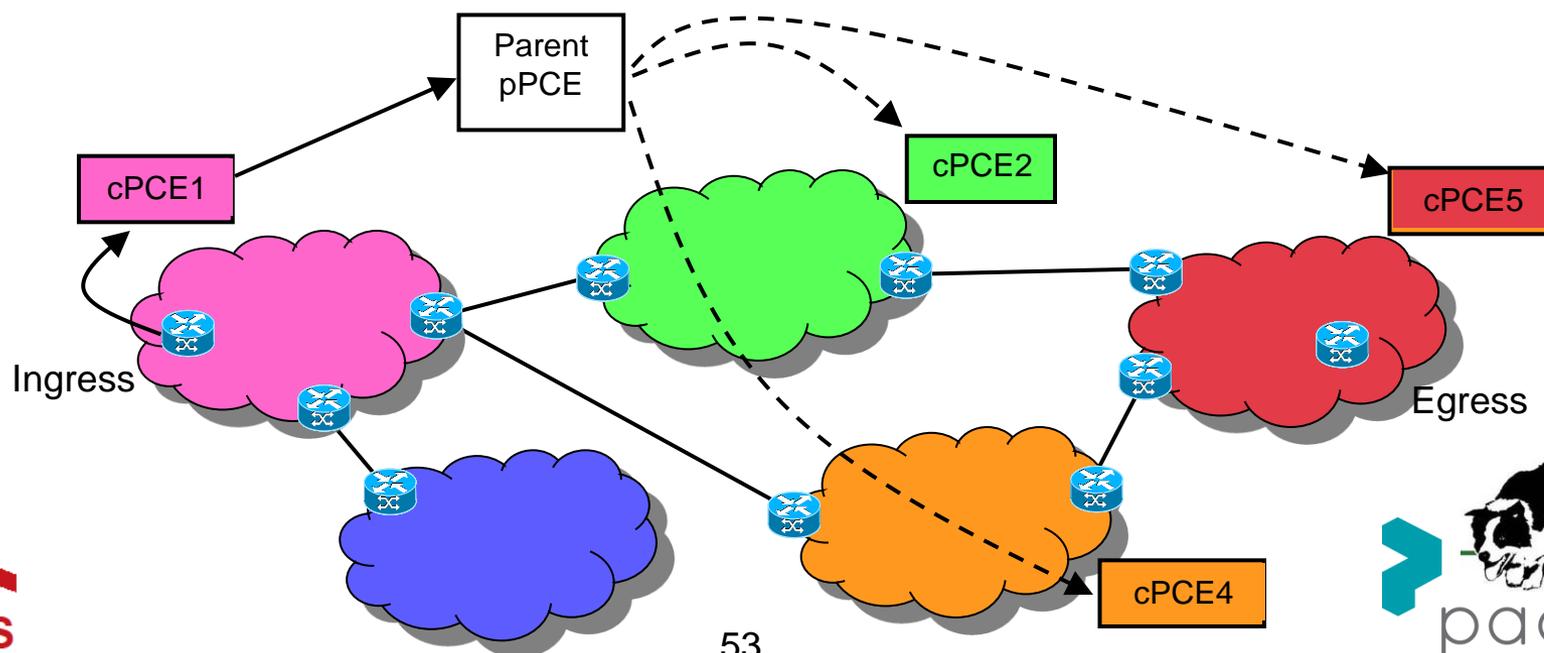
- The internal TE topology of a domain can't be seen from outside
  - Aggregation / abstraction hides information and leads to failed path setup
  - Flooding TE information breaks confidentiality and does not scale
- Means end-to-end paths are hard to compute
  - Which domains to use?
  - Which interconnection points to use?

# Existing Multi-Domain PCE Techniques

- Per domain path computation
  - With per domain the sequence of domains is known
  - Domain border nodes are also usually known
  - Computation technique builds path segments across individual domains
  - Domain choice is only possible with crankback
  - The mechanism does not guarantee an optimal path
- BRPC
  - Current definition assumes domain sequence is already known
  - BRPC is good for selecting domain border nodes
  - Computation technique derives optimal end-to-end path
  - BRPC could be applied to domain selection
    - Functions correctly (optimal solution)
    - Significant scaling issues

# Hierarchical PCE

- How do I select a path across multiple domains?
- Parent PCE (pPCE) has
  - An overview topology showing connectivity between domains
  - Communications with each Child PCE (cPCE)
- Parent can selectively and simultaneously invoke children to assemble an end-to-end path

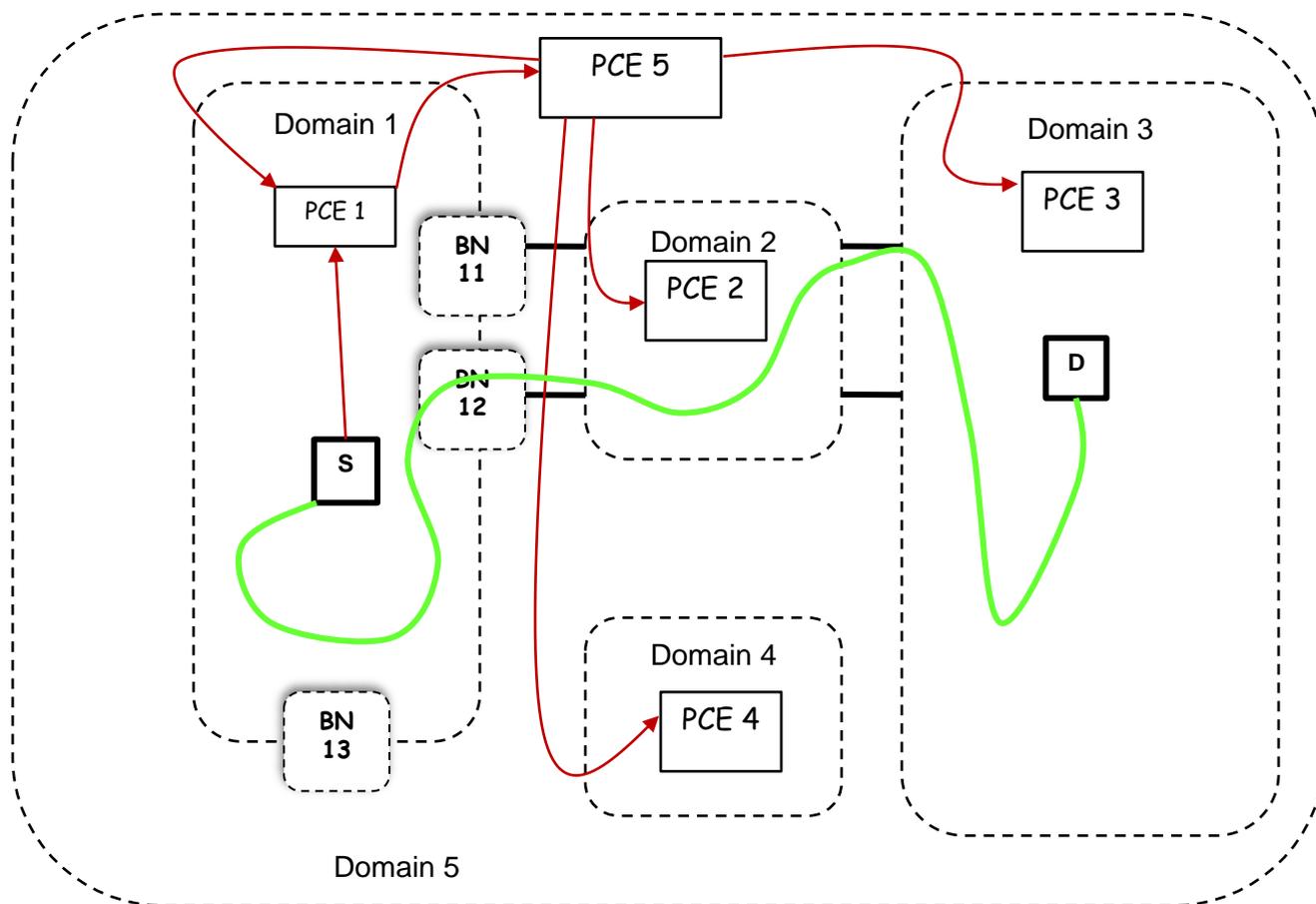


# Hierarchical PCE

- Each Child PCE is configured with the address of its parent PCE
- The Parent PCE maintains a topology map
  - The nodes are the Child domains
  - The map contains the inter-domain links
  - The TE capabilities of the links are also known
- Domain interconnection discovery or configuration
  - The Child PCE can report the following information to the Parent PCE
    - The identity of its neighbor domains
    - Details of the inter-domain TE links (as already advertised in the IGP)

