6MAN WG

Segment Routing IPv6

Authors and contributors:

Stefano Previdi (sprevidi@cisco.com)

Clarence Filfsils (cfilsfil@cisco.com)

Brian Field (Brian_Field@cable.comcast.com)

John Brzozowski (john_brzozowski@cable.comcast.com)

John Leddy (John_Leddy@cable.comcast.com)

Ida Leung (Ida.Leung@rci.rogers.com)

Roberta Maglione (robmgl@cisco.com)

* Eric Vyncke (evyncke@cisco.com)

Dave Barach (dbarach@cisco.com)

Mark Townsley (townsley@cisco.com)

Chris Martin (martincj@cisco.com)

Nagendra Kumar (naikumar@cisco.com)

David Lebrun (david.lebrun@uclouvain.be)

Pierre Francois (pierre.francois@imdea.org)

James Connolly (jconnolly@libertyglobal.com)

IETF91 – Honolulu, November 2014

Where is Traffic Engineering (TE) ?

- TE requires RSVP to install state in every the core routers
  - => 'low' convergence
  - => TE not widely deployed mainly because of scaling and complexity issues
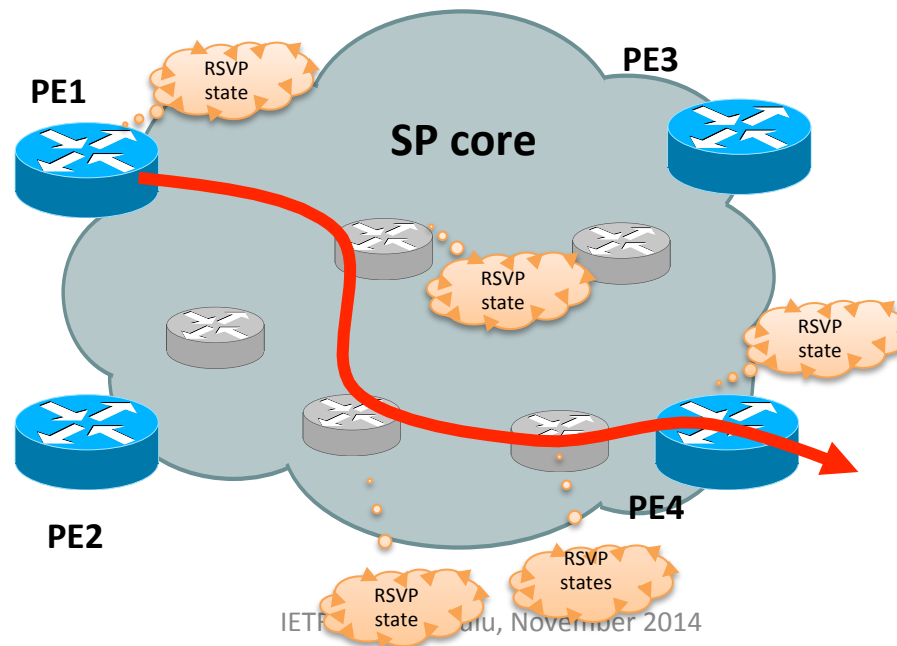
What Can We Do for Efficient/Flexible TE?

- Leverage IPv6 flexibility
  - Overload routing header, i.e. install state in the data packet

