MONSTER

Managing an Operator's Network with Software Defined Networking and Segment Routing

Ing. **Luca Davoli**

luca.davoli@studenti.unipr.it

Tutor: Prof. Ing. Luca Veltri

UNIVERSITÀ DEGLI STUDI DI PARMA
Dipartimento di Ingegneria dell'Informazione

7° Borsisti Day

20/01/2016
Roma – Consortium GARR

Borse di Studio
Orio Carlini
Consortium GARR

Borse di Studio Orio Carlini Consortium GARR

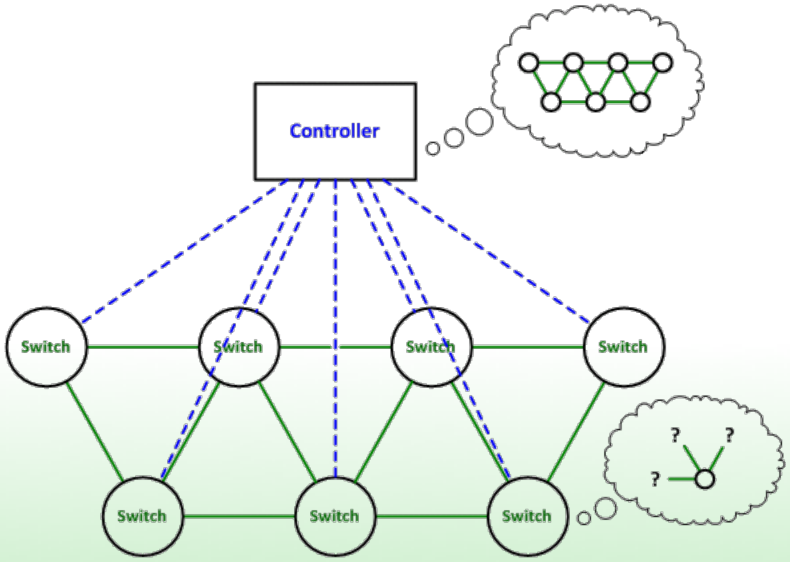
Overview

- Software-Defined Networking (SDN)
- Segment Routing (SR)
- SDN/SR architecture
- SR path selection
- Project extension proposal
- Publications

Software-Defined Networking (SDN)

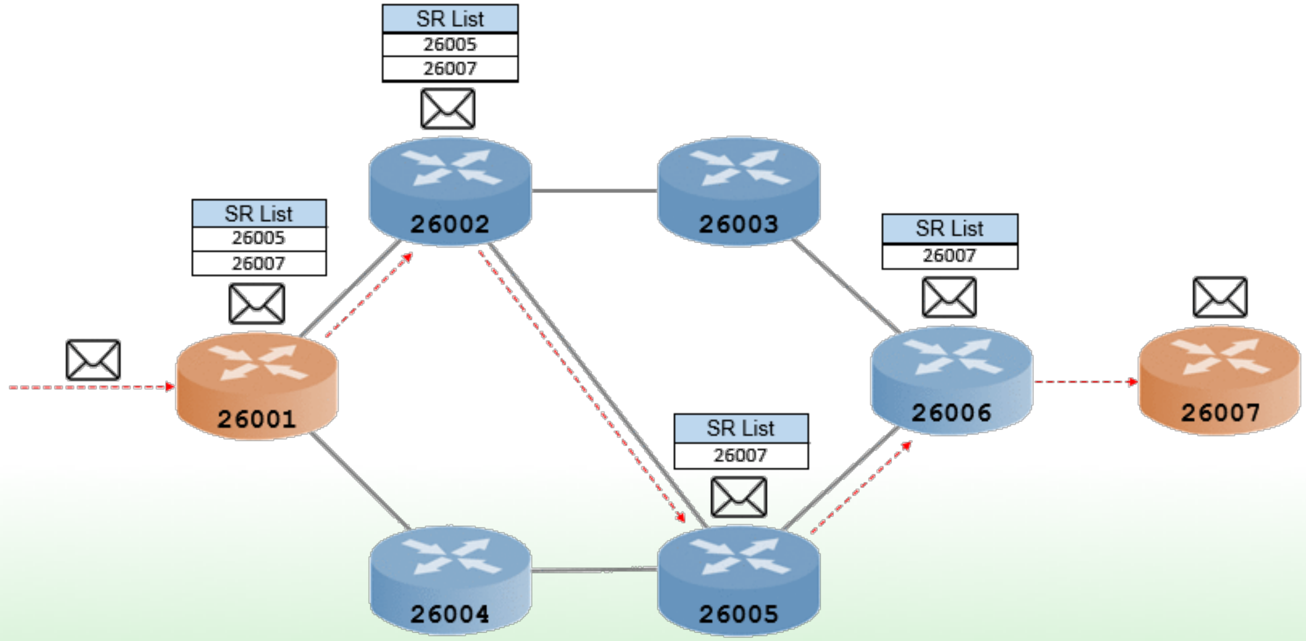
- Allows to decouple control layer (traffic control functions) from the underlying data layer (network hardware)
- Directly programmable, programmatically configured, centrally managed
- Interaction with APIs at both Northbound (NBI) and Southbound (SBI) interfaces

