Service Chaining in Multi-Layer Networks using Segment Routing and Extended BGP FlowSpec

F. Paolucci[1], **A. Giorgetti**[1], **F.Cugini**[2], **P. Castoldi**[1]
1: Scuola Superiore Sant'Anna, via Moruzzi 1, Pisa, Italy
2: CNIT, via Moruzzi 1, Pisa, Italy
e-mail: fr.paolucci@sssup.it

Abstract: Effective service chaining enforcement along TE paths is proposed using Segment Routing and extended BGP Flowspec for micro-flows mapping. The proposed solution is experimentally evaluated with a deep packet inspection service supporting dynamic flow enforcement.

OCIS codes: (060.0060) Fiber optics and optical communications; (060.4250) Networks;

1. Introduction

In next-generation metro and core networks, operators will be required to transport different application traffic, from/to specific client networks (e.g., data centers, 5G fronthaul clusters, content delivery networks - CDN) where each application may generate an huge number of low or medium bitrate flows (i.e., MicroFlows) subject to different end-to-end service requirements. Moreover, specific MicroFlows will also be required to traverse single or combined network functions (e.g., firewall, deep packet inspection, policers) before reaching the destination. The dynamic enforcement of such functions to selected flows, called Service Chaining (SC), is a hot topic[1]. However, efficient SC deployment is nowadays limited by traditional MPLS networks. Indeed, dynamic enforcement of service chaining is limited by the Label Switched Paths (LSP) granularity thus preventing specific per-flow service differentiation. MicroFlows originated in the access/aggregation network segment are not mapped onto core LSPs in a 1:1 match for scalability reasons (i.e., per-MicroFlow signaling is not realistic). Therefore, the selection of one or more existing LSPs at the edge of the core network is typically required when a new MicroFlow has to be served. Typically, this steering operation is currently based on local and manual policies (e.g., enforced by means of command line interface at the edge network elements). Flow specification extensions for Border Gateway Protocol (BGP FlowSpec) have been proposed in order to match a flow based on packet attributes and force it to specific actions, mainly addressing security issues[2].

In this work, for the first time, BGP Flowspec is extended to enable MicroFlow service chaining.

In particular, a traffic engineering (TE) control solution is proposed in the framework of the Stateful PCE architecture exploiting network information databases. The proposed solution exploits Segment Routing (SR)-based multi-layer networks[3]. SR is an emerging TE technique proposed by IETF that significantly simplifies the control plane operation and natively support SC enforcement without service node configuration. A SC deep packet inspection service node is implemented and validated in the testbed, along with the BGP FlowSpec steering procedure at the edge node.

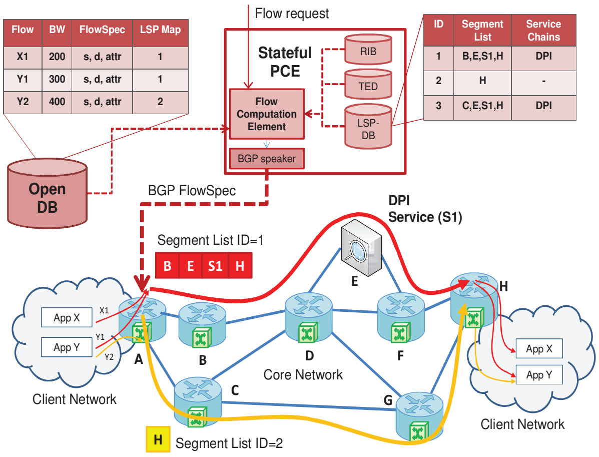
2. Extended BGP FlowSpec solution for Service Chaining in Segment Routing

The proposed network solution enabling effective SC and flow mapping enforcement is shown in Fig. 1. A core network connects different client networks. Several network functions can be available in the core network, e.g. deep packet inspection (service S1) is provided by node E in Fig. 1. Several applications run in the attached client networks generating traffic (Flow) requests. The Stateful PCE, is extended with an internal *Flow Computation Element* (FCE) module, processing the received Flow requests.

In the use-case of Fig.1 the network domain is based on Segment Routing (SR). In this case, TE does not require the utilization of signaling messages[3]. Specifically, a stack of MPLS labels (i.e., the segment list) is applied to each packet to enforce explicit routing of each Flow. Moreover, SR enables effective SC associating each service to a special MPLS label (i.e., service label) to be included in the segment list. This way, intermediate nodes (e.g., node E in Fig.1) do not require specific configurations for SC. In this scenario, the LSP-DB at the PCE stores the set of segment list (the SR-LSPs) currently used in the network. Upon flow computation the PCE communicates to the edge node the ID of the segment list to be enforced to the requested Flow.

