Support of RFC5440 in OpenDaylight

Dr. Olivier Dugeon, Orange Labs

olivier.dugeon@orange.com

Orangeexpert
future networks

orange™

Outline

Context & Use Cases

RFC5440

Traffice Engineering Database

SAMCRA algorithm

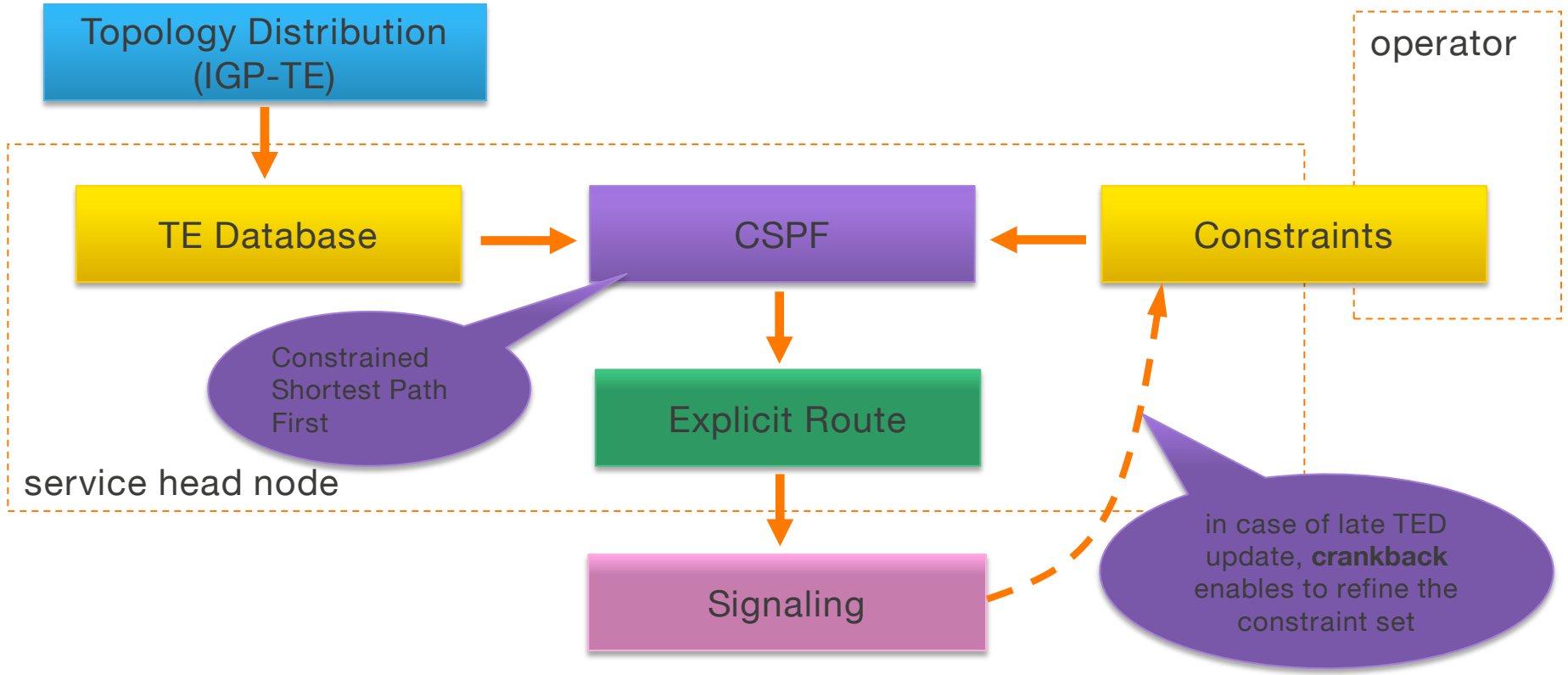
Experimental Prototype

Development issues

Demo

