Enhanced Virtual Networks (VPN+)

Stewart Bryant & Jie Dong (Huawei)

draft-bryant-rtgwg-enhanced-vpn

The Problem

- The problem that we need to solve is how to build a VPN that is enhanced to support:
    - Static and dynamic isolation,
    - Low latency,
    - Low packet drop,
    - Seamless integration of both physical and virtual network and service functions,
    - Simple creation, deletion and modification of the services.
- The incentive is to support 5G, but the capability is useful in its own right.

A Layered Solution

- We assume a substrate or underlay that we can draw on to provide the resources and features we need.
- Layer a number of VPNs on the underlay
- A VPN draws on the resources provided by the underlay specifically to this VPN to deliver a service with the required properties.
- We call this an enhanced VPN (VPN+)
- With VPN+, different subsets of the underlay resources are dedicated to different VPNs.
- The VPN+ solution needs tighter coupling with underlay. We cannot for example share the tunnel between enhanced VPNs which require hard isolation.
- In the following slides we explain the solution step by step.

Components of VPN+

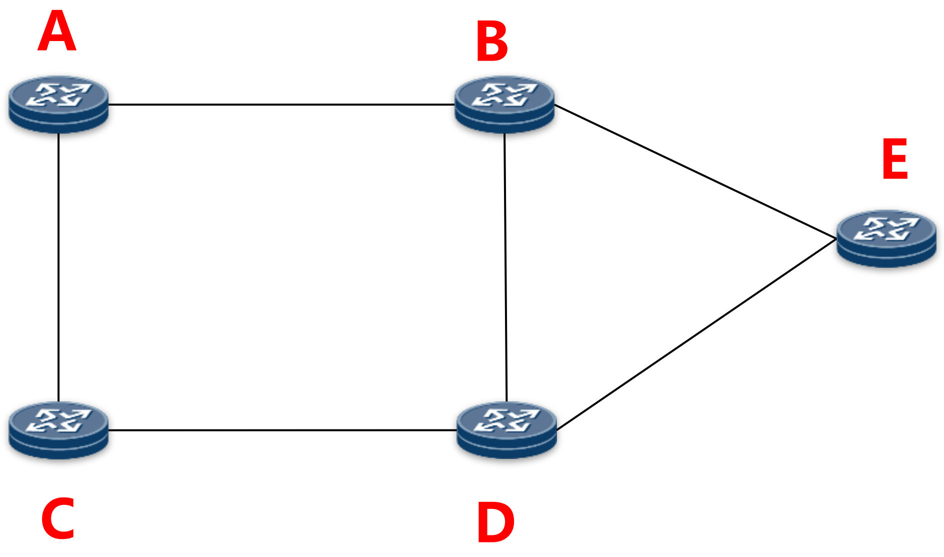
- **Use of Segment Routing Constructs**
  - Fine-grained steering of packets through network and compute resources provided by the underlay to achieve
    - isolation between VPNs
    - guaranteed latency
  - Less core state to support enhanced virtual networks
- **Support of Different Underlay**
  - Unified solution for both IP and MPLS
- **A Hybrid Control Plane**
  - SDN + distributed protocols

Stateful Virtual Networks

- A VPN is a network created by applying a multiplexing technique to the transport network (the underlay) in order to distinguish the traffic of one VPN from that of another.
- A VPN path that travels by other than the shortest path through the underlay normally requires state in the underlay to specify that path.
- State is normally applied to the underlay through the use of the RSVP signalling protocol, or directly through the use of an SDN controller.
- This state gets harder to manage as the number of VPN paths increases.
- Then as we increase the integration between the underlay and the overlay, difficulty of management is going to increase further.

Stateless Virtual Networks

Segment Routing allows us to construct any path through the underlay by expressing a series of next hops as a series of segment identifiers (SIDs) in the packet header.



To send from A to E via B, D & E: Node A prepends the ordered list of SIDs:
D, E and pushes the packet to B.

SID list {B, D, E} can be considered as a VPN path.

To create a VPN a set of SID Lists is created and provided to each ingress node of the VPN together with pkt selection criteria.

We can thus create a VPN with no state in the core.

