SRv6 for Mobile User-Plane

draft-matsushima-spring-dmm-srv6-mobile-uplane-**03**

IETF100

S.Matsushima, C.Filsfils, M.Kohno, D.Voyer

Feedbacks after IETF99

- **<u>Many people asked: System Impacts?</u>**
  - To current control-plane protocol.
  - To current RAN.
  - People really care degree of system impact to change U-plane protocol from current one.

- **<u>Benefits?</u>**
  - What is able to do with SRv6?
  - Isn’t that possible with current u-plane protocol?
  - Isn’t SRv6 just another tunneling protocol?

- **<u>Nobody asked: How SRv6 works for mobile user-plane</u>**
  - Sounds good. I did it.

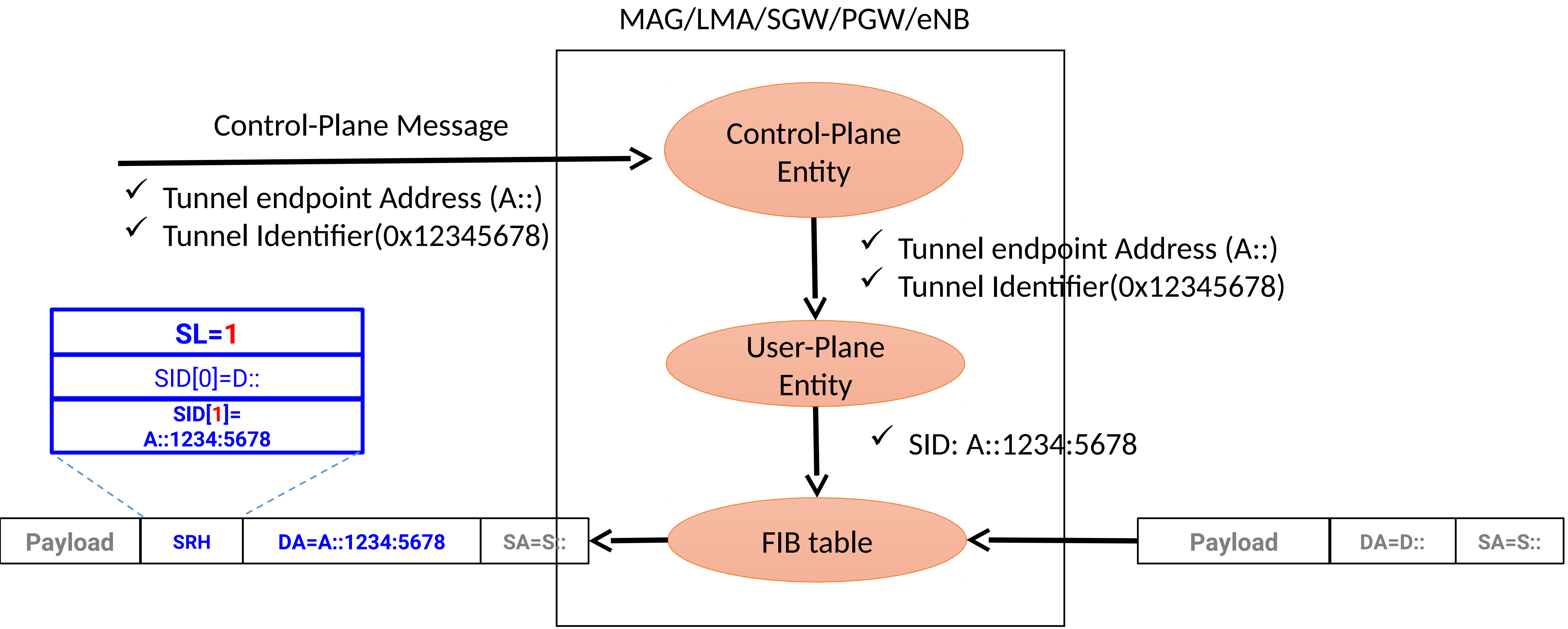
Updates to v03: Answer to the Feedback

- **Introduces "Basic Mode" User-Plane**
  - (It is supposed) **No impact** to control-plane, but no advanced SRv6 features in there.
  - Operator is able to gradually migrate from basic to more advanced mode.

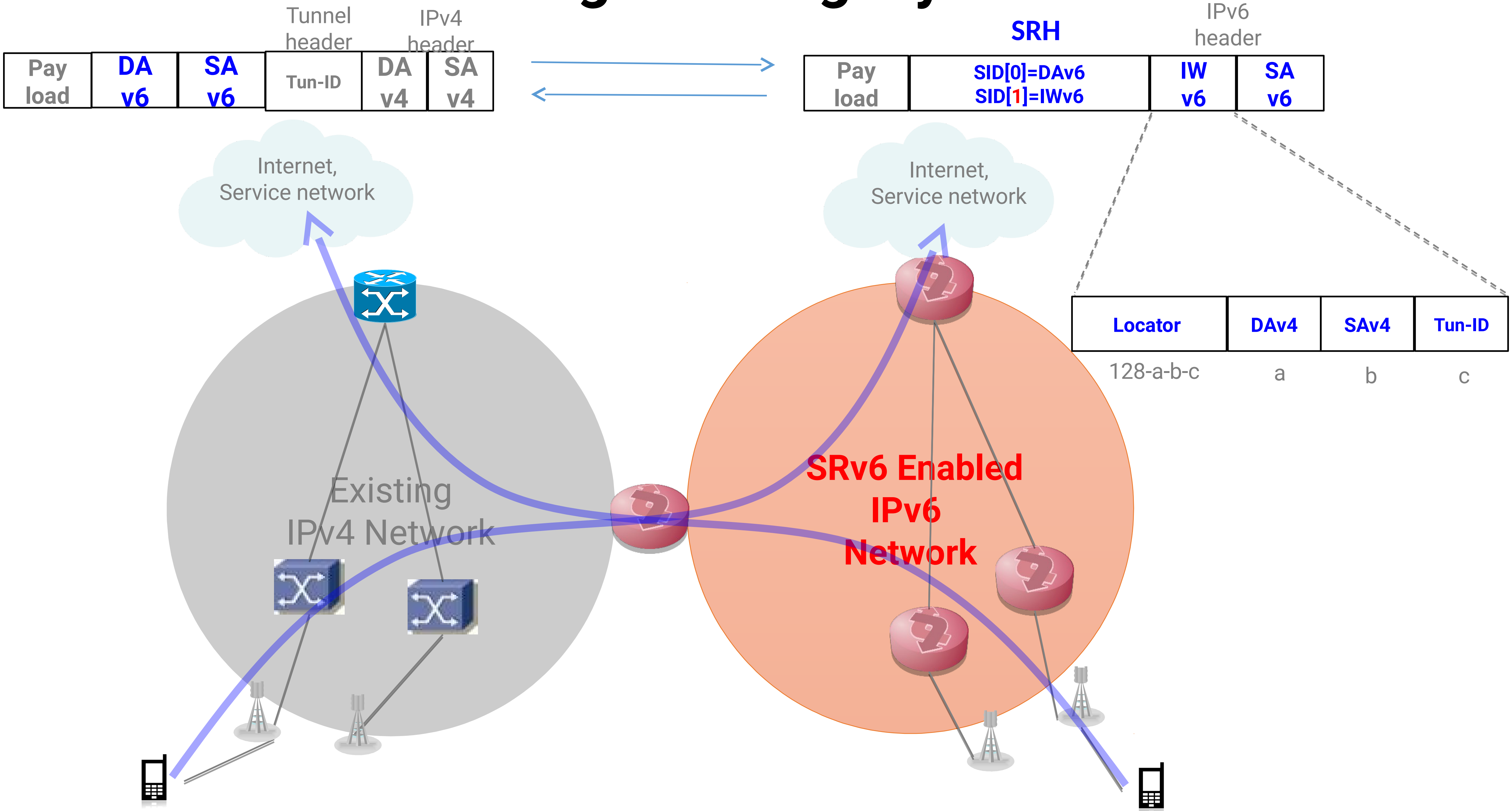
- **Introduces an Use Case "Stateless Interworking with Legacy Access"**
  - (It is supposed) **No impact** to current RAN in control-plane.