Next Steps

Constraint-Based Routing Principles in (G)MPLS



Multi-Path Problems

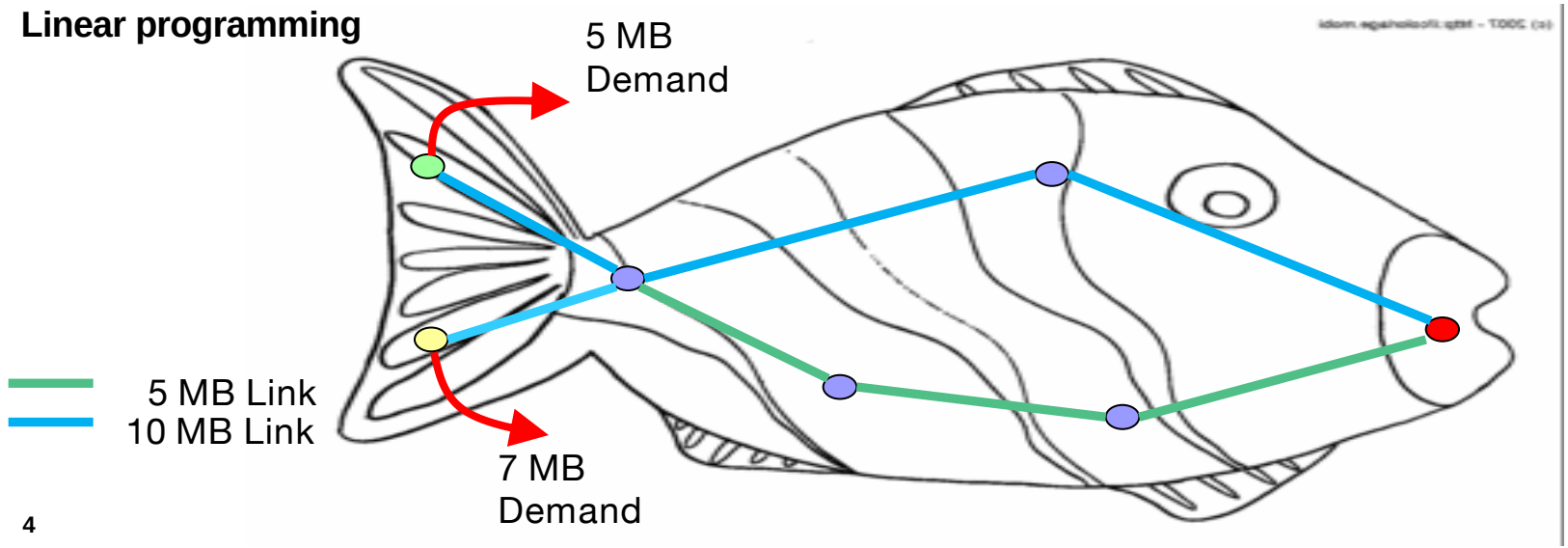
The classic “fish” problem of ordered provisioning

Diverse routes for protection paths

More sophisticated algorithms

k-shortest paths

Linear programming



Remind Me: What is PCE & PCEP?