Some Example

Start with VPN A+D+E

A has lists: {P, B, Q, D}, {P, B, S, E}

D has lists: {Q, B, P, A}, {E}

E has lists: {S, B, P, A}, {D}

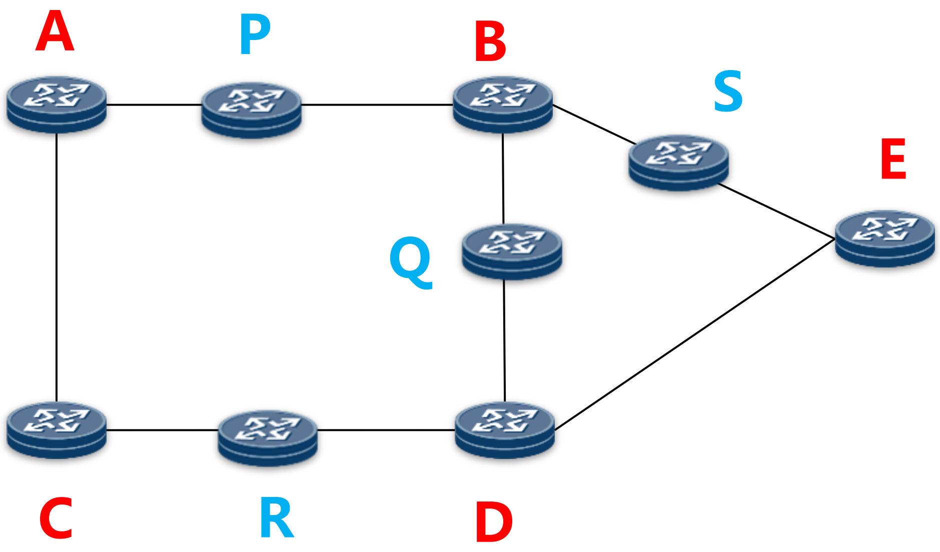
To create VPN C+D+B we give

C lists: {R, D}, {A, P, B}

D lists: {R, D}, {Q, B}

B lists: {Q, D}, {P, A, C}

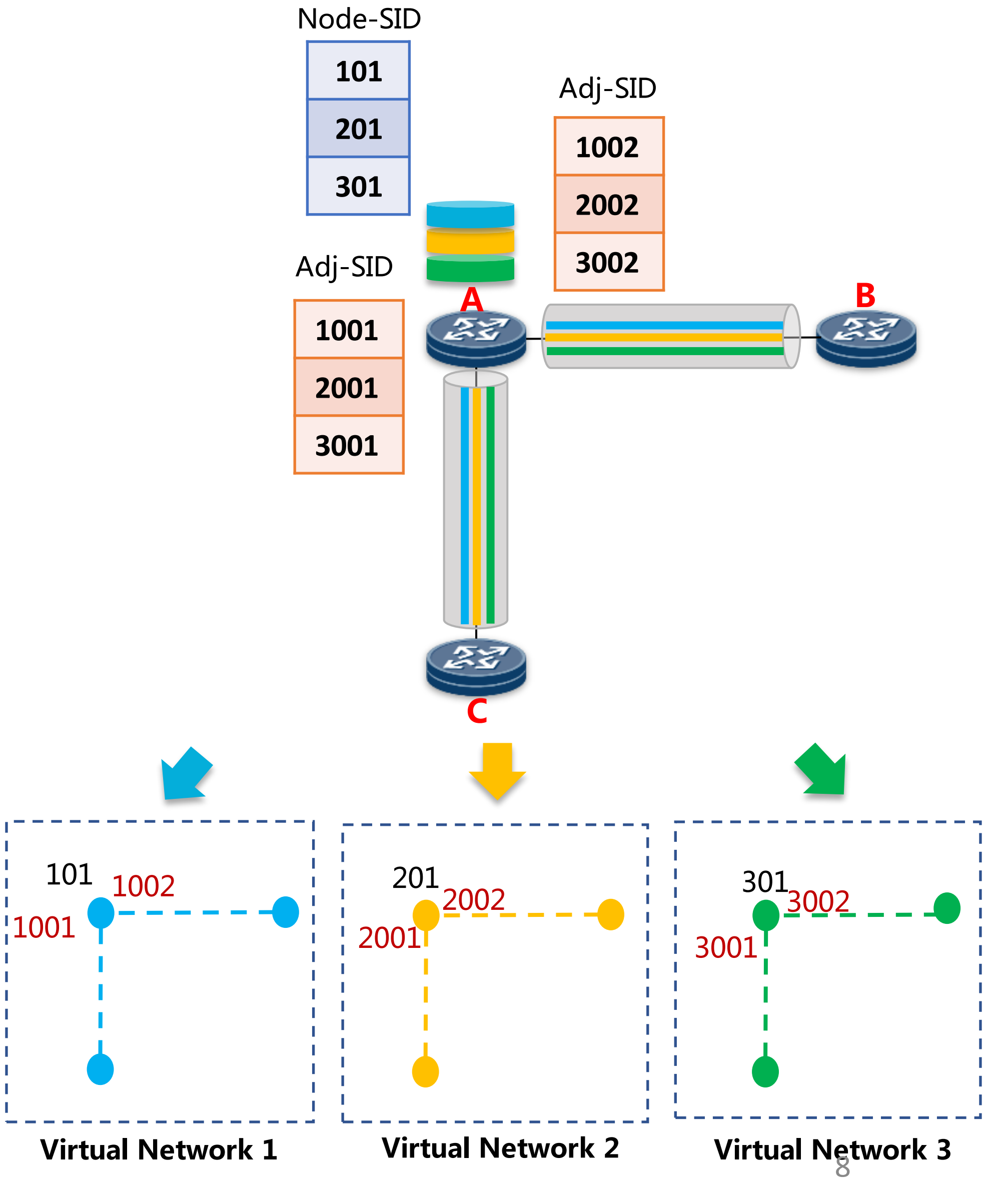
VPN C+D+B was created without touching the settings of the core routers.