- Overall perspective of the network
- Greater control on network behavior
- Greater simplicity on TE enforcement



Segment Routing (SR)

- Segment Routing (SR): based on source routing
  - Enhanced packet forwarding without any topological restrictions and additional signaling requirements
  - Segments based on MPLS or IPv6

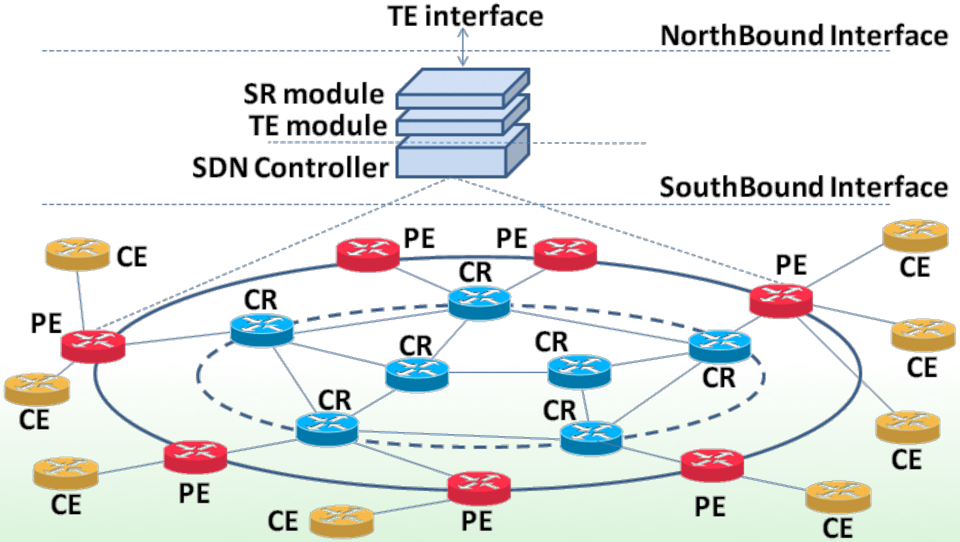


SDN & SR/TE

- ISP network managed by a (logically) centralized SDN controller

- Hybrid IP/MPLS/SDN nodes – OSHI
  - Provider Edge routers (PE)
  - Core Routers (CR)

- MPLS is used for TE
  - no change to the MPLS forwarding plane is required
  - no MPLS control plane has to be used



SDN & SR/TE (cont.)

- SDN controller is requested to allocate a set of traffic flows with a specified bit rate

- *Flow assignment* algorithm is first executed in order to compute the *TE paths*

- For each TE path, the corresponding *SR path* is calculated
  - list of *SIDs* that should be added to incoming packets for instructing them through the assigned TE path

→ Minimization of the number of required *SIDs*

SR path selection

- Traditional approaches for TE: per-flow routing state required within all network nodes

- Our approach: per-flow routing states are set-up only at the border of the network

- Designed and implemented a simple SR heuristic for SR path computation (Ref. [1][2])
- TE paths enforced by using Segment Routing (SR)
- Experimental analysis

- Open-source implementation GitHub

SR path selection (cont.)