- Remove state in the core
  - Push state at the edge or SDN controller
  - Simplification of the core
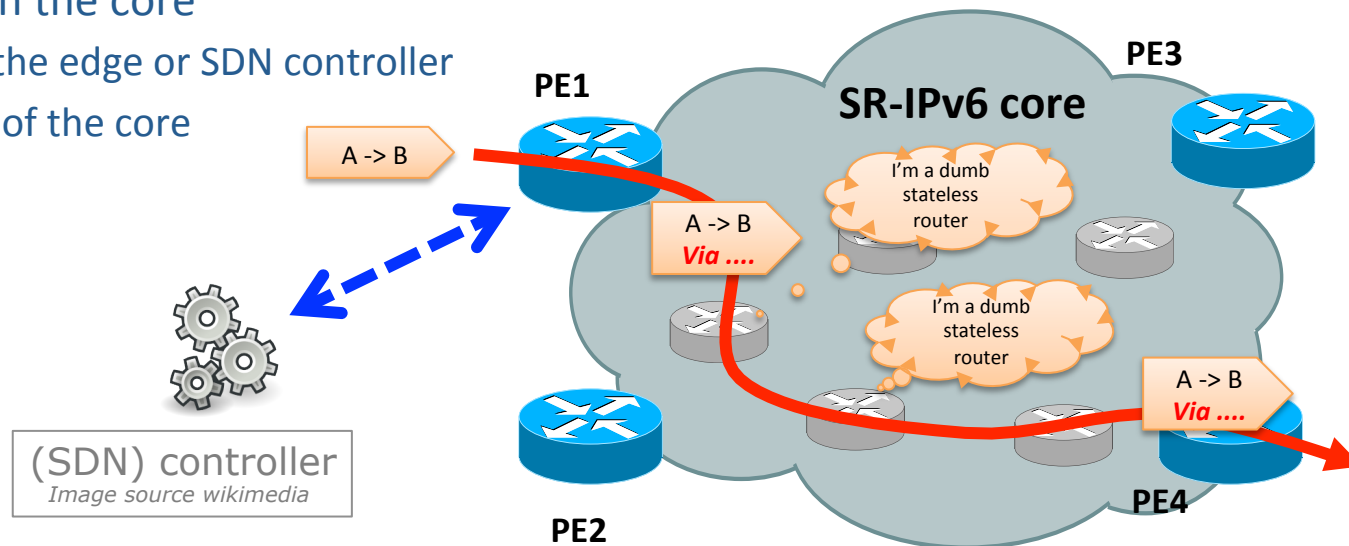
"End-to-End Traffic Engineering" from CPE/Set-up Box?

- What about mobile node away from SP network?
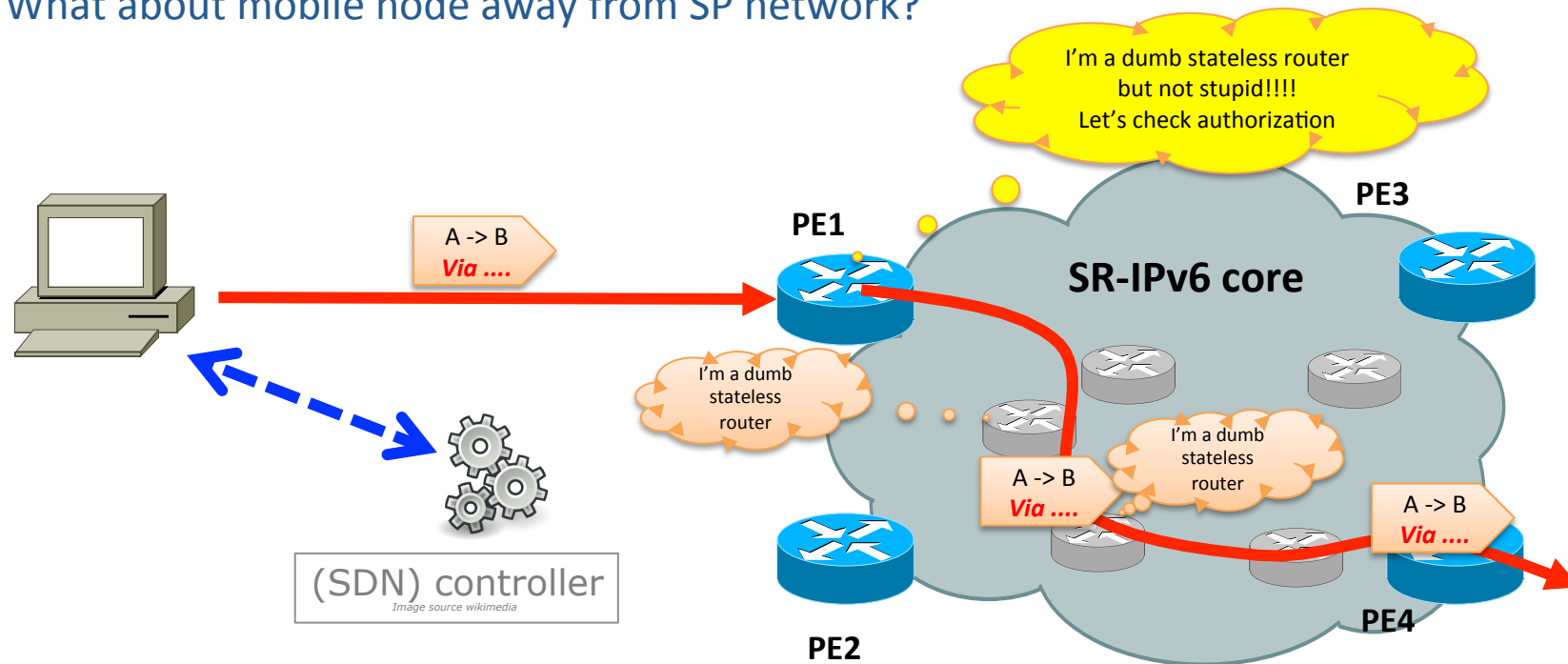
Segment Routing in a Nutshell