We can add endpoints to the VPNs, and move the paths around simply by providing new lists to the affected endpoints

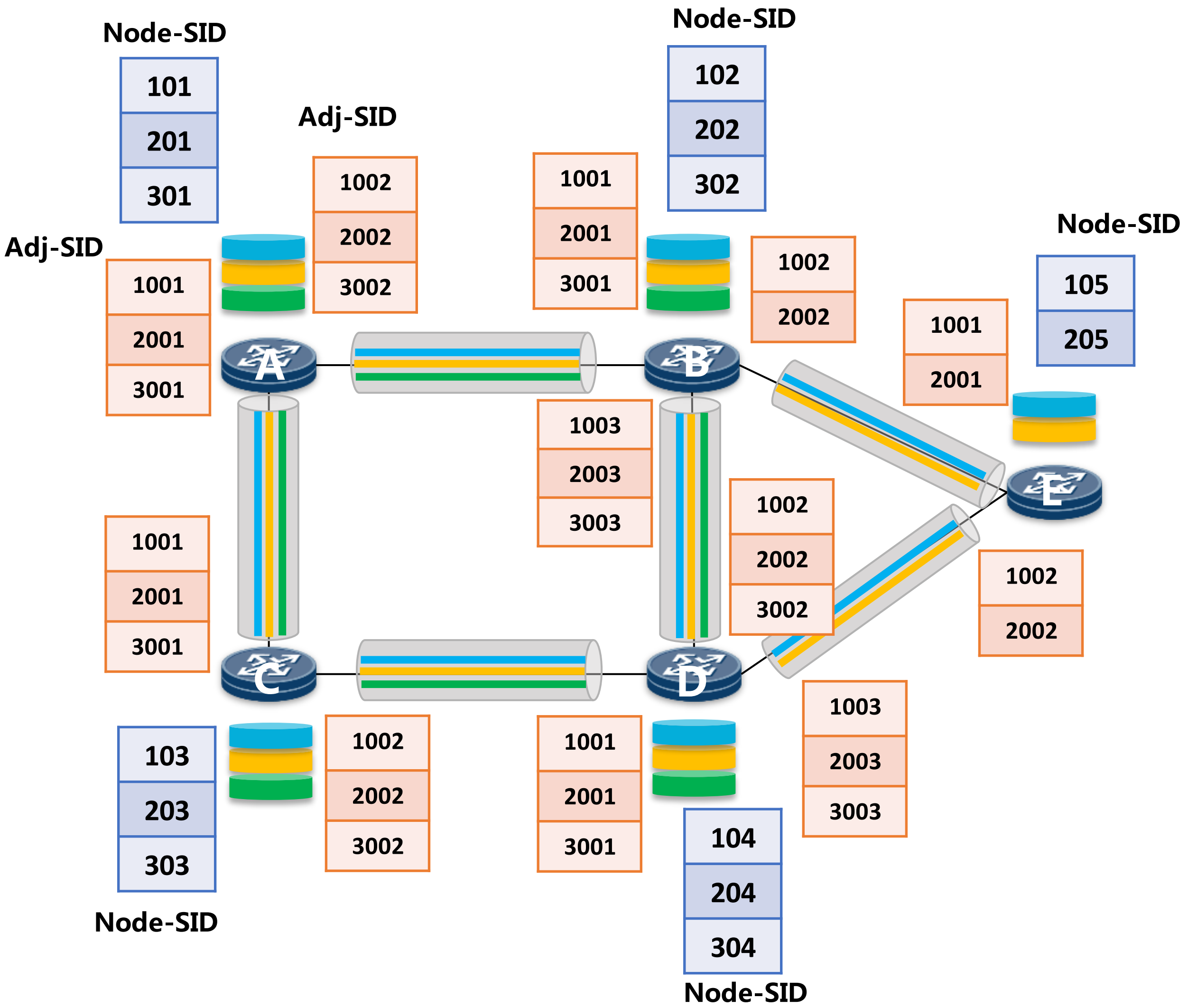
Segment Routing Extension for Isolation

- Use SIDs to represent different partitions of link and node resources
- In the diagrams on the right we have sliced the links and nodes into three parallel but isolated components.
- Virtual networks can be created by using different set of SIDs