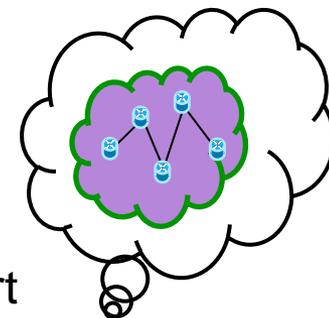
# Hierarchical PCE Procedures

1. Ingress LSR (S) sends a request to PCE1 for a path to egress (D)
2. PCE 1 determines egress is not in domain 1
3. PCE 1 sends computation request to parent PCE (PCE 5)
4. Parent PCE determines likely domain paths
5. Parent PCE sends edge-to-edge computation requests to PCE 2 responsible for domain 2, and to PCE 4 responsible for domain 4
6. Parent PCE sends source to edge request to PCE 1
7. Parent PCE sends edge to egress request to PCE3
8. Parent PCE correlates responses and applies policy requirements
9. Parent PCE supplies ERO to PCE 1



# The Stateful PCE

- The “classic” PCE uses network state stored in the TED
  - This information may be gathered from the network
  - Or from inventory, management systems, configuration export
- There is also transitory per-computation state in the PCE
  - This allows bulk computation
  - “Please compute a path considering this other LSP”
- A Stateful PCE is aware of LSPs in the network (LSP-DB)
  - Could retain knowledge of paths it previously computed
  - Or gather information about LSPs from the network
    - Extensions to BGP-LS
    - Possible extensions to PCEP
      - “Yes, I used that path you gave me”
      - “Here are some other LSPs I know about”
- A Stateful PCE is able to do more intelligent path computation
  - Especially re-optimisation and interactions between LSPs



PCE



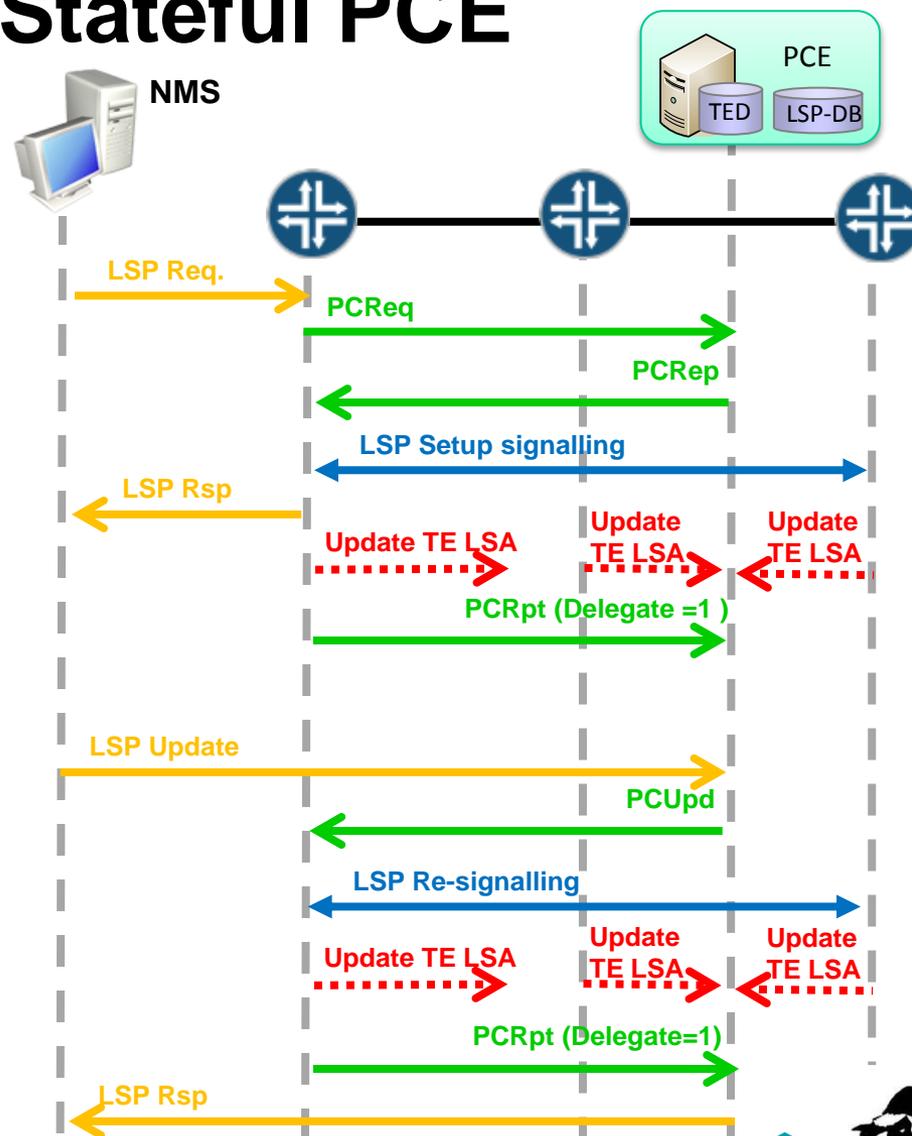
# The Active PCE

- An Active PCE is able to advise the network
  - About more optimal paths
  - When congestion is a problem
- As far as the protocol is concerned, it is only a small step
  - The PCC can say “Please worry about these LSPs for me.”
    - Delegation of LSPs from the PCC to the PCE
  - The PCE can say “Here is a path you didn’t ask for.”
    - For delegated LSPs or for new LSPs
- This enriches PCEP
  - From a request/response protocol
  - To become almost a configuration / provisioning protocol
- Architecturally it is “interesting”
  - PCEP used to be the language spoken by the computation engine (PCE)
  - Now it is the language spoken by the network management system (NMS) that has a computation component
  - That doesn’t make it wrong. It does make it different
- It also opens up PCEP as an SDN protocol as we will see later