When a new flow request is submitted, the FCE module performs flow computation, i.e., it identifies one or more established SR-LSPs over which to steer the Flow. If the new Flow cannot be served using active LSPs, one or more LSPs are properly initiated by PCE using the PCE Protocol with instantiation capabilities[4].

978-1-943580-23-1/17/$31.00 ©2017 Optical Society of America



Flow	BW	FlowSpec	LSP Map
X1	200	s, d, attr	1
Y1	300	s, d, attr	1
Y2	400	s, d, attr	2

ID	Segment List	Service Chains
1	B,E,S1,H	DPI
2	H	-
3	C,E,S1,H	DPI

Fig.1 Architecture

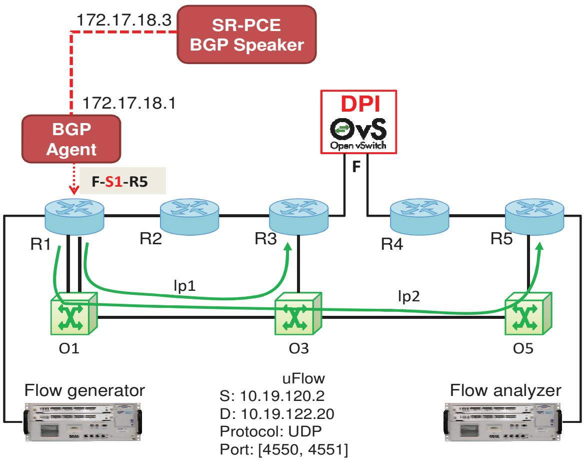


Fig. 2. Experimental testbed

The FCE module resorts to four databases to perform the Flows mapping: 1) the PCE Routing Information Base (RIB), to identify the edge nodes associated to traffic source and destination; 2) the SR-LSP database (LSP-DB), to select the active SR-LSPs that match the Flow constraints, edge points and required network services; 3) the Traffic Engineering Database (TED), to compute the path for possible new SR-LSPs; 4) the Flow database, storing all the installed flows with reserved bandwidth and the related hosting LSPs. Since the number of installed Flows in the core network may be huge, Flow database is proposed to be stored in an external Open Network Database[5]**.**

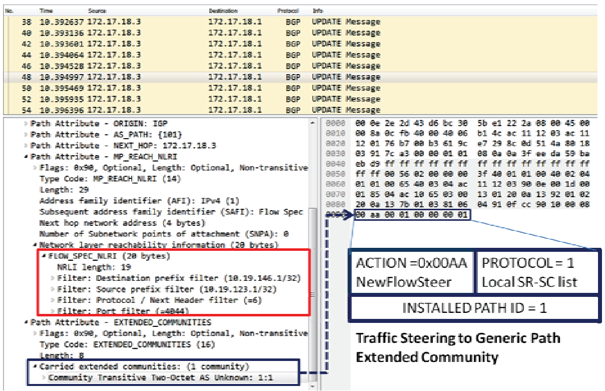
After flow computation, actual Flow steering is realized by means of the BGP protocol with flow specification attributes (i.e., BGP FlowSpec). The main advantage of this technique is that BGP is already widely utilized and available in most core network equipment. A BGP speaker located at the PCE collects the outputs of flow computation and sends one or more UPDATE messages only to the ingress edge node of the selected SR-LSPs. The UPDATE message encloses the information required to univocally identify the Flow, i.e., the FlowSpec match attributes (i.e. , source/destination IP address, IP protocol, layer4 ports, DSCP) and the related actions to be applied to the specific Flow.

To enable the described solution, we propose to extend BGP FlowSpec with three novel actions to create, remove and modify the Flow steering into existing SR-LSPs. Each action is encoded with three fields enclosing: 1) the action type; 2) the LSP ids into which the Flow has to be steered; and 3) the indication of the protocol instantiating the SR-LSP (e.g., PCEP, NETCONF, OpenFlow, MPLS, Segment Routing), allowing the steering independently from the protocol (and the technology) utilized to instantiate the paths.

