# Next Steps with draft-ietf-teas-ietf-network-slices

Adrian Farrel (Editor)

adrian@olddog.co.uk

TEAS Virtual Interim : September 27th, 2021

# Main Open Issues

- How do you define a service?
- What is an end-point?
- What is the realisation architecture?
    - What should we call the components of that architecture?

- There are probably plenty of other open issues, but this should be enough to talk about today!

# Network Slice Service – Top-level Definition

- We *almost* have agreement on a definition

> A service provider instantiates an IETF network slice service for a customer. The IETF network slice service is specified in terms of a set of the customer's endpoints ~~(CEs)~~, a set of one or more connectivity matrices (point-to-point (P2P), point-to-multipoint (P2MP), multipoint-to-point (MP2P), or multipoint- to-multipoint (MP2MP)) between subsets of these endpoints~~CEs~~, and a set of SLOs and SLEs for each endpoint~~CE~~ sending to each connectivity matrix. That is, in a given IETF Network Slice Service there may be one or more connectivity matrices of the same or different type, each connectivity matrix may be between a different subset of endpoints~~CEs~~, and for a given connectivity matrix each sending endpoint~~CE~~ has its own set of SLOs and SLEs, and the SLOs and SLEs in each set may be different. However, it is a free choice for a service provider to decide whether to implement a single connectivity matrix per IETF Network Slice Service, or to allow multiple matrices per slice.

- Debate about whether one service can contain multiple connectivity matrices
  - With this definition, an operator or implementer is free to have just one matrix per service
  - We hear some people say that a service may want to include multiple matrices
- But, note well!
  - What is a “connectivity matrix”? …

# Network Slice Service – Connectivity Matrix

- In this definition a connectivity matrix is *not* a network with delivery capabilities
    - Consider a mesh network where a packet may be routed from one endpoint to another endpoint over the mesh
    - That is analogous to a regular L3VPN
- This definition is more like a tunnel or LSP view
    - E.g., a P2MP connectivity matrix means that all packets from the ingress endpoint are delivered to all egress endpoints
        - Note that how an operator realises a P2MP connectivity matrix is entirely up to them
            - Ingress replication, multicast routing, P2MP tunnels, hub-and-spoke

# Network Slice Service – Service Request

- This is the subject of the NBI debate

- A question in the context of the framework document…
  - Should this document spend more time describing the *abstract* definition of a service specification?
    - A bit like an information model
  - Perhaps we are already there with:
    - Endpoints
    - Connectivity matrices
    - SLOs
    - SLEs

# Network Slice Endpoints (1 of 2)

- Debate around “where is an IETF slicing service delivered?”
- A lot of discussion relates this to CEs and PEs
  - This is good, but doesn’t fit with other realisations such as virtual functions
  - Nevertheless, it is useful to discuss in this context

```
    |<---------------------- (1) ---------------------->|
    |                                                   |
    | |<-------------------- (2) -------------------->| |
    | |                                               | |
    | |        |<----------- (3) ----------->|        | |
    | |        |                             |        | |
    | |        | |<--------- (4) --------->| |        | |
    | |        | |                         | |        | |
    V V   AC   V V                         V V   AC   V V
+-----+   |    +-----+                 +-----+    |   +-----+
|     |--------|     |                 |     |--------|     |
| CE1 |   |    | PE1 |. . . . . . . . .| PE2 |    |   | CE2 |
|     |--------|     |                 |     |--------|     |
+-----+   |    +-----+                 +-----+    |   +-----+
   ^              ^                       ^              ^
   |              |                       |              |
   |              |                       |              |
Customer       Provider                Provider       Customer
Edge 1         Edge 1                  Edge 2         Edge 2
```

1) CE is operated by the IETF Network Slice service provider. The edge of the IETF Network Slice may be within the CE.  The slicing process may utilize resources from within the CE such as buffers and queues on the outgoing interfaces.
2) The IETF Network Slice may extend as far as the CE, to include the AC, but not to include any part of the CE.  The CE is operated by the customer or the provider. Slicing the resources on the AC may require the use of traffic tagging (such as through Ethernet VLAN tags) or may require traffic policing at the AC link ends.