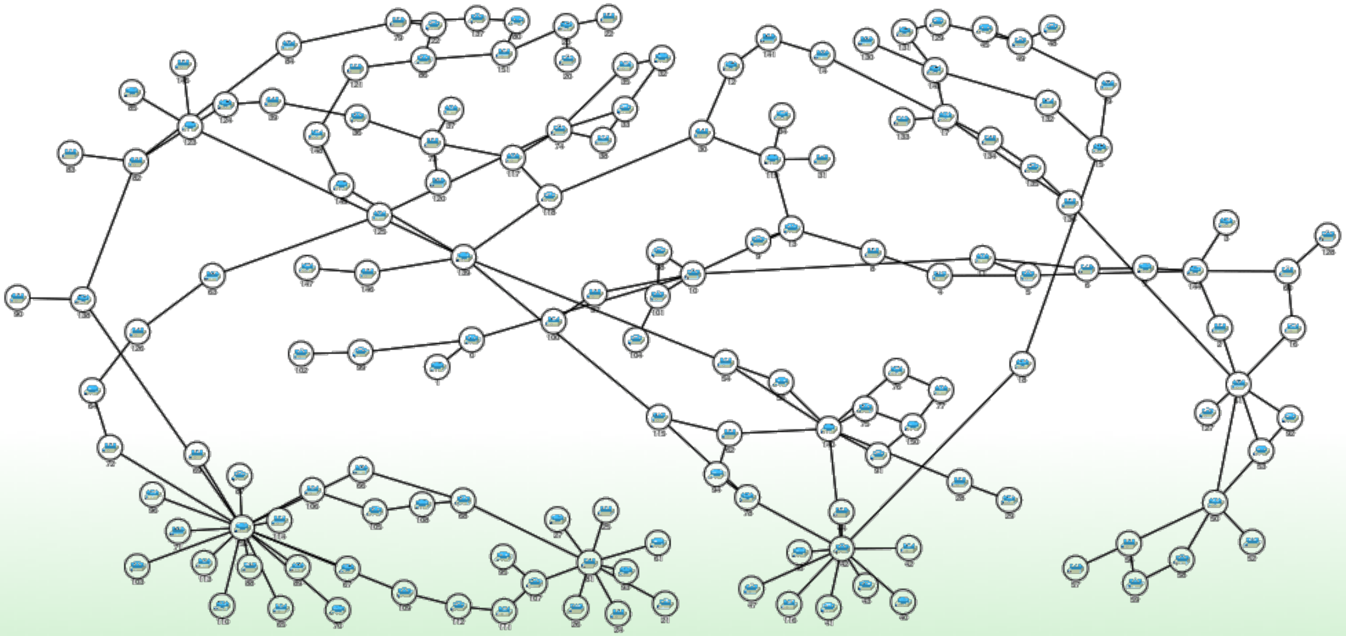
```
procedure INITIALIZATION
n_s = R_0; n_d = R_N; srp = {};

procedure SEGMENTALLOCATION
BEGIN
  p = tep(n_s, n_d);
  if ((SPN(n_s, n_d) == 1) AND (sp(n_s, n_d) == p)) then
    ADD n_d to srp;
    if (n_d == R_N) then
      goto FINISH;
    else
      n_s = n_d;
      n_d = R_N;
      goto BEGIN;
  else
    n_d = prec(p, n_d);
    goto BEGIN;
FINISH
```

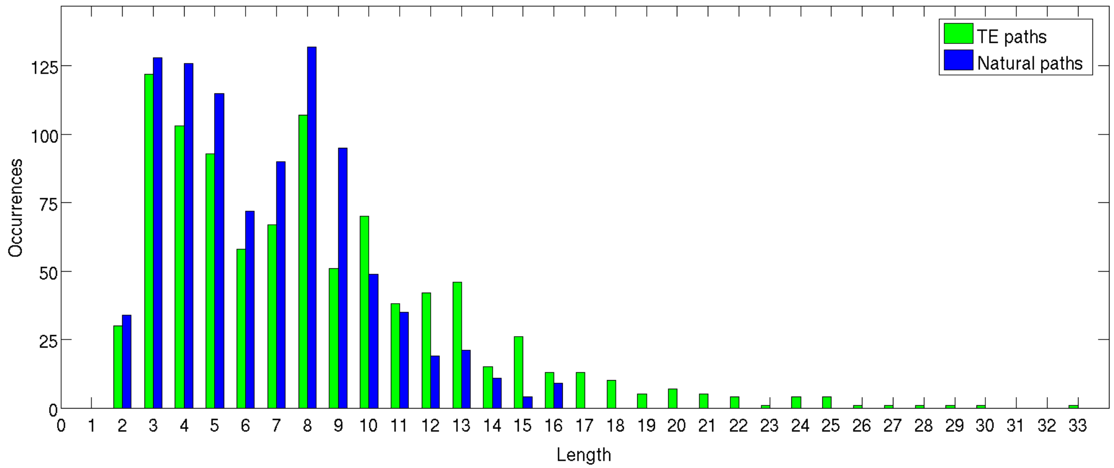
- $tep = tep(n_s, n_d)$
  - TE path that has to be set-up between n_s and n_d
- $tep(n_1, n_2)$
  - portion of the TE path starting from node n_1 and ending with node n_2
- $SPN(n_1, n_2)$
  - number of equal-cost shortest paths from n_1 to n_2, based on the routing tables set-up using a SPF algorithm (e.g., OSPF)
- $sp(n_1, n_2)$
  - if $SPN(n_1, n_2) \equiv 1$, it is the shortest path between nodes n_1 and n_2, otherwise it is not defined
- $prec(p, n)$
  - preceding node of node n along a path p
- srp
  - SR path containing the list of assigned *SIDs*

