Segment Routing (SRv6)

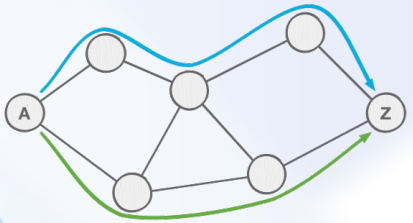
Milad Sharif

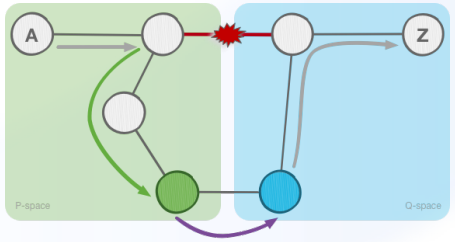
March 2018

Segment Routing — Introduction

- Source routing paradigm
  - Steers a packet through a ordered list of instructions, called *segments*.
  - Each instruction represents a specific function at a specific location in the network.
  - A function is locally defined on the node where it is executed

- Network programming beyond mere packet routing
  - Traffic engineering
  - Load Balancing
  - Fast reroute
  - Network function virtualization





Source: http://www.segment-routing.net/

Segment Routing — Forwarding Plane

- MPLS
  - MPLS data-plane without any changes
  - Segments are represented as MPLS labels (20-bit)

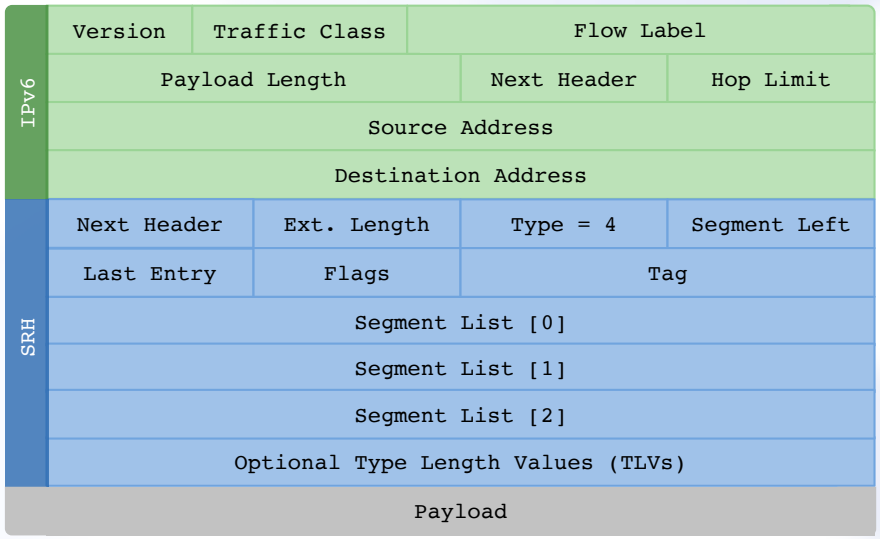
- IPv6
  - Using segment routing extension header (SRH)
  - Ordered list of segments (128-bit) ⟶ scalable
  - User-defined functions

SRv6 — Segments

- Each Segment identifier (SID) is an 128-bit IPv6 Address
- Each SID is represented by LOC:FUNCT:ARGS
  - Locator — route the segments to its parent node
  - Function — opaque identifier of a local action at the parent node
  - Argument (optional) — local function argument
- Specific SID formatting only needs to be understood by the parent node
- A local SID of a node can be an IPv6 address associated to a local interface

SRv6 — Routing Extension Header

- A new type of the Routing Header
  - Segments Left: contains the index of the next segment to inspect
  - Last Entry: contains the index of the last element of the Segment List
  - Tag: tag a packet as part of a class or group of packets
  - Segment List: The SID list is encoded in reverse order
  - TLVs can be used to convey contextual information HMAC, Metadata, OAM values (e.g. INT)