- **Introduces "Aggregate Mode" User-Plane**
  - Benefits **seamless deployment** of service-chain, VPNs and TE within the mobile user-plane.
  - Complicated? Mobile control-plane can focus to only manage mobility and keep simple.
    - Implementing other service policies to the user-plane can be done by separate systems.
  - Complicate text? May need to find concise way to describe the procedures.

Leveraging Current Control-Plane



Stateless Interworking with Legacy Networks



Updates to v03: Technical Progress

- **Introduces New SRv6 Functions: “End.TM” and “T.Tmap”**
  - To support Stateless Interworking with legacy user-plane with some parameters.

End.TM

(Endpoint function with encaps for mapped tunnel)

SRv6 -> Legacy

```
IF NH=SRH & SL > 0 THEN
   decrement SL
   update the IPv6 DA with SRH[SL]
   push header of TUN-PROTO with tunnel ID from S            ;; Ref1
   push outer IPv4 header with SA, DA from S
ELSE
   Drop the packet

 Ref1: TUN-PROTO indicates target tunnel type.
```

T.Tmap

(Transit behavior with decaps tunnel and map SRv6 policy)

Legacy-> SRv6

```
  IF P.PLOAD == TUN-PROTO & T.PLOAD == IPv6 THEN    ;; Ref1, Ref1bis
     pop the outer IPv4 header and tunnel headers
     copy IPv4 DA, SA, TUN-ID to form SID B with IW-IPv6-Prefix
     insert the SRH (D, B; SL=1)                    ;; Ref2, Ref2bis
     set the IPv6 DA = B
     forward along the shortest path to B
  ELSE
     Drop the packet

 Ref1: P.PLOAD and T.PLOAD represent payload protocol of the receiving
 packet, and payload protocol of the tunnel respectively.
```

Work in Progress

- **<u>QoS and Accounting</u>**
  - Enables SID to represent QoS and accounting policy.

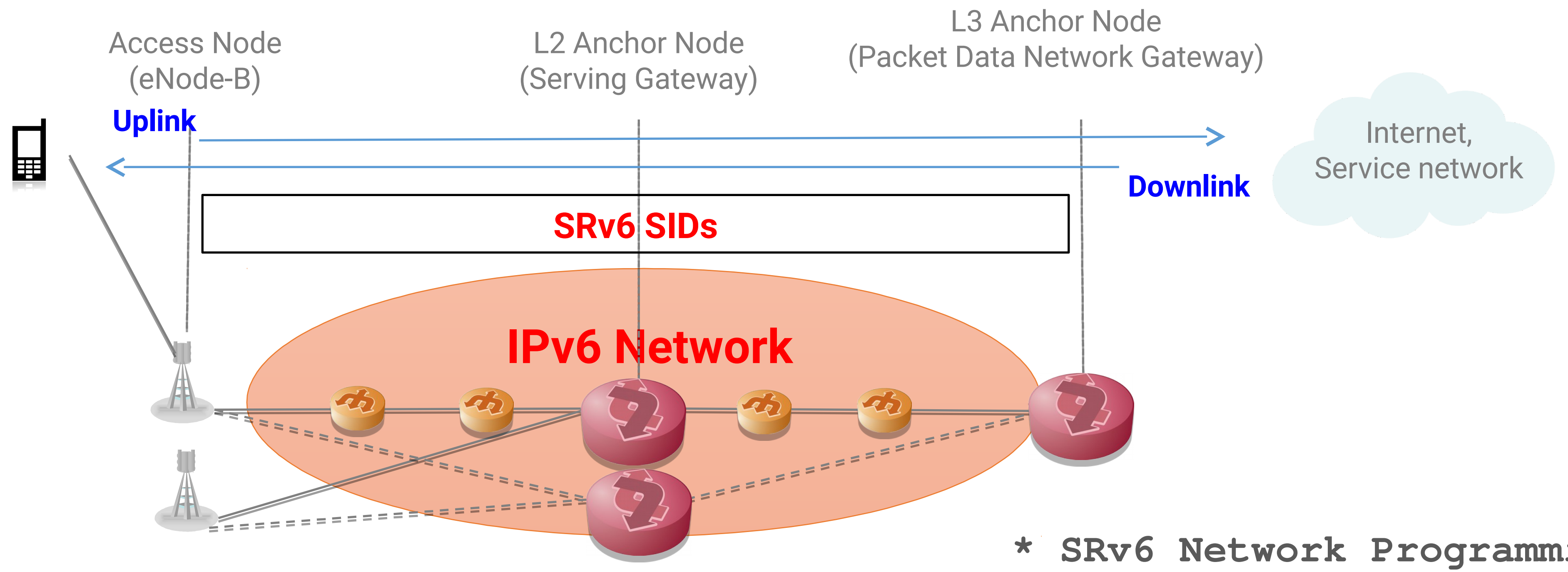
- **<u>E2E SR and Network Slicing</u>**
  - Enables Apps running on MN be able to designate slices.

- **<u>IPv4 Support</u>**
  - Carries IPv4 user packets.
  - Many IPv6 transition solutions make it can be considered as an user application on IPv6.
    - MAP-E, MAP-T, NAT64, 464XLAT and DS-Lite.

- **<u>Collaborations</u>**
  - 3GPP CT4 to initiate study work of user-plane protocol.

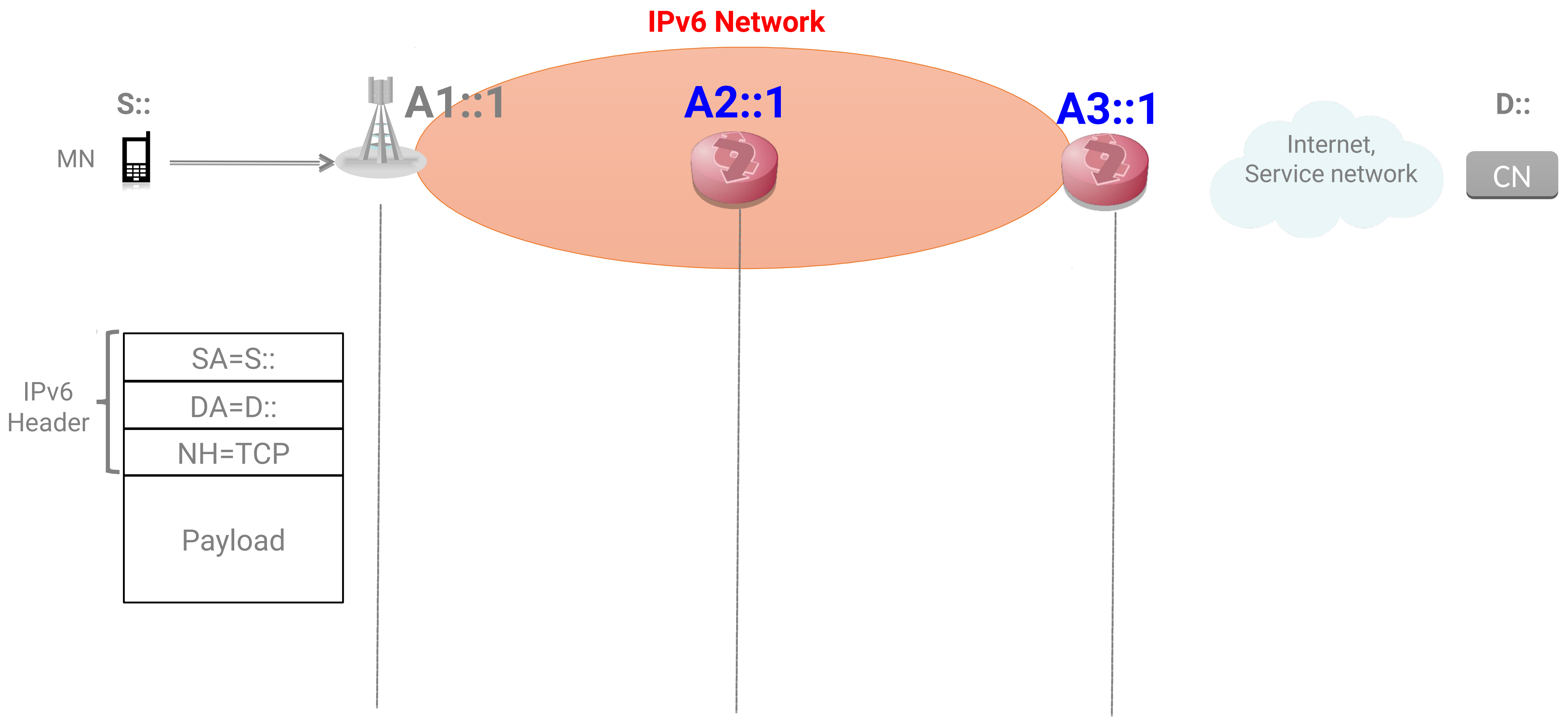
Basic Mode

	Uplink	Downlink
Access Point	T.Insert	End
L2 Anchor Node	End.B6	End.B6
L3 Anchor Node	End.T	T.Insert