Experimental analysis

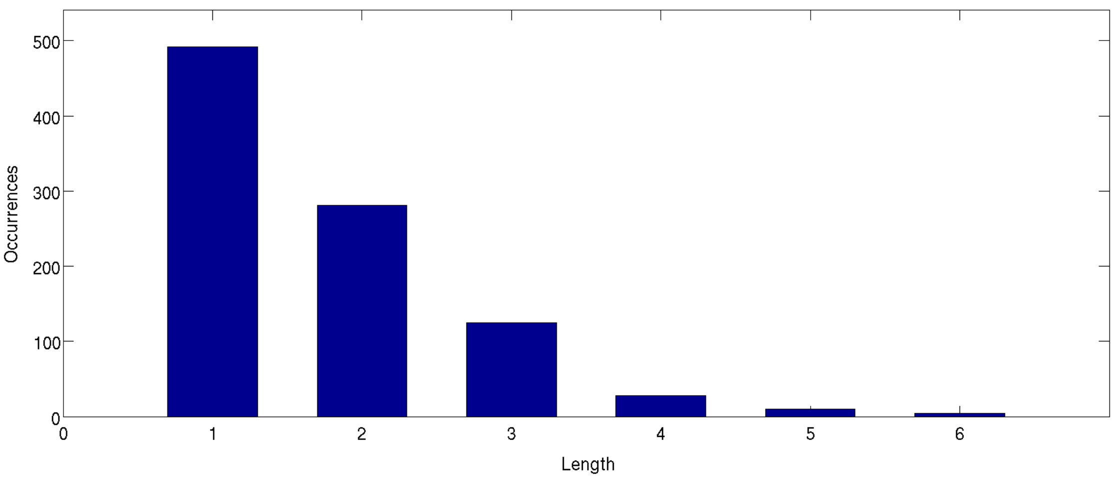
- Relatively large scale topology: 153 nodes, 354 links
- Total generated flows: 2460
- Admitted flows: 940 (according to the *Flow Assignment* algorithm)



Experimental analysis (cont.)