SRv6 — Local SID table

- Contains all the SRv6 segments instantiated at the parent node
  - Define local SIDs that are explicitly instantiated
  - Specify which instruction is bound to each of the instantiated SIDs
  - Store the parameters associated with such instruction (i.e. OIF, NextHop, ...)
- By default it is NOT populated with all the addresses of local interfaces
- An operator MUST explicitly instantiate an SRv6 SID
- SIDs (and corresponding endpoint functionalities) are expected to be signaled in BGP

  https://tools.ietf.org/html/draft-dawra-idr-bgpls-srv6-ext-00

SRv6 — Nodes

- Source node
  - A host originating an IPv6 packet or SR domain ingress router encapsulating a received IPv6 packet
  - DA of the packet is set with the value of the first segment
  - Segments Left field is set to n-1 where n is the number of elements in the Segment List
- Transit node
  - Forward the packet
  - Must NOT inspect the underneath routing header
- Endpoint node
  - The node whose local SID table contains an entry for the DA of the packet
  - Executes the function bound to the SID

```
1.  IF SegmentsLeft > 0 THEN
2.      decrement SL
3.      update the IPv6 DA with SRH[SL]
4.      FIB lookup on updated DA
5.      forward accordingly to the matched entry
6.  ELSE
7.      drop the packet
```

SRv6 — Endpoint functionalities

Name	Functions associated with a SID
END	Endpoint
END.X	Endpoint with Layer-3 cross-connect
END.T	Endpoint with specific IPv6 table lookup
END.DX2	Endpoint with decapsulation and Layer-2 cross-connect
END.DX2V	Endpoint with decapsulation and VLAN L2 table lookup
END.DT2U	Endpoint with decapsulation and unicast MAC L2 table lookup
END.DT2M	Endpoint with decapsulation and L2 table flooding
END.DX6	Endpoint with decapsulation and IPv6 cross-connect
END.DX4	Endpoint with decapsulation and IPv4 cross- connect
END.DX2	Endpoint with decapsulation and Layer-2 cross-connect
End.DT6	Endpoint with decapsulation and specific IPv6 table lookup
End.DT4	Endpoint with decapsulation and specific IPv4 table lookup
End.DT46	Endpoint with decapsulation and specific IP table lookup

Name	Functions associated with a SID
End.B6	Endpoint bound to an SRv6 policy
End.B6.Encaps	Endpoint bound to an SRv6 encapsulation policy
End.B6.Encaps.Red	Endpoint bound to an SRv6 reduced encapsulation policy
End.BM	Endpoint bound to an SR-MPLS policy
End.S	Endpoint in search of a target in table T
T	Transit
T.Insert	Transit with insertion of an SRv6 Policy
T.Insert.Red	Transit with reduced insertion of an SRv6 Policy
T.Ecaps	Transit with encapsulation in an SRv6 Policy
T.Encaps.Red	Transit with reduce encaps in an SRv6 Policy
T.Encaps.L2	Transit with encapsulation of L2 frames
T.Encaps.L2.Red	Transit with reduce encaps of L2 frames in an SRv6 Policy

https://tools.ietf.org/html/draft-filsfils-spring-srv6-network-programming-04

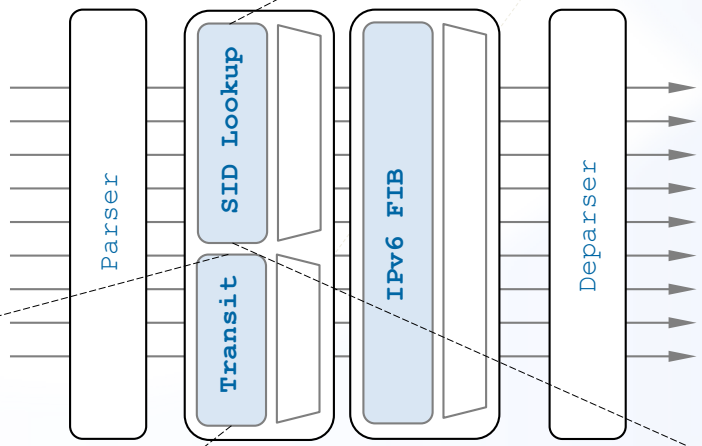
P4 data-plane

- Based on SRv6 Network Programming-v1

  https://tools.ietf.org/html/draft-filsfils-spring-srv6-network-programming-01

- 10 segments in two SRH with L2/L3 forwarding

- ~400 lines of P4




```
table srv6_transit {
  key = {
    ipv6.dstAddr : lpm;
  }
  actions = {
    t;
    t_insert;
    t_encaps;
  }
}
```