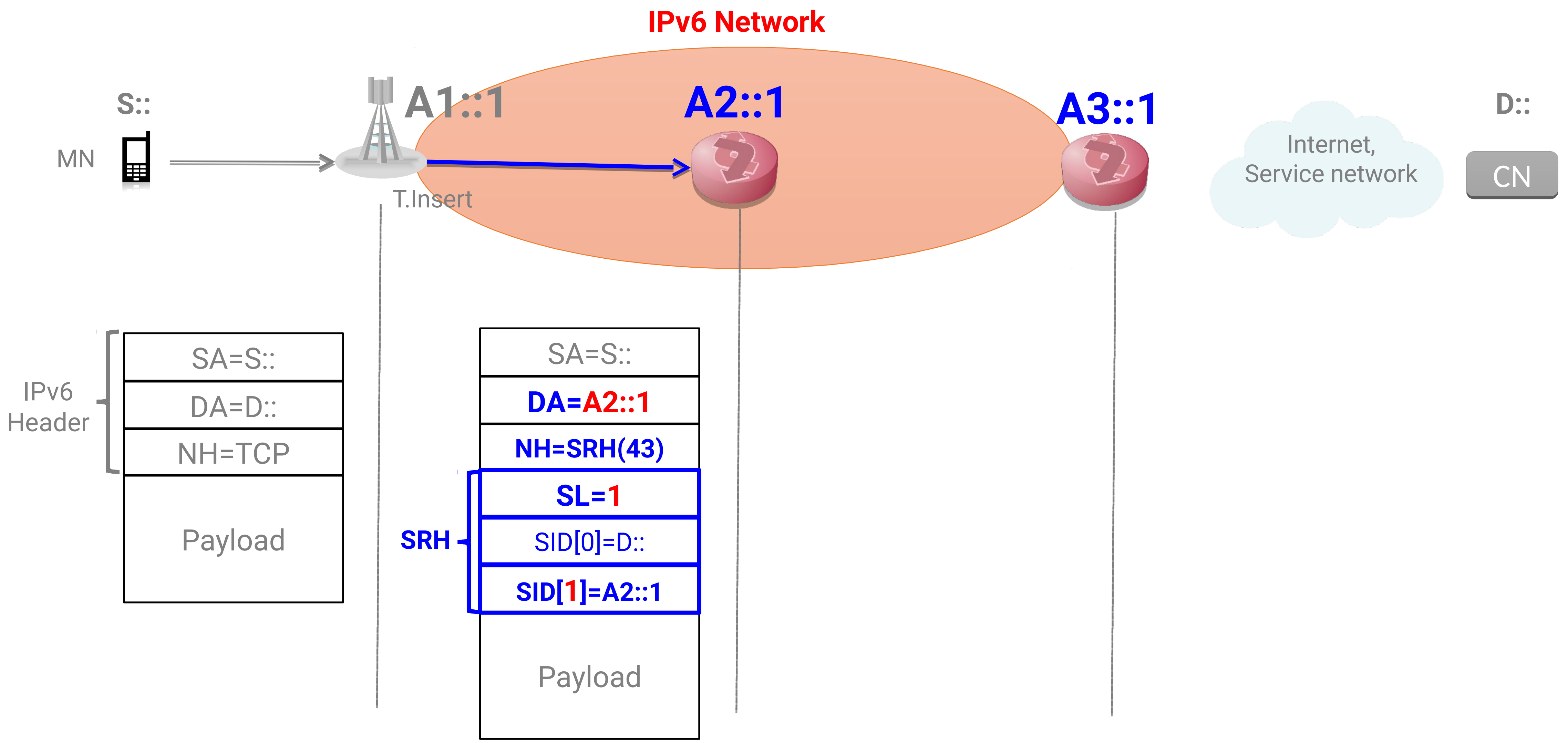


* SRv6 Network Programming

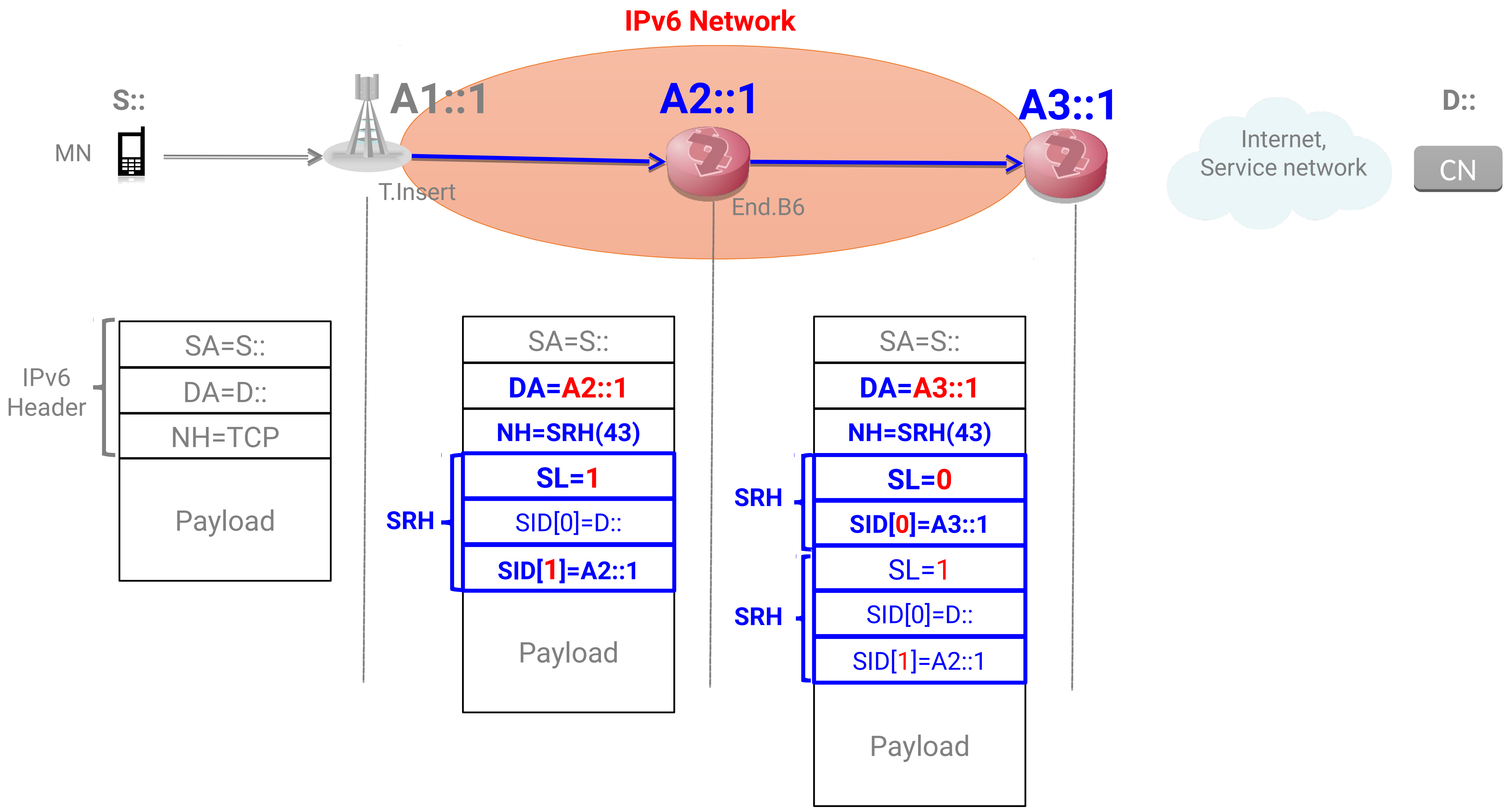
Basic Mode User-Plane Flows (Uplink)