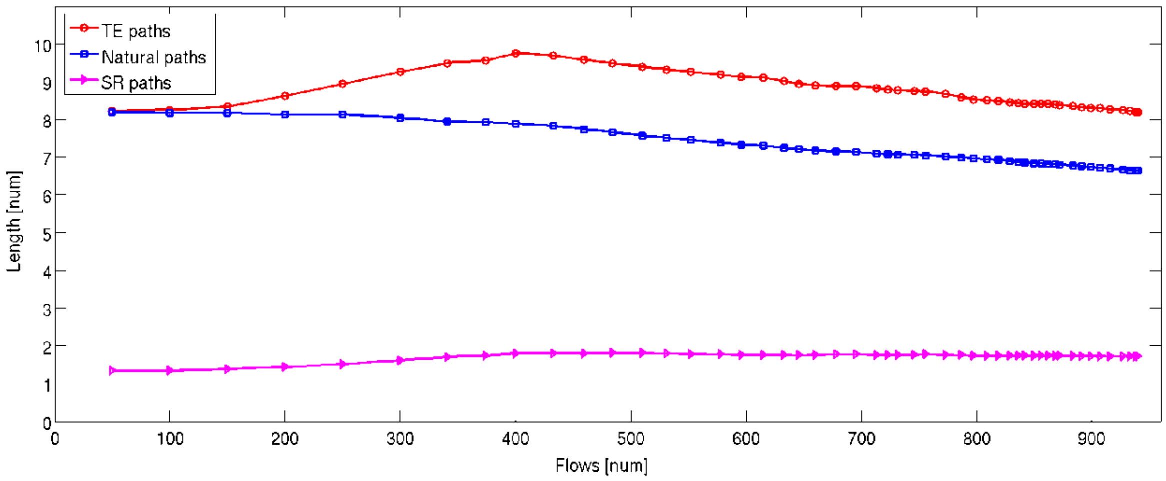


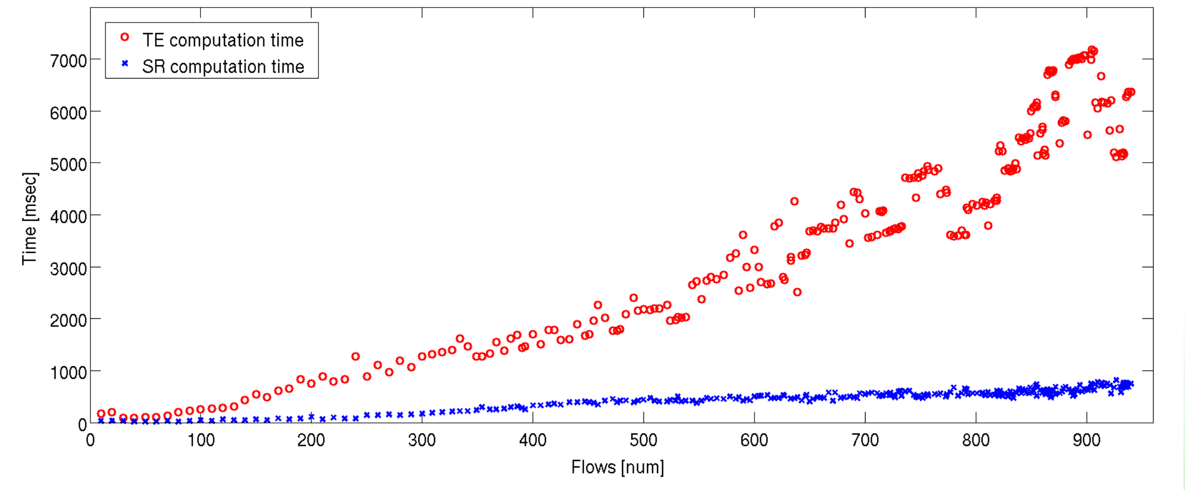
- Distribution of path lengths for:
  - the TE paths
  - the *natural paths* (i.e., the shortest path from ingress PE router to the egress PE router)



- Distribution of SR path lengths, resulting from the *SR assignment* algorithm

Experimental analysis (cont.)





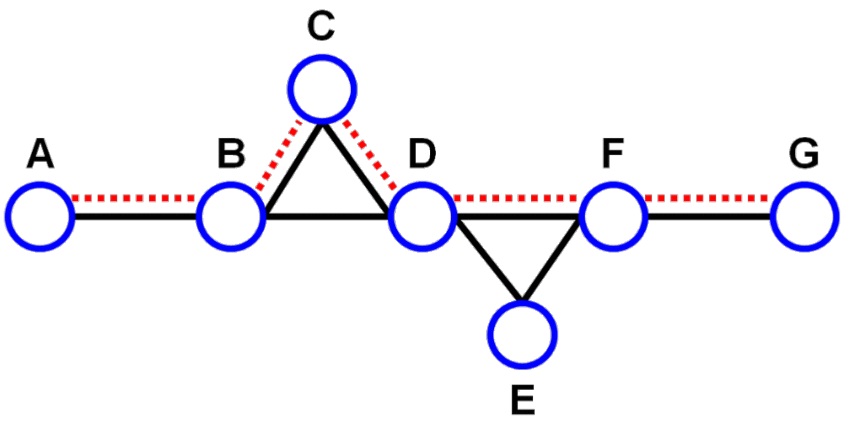
- Mean length, for different number of allocated flows, of:
  - TE paths
  - Natural paths
  - SR paths
- Processing time of the TE/SR algorithms
  - SR assignment algorithm time negligible with respect to the flow assignment heuristic time
  - Less than 8 sec