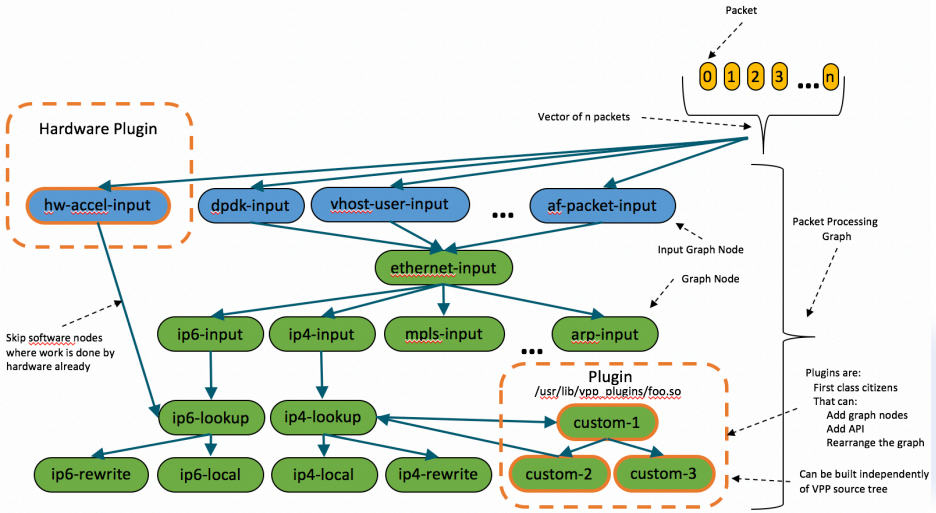
```
table srv6_local_sid {
  key = {
    ipv6.dstAddr : lpm;
    ipv6_srh.valid : ternary;
    ipv6_srh.segLeft : ternary;
    ipv6_srh.nextHdr : ternary;
  }

  actions = {
    drop_;
    transit;
    end;                /* END */
    end_x;              /* END.X */
    end_t;              /* END.T */
    end_dx2;            /* END.DX2 */
    end_dx4;            /* END.DX4 */
    end_dx6;            /* END.DX6 */
    end_dt4;            /* END.DT4 */
    end_dt6;            /* END.DT6 */
    end_b6;             /* END.B6 */
    end_b6_encaps;      /* END.B6.ENCAPS */
  }

  const default_action = transit;
}
```

VPP software data-plane

- Fully Programmable Data Plane Implementation using Directed Graphs (Graph Nodes)
- SRv6 is implemented as part of release 17.04
- VPP as a P4 target
  - PVPP: A Programmable Vector Packet Processor (SOSR '17)

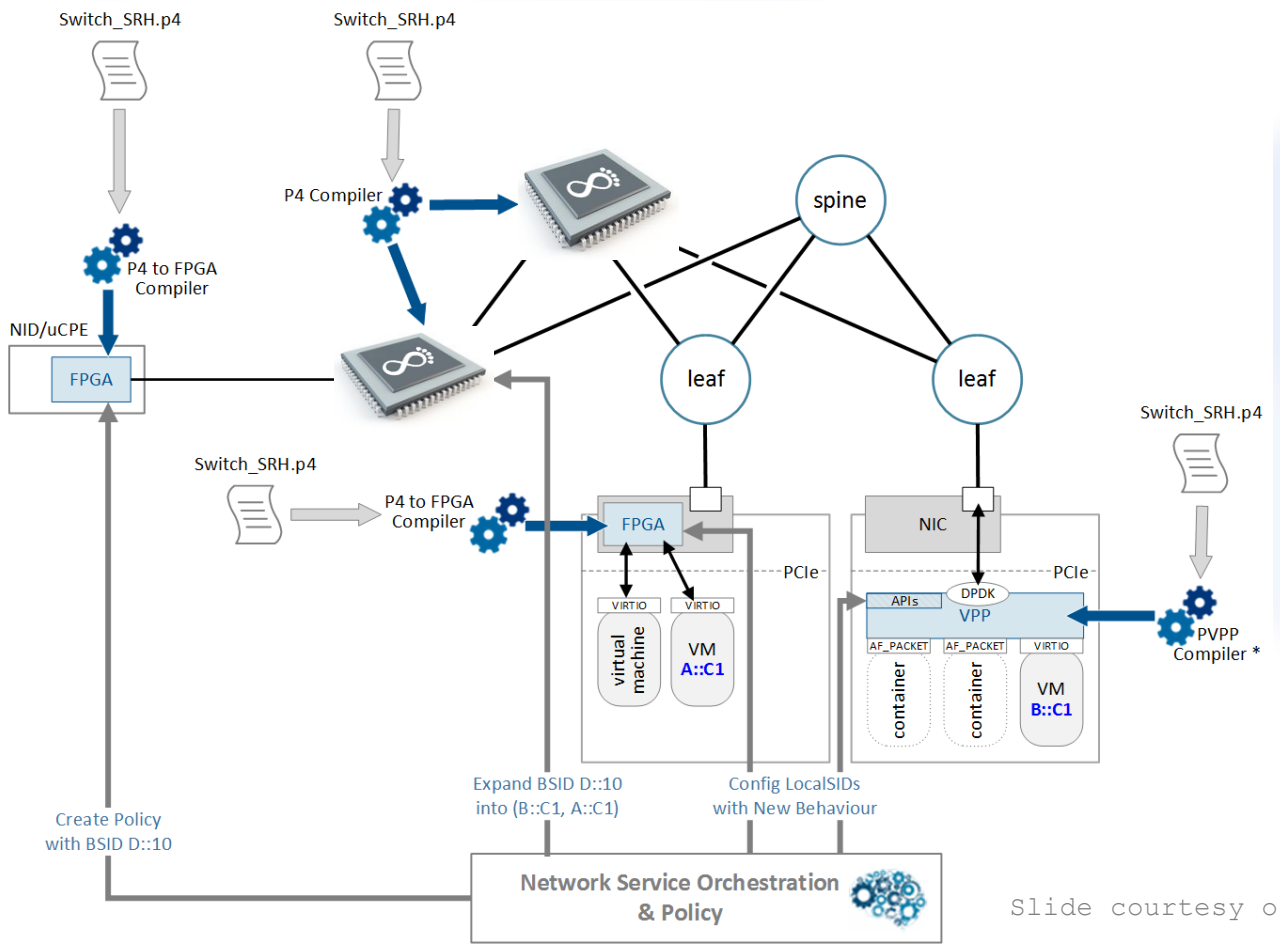
  http://www.cs.princeton.edu/~mshahbaz/papers/sosr17demos-pvpp.pdf
- Deployable in Multiple Scenarios
  - Hypervisor/Host Networking
  - Embedded Systems
  - VNF

