Extensions to the Path Computation Element protocol (PCEP) for residual path bandwidth support

draft-lazzeri-pce-residual-bw-00

Francesco Lazzeri Daniele Ceccarelli	Ericsson
Young Lee	Huawei

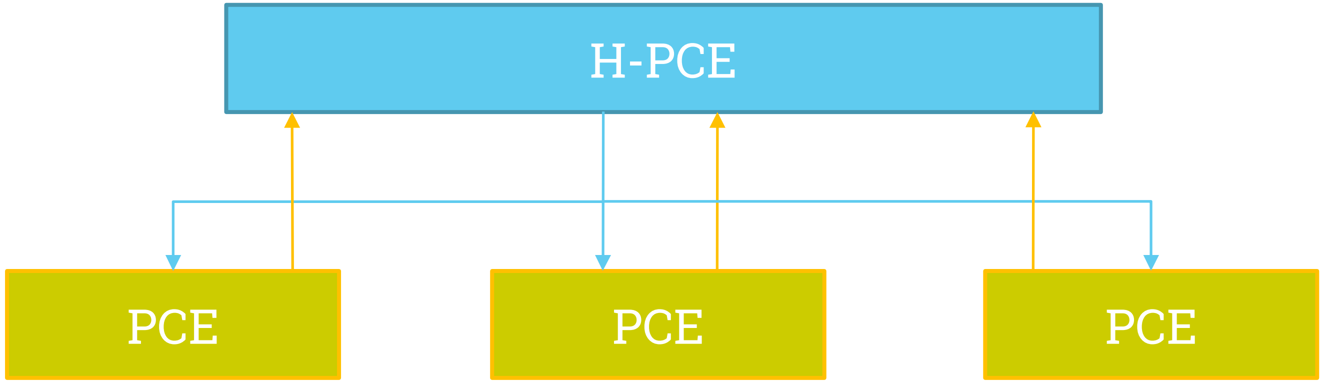
Introduction

- Proposal for the introduction of additional metrics in PCEP
  - **Path unreserved bandwidth at a given priority**: the minimum value of the unreserved bandwidth at that priority among all the links along the path
  - **Path residual bandwidth**: the minimum value of the free physical bandwidth among all the links along the path
- Rationale for the introduction of these metrics
  - When metrics are returned as a result of the path computation (using C bit as specified in RFC5440) they can **be used to know how much traffic can still be routed through the path just computed** (avoiding to recompute it if new similar requests happen)
  - **Optimizing a path against unreserved or residual bandwidth allows a better usage of the network resources**, reducing network blockability
  - **Putting constraints on the values of unreserved or residual bandwidth also helps preventing network bottlenecks**

Background

- RFC5541 defines something similar : the objective function MBP (maximum residual bandwidth path)
  - It enables to compute the path maximizing the minimum value of the residual bandwidth (that is the physical bandwidth left free on the links along the path)
  - The path with the largest “bottleneck” is returned
- However:
  - The value of the bottleneck (which is the main objective of this proposal) **cannot be returned as MBP is an OF and not a metric**
  - **Only the physically available bandwidth is taken into account**: it’s also needed (and possibly more important) to consider the available bandwidth per priority level
  - **It’s not possible to put constraints on the bottleneck of the path**, that is finding a path which optimizes another metric and has a bandwidth bottleneck not narrower than a given value.