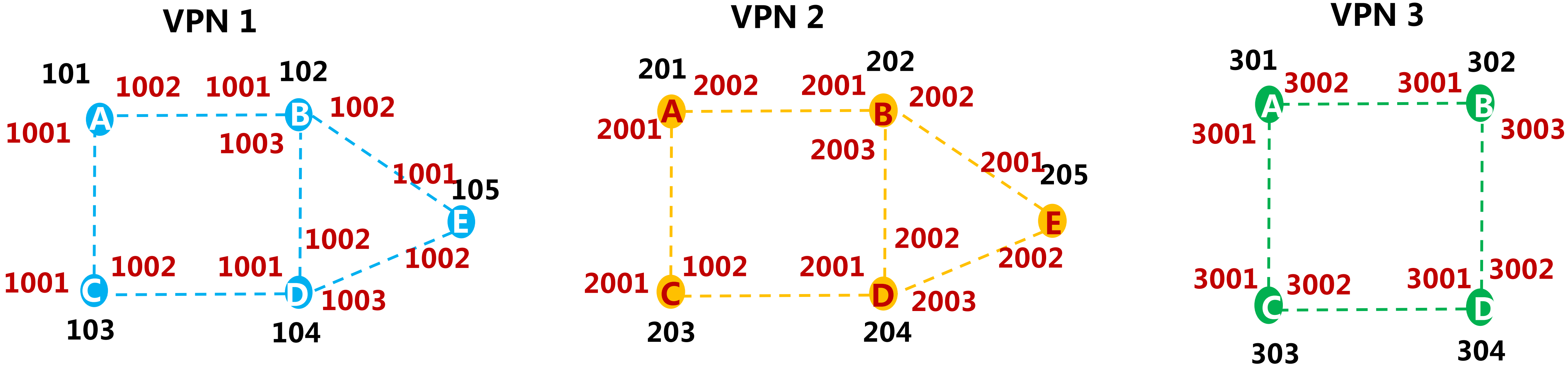
The Underlay Network

- Different SIDs allocated for every partitioned link/node resource



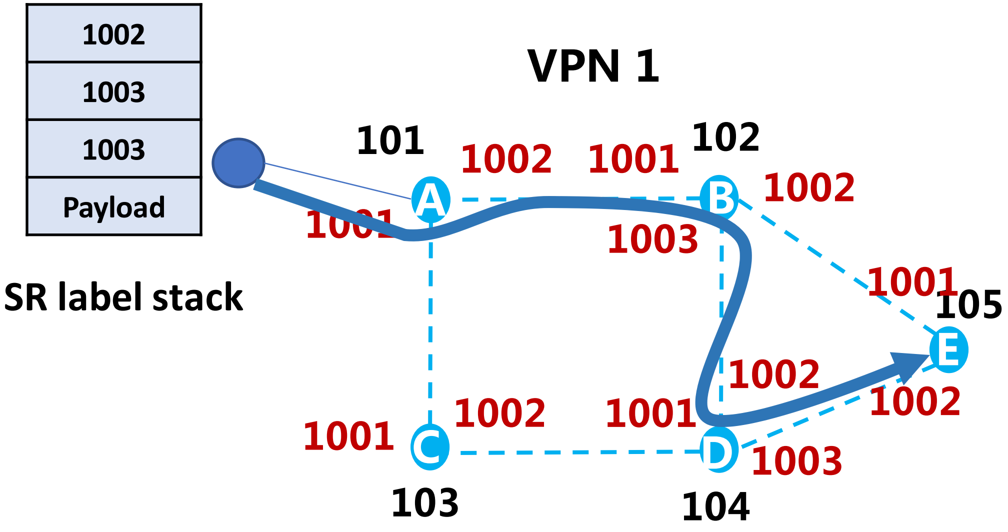
The Enhanced VPNs

- Isolated SR networks
  - Different VPNs bind with different sets of SIDs, i.e. isolated resources