fd.io
The Universal Dataplane



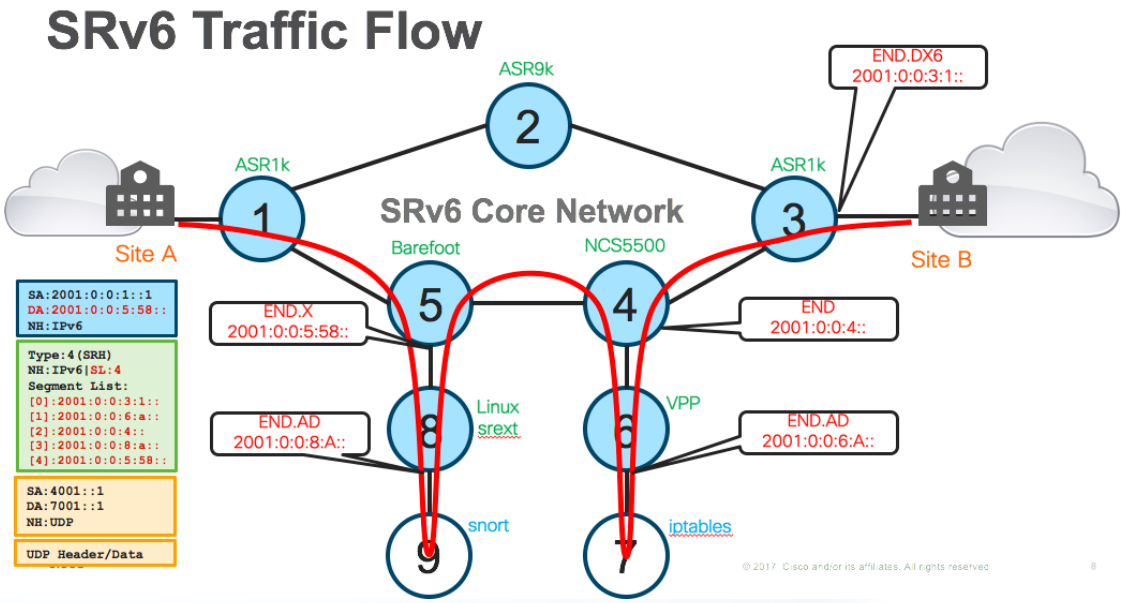
BAREFOOT NETWORKS

Extending the network with a new behavior



Slide courtesy of Daniel Bernier

Sigcomm'17 demo — Interoperability



Questions!!!