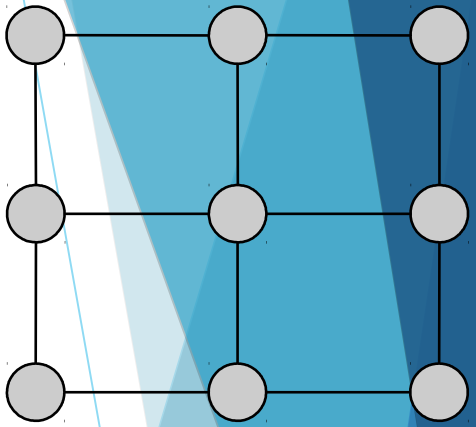
Use case

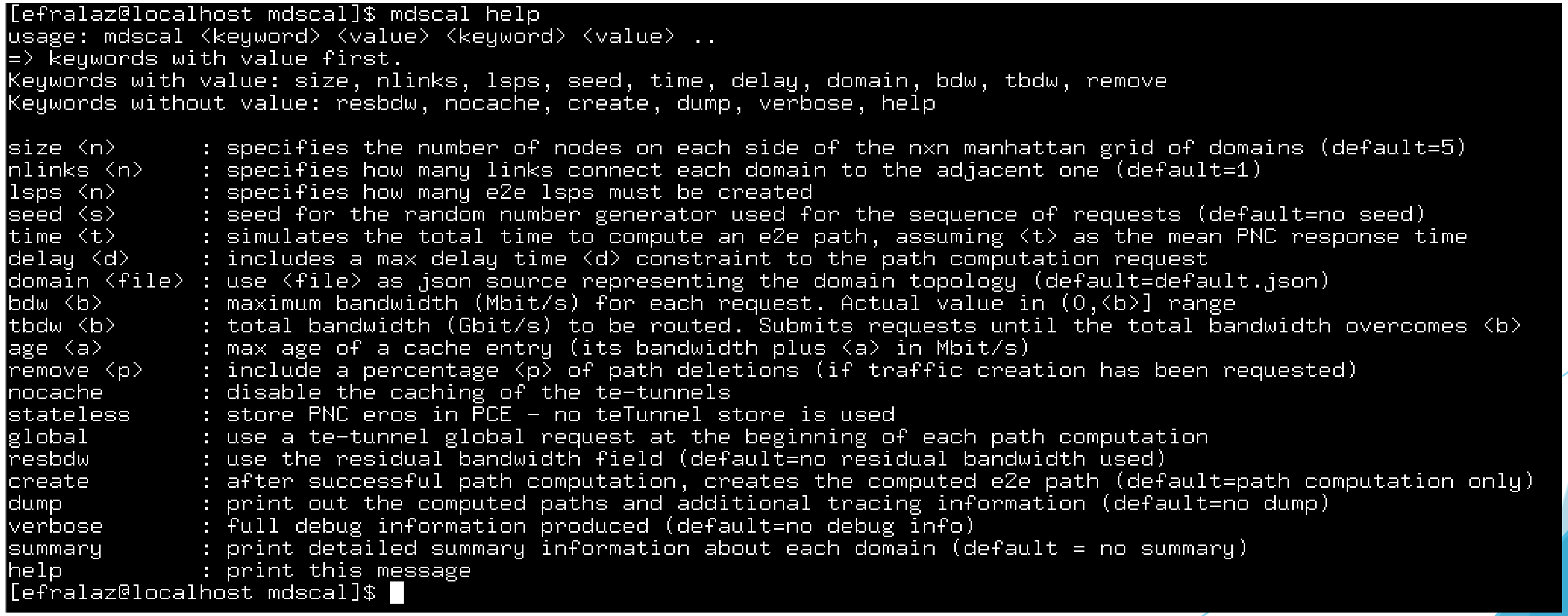


- H-PCE is a path computation element in charge of computing **end-to-end paths** across a network with a number of domains
- Part of or all the domains can't/**won't provide full topology information to the H-PCE**, so H-PCE must request path computation to the PCEs (**ACTN black/grey topology**)
- If information about unreserved bandwidth can be returned along with the computed path, H-PCE can use this information to **avoid requesting again the same (or similar) path computation** and reuse instead the already computed path.
- If for some reason (e.g. faults or deployment of traffic consuming the bandwidth of the stored path) the unreserved bandwidth becomes insufficient, the relevant LSP deployment will fail and H-PCE will request again the path computation, getting a fresh path and the relevant unreserved bandwidth, as needed.

Assessment

- Simulator including one H-PCE and a configurable number of PCEs
  - Manhattan-like network of domains.
  - All domains are “black”, therefore no topology information is provided to H-PCE
  - Different sizes of the domain grid can be tested
  - Each domain includes an arbitrary network described by a user file.
  - Configurable number of path computations can be run. Random seed configurable to produce the same sequence of requests
  - Simulation of LSP creation and LSP removal is possible
  - Different ways to request paths to domain (single requests versus cumulative)





```
[efralaz@localhost mdscal]$ mdscal help
usage: mdscal <keyword> <value> <keyword> <value> ..
=> keywords with value first.
Keywords with value: size, nlinks, lsps, seed, time, delay, domain, bdw, tbdw, remove
Keywords without value: resbdw, nocache, create, dump, verbose, help

size <n>      : specifies the number of nodes on each side of the nxn manhattan grid of domains (default=5)
nlinks <n>    : specifies how many links connect each domain to the adjacent one (default=1)
lsps <n>      : specifies how many e2e lsps must be created
seed <s>      : seed for the random number generator used for the sequence of requests (default=no seed)
time <t>      : simulates the total time to compute an e2e path, assuming <t> as the mean PNC response time
delay <d>     : includes a max delay time <d> constraint to the path computation request
domain <file> : use <file> as json source representing the domain topology (default=default.json)
bdw <b>       : maximum bandwidth (Mbit/s) for each request. Actual value in (0,<b>] range
tbdw <b>      : total bandwidth (Gbit/s) to be routed. Submits requests until the total bandwidth overcomes <b>
age <a>       : max age of a cache entry (its bandwidth plus <a> in Mbit/s)
remove <p>    : include a percentage <p> of path deletions (if traffic creation has been requested)
nocache       : disable the caching of the te-tunnels
stateless     : store PNC eros in PCE - no teTunnel store is used
global        : use a te-tunnel global request at the beginning of each path computation
resbdw        : use the residual bandwidth field (default=no residual bandwidth used)
create        : after successful path computation, creates the computed e2e path (default=path computation only)
dump          : print out the computed paths and additional tracing information (default=no dump)
verbose       : full debug information produced (default=no debug info)
summary       : print detailed summary information about each domain (default = no summary)
help          : print this message
[efralaz@localhost mdscal]$ █
```