The described control scheme also enables TE solutions. In Fig. 1, two segment lists (i.e., SR-LSPs) are used between edge nodes A and H, each one allowing 1000 BW units. The segment list B-E-S1-H has ID=1 and specifies the red SR-LSP including the execution of the DPI service (service label S1) at node E. The segment list H has ID=2 and is composed only by the destination node identifier (i.e., the A-H shortest path), thus identifying the yellow SR-LSP. Moreover, three Flows generated by two applications (i.e., app X and Y) are already established, as shown in the OpenDB table, and associated with one of the SR-LSPs. When a new Flow arrives requiring DPI, FCE selects SR-LSP with ID=1. However, if requested bandwidth exceeds 500 BW units, FCE should compute a new segment list, for instance C-E-S1-H with ID=3 enforcing path A-C-D-E-F-H, e.g. to avoid the bottleneck link B-D, and assuring the DPI service at node E. If DPI is not required, segment list with ID=2 is selected.

3. Experimental Demonstration

The proposed SC-enabled architecture have been evaluated in the multi-layer network testbed shown in Fig. 2. The data plane is composed by Juniper routers running OSPF-TE equipped with a set of agents enabling the utilization of SR[3] and Ericsson SPO 1400 ROADMs. A server with two Gigabit Ethernet interfaces is included (i.e., node F in Fig. 2) implementing a simplified DPI and SR functionalities by using OpenvSwitch v2.4[6]. The DPI participates to the OSPF-TE instance announcing node F with DPI service (service label S1). The network is controlled by a stateful SR-PCE extended with a BGP FlowSpec Speaker developed in C++. Edge node R1 is equipped with an agent running PCEP and extended BGP FlowSpec. Node agent elaborates BGP FlowSpec update messages and enforce flow steering by sending interactive CLI-based scripts to edge routers. Traffic flows are injected by a Spirent SPTN4U traffic generator and analyzer connected to edge nodes R1 and R5.



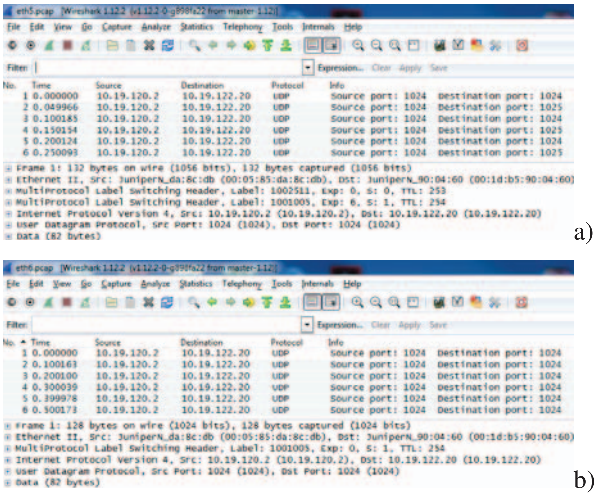
```
No.  Time       Source       Destination  Protocol Info
 38  10.392637  172.17.18.3  172.17.18.1  BGP      UPDATE Message
 40  10.393136  172.17.18.3  172.17.18.1  BGP      UPDATE Message
 42  10.393601  172.17.18.3  172.17.18.1  BGP      UPDATE Message
 44  10.394064  172.17.18.3  172.17.18.1  BGP      UPDATE Message
 46  10.394528  172.17.18.3  172.17.18.1  BGP      UPDATE Message
 48  10.394997  172.17.18.3  172.17.18.1  BGP      UPDATE Message
 50  10.395469  172.17.18.3  172.17.18.1  BGP      UPDATE Message
 52  10.395935  172.17.18.3  172.17.18.1  BGP      UPDATE Message
 54  10.396396  172.17.18.3  172.17.18.1  BGP      UPDATE Message
```

```
▹ Path Attribute - ORIGIN: IGP
▹ Path Attribute - AS_PATH: {101}
▹ Path Attribute - NEXT_HOP: 172.17.18.3
◢ Path Attribute - MP_REACH_NLRI
  ▹ Flags: 0x90, Optional, Length: Optional, Non-transitive
    Type Code: MP_REACH_NLRI (14)
    Length: 29
    Address family identifier (AFI): IPv4 (1)
    Subsequent address family identifier (SAFI): Flow Spec
    Next hop network address (4 bytes)
    Number of Subnetwork points of attachment (SNPA): 0
  ◢ Network layer reachability information (20 bytes)
    ◢ FLOW_SPEC_NLRI (20 bytes)
        NRLI length: 19
      ▹ Filter: Destination prefix filter (10.19.146.1/32)
      ▹ Filter: Source prefix filter (10.19.123.1/32)
      ▹ Filter: Protocol / Next Header filter (=6)
      ▹ Filter: Port filter (=4044)
◢ Path Attribute - EXTENDED_COMMUNITIES
  ▹ Flags: 0x90, Optional, Length: Optional, Non-transitive
    Type Code: EXTENDED_COMMUNITIES (16)
    Length: 8
  ◢ Carried extended communities: (1 community)
    ▹ Community Transitive Two-Octet AS Unknown: 1:1
```

```
0000  00 0e 2e 2d 43 d6 bc 30  5b e1 22 2a 08 00 45 00
0010  00 8a 0c fb 40 00 40 06  b1 4c ac 11 12 03 ac 11
0020  12 01 76 b7 00 b3 61 9c  e7 29 8c 0d 51 4a 80 18
0030  03 91 7c a3 00 00 01 01  08 0a 0a 3f ee da 59 ba
0040  eb d9 ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
0050  ff ff 00 56 02 00 00 00  3f 40 01 01 00 40 02 04
0060  01 01 00 65 40 03 04 ac  11 12 03 90 0e 00 1d 00
0070  01 85 04 ac 10 65 03 00  13 01 20 0a 13 92 01 02
0080  20 0a 13 7b 01 03 81 06  04 91 0f cc 90 10 00 08
      00 aa 00 01 00 00 00 01
```