The Enhanced VPNs

- Isolated data plane based on SR forwarding
  - Strict path: A-B-D-E



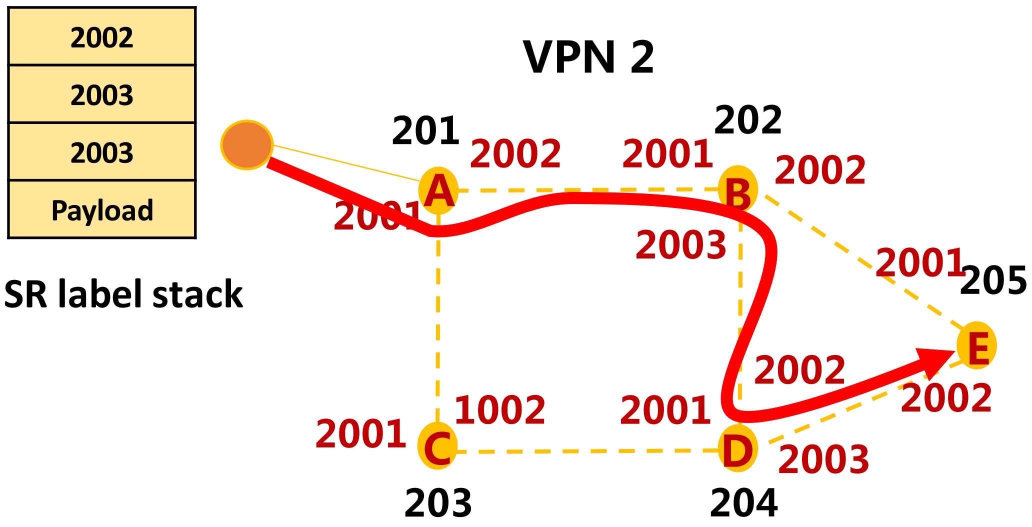
1002

1003

1003

Payload

SR label stack



2002

2003

2003

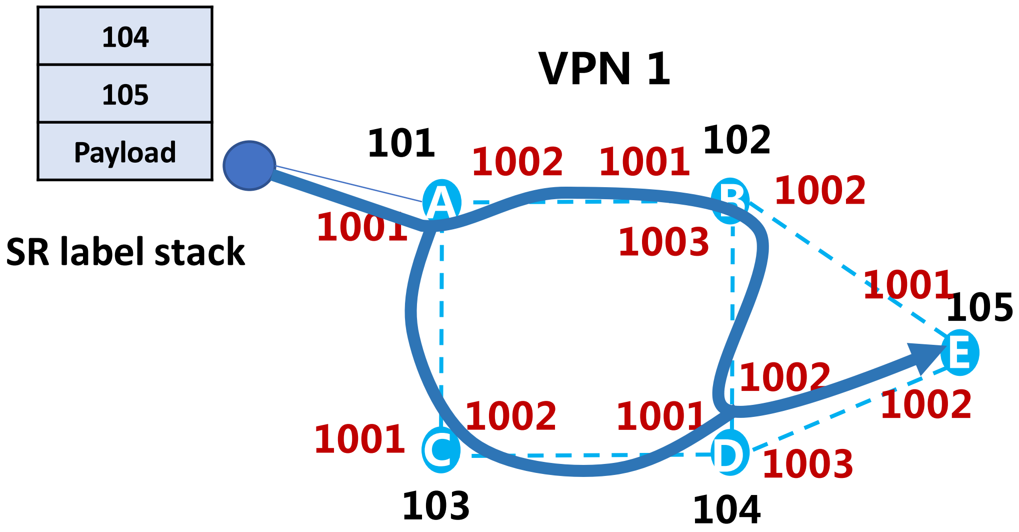
Payload

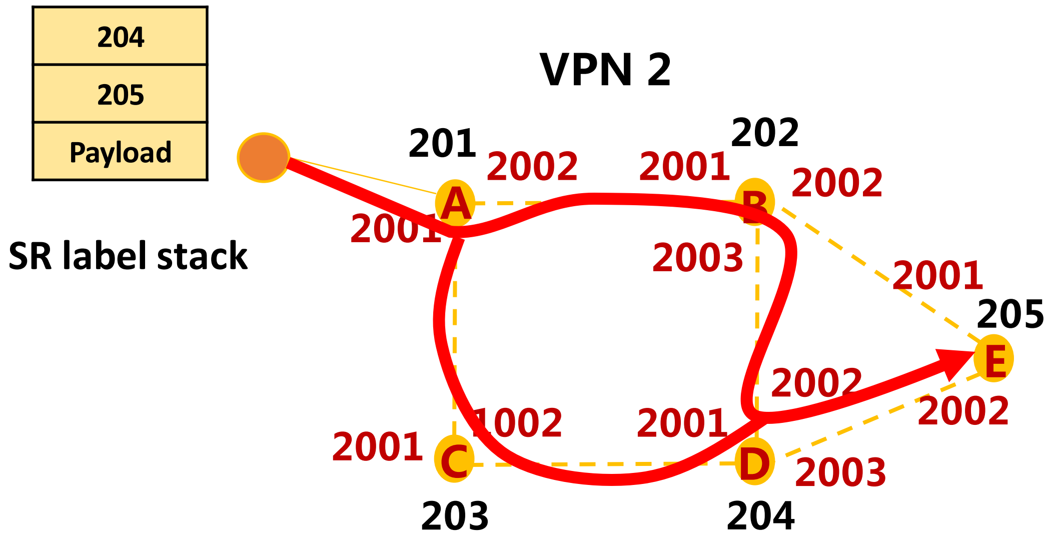
SR label stack

The Enhanced VPNs

- Isolated data plane based on SR forwarding

  - ● Loose path: A-D-E





- Calculation of loose path is constrained to specific VPN topology

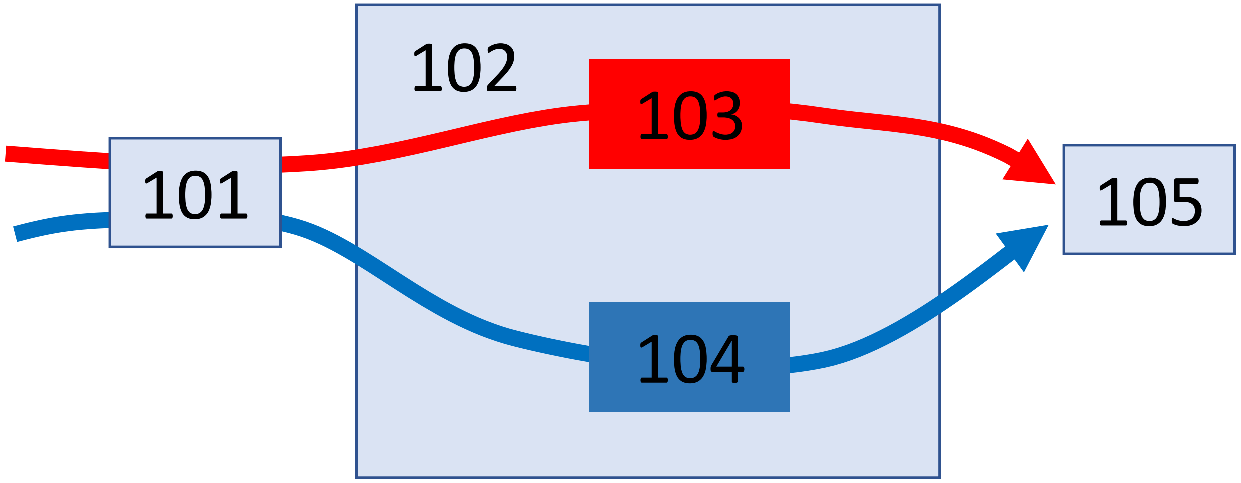
Integrate SFC and VPN

- Within a VPN we have some service functions.
- Suppose we have some function F, which resides on node 3, identified by SID 301. By including 301 at node 3 in the instruction list we add this service function to the path.