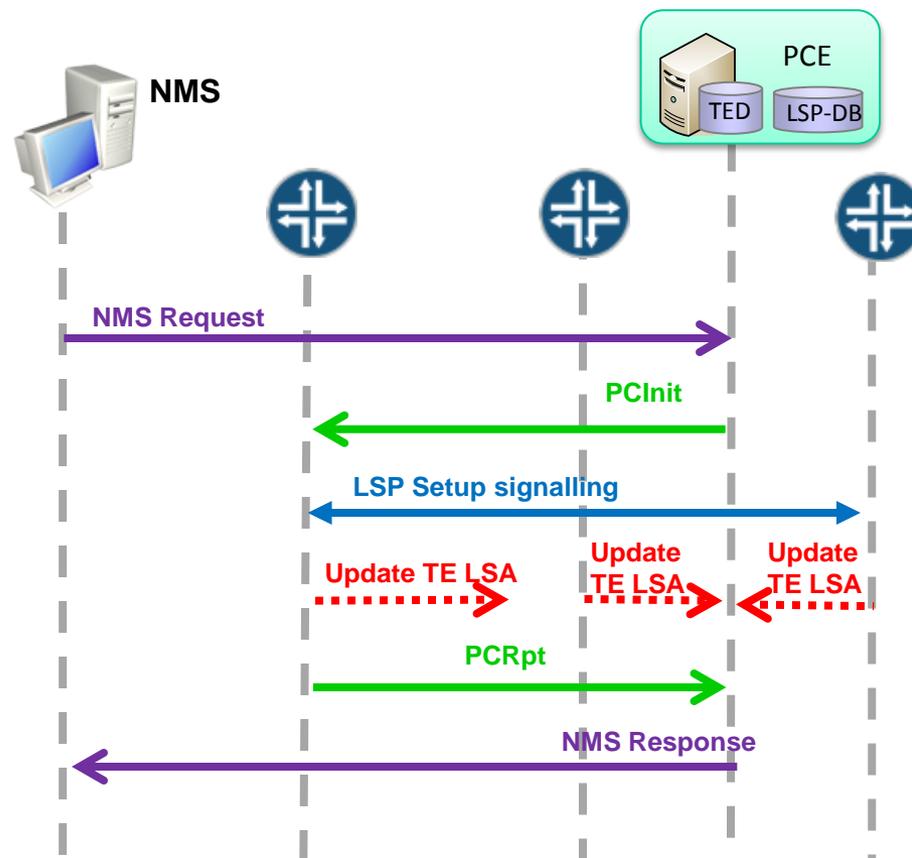
# Active Stateful PCE

- NMS requests an LSP
- PCE is used as normal
- LSP is set up
- TED is updated via IGP
- New PCRpt
  - Supply data to LSP-DB
  - Delegate control
- NMS requests update to LSP
- PCE computes updated path
  - New PCUpd to tell ingress
- LSP is ressignalled
- TED is updated via IGP
- PCRpt
  - Supply data to LSP-DB
  - Continue delegated control



# LSP Initiation from Active Stateful PCE

- So what if a PCE could initiate an LSP?
  - It fundamentally changes the nature of the PCE
- NMS requests an LSP
- PCE computes a path
- New PCInit message
- Instruction to set up an LSP
  - Instruction to set up an LSP
- LSP is set up
- TED is updated via IGP
- PCRpt
- It sets the scene for SDN
  - More of that later



# State of the Nation

## What Have We Built and Deployed?

- Packet networks have not been a roaring success for PCE
  - Initially, only Cisco implemented
  - It is implemented and deployed
  - Main use cases are
    - Dual-homed IGP areas
    - Centrally controlled TE domains
- There is a huge amount of research and experimentation
  - More than 20 projects funded by the EU have PCE as a core component
  - A number of operators have experimented in depth
- Commercial and Open Source Implementations
  - Software stacks from Metaswitch and Marben
    - But these are PCEP implementations, not full PCEs
  - Several Open Source implementations exist
  - Hardware vendors
  - Network operators
- The best take-up for PCE so far is in optical networks

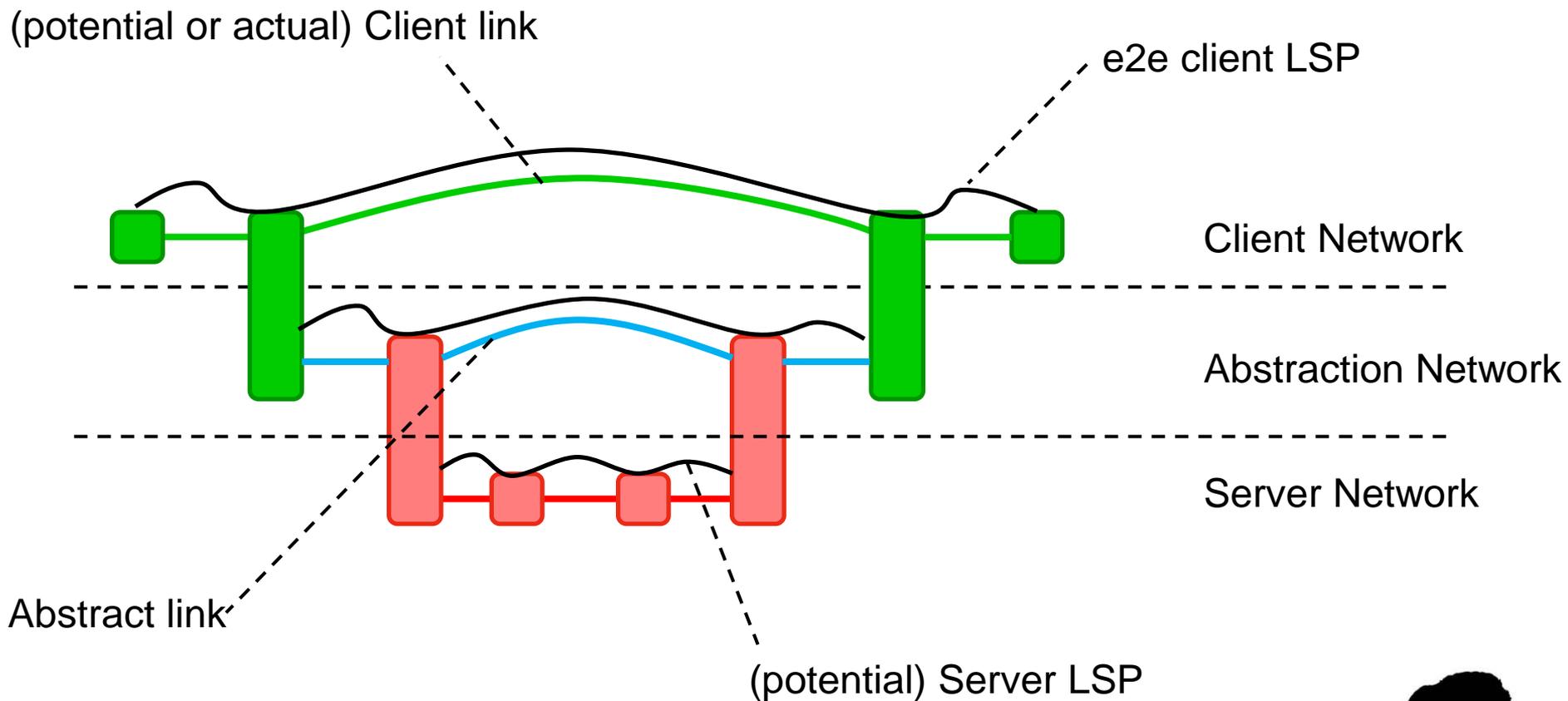
# What Are We Coding and Testing Now?

- Vendor constraints
  - An old Internet-Draft just published as RFC 7150
  - Allows standards-based PCEP implementation with vendor-specific hardware
  - Particularly handy for optical equipment
- GMPLS extensions
  - Surprising that this is still under development?
  - Most things work fine already
  - Some gap filling needed for special or minor cases (draft-ietf-pce-gmpls-aps-req)
    - Additional TE quality constraints (e.g., VCAT or single channel)
    - Adaptation capabilities
    - Asymmetric bandwidth
    - etc.
- Hierarchical PCE
  - The architecture is done (RFC 6805)
  - Solutions work progresses with implementations (draft-ietf-pce-hierarchy-extensions)
- Stateful PCE
  - Architecture, requirements, and protocol (draft-ietf-pce-stateful-pce)
  - Early deployments