From RFC 4655

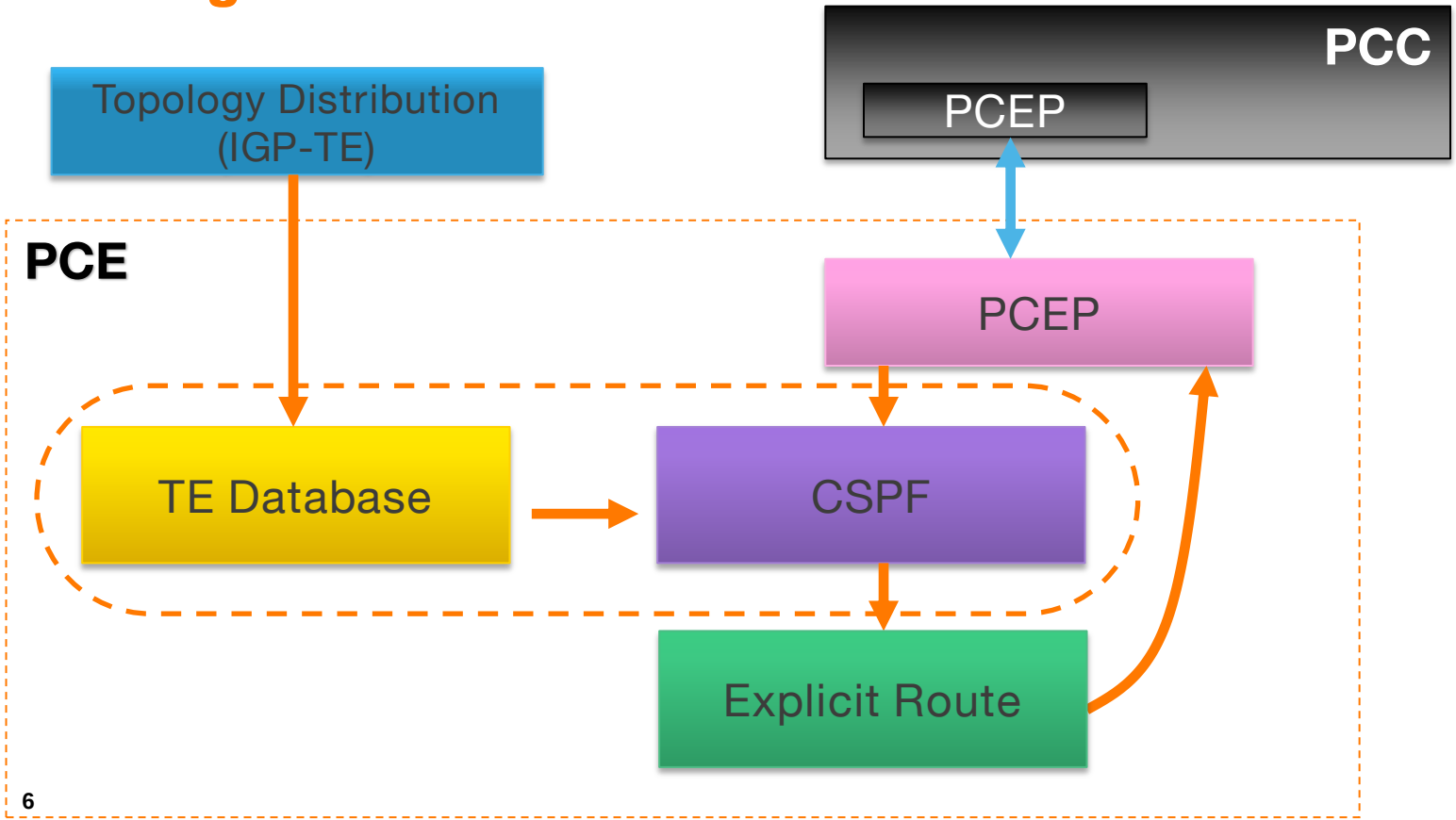
“PCE: Path Computation Element. An entity (component, application, or network node) that is capable of computing a network path or route based on a network graph and applying computational constraints”

From RFC 5440

“This document specifies the Path Computation Element (PCE) Communication Protocol (PCEP) for communications between a Path Computation Client (PCC) and a PCE, or between two PCEs”

“Such interactions include path computation requests and path computation replies as well as notifications of specific states related to the use of a PCE in the context of Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineering”