- We thus integrate SFC and SR to create VPNs that include service functions.
- VPNs can make use of dedicated or shared service functions.
- We now have stateless VPNs with service functions.
  - Detailed mechanism described in draft-xu-mpls-service-chaining.

Steering Through Resources

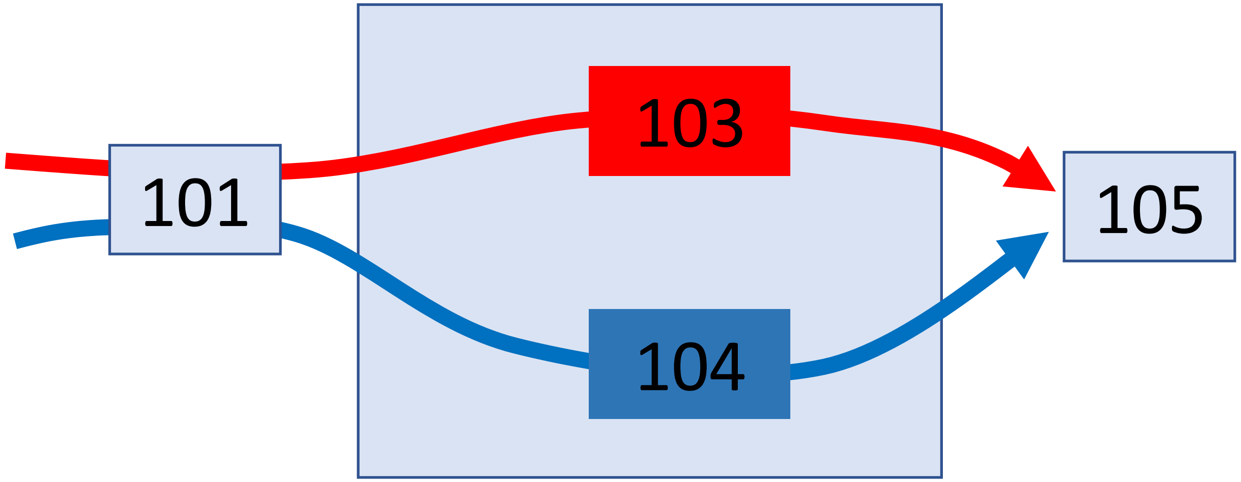
- We can extend the mechanism to route a packet through any identifiable resource within the enhanced VPN. For example:
  - Physical and virtual interfaces
  - NPU
  - CPU core
  - Lookup engine
- We can also use the approach to tune the behaviour at a resource.
- This allows us to provide the isolation and performance guarantees we need.