#Flows	BGP Flowspec	Global Steering
1	144 us	2.43 s
10	4.1 ms	2.71 s
100	45 ms	5.37 s

Fig. 3 BGP Flowspec extensions and steering scalability



eth5.pcap [Wireshark 1.12.2 (v1.12.2-0-g898fa22 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save

```
No. Time      Source       Destination   Protocol Info
  1 0.000000  10.19.120.2  10.19.122.20  UDP      Source port: 1024  Destination port: 1024
  2 0.049966  10.19.120.2  10.19.122.20  UDP      Source port: 1024  Destination port: 1025
  3 0.100185  10.19.120.2  10.19.122.20  UDP      Source port: 1024  Destination port: 1024
  4 0.150154  10.19.120.2  10.19.122.20  UDP      Source port: 1024  Destination port: 1025
  5 0.200124  10.19.120.2  10.19.122.20  UDP      Source port: 1024  Destination port: 1024
  6 0.250093  10.19.120.2  10.19.122.20  UDP      Source port: 1024  Destination port: 1025
⊞ Frame 1: 132 bytes on wire (1056 bits), 132 bytes captured (1056 bits)
⊞ Ethernet II, Src: JuniperN_da:8c:db (00:05:85:da:8c:db), Dst: JuniperN_90:04:60 (00:1d:b5:90:04:60)
⊞ MultiProtocol Label Switching Header, Label: 1002511, Exp: 0, S: 0, TTL: 253
⊞ MultiProtocol Label Switching Header, Label: 1001005, Exp: 6, S: 1, TTL: 254
⊞ Internet Protocol Version 4, Src: 10.19.120.2 (10.19.120.2), Dst: 10.19.122.20 (10.19.122.20)
⊞ User Datagram Protocol, Src Port: 1024 (1024), Dst Port: 1024 (1024)
⊞ Data (82 bytes)
```

a)

eth6.pcap [Wireshark 1.12.2 (v1.12.2-0-g898fa22 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save

```
No. ▲ Time    Source       Destination   Protocol Info
  1 0.000000  10.19.120.2  10.19.122.20  UDP      Source port: 1024  Destination port: 1024
  2 0.100163  10.19.120.2  10.19.122.20  UDP      Source port: 1024  Destination port: 1024
  3 0.200100  10.19.120.2  10.19.122.20  UDP      Source port: 1024  Destination port: 1024
  4 0.300039  10.19.120.2  10.19.122.20  UDP      Source port: 1024  Destination port: 1024
  5 0.399978  10.19.120.2  10.19.122.20  UDP      Source port: 1024  Destination port: 1024
  6 0.500173  10.19.120.2  10.19.122.20  UDP      Source port: 1024  Destination port: 1024
⊞ Frame 1: 128 bytes on wire (1024 bits), 128 bytes captured (1024 bits)
⊞ Ethernet II, Src: JuniperN_da:8c:db (00:05:85:da:8c:db), Dst: JuniperN_90:04:60 (00:1d:b5:90:04:60)
⊞ MultiProtocol Label Switching Header, Label: 1001005, Exp: 0, S: 1, TTL: 254
⊞ Internet Protocol Version 4, Src: 10.19.120.2 (10.19.120.2), Dst: 10.19.122.20 (10.19.122.20)
⊞ User Datagram Protocol, Src Port: 1024 (1024), Dst Port: 1024 (1024)
⊞ Data (82 bytes)
```