Borse di Studio Orio Carlini Consortium GARR

SR path selection

- In general a SR path may require both global and local SIDs

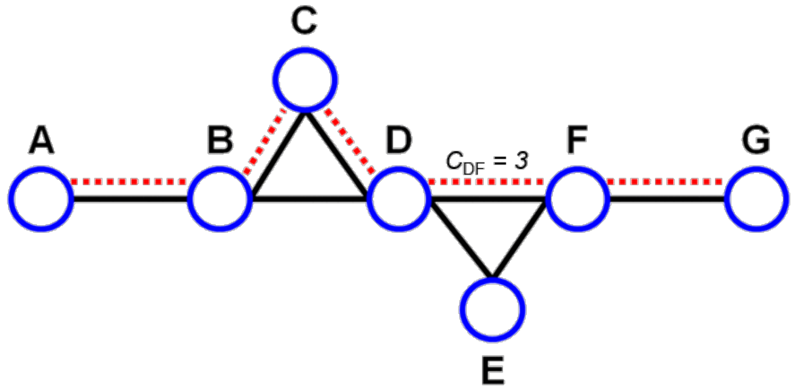
- Equal links cost
- $cost_{link} = K$ → global SIDs
  - $path_{TE}$ = { *A*, *B*, *C*, *D*, *F*, *G* }
  - $path_{NA}$ = { *A*, *B*, *D*, *F*, *G* }
  - SID_{list} = { *C*, *G* }



- Issue: It requires the distribution of global SIDs
- → global SIDs deterministically derived from node IDs

SR path selection (cont.)

- Different links cost
- $cost_{e(D,F)} = 3$ → local or global Adj-SIDs required (***XY***)
  - $path_{TE} = \{ A, B, C, D, F, G \}$
  - $path_{NA} = \{ A, B, D, E, F, G \}$
  - $SID_{list} = \{ C, D,$ ***DF***$, G \}$



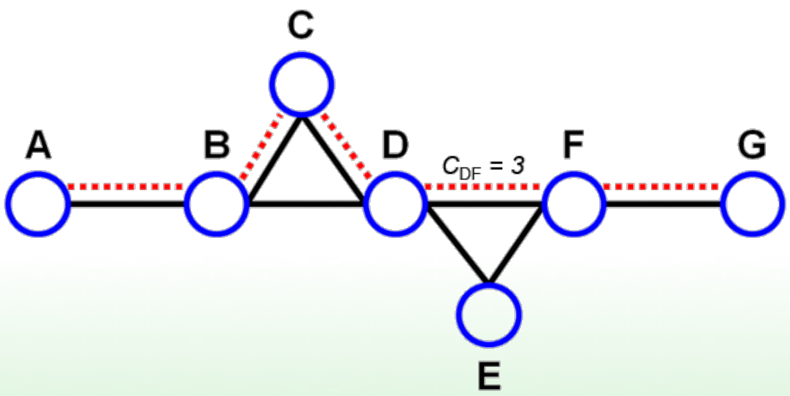
- Issue: increased number of segments needed
- Issue: distribution of SIDs issue not totally solved

SR path selection (cont.)

- Distribution of SIDs implies significant extensions to:
  - the routing protocols
  - the routing daemons running into the routers
- Try to use <u>only</u> global segments whose SIDs can be automatically generated in a distributed fashion, with:
  - no need of explicit advertising
  - no extensions to routing protocols
- *Direct-Link Segment* (Ref. [3][4])
  - Identifies a target destination node to be reached (similar to a Node segment)
  - Direct link toward the destination node: it forces the node to use the direct link rather than the shortest path dictated by the routing protocol
  - No direct link toward the target node: same behavior of a Node segment

SR path selection (cont.)

- *DL-SID = direct-link* SID
- New class of SIDs with global significance and corresponding to the *direct-link* segments
- It needs to identify the target node, like a Node-SID, and to carry further information that identifies it as *direct-link* SID
- $cost_{e(D,F)} = 3 \rightarrow DL\text{-}SID(F) \equiv (\mathbf{F^*})$
  - $path_{TE} = \{ A, B, C, D, F, G \}$
  - $path_{NA} = \{ A, B, D, E, F, G \}$
  - $SID_{list} = \{ C, \mathbf{F^*}, G \}$



SR path selection (cont.)