Example of steering through specific resources



In VPN A we allocate path (101, 102, 103, 105), and in VPN B we allocate path (101, 102, 104, 105).

Of course we can do a similar thing with RSVP-TE if we wish.



Objects can be sub-objects of nodes as above, or they can be objects in their own right.

Support of Different Underlay

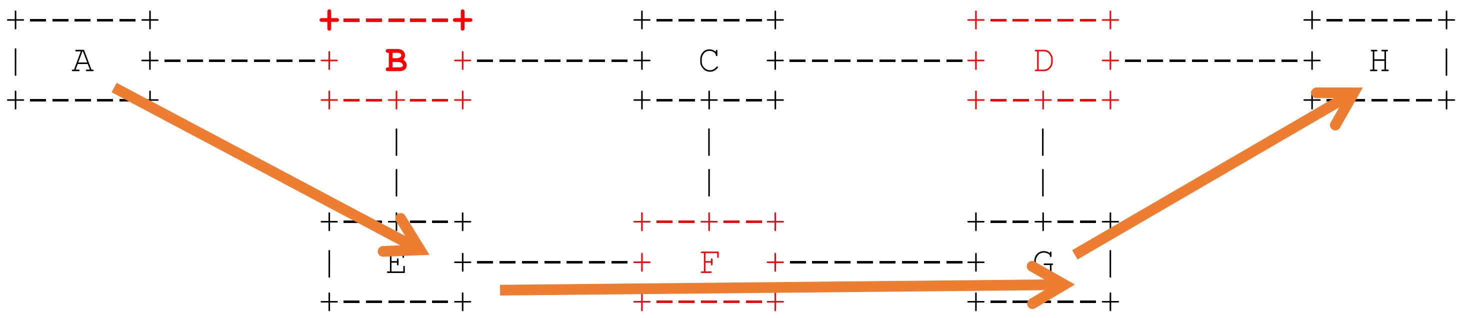
- MPLS is a widely used VPN underlay.
- In some cases we need to run over an IP underlay, and in some cases we need to bridge two MPLS domains via an IP network.
- There are a number of ways of doing this, but one of the simplest is to run the SR based VPN directly over IP as described in
  - draft-xu-mpls-unified-source-routing-instruction and
  - draft-bryant-mpls-unified-ip-sr

IP Encapsulation

```
+----------------------+
|  IP (v4 or v6)header |
+----------------------+
|         UDP          |
+----------------------+
|       MPLS Label     |
.         Stack        .
+----------------------+
|                      |
.        Payload       .
|                      |
+----------------------+
```

- The KEY point of the MPLS Label Stack is that it is hardware friendly, existing way of carrying a series of 20 bit instructions (SFid, SID, etc).
- The UDP header is used to provide an ECMP method that works with existing IP forwarders. It is also middlebox friendly.
- The IP header can be IPv4 or IPv6.

Tunnelling MPLS-SR over an IP Network



```
+---------+
|IP(A->E)|
+---------+               +---------+
|  L(G)   |               |IP(E->G)|
+---------+               +---------+     +---------+
|  L(H)   |               |  L(H)   |     |IP(G->H)|
+---------+               +---------+     +---------+
| Packet  |     --->      | Packet  | --->| Packet  |
+---------+               +---------+     +---------+
```