Path Computation Element Building Blocks

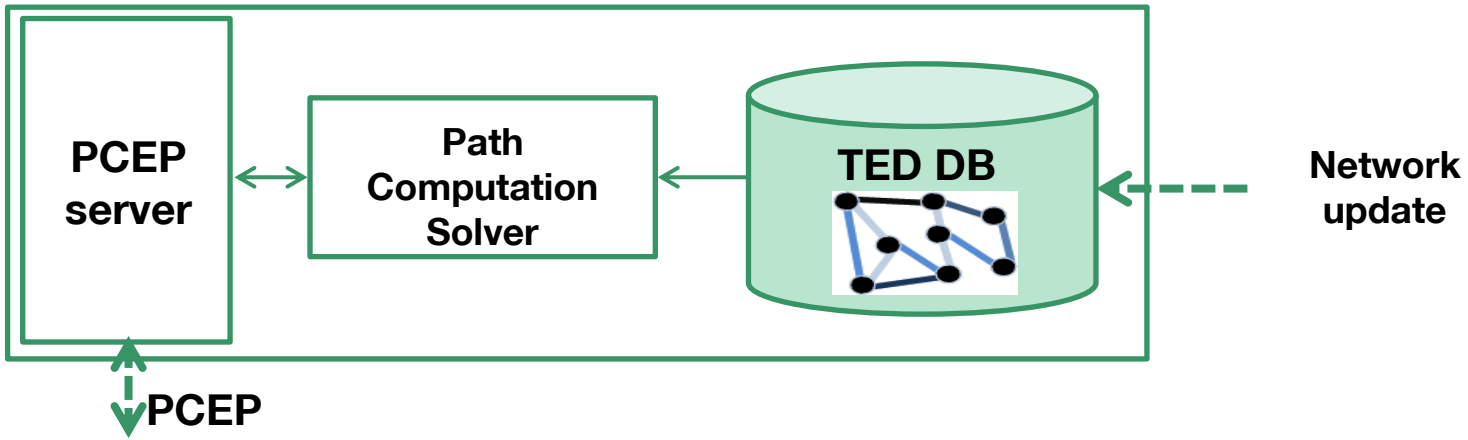


The TED & PCE

Traffic Engineering Database (TED) is essential internal component of a PCE

Provides the updated snapshot of the controlled network and its resources

PCE algorithms resort to TED as primary information source input



Model the network topology as a Connected Graph for performance efficiency

Automatic TED fulfilment

Listen to BGP Link-State information from the Data Store

Start with Linkstate Topology

- **Missing information (e.g. Node SID, Extended Metrics)**
- **Sometimes Link-State is empty (See Later for the cause)**

Finally used Linkstate Routes

- **All information from BGP-LS are there**
- **Already up to date**

TED is updated once RSVP-TE tunnel is setup

Done automatically as routers update their TE links

- **Need improvement for Segment Routing as there is no more signalling**

Self-Adaptive Multiple Constraints Routing Algorithm (SAMCRA)

Multiple Constraints Routing is a N/P Complete problem

Intensively studied in the past

Many algorithms have been proposed in the past trying to optimize one or another objective

We chosen SAMCRA proposed by Fernando Kuiper https://fernandokuipers.nl/

It is an exact Qos Routing algorithm (mathematical proof)

Even if it is complex, it has been proof as one of the best algorithm

Handle multiple constraints as a composite metric

Separate Concave metric (i.e. bandwidth) from Additive metric (e.g. delay)

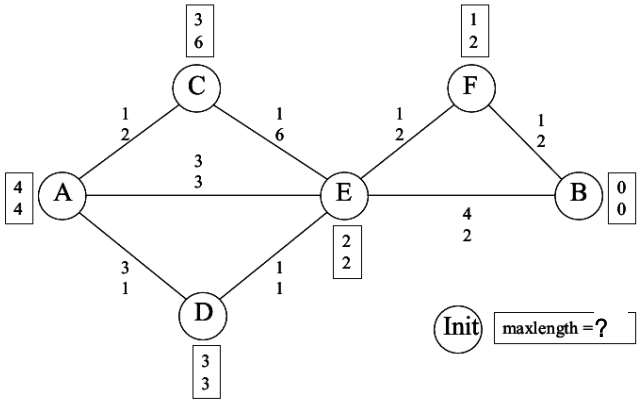
Use the notion of ‘Dominate’ node to select the next hop