# Into The Future

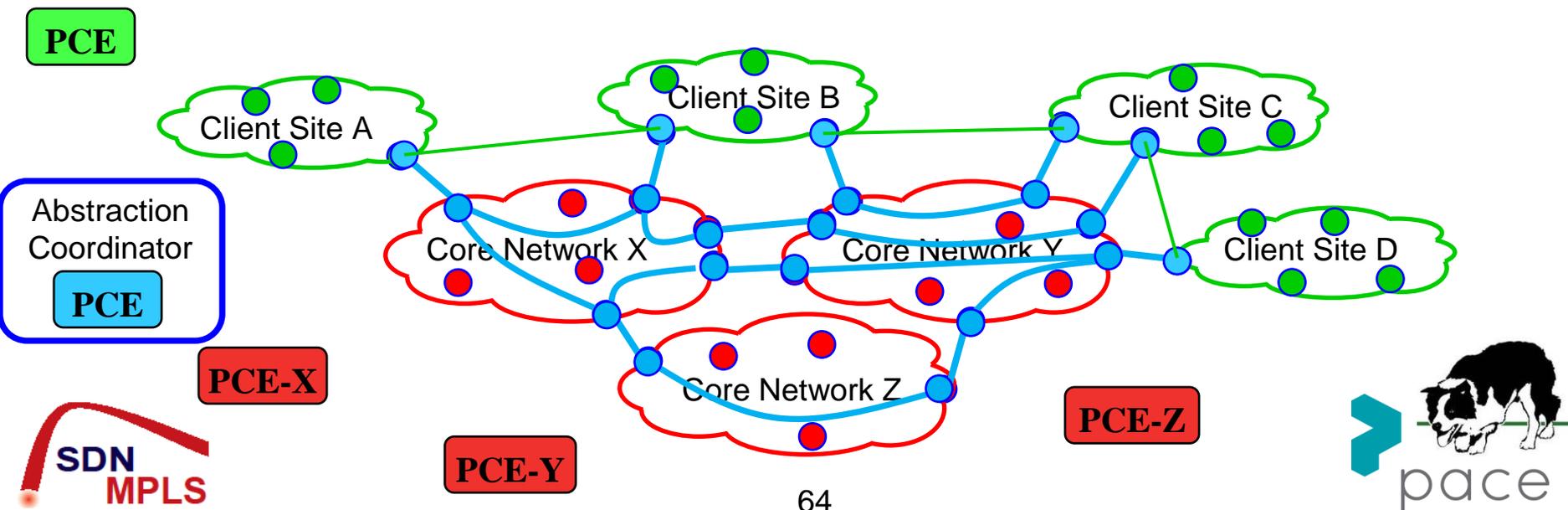


# PCE with Inter-Domain TE Abstraction Architectural Overview



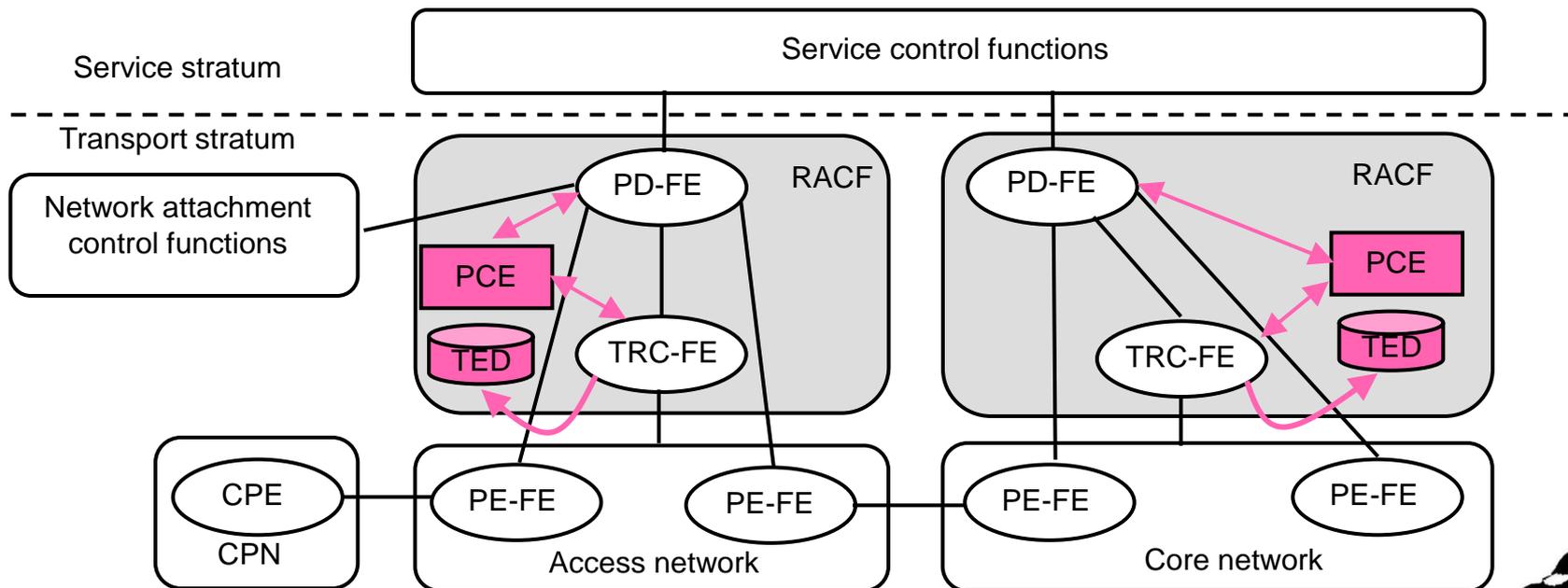
# The Abstract Network

- The client network has become partitioned
  - New client layer links are needed
- Core networks advertise potential connectivity
  - Abstraction layer network is constructed
- Abstraction coordinator computes paths in abstraction layer
- Links in client network are created



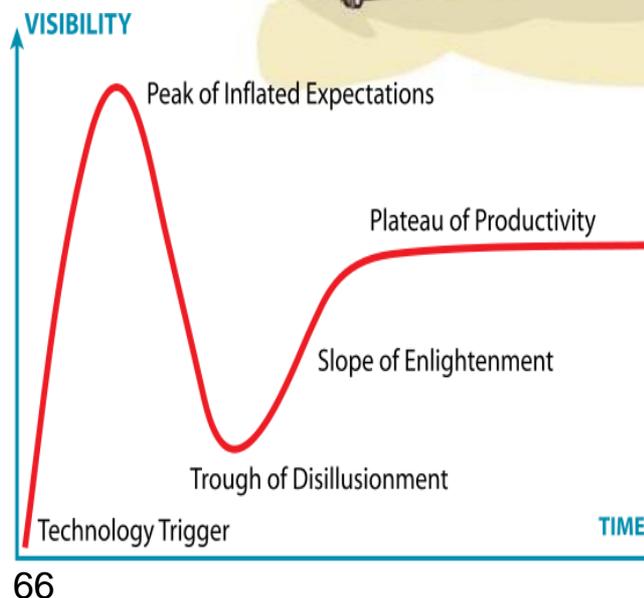
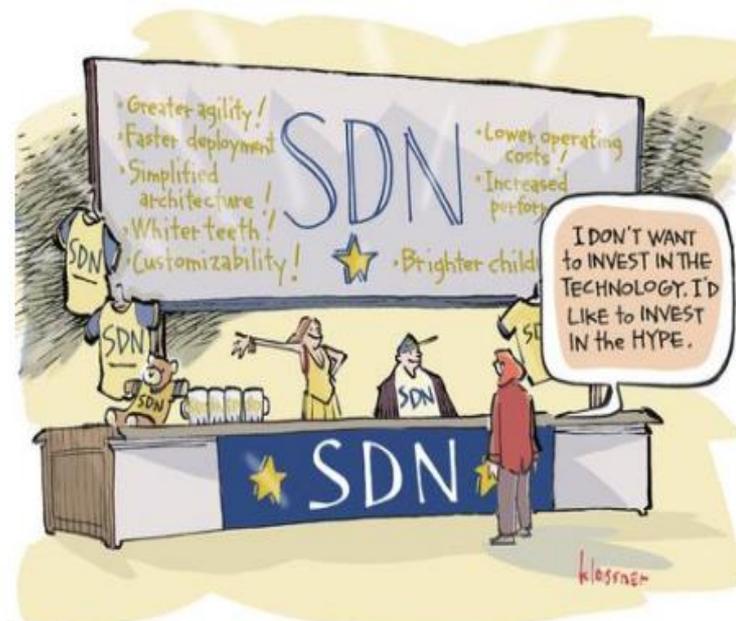
# Service Management – An Old View of SDN

- ITU-T's Resource and Admission Control Function (RACF)
  - Plans and operates network connectivity in support of services
- Policy Decision Functional Entity
  - Examines how to meet the service requirements using the available resources
- Transport Resource Controller Functional Entity
  - Provisions connectivity in the network (may use control plane)



# SDN : A Reality Check

- Why Software Defined Networking?
  - There's a hype in the industry!
  - Where are we on the hype cycle?
- Dispelling some myths
  - SDN is not a provisioning system or configuration management tool
- SDN is not a new protocol
- You do not buy "off the shelf" SDN
  - It's an architectural approach



# What do we mean by “SDN”?

## Software

- It's all software!
- We are looking for automation
- Tools and applications
- The goals of commercial SDN networks
  - Make our networks better
  - Rapidly provide cool services at lower prices
  - Reduce OPEX and simplify network operations
  - Enable better monitoring and diagnostics
  - Make better use of deployed resources
- Converged services are the future
- Converged infrastructure is the future
- There is a significant element of centralisation

## Driven or Defined

- Does it matter?