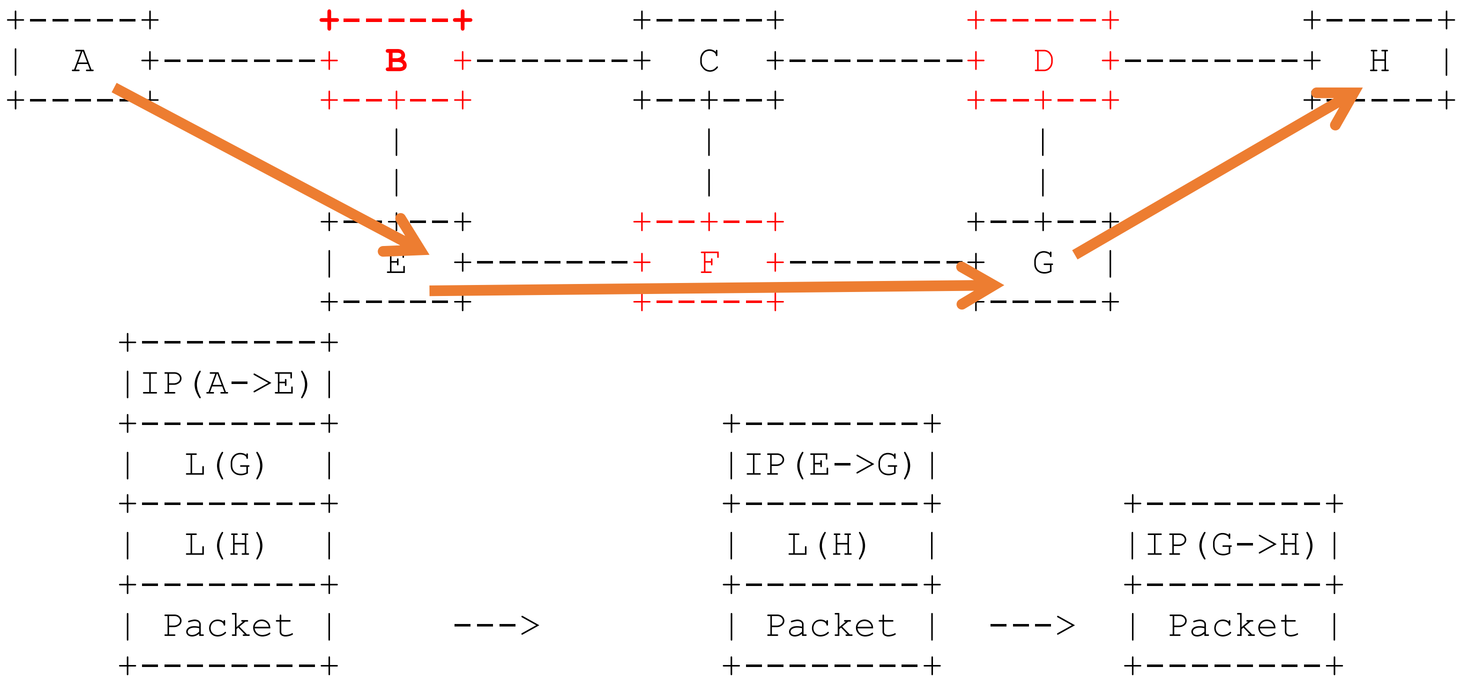
B, F & D NOT MPLS-SR Capable

L(E)-> Next Hop E

L(G)-> Next Hop G

L(H)-> Next Hop H

SR in an IP Network



```
+------+      +------+      +------+      +------+      +------+
|  A   +------+  B   +------+  C   +------+  D   +------+  H   |
+------+      +--+---+      +--+---+      +--+---+      +------+
                 |             |             |
                 |             |             |
              +--+---+      +--+---+      +--+---+
              |  E   +------+  F   +------+  G   |
              +------+      +------+      +------+
     +---------+
     |IP(A->E)|
     +---------+               +---------+
     |  L(G)   |               |IP(E->G)|
     +---------+               +---------+        +---------+
     |  L(H)   |               |  L(H)   |        |IP(G->H)|
     +---------+               +---------+        +---------+
     | Packet  |      --->     | Packet  |  --->  | Packet  |
     +---------+               +---------+        +---------+
```

B, F & D Simply forward IP packets

E & G Interpret the 20 labels as :

L(G)-> Next Hop G

L(H)-> Next Hop H

Summary of Unified Encapsulation

- A single compact data plane format can support
  - Interconnection of disjoint MPLS-SR islands
  - Service Function Chaining
  - Segment Routing version X.
- The required data-plane specifications mostly exist (RFC7510 MPLS-over-UDP).
- It is important to focus on the 20 bit instructions, not the packaging of those instructions into a RFC3032 format. This packaging is just a convenience – Compact design makes the forwarder simpler and cheaper.
- It is also important to remember that the use of RFC3032 format DOES NOT imply that we always use the MPLS control protocols.
- This unification approach has many benefits, and is worthy of further development.

Control Plane(s)

- Initial focus is on the data-plane. Matching control plane will follow.
- The underlay follows the best current practise in SDN design:
  - IGP to establish base connectivity and physical connectivity to the SDN controller
  - BGP-LS to advertise topology and status information to controller
  - PCE/Controller to compute the SR path within each virtual network, and program the path to the edge nodes
- The overlay can use a shared or dedicated control plane and run as an isolated component

Conclusion

This design gives us:

- Virtual networks with hard isolation (compared to classic VPN)
- Both hard isolation and statistical multiplexing.
- Tighter integration between underlay and overlay.
- Resource reservation on a per VPN basis.
- Less state in the core compared to other techniques.
- Integrates VPNs and service functions and provides guaranteed performance.

Thank You

Further Questions?