Results

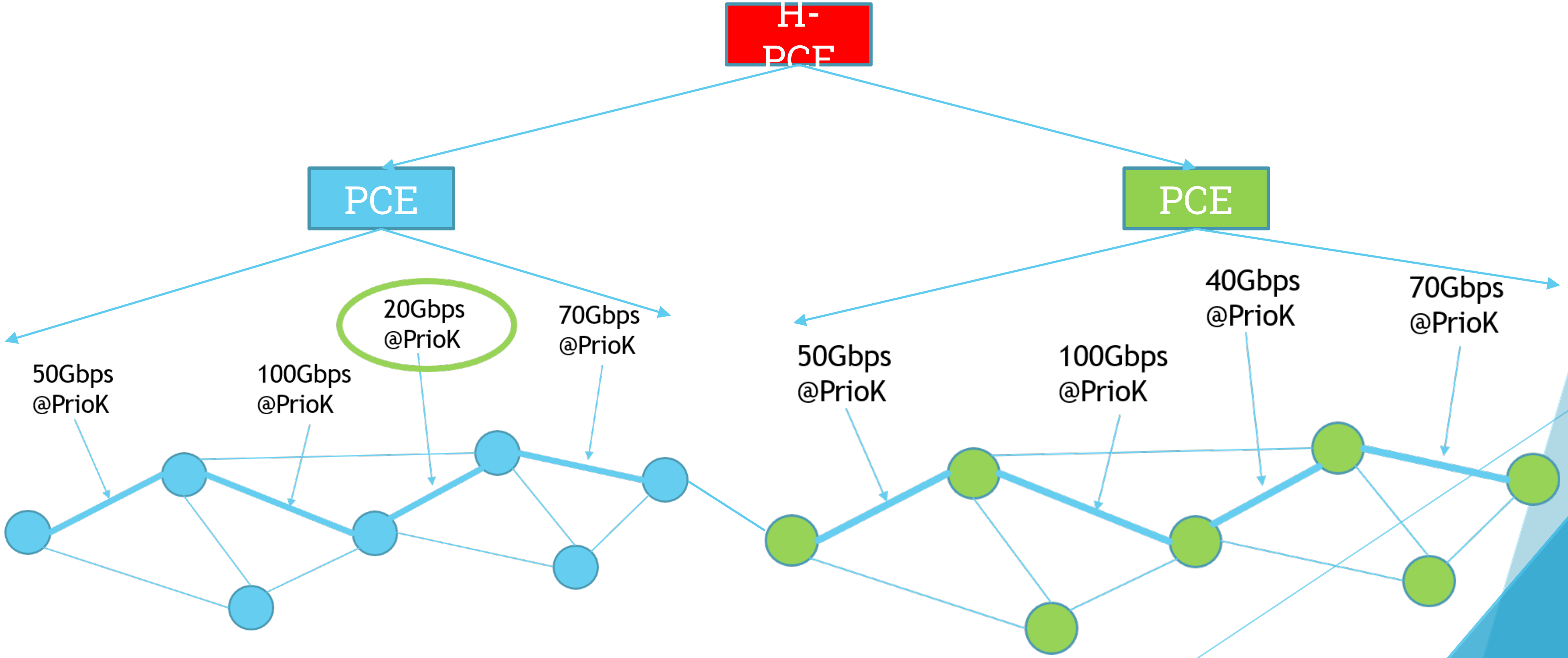
- 10x10 network for each domain with 10Gb/s links.
- 3x3 domains and 4x4 domains
- 1000 path computations with LSP deployment (1 Gb/sec bandwidth).
- LSPs removed randomly after deployment with probability 0.2
- Comparison between no-caching, caching without and with residual bandwidth usage. The following figures have been estimated:
  - Number of path computation requests on PCEs per domain and total
  - Estimated e2e path computation time (mean and maximum) given a fixed PCEP return trip time (700 ms)