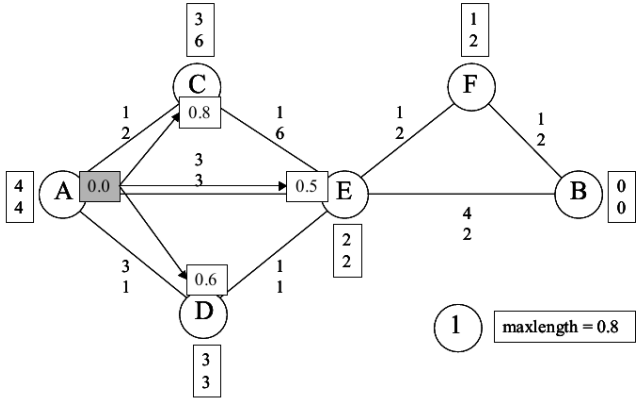
Improve with ‘look-a-head’ variant

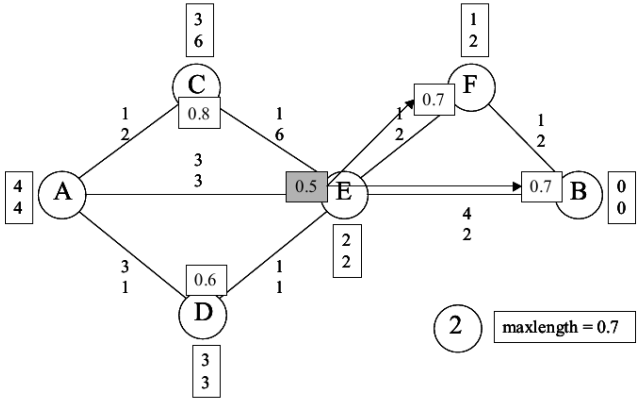
During Path Computation, evaluate with dijkstra algorithm the remaining part of the path between current Node and the destination to improve selection of nexthop in case of composite metric is equivalent and there is no real dominate node

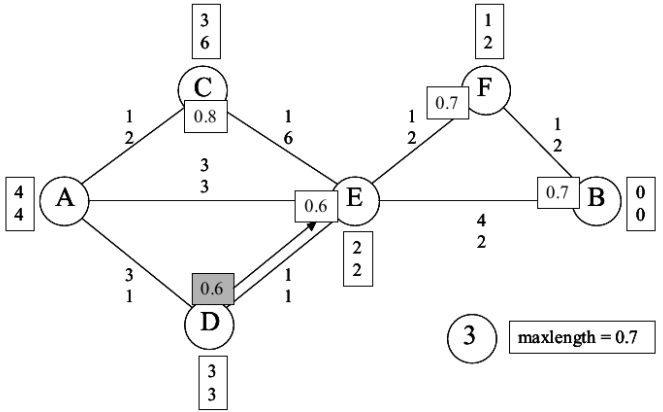
SAMCRA example (source F. Kuiper thesis)