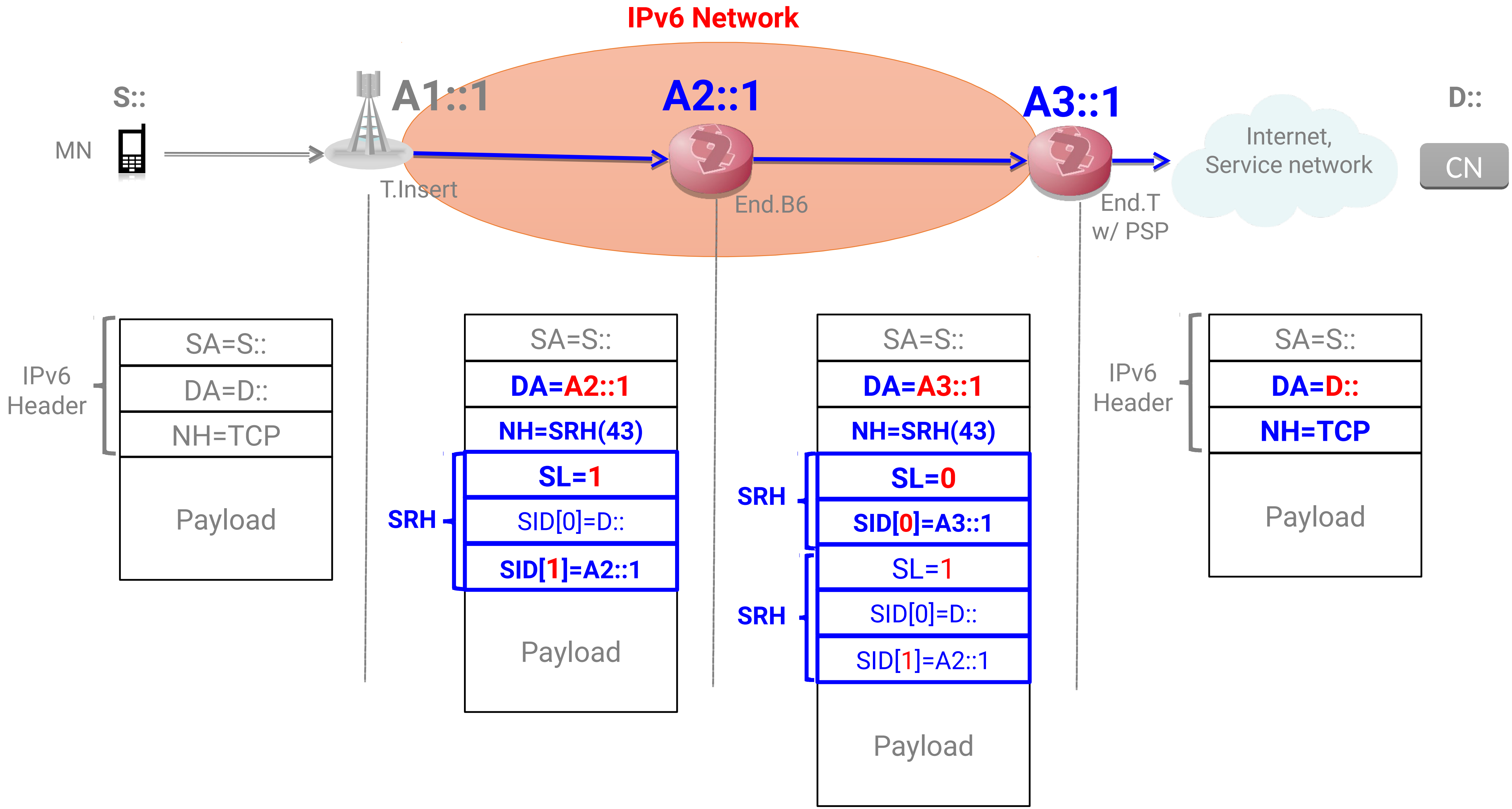
Basic Mode User-Plane Flows (Uplink)



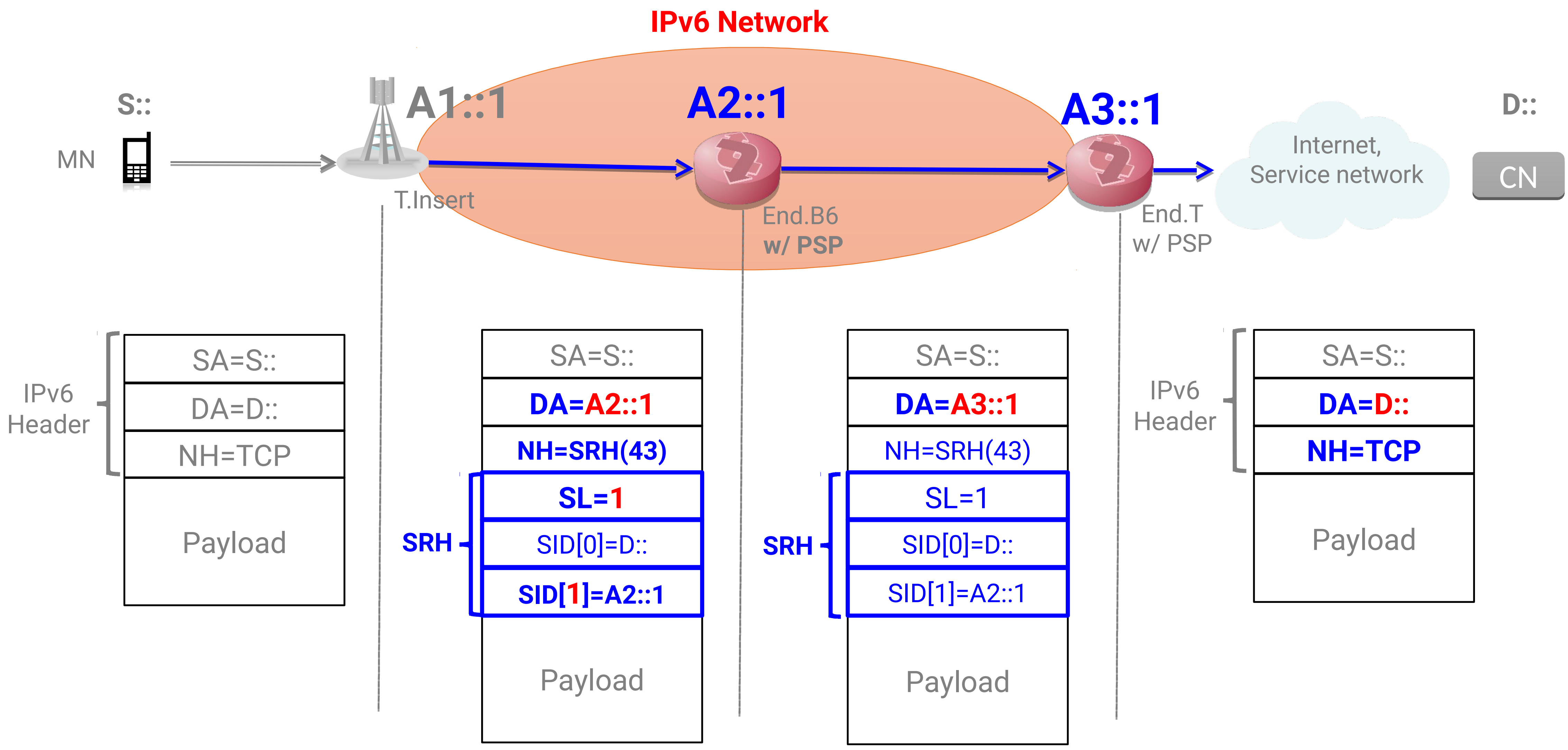
Basic Mode User-Plane Flows (Uplink)