## Networks

- Management of forwarding decisions
- Control of end-to-end paths
- Whole-sale operation of network

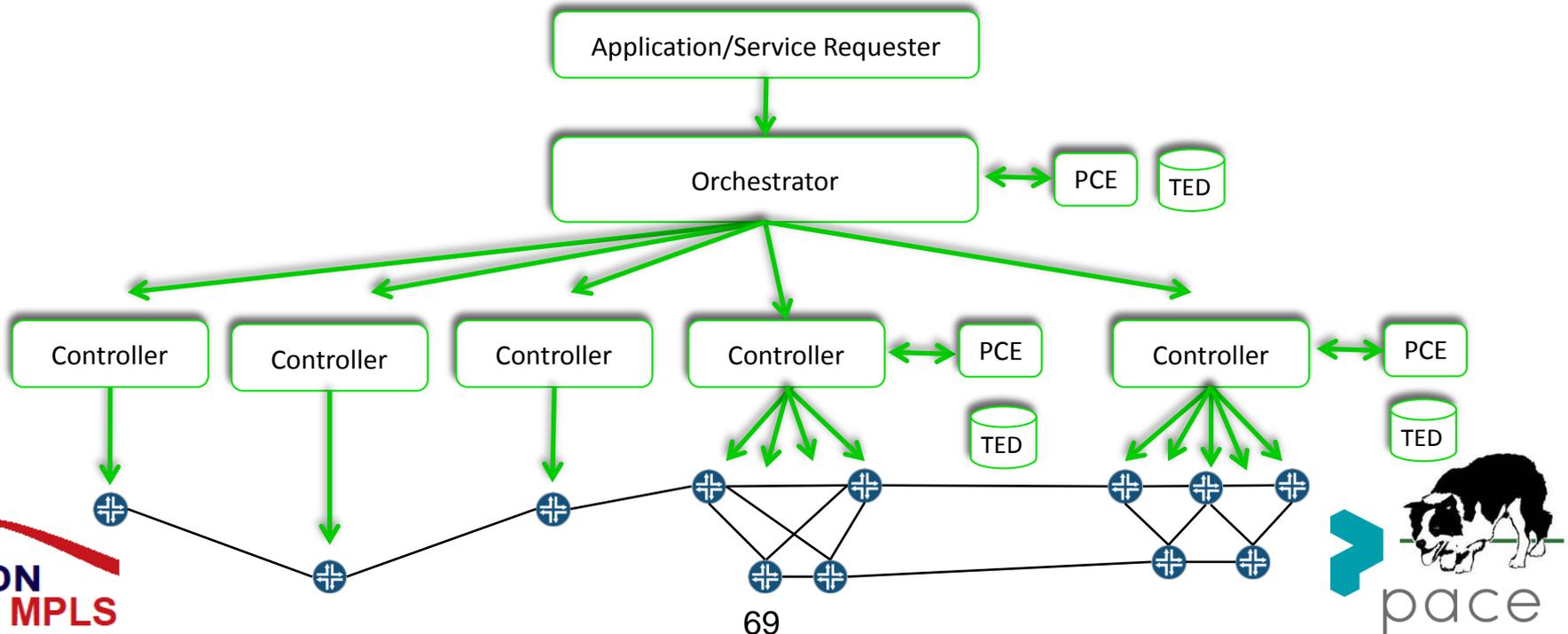


# Core Principles of SDN

- Centralized control,
  - Especially of end-to-end services
- Network convergence
  - Converged services and infrastructure
- Network “abstraction”
  - Partitioning of resources
- Network automation
- Application-to-network relationship
  - Provides access to the forwarding plane of network devices
- Momentum being driven for commercial benefits:
  - Rapidly provide cool services at lower prices
  - Reduce OPEX and simplify network operations
  - Enable better monitoring and diagnostics

# Bringing PCE to the SDN Feast

- It is crazy to talk about SDN without path computation
- PCE is an essential element for planning services in *any* network
- An Orchestrator cannot orchestrate without determining how traffic will flow
  - And that means that an Orchestrator needs path computation function
  - The PCE may be built into the Orchestrator or live as a separate component
- If a Controller controls more than one node must determine how traffic flows
  - And that means that a Controller may need path computation function
  - The PCE may be built into the Controller or live as a separate component



# PCEP as an SDN Protocol?

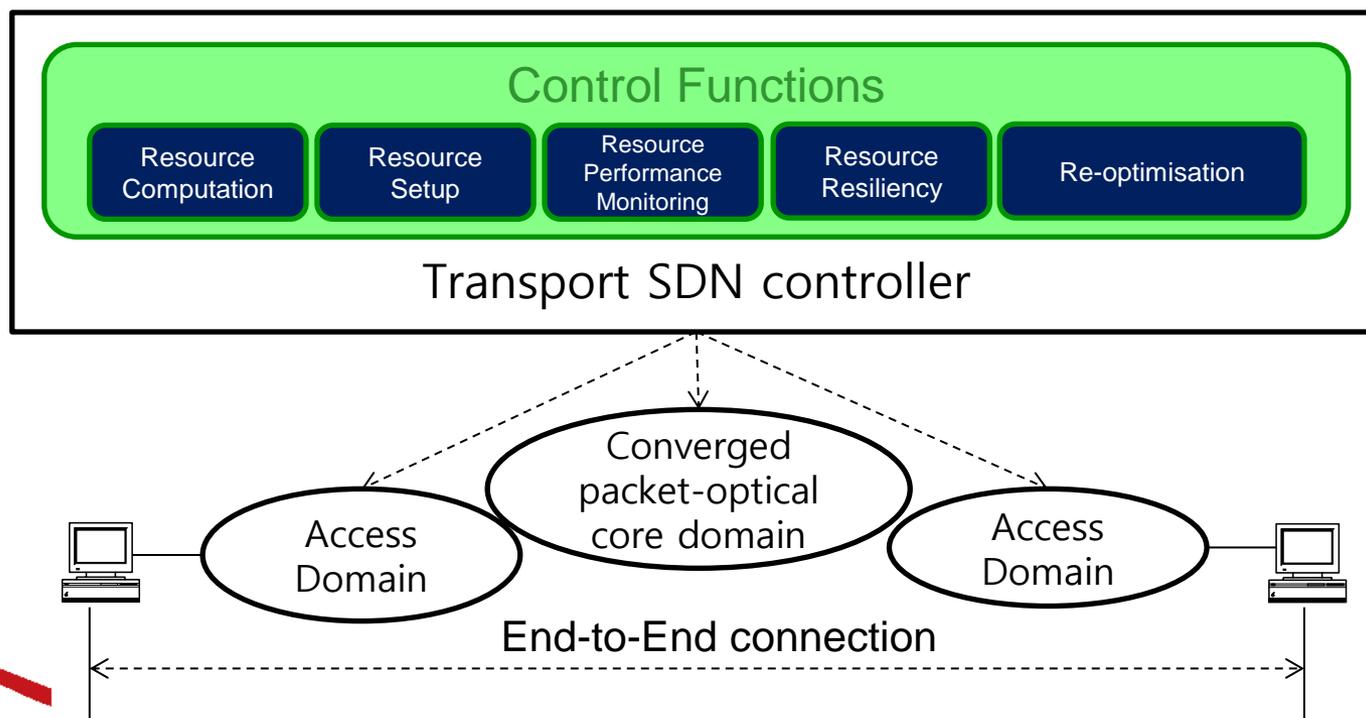
- We already talked about LSP initiation from a stateful PCE
  - In protocol terms it is a simple step
  - Instead of suggesting LSPs, a PCE can provision LSPs
- Now PCEP can be seen as a full-scale provisioning protocol
  - I can provision anything for which I might have asked for a path
    - End-to-end LSPs (signalled by the control plane)
    - A fragment or segment of an LSP
    - The forwarding instructions on a single node
- Now PCE can be integral to the SDN components
  - I can use PCEP as an SDN Controller protocol
  - And/or as the Orchestrator-to-Controller protocol
- This raises the question of “competition” with OpenFlow
  - But there are plenty of other competitors
    - Perhaps the strongest is Netconf/YANG