- Segment Routing:
    - **Source based routing model** where the source chooses a path and encodes it in the packet header as an ordered list of segments
        - Removes routing states from any node other than the source
    - A segment is an instruction applied to the packet
        - Instruction can be of any type: IGP, BGP, Service, Locator, Egress link, …
    - Segment Routing leverages the source routing architecture defined in RFC2460 for IPv6
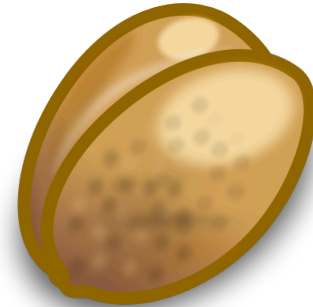
Source: wikimedia

Current SR-IPv6 Drafts

- draft-filsfils-spring-segment-routing
  - Describes the overall Segment Routing architecture

- draft-ietf-spring-ipv6-use-cases
  - describes the SR-IPv6 use cases

- draft-previdi-6man-segment-routing-header
  - describes a new type of the Routing Header (SRH)

- draft-vyncke-6man-segment-routing-security
  - describes the security mechanisms applied to the SRH

Segment Routing Header (SRH)

Segment Routing for IPv6 Dataplane

- A Segment is identified through its IPv6 address
  - No mapping needed between Segment IDs (SIDs) and node's addresses
  - Simplifies signaling, address == SID
- New Routing Extensions Header type
  - Segment Routing Header (SRH)
  - Contains Segment List, Policy List, and a few other bits…

SRH

- **Next Header:** 8-bit selector. Identifies the type of header immediately following the SRH

- **Hdr Ext Len**: 8-bit unsigned integer. Defines the length of the SRH header in 8-octet units, not including the first 8 octets

- **Type**: TBD by IANA (SRH)

- **Next Segment**: index, in the Segment List, of the next active segment in the SRH

- **Last Segment:** index, in the Segment List, of the last
segment of the path

- **Flags:** 16 bits of flags. *See SPRING WG presentation.*

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header   |  Hdr Ext Len  | Routing Type  | Next Segment  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Last Segment  |             Flags             |  HMAC Key ID  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Segment List[0] (128 bits ipv6 address)            |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
                              ...
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Segment List[n] (128 bits ipv6 address)            |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Policy List[0] (128 bits ipv6 address)             |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Policy List[1] (128 bits ipv6 address)             |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Policy List[2] (128 bits ipv6 address)             |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                          HMAC                               //
//                  (256 bits, optional)                       //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

SRH

- Segment List[n]: 128-bit IPv6 addresses representing each segment of the path
- Policy List[n]: See SPRING WG presentation
- HMAC: SRH authentication (optional)