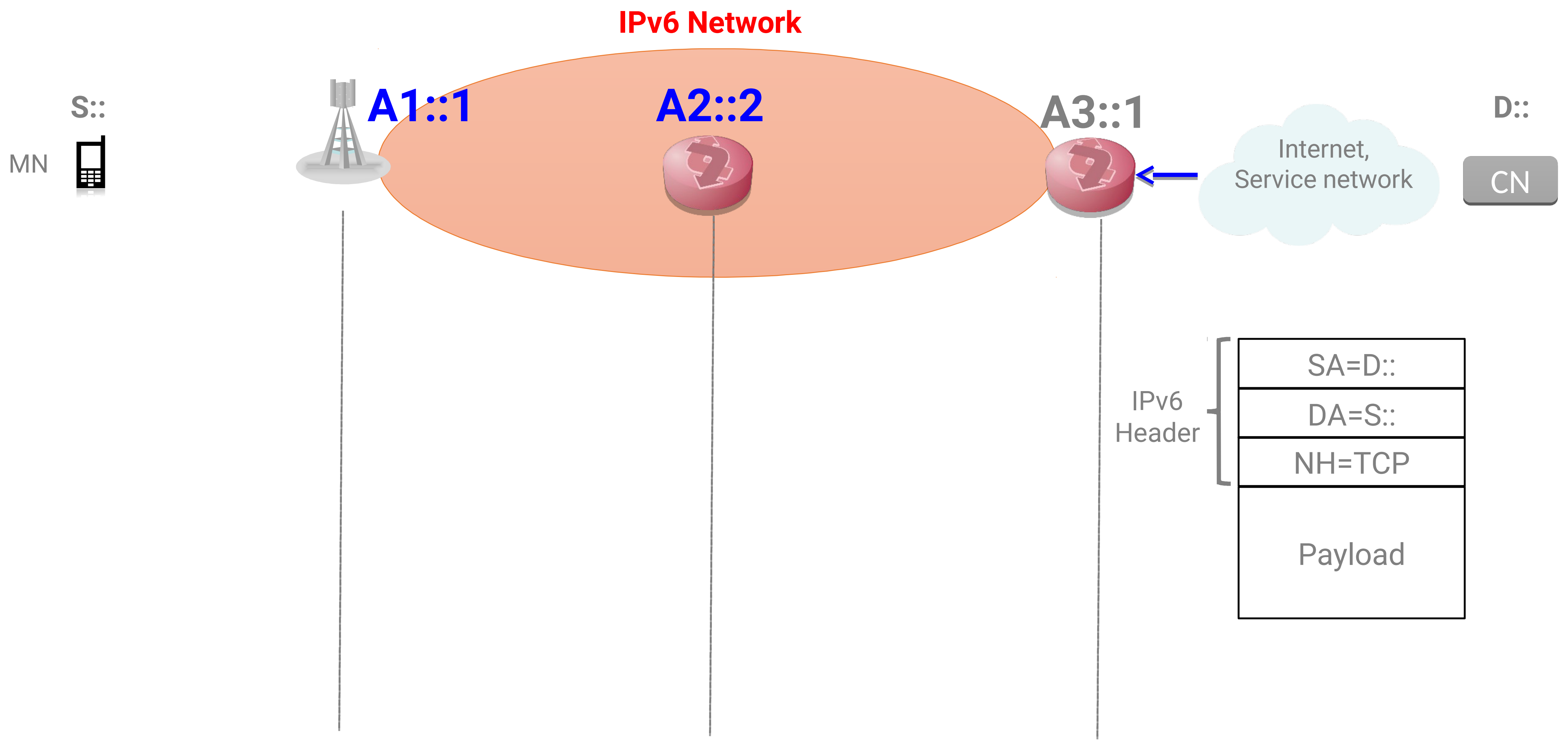
Basic Mode User-Plane Flows (Uplink)



Basic Mode User-Plane Flows (Uplink)



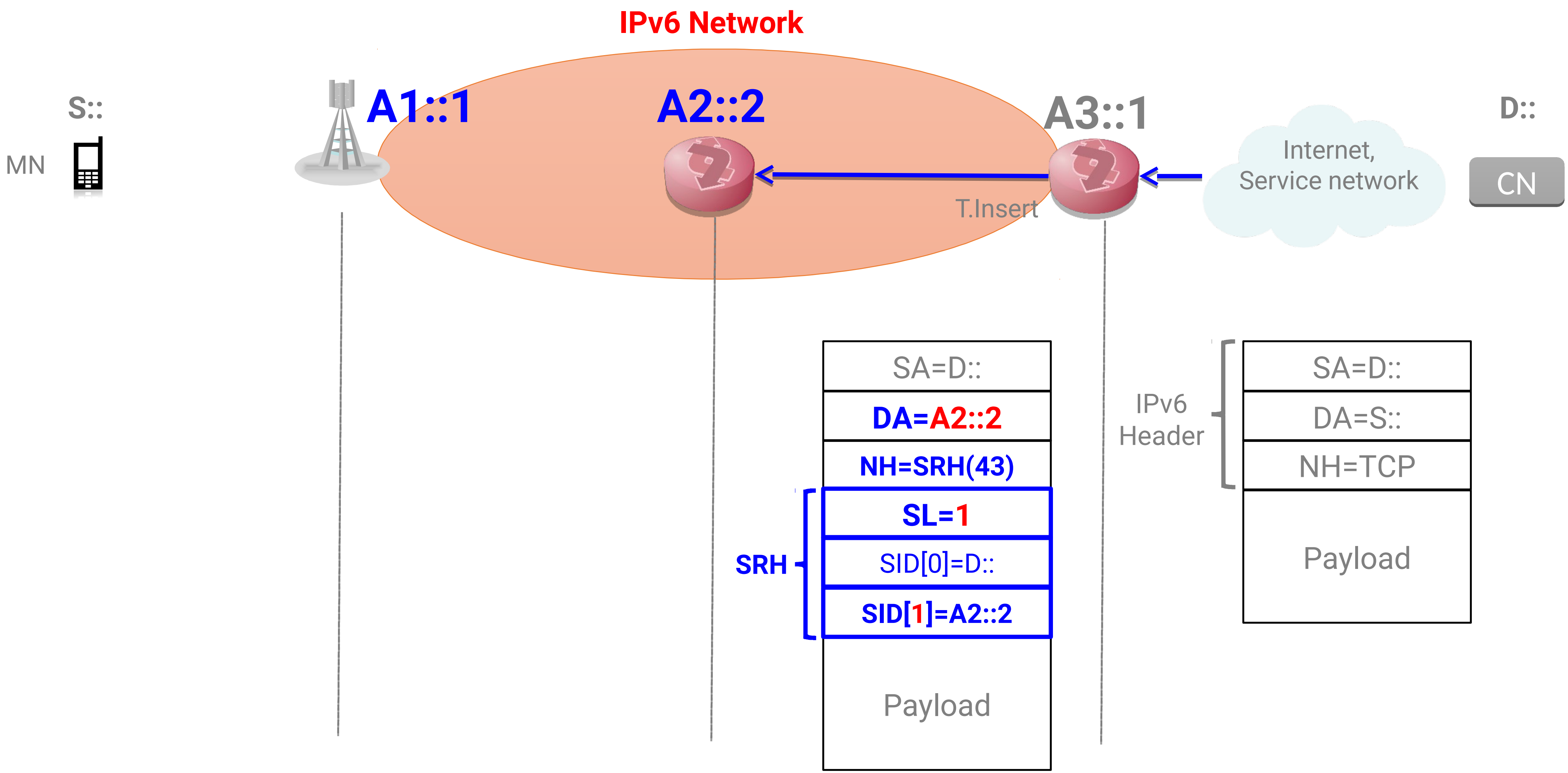
Basic Mode User-Plane Flows (Downlink)