# Can We Define “NFV”?

- Operators use a variety of proprietary appliances to provide network functions when delivering services
- Deploying a new network function often requires new hardware components
  - Integrating new equipment into the network takes space, power, and the technical knowledge
  - This problem is compounded by function and technology lifecycles which are becoming shorter as innovation accelerates
- The concept of virtualization is well-known and has been used for many years
  - Operating system virtualization (Virtual Machines)
  - Computational and application resource virtualisation (Cloud Computing)
  - Link and node virtualisation (Virtual Network Topologies)
  - Data Center Virtualisation (Virtual Data Center)
- Network Function Virtualization
  - Virtualize the *class* of network function
  - Replace specialist hardware with instances of virtual services provided on service nodes in the network
  - Enables high volume services and functions on generic platforms
- Virtualizing network connectivity for services and applications is just another facet of NFV

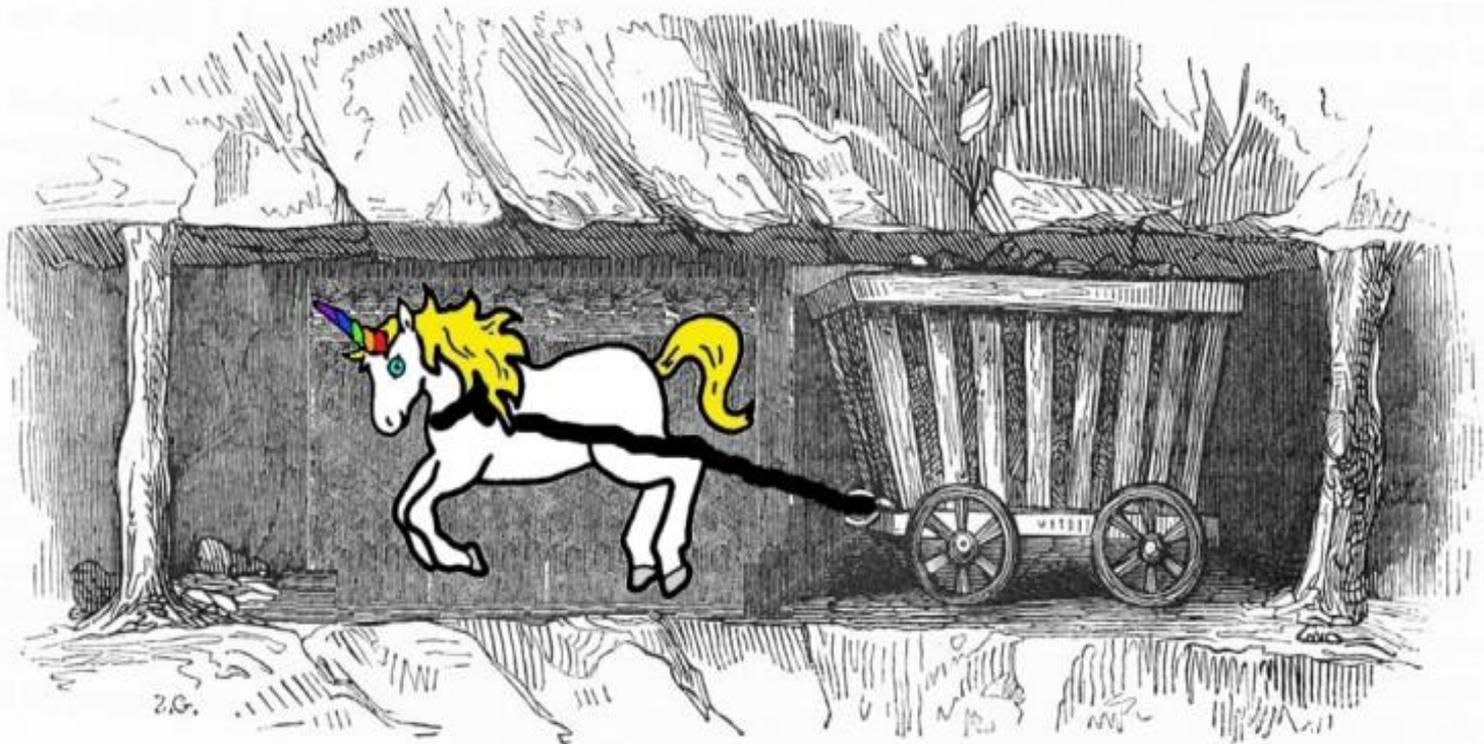
# SDN & PCE as enablers for Network Virtualization

- Consider Transport SDN as an example
  - Integrates Packet, TDM, and Optical Layer into a single converged network
  - Requires centralized control functions including resource computation
  - Uses southbound control interface



# Harnessing the Unicorn

- We've established that PCE is a wonderful thing
- We know that SDN and NFV offer a bright future for networking
- How do we bring PCE fully into the picture and make it work for us?