Find Path from A to B with constraints (10,10): Step Init to 3



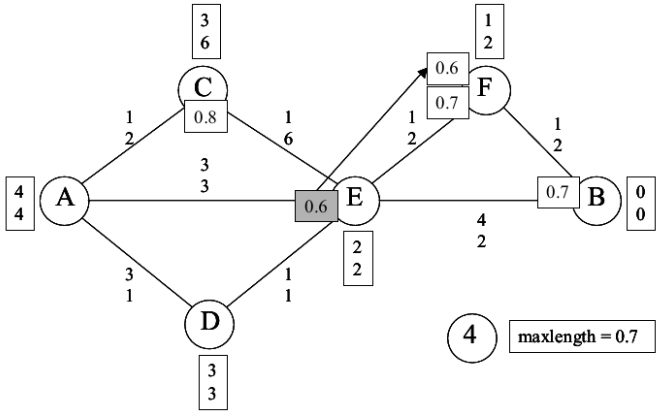


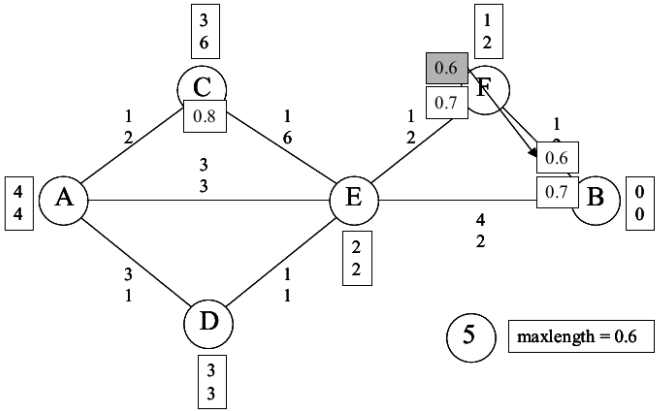


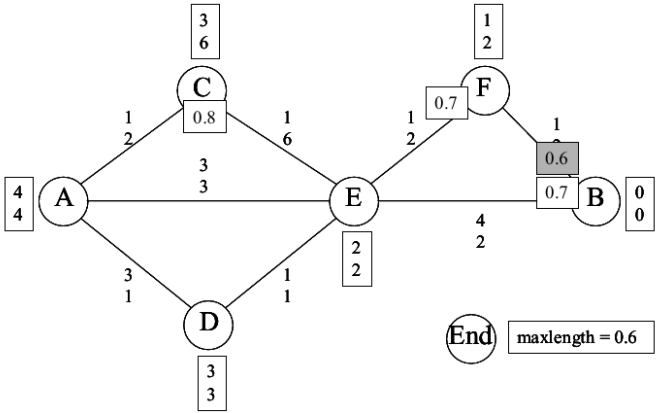


SAMCRA example (source F. Kuiper thesis)

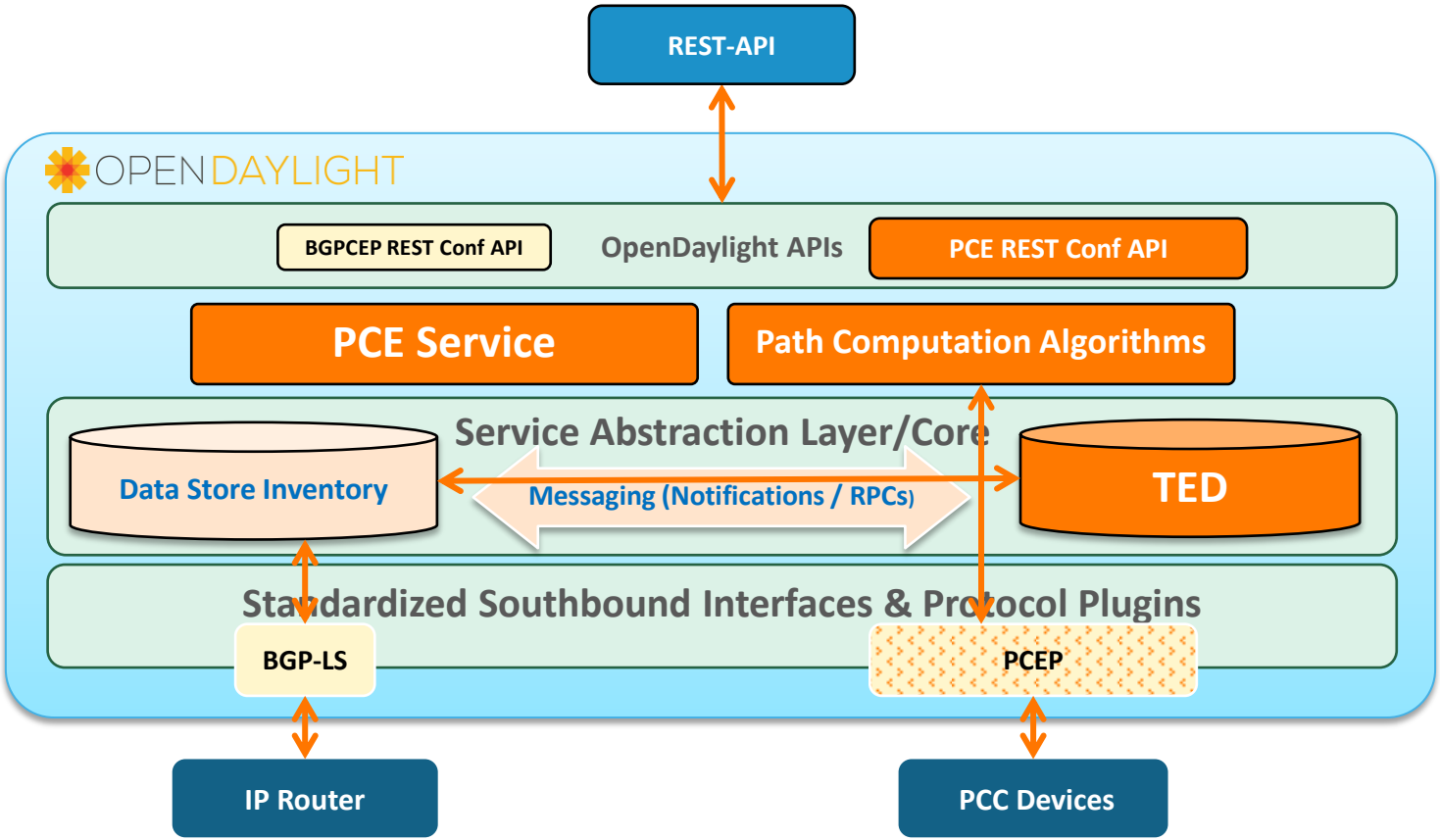
Find Path from A to B with constraints (10,10): Step 4 to End