b)

Fig. 4 DPI Service captures (a. in, b:out)

Two 10G lightpaths lp1 and lp2 are installed in the optical network, providing optical bypass and 2 TE links at routers OSPF-TE instance interconnecting R1-R3 and R1-R5, respectively. Therefore, flows requiring DPI will be sent to lp1 through the F-S1-R5 segment list (i.e., R1-F shortest path is the two-hops R1-R3-F route through lp1 bypass) without the need of configuring neither intermediate nodes nor the DPI server.

Fig. 3 shows a Wireshark capture collected at the BGP Speaker of the stateful PCE (IP 172.17.18.3). The capture shows the UPDATE messages sent to the agent (172.17.18.1). In particular, one message is expanded showing the FlowSpec NLRI field describing the Flow match filter (i.e., source 10.19.123.1 /32, destination 10.19.146.1 /32, protocol TCP, TCP port 4044). Moreover, the novel Extended Communities Path Attribute is shown, enforcing flow steering action to a generic path. In this case the ACTION field is set to New Flow Steer. The Protocol field is set to Local SR-SC list (i.e., the hosting path id is identified within the set of segment lists including special SC labels installed locally). Finally, the installed path identifier is enclosed. This way, the router identifies the target path besides the origin and the protocol used to instantiate it. Scalability tests against the implemented BGP Flowspec have been performed by sending bulks of different Flow steering messages: results are reported in the table of Fig. 3. The second column shows the time needed to process the BGP Flowspec updates, including extensions. The third column shows the global steering time including router configuration. Results show that the BGP Flowspec processing time is almost linearly increasing with the number of updates (1 update requires less than 150 us, 100 Flows are processed in 45 ms). The total configuration requires up to 5.4s for 100 Flows, mainly due to the router commit time. Fig. 4 reports the capture of packets entering (Fig. 4a) and exiting (Fig. 4b) the DPI service provided by node F. Two SC Flows are steered in the network and arrive at the service with the same MPLS stack composed of two labels (i.e., service label 1002511, and R5 label 1001005). The first flow is directed to UDP port 1024, the second to UDP port 1025. After DPI inspection of the packets content, the server decides to drop the second Flow. Thus only packets belonging to the first flow are forwarded after popping the MPLS label 1002511 (label S1) used to require the application of the DPI service.

4. Conclusions

For the first time, service chaining was enabled by traffic steering through extended BGP Flowspec. Experimental results on a Segment Routing multi-layer network demonstrate the scalability of the proposal, which does not trigger any control plane configuration in intermediate nodes.

Acknowledgements. This research has been partially supported by the EU H2020-ICT-14 project 5GEx (grant n. 671636).

References

[1] W. John et al., "Research Directions in Network Service Chaining", SDN Future *Network&Services Conf.*, 2013.
[2]. P.Marques at al., RFC 5575, IETF.
[3] A.Sgambelluri, F. Paolucci, A. Giorgetti, F. Cugini, and P. Castoldi., "Experimental Demonstration of Segment Routing," *Journal of Lightwave Technology*, Vol. 34, no. 1, pp. 205-212, 2016.
[4] O. Gonzalez De Dios et al., " Multipartner Demonstration of BGP-LS-Enabled Multidomain EON Control and Instantiation With H-PCE [Invited]", *IEEE/OSA Journal of Optical Communications and Networking (JOCN)*, Vol. 7, n. 12, 2015.
[5] F. Paolucci, F. Cugini, G. Cecchetti, and P. Castoldi, "Open Database for Interconnected Traffic Engineered Multi-Layer Networks," Proc. *OFC 2016*, paper Th4G.5, 2016.
[6] OpenvSwitch, www.openvswitch.org