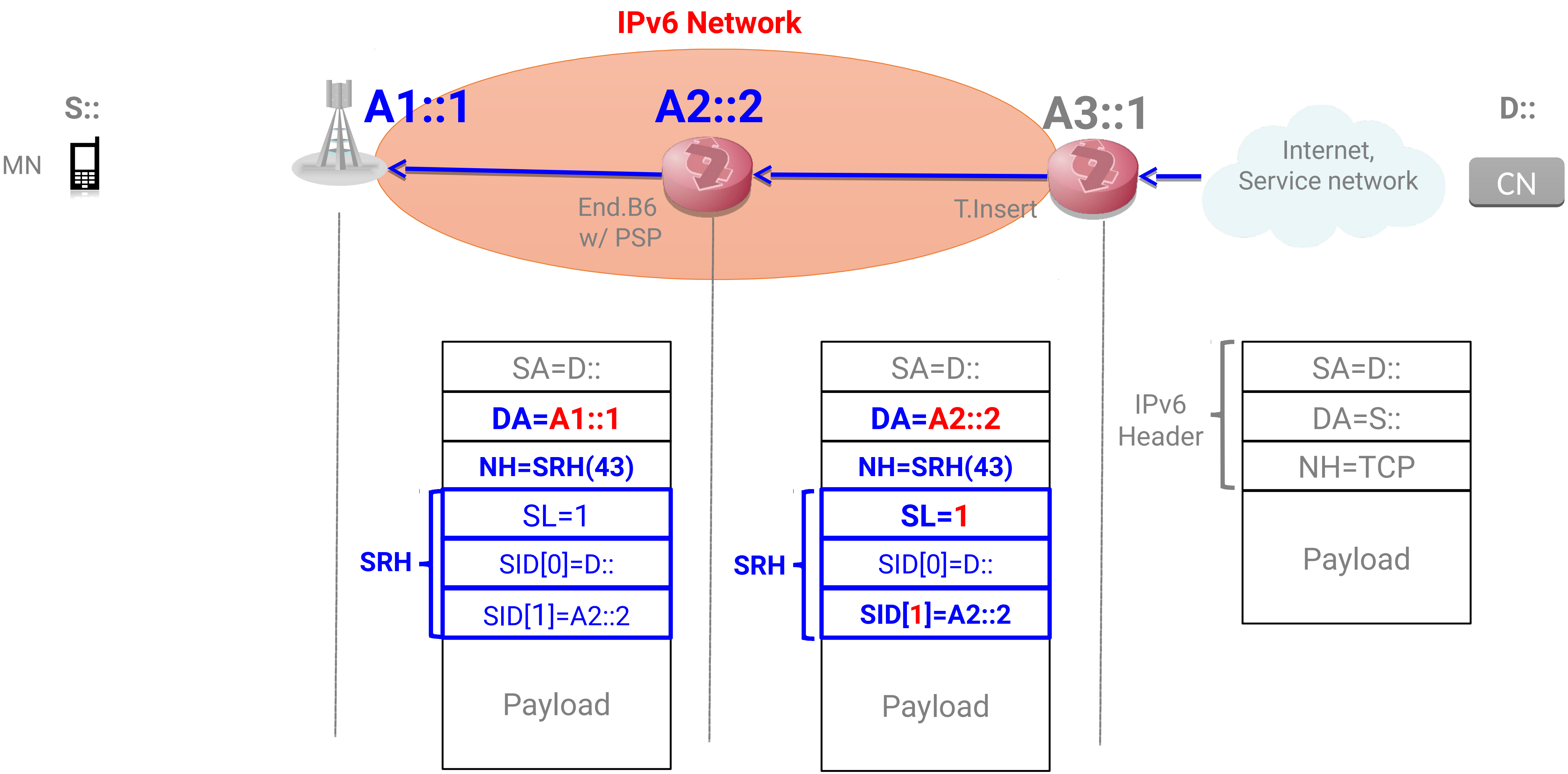
IPv6 Header

SA=D::
DA=S::
NH=TCP
Payload

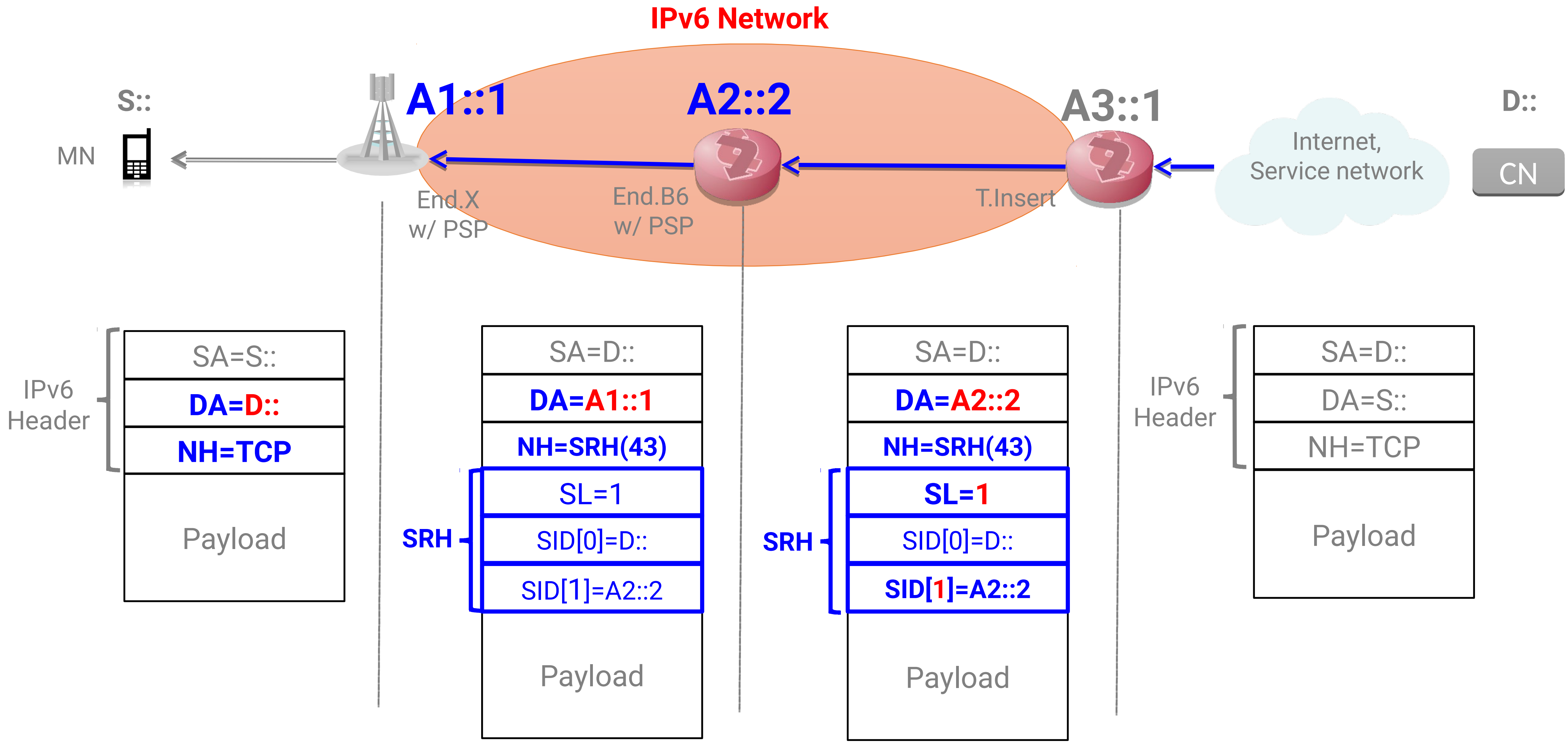
Basic Mode User-Plane Flows (Downlink)



Basic Mode User-Plane Flows (Downlink)