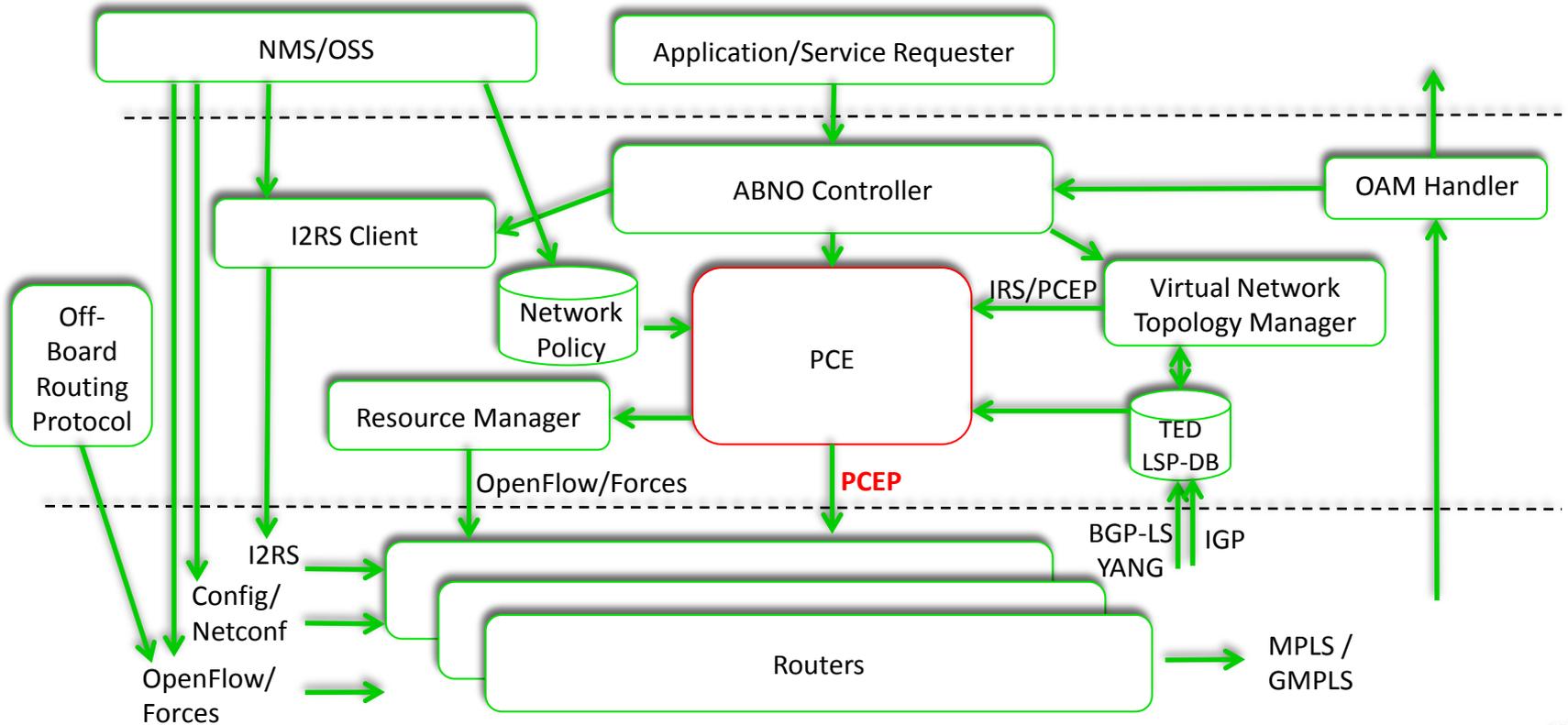
# Building a Functional Architecture

- The purpose of a functional architecture is to decompose a problem space
  - Separate distinct and discrete functions into separate components
  - Identify the functional interactions between components
- An architecture is not a blue-print for implementation!
  - Components are abstract functional units
  - They can be realized as separate software blobs on different processors
  - Or they can all be rolled into one piece of spaghetti code
  - And they can be replicated and distributed, or centralized
- A protocol provides a realization of the interaction between two functional components
  - You only need to use it when the components are separated
- There have been many useful attempts to document architectures for SDN and NFV
- Our work has tried to present a wider picture
  - Address a range of network operation and management scenarios
  - Encompass (without changing) existing profiles of the architecture
  - Embrace SDN and NFV without becoming focused or obsessed with them
  - Highlight existing protocols and components

# Application-Based Network Operation (ABNO)

- A PCE-based Architecture for Application-based Network Operations
  - draft-farrkingel-pce-abno-architecture
- Network Controller Framework
  - Avoiding single technology domain “controller” architecture
  - Reuse well-defined components and protocols
    - Discovery of network resources and topology management.
    - Routing and path computation
    - Multi-layer coordination and interworking
    - Policy Control
    - OAM and performance monitoring
- Support a variety of southbound protocols
  - Leveraging existing technologies, support new ones
- Integrate with defined and developing standards across SDOs

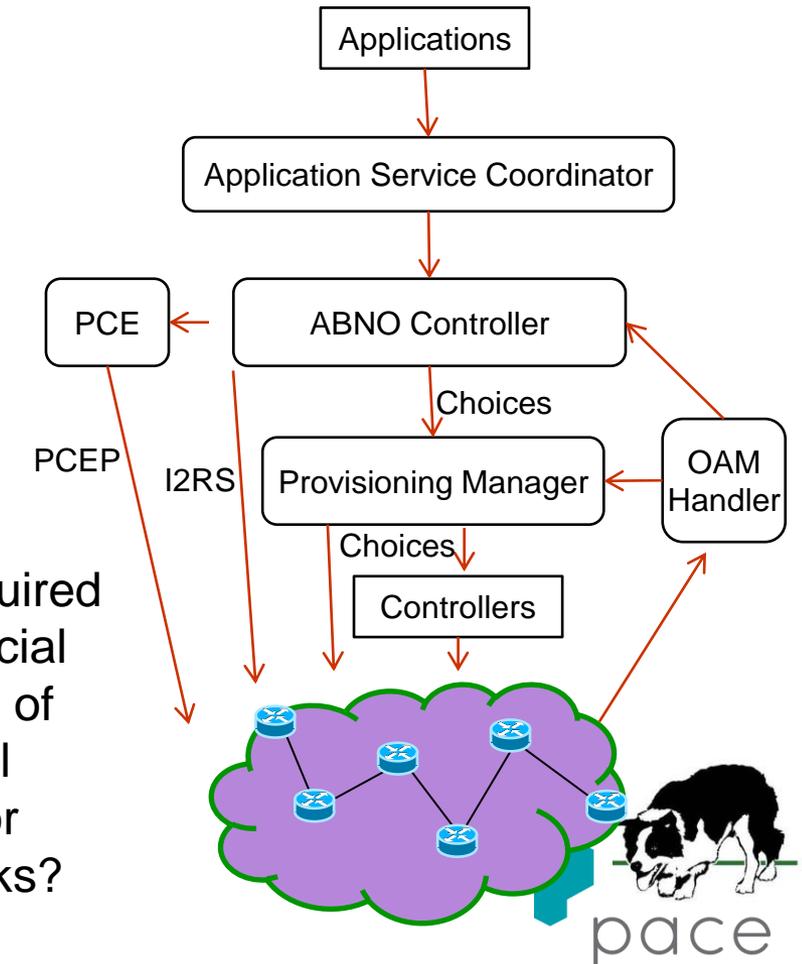
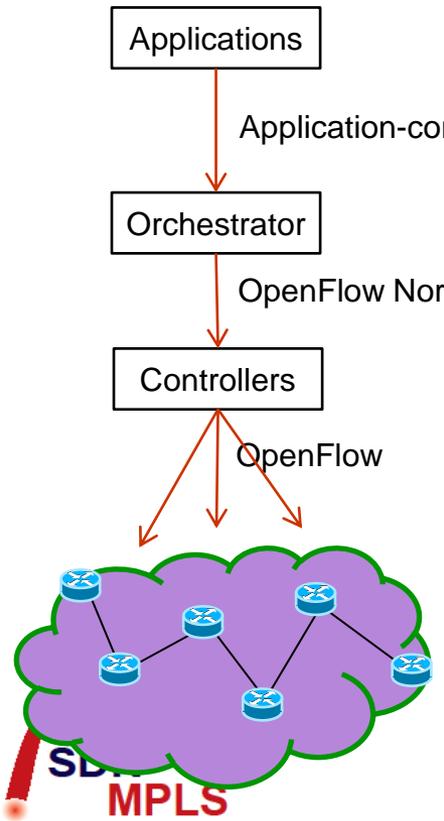
# Application-Based Network Operation (ABNO)



# Compare ABNO with SDN Architecture

- A richer function-set based on the same concepts
- Enables the use of OpenFlow and other protocols
- There are implementation/deployment choices to be made