# Network Slice Endpoints (2 of 2)

```
    |<---------------------- (1) ---------------------->|
    |                                                   |
    | |<-------------------- (2) -------------------->| |
    | |                                               | |
    | |        |<----------- (3) ----------->|        | |
    | |        |                             |        | |
    | |        | |<--------- (4) --------->| |        | |
    | |        | |                         | |        | |
    V V   AC   V V                         V V   AC   V V
+-----+   |    +-----+                 +-----+    |   +-----+
|     |--------|     |                 |     |--------|     |
| CE1 |   |    | PE1 |. . . . . . . . .| PE2 |    |   | CE2 |
|     |--------|     |                 |     |--------|     |
+-----+   |    +-----+                 +-----+    |   +-----+
   ^              ^                       ^              ^
   |              |                       |              |
   |              |                       |              |
Customer       Provider                Provider       Customer
Edge 1         Edge 1                  Edge 2         Edge 2
```

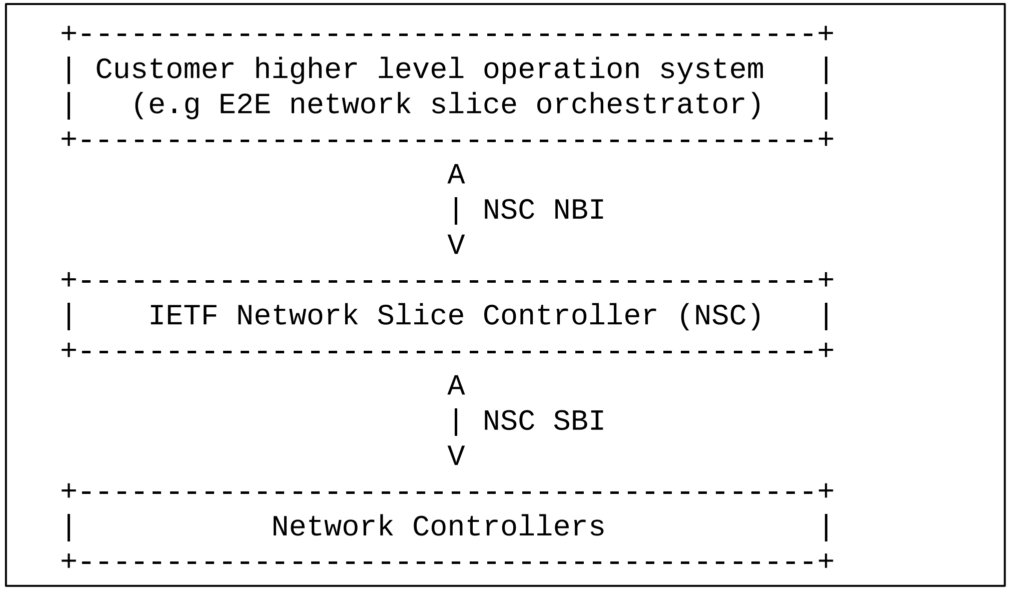
3) The endpoints of the IETF Network Slice are the customer-facing ports on the PEs. Managed in a way similar to a port-based VPNs: each port (AC) or virtual port (e.g., VLAN tag) identifies the IETF Network Slice and maps to an IETF Network Slice endpoint.

4) The endpoint of the IETF Network Slice may be within the PE. The PE classifies the traffic coming from the AC according to information (such as the source and destination IP addresses, payload protocol and port numbers, etc.) in order to place it onto an IETF Network Slice.