- Limitation: it does not allow the handling of multiple parallel links between two routers at L3 (i.e., with different IP addresses)
  - If present, they must be handled at L2 and seen at IP level as a single link
  - Multiple parallel links bonded at L2 is a typical solution for operators → may be not a critical limitation

	Traditional local Adj-SID-based Segment Routing	***DL-SID*-based Segment Routing**
Need to configure nodes with SIDs	Yes, $O(\eta)$ nodes	No
Need to advertise SIDs	Yes	No
Routing state	$O(\eta)$	$O(2\eta)$
SR path length for a path of λ *links*	$L_{Adj\text{-}SID} \leq 2\lambda$	$L_{DL\text{-}SID} \leq L_{Adj\text{-}SID}$

SR path selection (cont.)

Algorithm 1 Pseudo-code of SR path assignment for traditional SR.

```
function T_SRP: (TEP(S, D)) → SRP
    x = s; y = d; srp = {};

    START:
    p = tep(x, y);
    //check if the sub-path p is the only shortest path
    if ((SPN(x, y) == 1) AND (sp(x, y) == p)) then
        ADD y srp;
        goto ADDED:
    else
        //check if the sub-path p is just one link
        if (prec(p, y) == x) then
            ADD Adj-SID of e(x, y) to srp;
            goto ADDED:
        else
            //no segment added, try with a shorter path
            //(from x to the node that precedes y)
            y = prec(p, y);
            goto START:
    ADDED:
    if (y ≠ d) then
        //consider the remaining part of the path
        x = y;
        y = d;
        goto START:
    return srp;
```

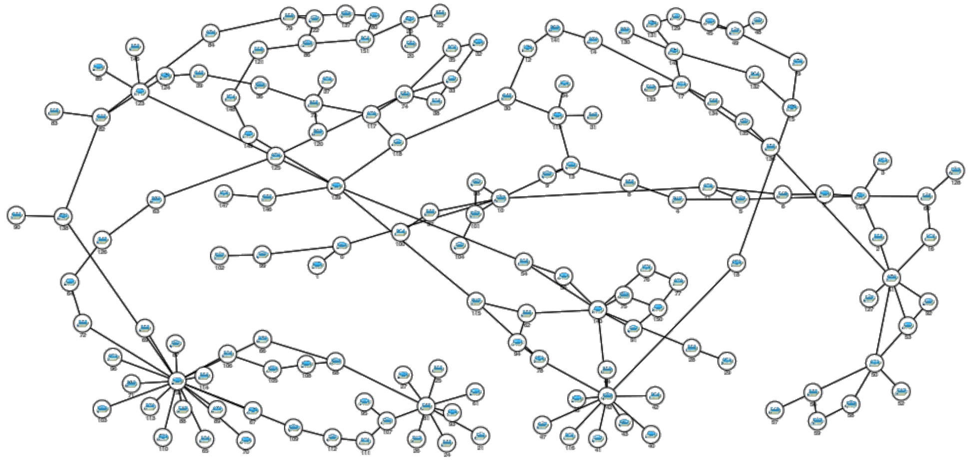
Algorithm 2 Replacement of adjacency SIDs with direct-link SID.