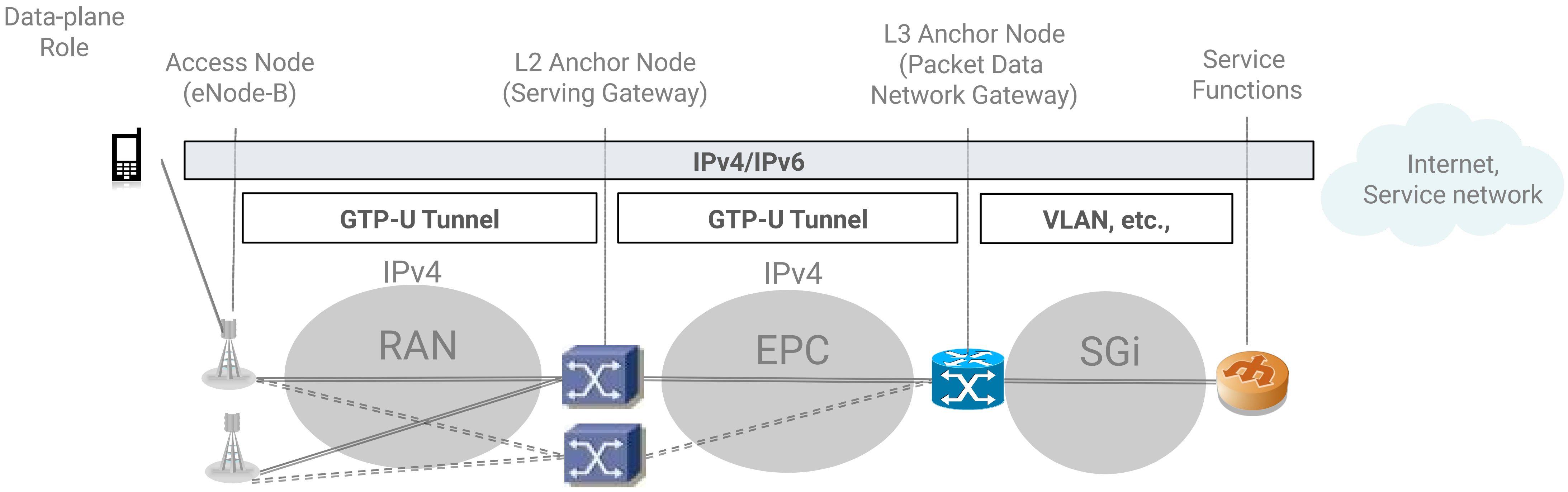


Basic Mode User-Plane Flows (Downlink)



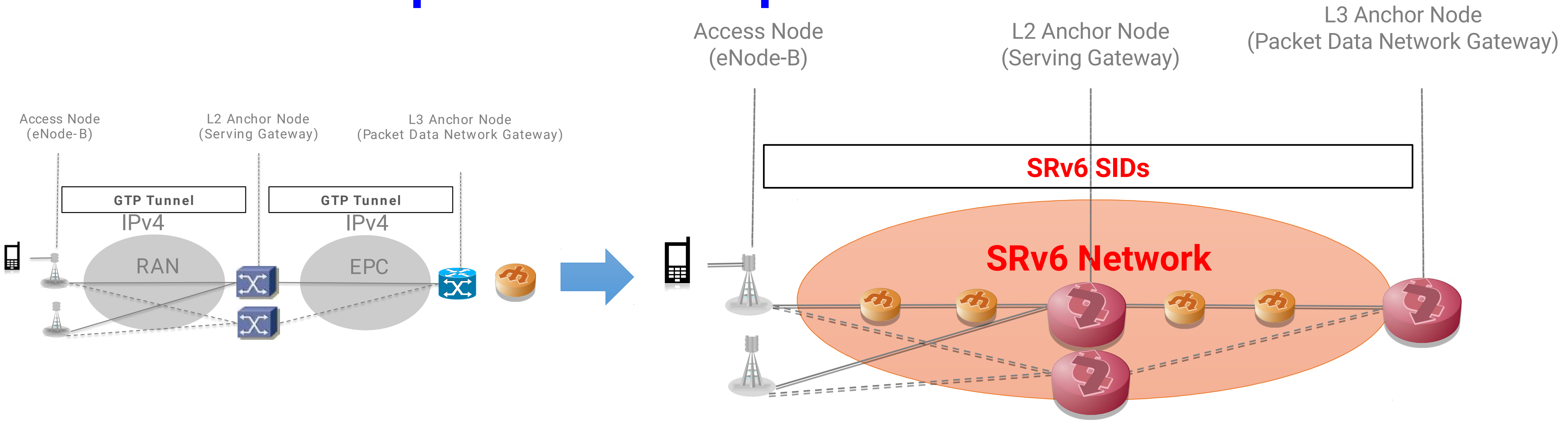
A Current Mobile Network Example

- **Well fragmented to RAN, EPC and SGi.**
- **Per-session tunnel creation and handling.**
- **Non-optimum data-path.**



What if SRv6 Becomes An Alternative of GTP-U Tunnel?

- ~~Well fragmented to RAN, EPC and SGi.~~
- ~~Per-session tunnel creation and handling.~~
- ~~Non-optimal data-path.~~
- **IPv6 integrates networks of the mobile and others.**
- **A SID represents data-plane role and function.**