- Conclusions
  - We should embrace all four endpoint locations
  - We should not talk about CE/PE (except as examples)
  - We should use terms such as
    - Slice customer edge
    - Slice provider edge

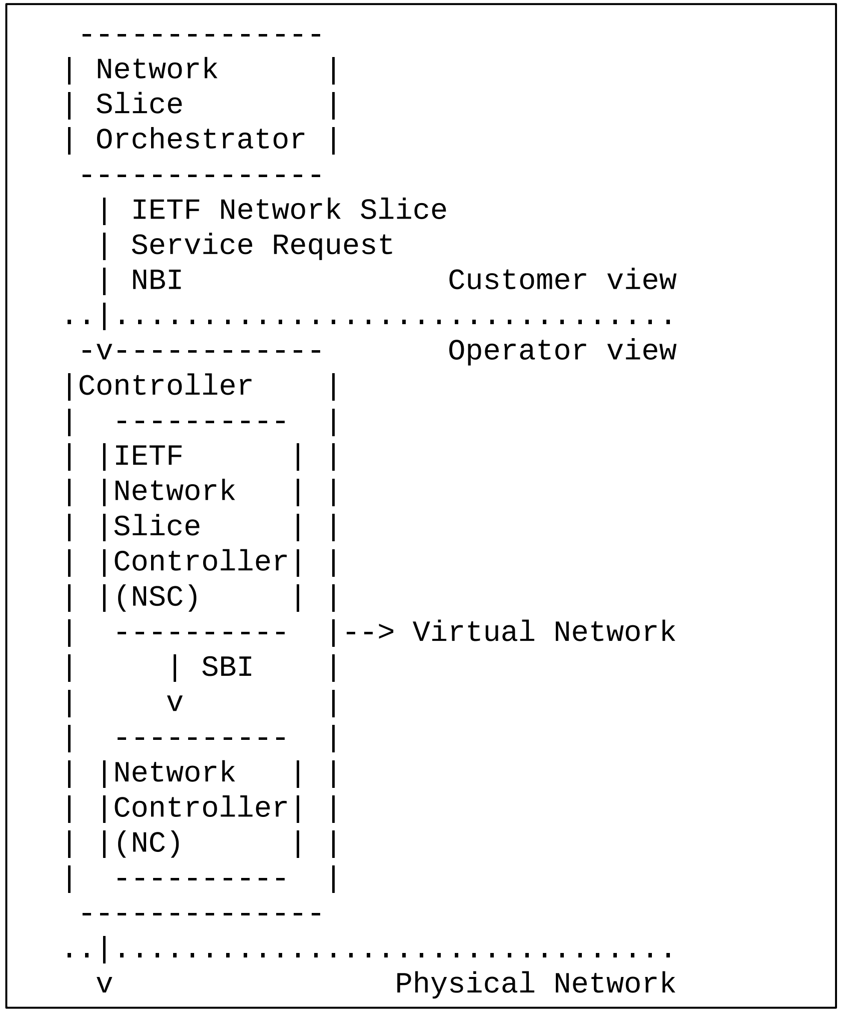
# Management Architecture

- I think we have this covered
  - See also draft-barguil-teas-network-slices-instantation



```
+-------------------------------------------+
| Customer higher level operation system    |
|   (e.g E2E network slice orchestrator)    |
+-------------------------------------------+
                      A
                      | NSC NBI
                      V
+-------------------------------------------+
|    IETF Network Slice Controller (NSC)    |
+-------------------------------------------+
                      A
                      | NSC SBI
                      V
+-------------------------------------------+
|            Network Controllers            |
+-------------------------------------------+
```

From an implementation point of view this probably looks like this



```
 --------------
| Network      |
| Slice        |
| Orchestrator |
 --------------
  | IETF Network Slice
  | Service Request
  | NBI               Customer view
..|..................................
 -v------------       Operator view
|Controller    |
|  ----------  |
| |IETF      | |
| |Network   | |
| |Slice     | |
| |Controller| |
| |(NSC)     | |
|  ----------  |--> Virtual Network
|     | SBI    |
|     v        |
|  ----------  |
| |Network   | |
| |Controller| |
| |(NC)      | |
|  ----------  |
 --------------
..|..................................
  v                Physical Network
```