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header   | Hdr Ext Len   | Routing Type  | Next Segment  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Last Segment  |            Flags              |  HMAC Key ID  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Segment List[0] (128 bits ipv6 address)            |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
                              ...
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Segment List[n] (128 bits ipv6 address)            |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Policy List[0] (128 bits ipv6 address)             |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Policy List[1] (128 bits ipv6 address)             |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Policy List[2] (128 bits ipv6 address)             |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                          HMAC                               //
//                  (256 bits, optional)                       //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
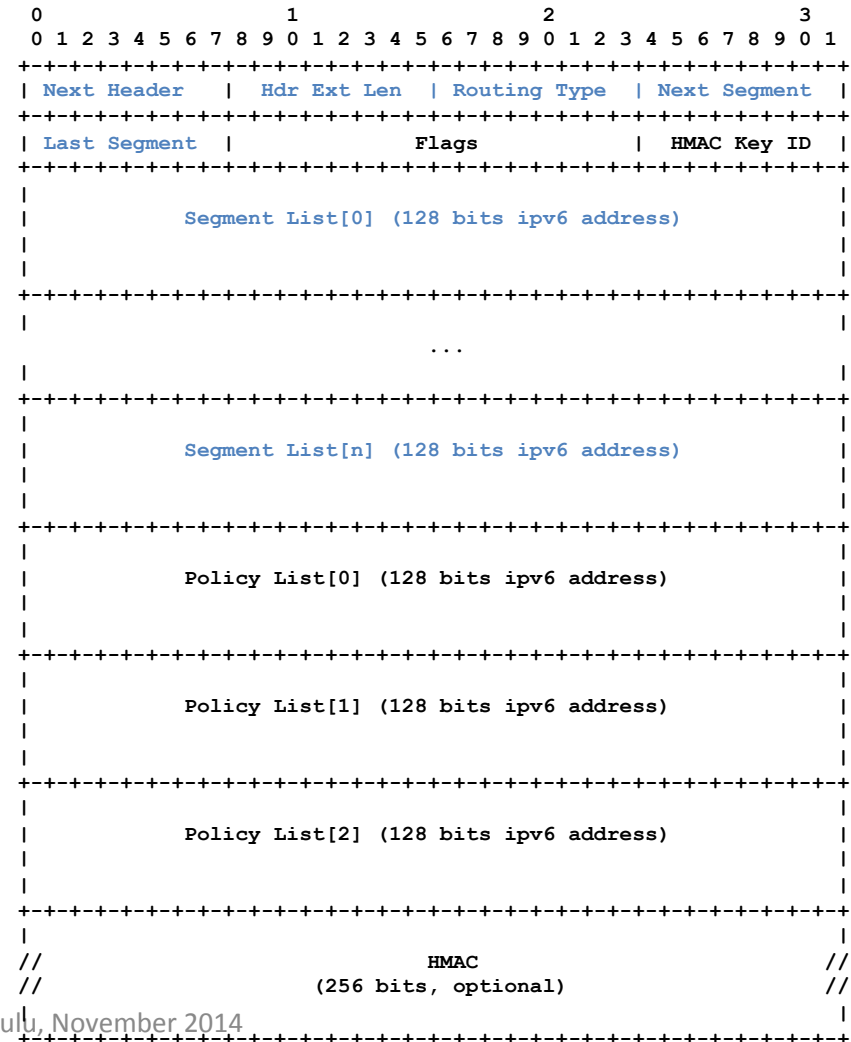
Myth #1: Ext Hdr are dropped on the Internet

- draft-gont-v6ops-ipv6-ehs-in-real-world
  - About 20-40% of packets with Ext Hdr are dropped over the Internet
- SRH works only within one administrative domain
  - => not an issue as operator set the security/drop policy
- Test on your own: http://www.vyncke.org/sr.php
  - And let us know !

Myth 2: RH has been deprecated

- RFC 5095 only deprecates RH-0
  - Amplification / reflection attacks
- RH-2, mobile IPv6 is OK
  - Only one segment
- RH-3, RPL is also OK
  - Within a single administrative domain, not on the Internet

Segment Routing Security

- Addresses concerns of RFC5095
  - HMAC field to be used at ingress of a SR domain in order to validate/authorize the SRH
  - Inside SR domain, each node trust its brothers (RPL model)

- HMAC requires a shared secret (SDN & SR ingress routers)
  - Outside of current discussions
  - Pretty much similar to BGP session security or OSPFv3 security

Implementations

- Multiple implementations exist and interoperability has been demonstrated during IETF-90
  - Cisco,
    Comcast,
    Ecole Polytechnique (Paris),
    UCLouvain (LLN, Belgium)
- Demonstrated interoperability between multiple, independent IPv6 Segment Routing implementations (routers and hosts)
- Illustrate interoperability between SR and non-SR capable routers and hosts
- Illustrate how SR can be leveraged for video content delivery through SR capable caches

Questions?

Ask: Becoming a WG document?

(SPRING are WG doc)

Thanks!