SRv6 in A Nutshell

SRH (Segment Routing Header)

```
 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  | Segments Left |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Last Entry   |     Flags     |              Tag              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Segment List[0] (128 bits IPv6 address)            |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
                              ...
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Segment List[n] (128 bits IPv6 address)            |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                                                             //
//         Optional Type Length Value objects (variable)       //
//                                                             //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4

Segment ID (SID)

SRv6 in A Nutshell (Cont'd)

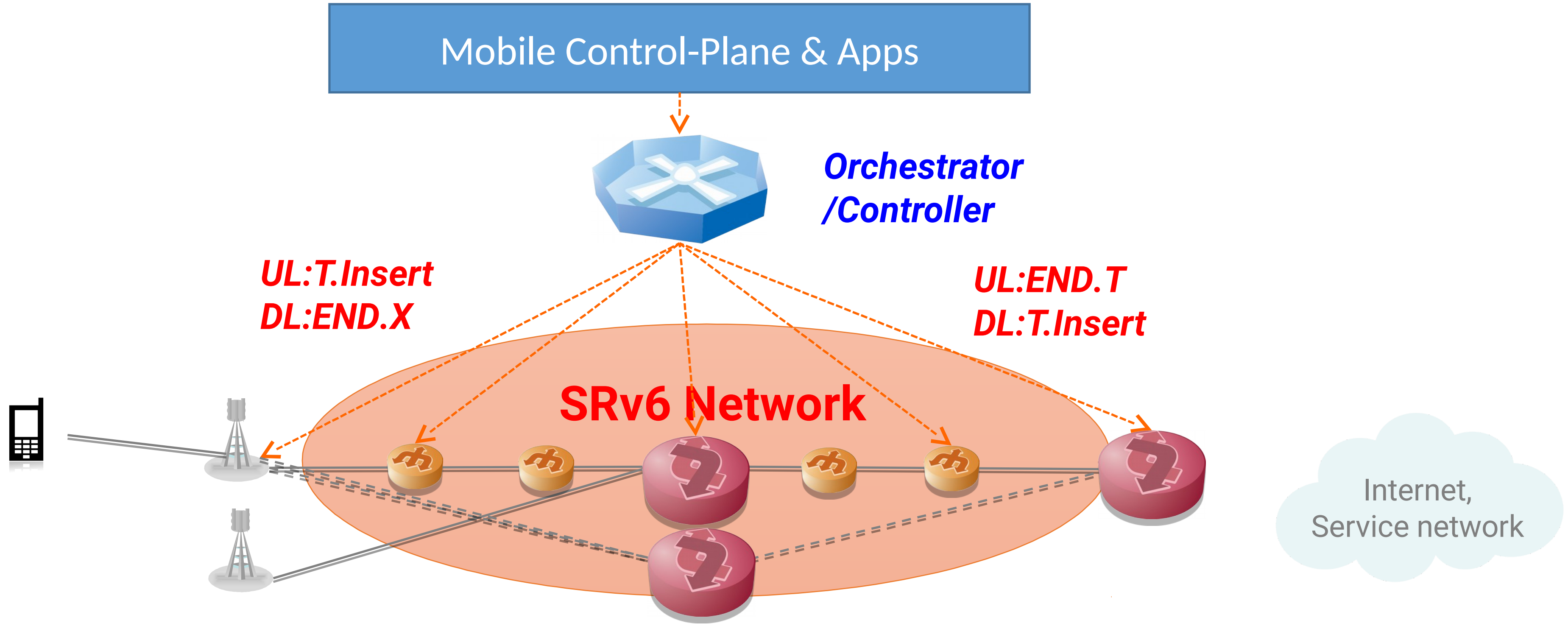
SRv6 Function* Name	Forwarding
END	Lookup SRH
END.X	L3 cross-connect to next-hop
END.T	L3 lookup IPv6 table
END.DT6	Decap outer IPv6 hdr and lookup IPv6 table
END.DT4	Decap outer IPv6 hdr and lookup IPv4 table
END.DX6	Decap outer IPv6 hdr and IPv6 cross-connect
END.DX4	Decap outer IPv6 hdr and IPv4 cross-connect
END.B6	Bound to an SRv6 policy(SID list)

SRv6 Function* Name	Forwarding
T	Pure IPv6 transit
T.Insert	Insert an SRv6 policy (SID list)
T.Encaps	Encap SRv6 policy (SID list) by outer IPv6 hdr

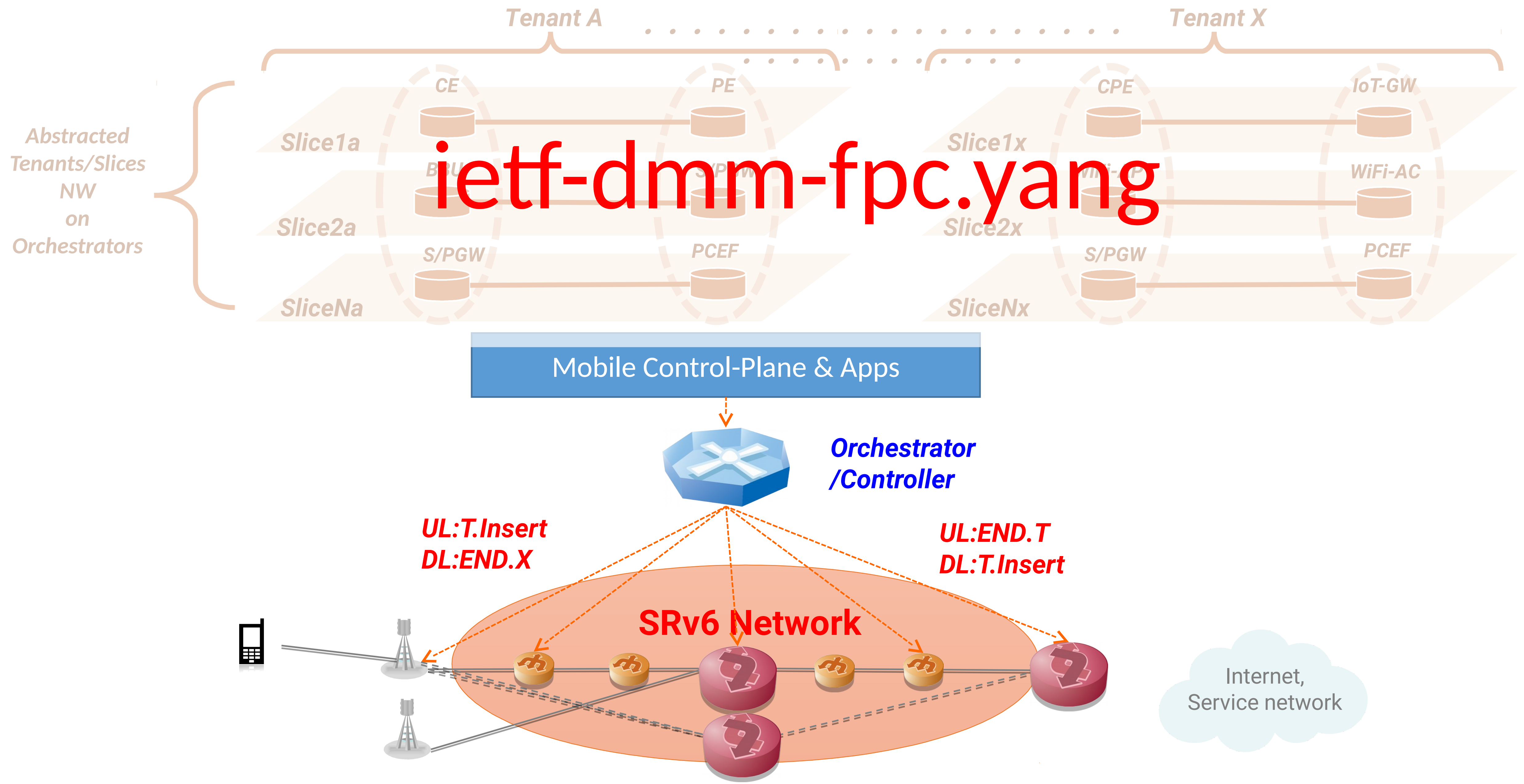
* SRv6 Network Programming

E2E Mobile Orchestration with SRv6

- **Data-plane nodes are NOT dedicated to specific roles.**
  -> SID represents each data-plane role.
- **Orchestrator puts SIDs to the nodes with its functions**
  -> It requires some data models to instantiate the data-plane

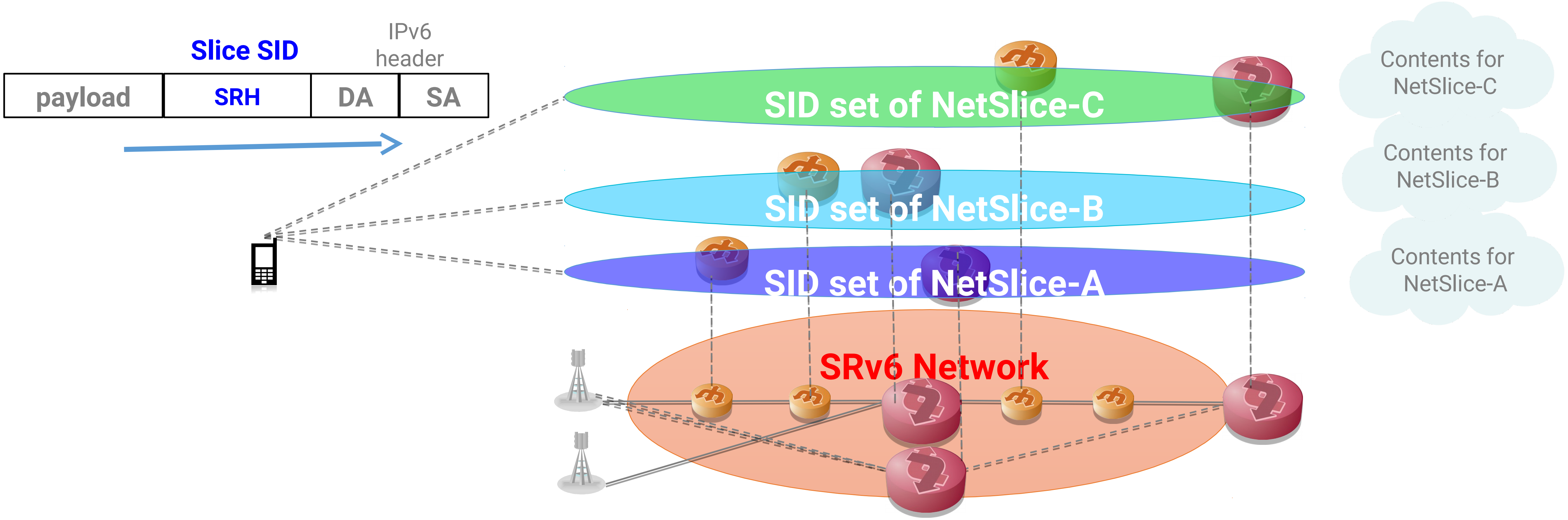


Data Model for Mobile Orchestration with SRv6



SRv6 for Network Slicing

- **A set of SIDs represents Network Slice.**
  -> Sharing same prefix among SIDs in a slice would work.
- **Then user packets could also indicate Slices by SID.**
  -> Applications in a MN could be able to use SID to do that.



References

- IPv6 Segment Routing Header (SRH)
  - draft-ietf-6man-segment-routing-header
- SRv6 Network Programming
  - draft-filsfils-spring-srv6-network-programming
- ietf-dmm-fpc.yang
  - A SDO neutral mobile data-plane model as a part of the FPC work in IETF DMM working group.
  - draft-ietf-dmm-fpc-cpdp

Summary

- **SRv6 is expected to make mobile network to be:**
  - Simple to operate in E2E basis.
  - Flexible where to deploy various functions.

- **SID Functions for mobile data-plane represent:**
  - Access point, L2 Anchor, and L3 Anchor node.
  - Interworking node in stateless manner with some new SRv6 function and parameters.

- **Basic Mode vs. Aggregate Mode**
  - Basic mode works with existing c-plane protocol and interwork with current user-plane.
  - Aggregate mode introduces advanced features of SRv6 to seamless deployment which are service chain, VPNs, TE etc,. with mobility management.

Next Step

- Be a start point for user-plane optimization work?