Size	Total # requests	Mean time (ms)	Max time (ms)
3x3 No-caching	13880	3667	8000
3x3 Caching w/o resbw	4298	1380	7000
3x3 Caching with resbw	3660	1091	5600
4x4 No-caching	35238	30790	73500
4x4 Caching w/o resbw	6100	6350	51100
4x4 Caching with resbw	4002	4464	51100

Hidden

Changes to the protocol

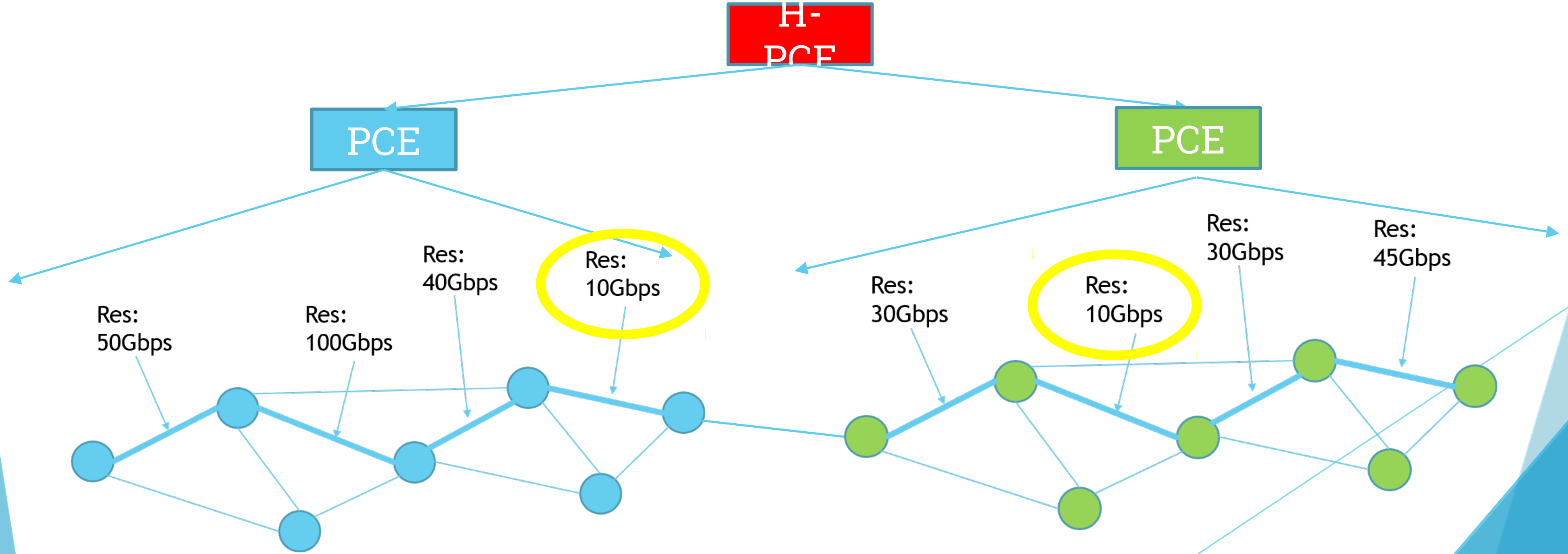
- Two new metric types are added to the METRIC object of RFC5440
  - **The path unreserved bandwidth at a given priority k** *is defined as the minimum value of the unreserved bandwidth at priority k among all the links along a path.*



Hidden

Changes to the protocol

- Two new metric types are added to the METRIC object of RFC5440
  - **The path unreserved bandwidth at a given priority k** *is defined as the minimum value of the unreserved bandwidth at priority k among all the links along a path.*
  - **The path residual bandwidth** *is defined as the minimum value of the residual bandwidth among all the links along the E2E path*



Changes to the protocol

- Both path unreserved and path residual bandwidth have the same format:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Path Unreserved/Residual Bandwidth                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

  - Two new metric types are added to the METRIC object of RFC5440
  - The bandwidth is expressed in bytes/sec and encoded in 32 IEEE floating point format
  - For non PSC requests the field could include (subject to further study) the count of available timeslots or lambdas (as included in the PCReq BANDWIDTH object), keeping the same format.
  - The priority to be used for path unreserved bandwidth is the one mentioned on the LSPA object (if any, otherwise 0).

Next steps

- Simple extension with no dependencies
- Proven improvement in number of iterations and path computation time
- Interest in the WG?