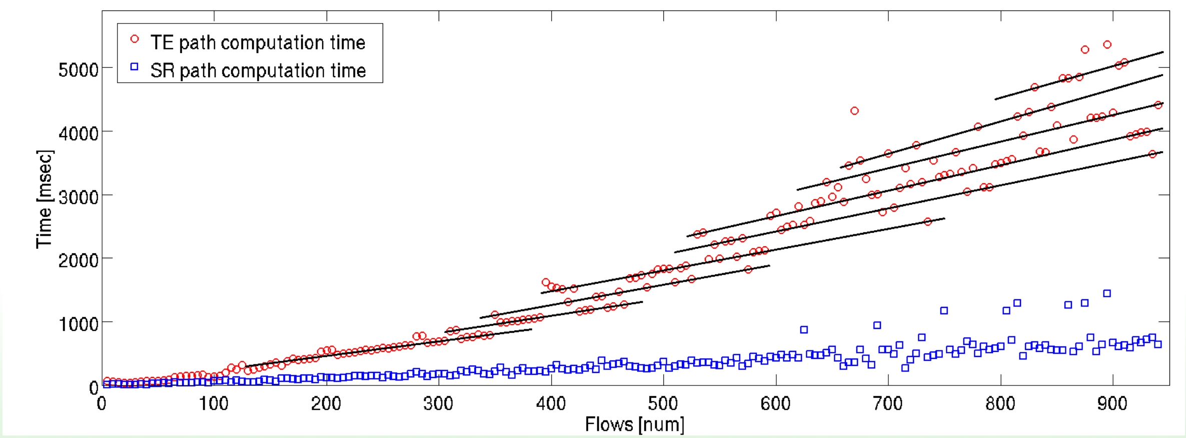
```
function DL_SRP: SRP → DLSRP
    dlsrp = {};

    for (i = 0; i < srp.length; i++) do
        if (srp[i] is an Adj-SID) then
            d = destination of srp[i];
            ADD d* to dlsrp;
        else
            if (srp[i + 1] is not an Adj-SID) then
                ADD spr[i] to dlsrp;
            else
                if (SPN*(srp[i − 1], srp[i + 1]) > 1 OR
                    sp*(srp[i − 1], srp[i + 1]*) ≠ tep(srp[i − 1], srp[i + 1])) then
                    ADD srp[i] to dlsrp;
    return dlsrp;
```

- *sp*(x, y*)*
  - the set of *direct-links biased* shortest paths from *x* to *y*; a *direct-links biased* shortest path is built heading from *x* to *y* on a shortest path, unless there is a *direct link* from an intermediate node to *y*, which is always followed
- *SPN*(x, y*)*
  - number of *direct-links biased* shortest paths *sp*(x, y*)*

Experimental analysis





- Processing time of the TE/SR algorithms
  - SR assignment algorithm time negligible with respect to the flow assignment heuristic time
  - Step-wise dependence on the number of iterations of the heuristic optimization cycle (parallel lines correspond to a given number of cycles)
  - Adequate for periodic (e.g., nightly) reallocation procedures

Project extension proposal

- SDN/SR architecture experimental validation based on IPv6 network
  - Alternative to MPLS adoption
  - IPv6 Segment Routing extension (SRH)
  - Linux-based IPv6 SR implementation
  - SR and SDN/SR practical experimentation and validation

Borse di Studio Orio Carlini Consortium GARR

Project extension proposal (cont.)

- Integration of SDN/SR with NFV
  - Network elements on COTS assets
  - Network Functions (NFs) as Virtual NF (VNFs)
  - NFs managed with SDN approach
  - Usage of SR for implementing Network Function Chaining

Publications

[1] **L. Davoli**, L. Veltri, P. L. Ventre, G. Siracusano, and S. Salsano, “Traffic Engineering with Segment Routing: SDN-based Architectural Design and Open Source Implementation,” IEEE *European Workshop on Software Defined Networks (EWSDN), Poster Session*, pp. 111-112. Sep. 30-Oct. 2, 2015, Bilbao. DOI:10.1109/EWSDN.2015.73

[2] **L. Davoli**, L. Veltri, P. L. Ventre, G. Siracusano, and S. Salsano, “Traffic Engineering with Segment Routing: SDN-based Architectural Design and Open Source Implementation,” *arXiv CoRR,* vol. abs/1506.05941, 2015. arXiv:1506.05941

[3] S. Salsano, L. Veltri, **L. Davoli**, P. L. Ventre, and G. Siracusano, "PMSR - Poor Man’s Segment Routing, a minimalistic approach to Segment Routing and a Traffic Engineering use case“, *IEEE/IFIP Network Operations and Management Symposium (NOMS 2016)*, April 2016. (**ACCEPTED**)

[4] S. Salsano, L. Veltri, **L. Davoli**, P. L. Ventre, and G. Siracusano, "PMSR - Poor Man’s Segment Routing, a minimalistic approach to Segment Routing and a Traffic Engineering use case“, *arXiv CoRR,* vol. abs/1512.05281, 2015. arXiv:1512.05281

Borse di Studio Orio Carlini Consortium GARR

MONSTER

Managing an **O**perator's **N**etwork with **S**oftware Defined Ne**t**working and S**e**gment **R**outing

Thanks for your attention!