PCE architecture



PCE services

RFC5440 compatibilities

Able to give a computed path to a PCC following a PCReq message

- **PCRep message convey ERO or NO-PATH**

Path Computation Service REST - API

Setup/Remove LSP tunnel by trigger Path Computation Algorithm

- **PCInit method used with computed ERO**

Request a Path similar to a PCReq message

- **ERO is provided in the answer as json**

TED REST – API

Get Graph detail

Rest Graph

Setup Graph though specific topology file

DEMO

Development Issues (1/2)

Impossibility to model connected Graph in Yang

Yang define flat tree only

Using Hash Array could overcome the limitation but at the price of performance

- **Not possible on large scale network (more than 1000 nodes / 10000 links)**

Why Bandwidth is not converted to Float?

Bandwidth as defined in RFCs use IEEE floating point format (32bits)

- **Why yang-tools not convert bandwidth in Java float type?**
- **Keeping Bandwidth as a 4 bytes tuple leads to poor performance when computing path**
  - Need to convert 4 bytes tuple to float each time a link is explored
- **Impose to use big decimal instead in the TED Graph to avoid conversion**

Development Issues (2/2)

Grouping in yang model are not converted as generic Class

Typedef in yang is reserved to pre-defined type

- **Yang does not authorize typedef for grouping while it could be useful**
- **Yang tools duplicate grouping per container which is not useful in certain cases:**
  - E.g. route object could not be easily copied from ERO to RRO

BGP Link state Topology is not handle late registry

Blueprint starts bundle with delay which avoid PCEP feature used BGP link state registry

- **Path feature register TED listener in BGP link state but never received data store notification**
  - BGP link state send notification about new topology only to already registered functions
  - Late registry are not advertise about existing topology in the data store
  - Sometimes problem occurs with Link State functions

Next Steps

Clean & Rebase code on master

Publish alpha version on github prior to submit code to ODL Gerrit to get feedback

Enhance yang model of TED

Define Vertex and Edge yang model

- **Store flat Graph in Data Store**
  - To Get/Modify/Create directly Graph
- **Keep connected Graph in memory**
  - For better performance
- **Add IPv6 & Segment Routing information**

Add Path in Configuration Data Store

Persistent Tunnel Setup across ODL restart

Improve Path Algorithms

Add IPv6 & Segment Routing

- **SAMCRA is agnostic about IP address, SID …**
  - But ERO must be formatted accordingly

Add Path Algorithm registry

- **To allow contributors adding their own algorithms easily**
  - No need to re-code topology acquisition
  - Support for Objective Function (PCReq Object)

Thanks

This work has been performed in the framework of the H2020-ICT-2014 project 5GEx http://www.5gex.eu (Grant Agreement no. 671636), which is partially funded by the European Commission.