Minimum required for SDN controller of infrastructure



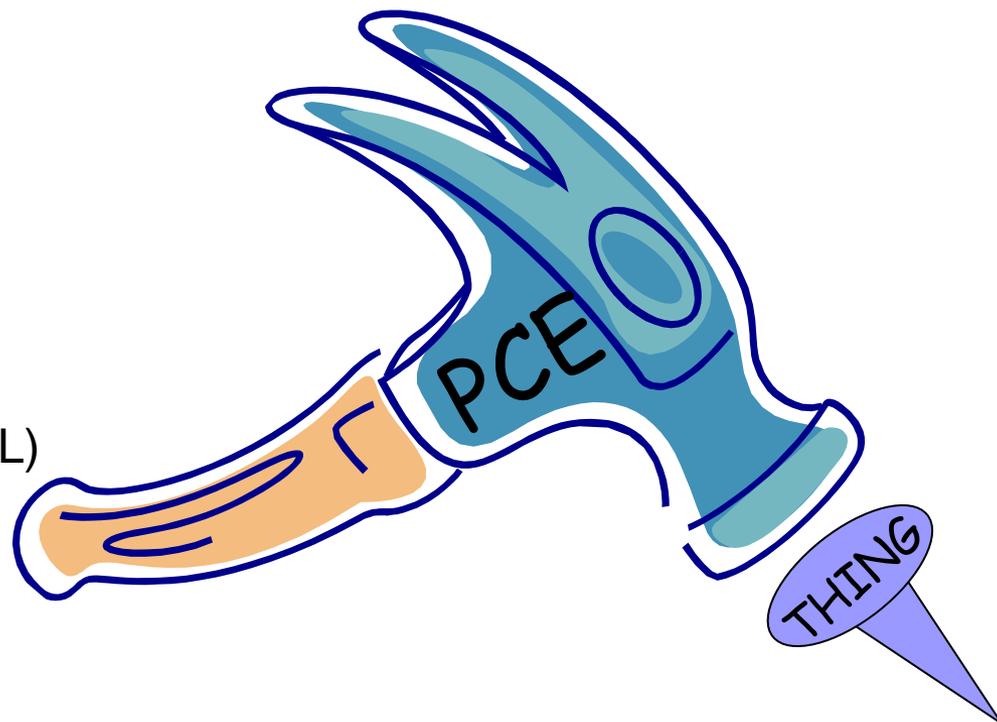
What is required for commercial deployment of SDN control platforms for real networks?

# PCE and Emerging Technologies



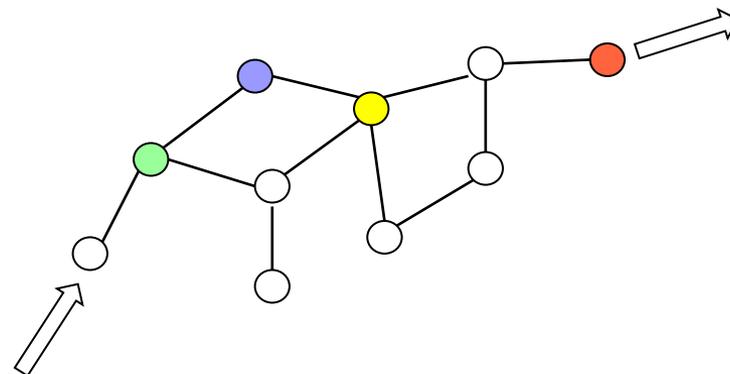
# The Future of PCE

- Ideas and applications for PCE keep growing
- Every new technology looks to use PCE
  - IP-Over-Flexigrid Networks
  - Ever more SDN ideas
  - NFV
  - IP Fast Reroute
  - MPLS Source Routing
  - Multi-segment Pseudowires
  - Managed Ethernet (e.g. TRILL)
  - Power-aware networking
  - Internet of Things
- Even applied to non-telecoms applications
  - Utility companies



# Source Routing

- MPLS-TE and GMPLS are a form of source routing
  - A packet is placed on an LSP according to a choice at the source
  - Once on the LSP, the packet's path is predetermined
  - The Explicit Route in signalling has determined the path
  - The mechanisms uses PCE at LSP establishment
  - Call this “per-flow source routing”
- “Per-packet source routing” is being re-investigated
  - Familiar to old IPv4 enthusiasts
  - Each packet carries the path to follow
  - Now investigated for MPLS networks
    - Pop a label and forward the packet
    - Interesting for ECMP, FRR, TE
  - Many things think about
    - How deep is the label stack?
    - Can each packet take a different path?
    - How does PCE supply the paths to use?

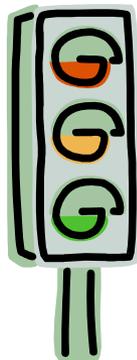
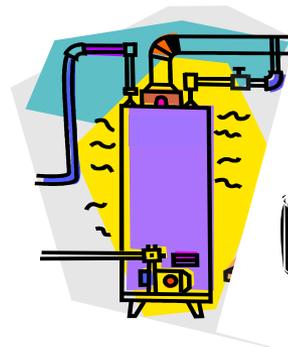


# Power-Aware Networking

- Route traffic to reduce network-wide power consumption
- There are many unanswered questions
  - Will I turn off a router line card?
  - Does it make sense to turn down provisioned optical resource?
  - What aspects of equipment consume the power?
  - Does a card at 100% consume more power and need more cooling than two cards at 50%
- Early research shows **potential** savings of up to 25% in IP networks
  - That probably means real savings as low as 10%
- If this catches on then PCE will be important

# Internet of Things

- Interconnection of people and devices
  - Key features are low power, low memory, and low-quality links
- Interconnection may be:
  - Route-over
  - Mesh-under
- Routing protocols use power and memory
- Packets may be source-routed
- New IETF work “6TSCH”
  - IEEE802.15.4e TSCH time synchronised channel-hopping wireless
  - PCE provides MAC layer transmission scheduling and end-to-end routing

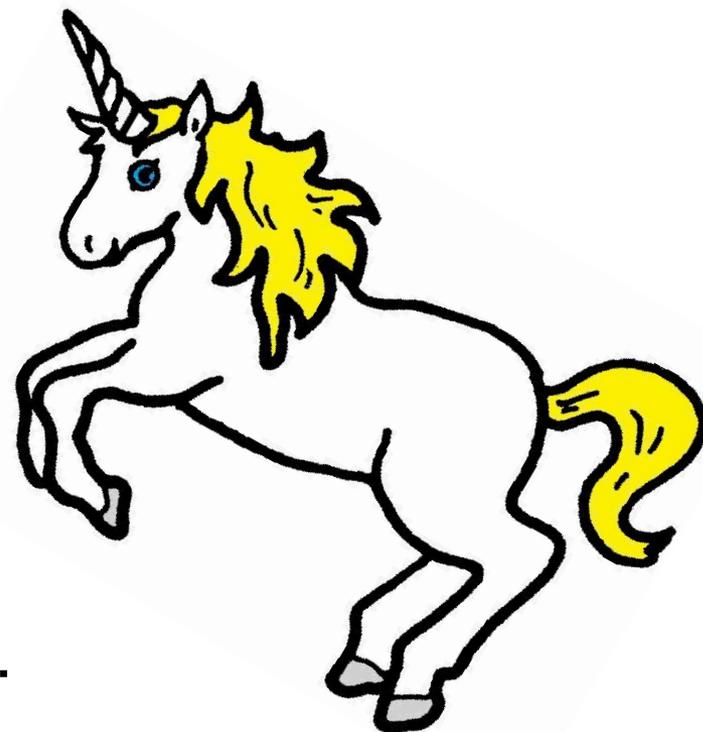


## Is The Future Bright for PCE?

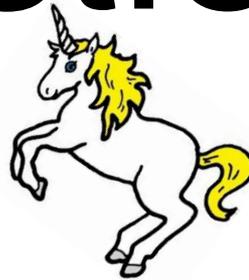
- There are many applications that call for the computation of paths or the selection of resources
- There is no real purpose in debating the definition of PCE
  - However, it is helpful to state what we mean when we say “PCE”
  - It is important to clarify what **you** mean when you talk about PCE
- A key differentiator is the level of function in the interfaces
  - Base PCEP (RFC 5440)
  - Extended PCEP for stateful PCE (draft-ietf-pce-stateful-pce)
  - Additional PCEP function
  - Other protocols
- There is still no “killer application” for PCE
  - All the great ideas are just great ideas until they are implemented and widely deployed

# PCE Solves All Known Problems

- It is true that PCE can be used in a surprisingly large number of environments
  - Much more than we considered when we worked on RFC 467
- It appears that PCE is real
  - Some actual deployments
  - A lot of potential
- PCE is my baby and of course...
  - My baby is beautiful
- Despite evidence to the contrary...
  - **PCE is not magic**



# Questions?



Follow-up

[adrian@olddog.co.uk](mailto:adrian@olddog.co.uk)