# Realisation Architecture – The problem

- Aim for a common and generic architecture
  - Terminology and components we can all agree on
  - Something for technology solutions to reference
    - A solution can use its own terminology and map to this reference model
    - A solution can use this common terminology
- The challenge has been that solutions work is quite advanced
  - Some “entrenching” on terminology and models
  - Need to step back and take a broader, functional view
- Some good offline discussions between authors of
  - draft-ietf-teas-enhanced-vpn and related drafts
  - draft-bestbar-teas-ns-packet and related drafts
  - ➢Progress made on understanding the concepts
  - ➢Still not reached any firm conclusions

# Realisation Architecture: Things We Can Agree On

- There may be a large number of network slice services
    - The operator cannot micro-manage the network for all these slices
    - Some form of grouping of slice services is required
        - According to common properties
    - Each service needs to be mapped to a set of network resources that support the group
- So far, so good, but
    - Do we need to break out different flows from within a slice service and map them onto different sets of network resources?

# Realisation Architecture: Questions to discuss

- How do we decide which set of resources to use to support a group?
- Do we build that set…
  - In advance as a managed profile of the network
  - On-demand as new classes of slice service arrive
  - Dynamically to flexibly support each new slice service
- Probably we have to be flexible to all of these approaches
- 1:1, n:1, n:m mapping of service to resource set?
  - n:1 seems to be the obvious, scaling solution
  - What would n:m mean and why would we do it?
    - But it is only a function of mapping/classification and that is an implementation choice
- What do we call our constructs?
  - Before we dive in, recall that a slice service is a (set of) “connectivity matrix”

# Realisation Architecture : Names

The network slice (service) as viewed by the customer.

Controller programs the operations at each stage.

The architecture must not care about the network technology or how it is operated.

```
                    --      --      --
                   |CE|    |CE|    |CE|
                    --      --      --
                  AC :    AC :    AC :
                  ----------------------        -------
                 ( |PE|....|PE|....|PE| )      ( IETF  )
                (   --:     --     :--   )    ( Network
                (     :............:     )    (  Slice  )
                 (  IETF Network Slice  )      (       )  Customer
                  ----------------------        -------     View
               ........................\........./..................
                                        \       /        Controller
       >>>>>>>>>>>>>>>  Grouping/Mapping v     v            View
      ^             -----------------------------------------
      ^            ( |PE|.......|PE|........|PE|.......|PE|  )
 ------------     (   --:        --         :--         --    )
|            |    (     :...................:                 )
| Controller |     (        Network Resource Partition       )
|            |      -----------------------------------------
|            |                             ^
|            |>>>>>  Resource Partitioning |
 ------------        of Available Topology |
  v   v                                    |
  v   v            -----------------------------      --------
  v   v           (|PE|..-..|PE|... ..|PE|..|PE|)    (        )
  v   v          ( :--  |P|  --   :-:  --   :--  )  (  Filter  )
  v   v          ( :.-   -:.......|P|       :-   )  ( Topology )
  v   v          (  |P|...........:-:.......|P|  )   (        )
  v   v           (  -    Filter Topology       )     --------
  v   v            -----------------------------       ^
  v    >>>>>>>>>>>>  Topology Filter ^                /
  v            .......................\............../..............
  v                                    \            /  Physical Network
  v            ------------------------------------------------
  v           ( |PE|.....-.....|PE|.......    |PE|.......|PE|  )
  v          (   --     |P|     --       :-...:--     -..:--    )
   >>>>>>>>>(    :       -:..............|P|.........|P|         )
 Program    (    -.......................:-:..-       -          )
 the Network (  |P|..........................|P|......:         )
              (  -                            -                )
               ------------------------------------------------
```

Set of network resources available to support a group of network slice services.

Optional stage like coloured graphs in multi-topology, or like clustering all available resources with a specific property.

# Realisation Architecture : Work Flow

- Not sure this is needed
  - I used to think it would be helpful
- Now appears that the operations can take place in any order
  - For example…
  - Build a resource partition before any services are requested
  - Program the network (e.g., SR policies, TE-LSPs) in advance
  - React to a slice service request as provision the network accordingly
  - Etc.