Unified YANG Model for OAM

draft-tissa-lime-yang-oam-model
IETF 91
Hawaii 2014

To Keep in mind

- Proposal here is to Use the management framework of CFM, not the protocol aspects of CFM.

Bit of History

- Started as a discussion among few people
  - Many different OAM protocols
  - Costing money to Operators to train people
  - Troubleshooting is cumbersome
  - Hard to provide end-end SLA when multiple layers are involved
- Two schools of Thoughts
  - Develop a brand new universal OAM protocol
  - Create a management framework that encapsulate complexities and provide a uniform framework
    - draft-tissa-lime-yang-oam-model follow this.
    - Work pre-dates to LIME/TIME

Goal

- Create a YANG model that provides set of unified interfaces across various OAM technologies.
- Facilitate “nested OAM” across layers that deploys different OAM technologies.
- Resulting in:
  - Programmability for monitoring and troubleshooting
  - Reduced OpEx

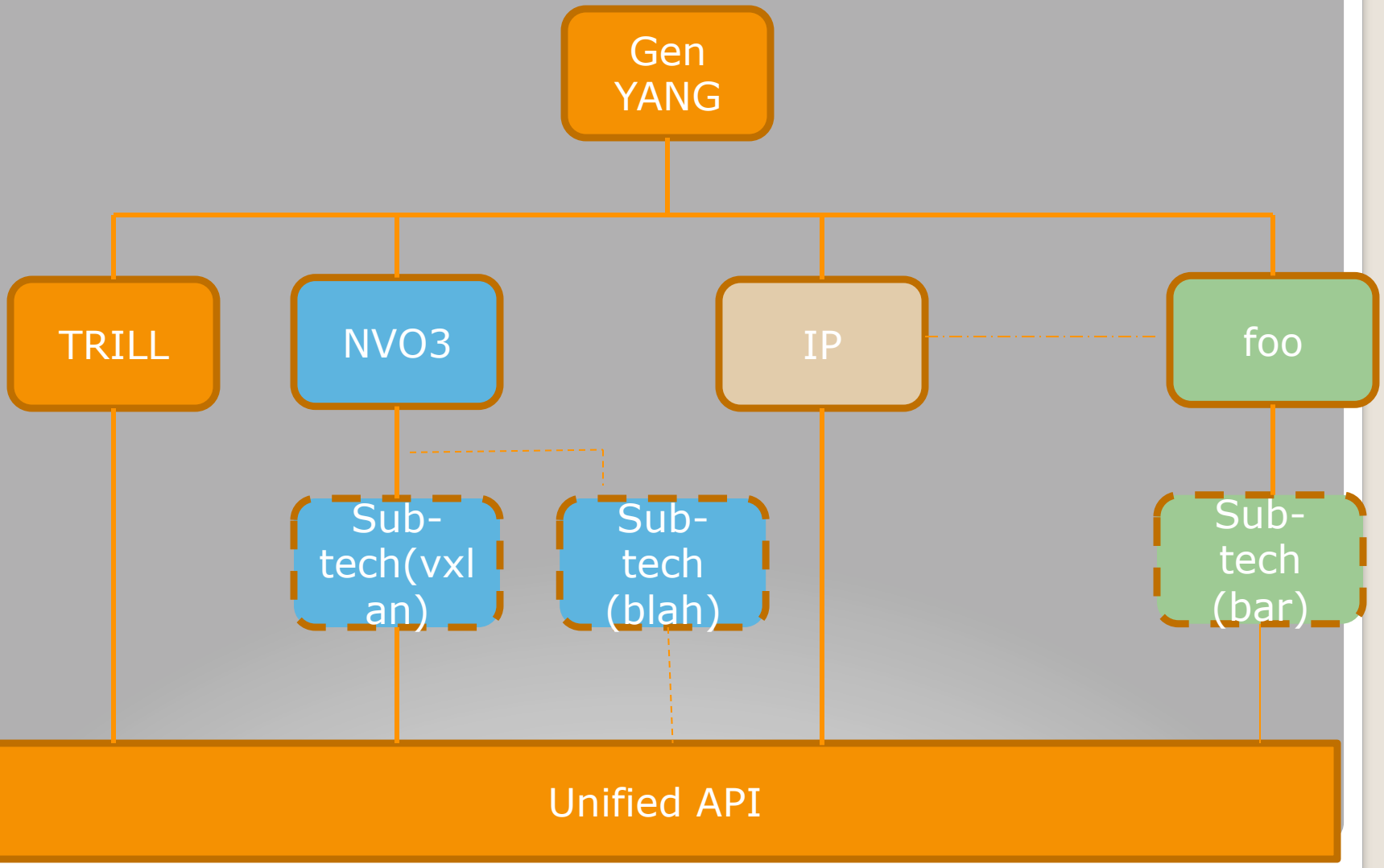
Architecture

- Top Level YANG Model provides the overall framework and setup the structures
- Separate YANG model for each technology area that
  - augments the top level model to include technology specific aspects

Application

- draft-tissa-lime-yang-oam—model – presents the top level YANG model
- draft-ietf-trill-yang-oam-00 and draft-tissa-nvo3-yang-oam provide extensions for TRILL and NVO3

YANG Model


Gen YANG
TRILL
NVO3
IP
foo
Sub-tech(vxlan)
Sub-tech (blah)
Sub-tech (bar)
Unified API

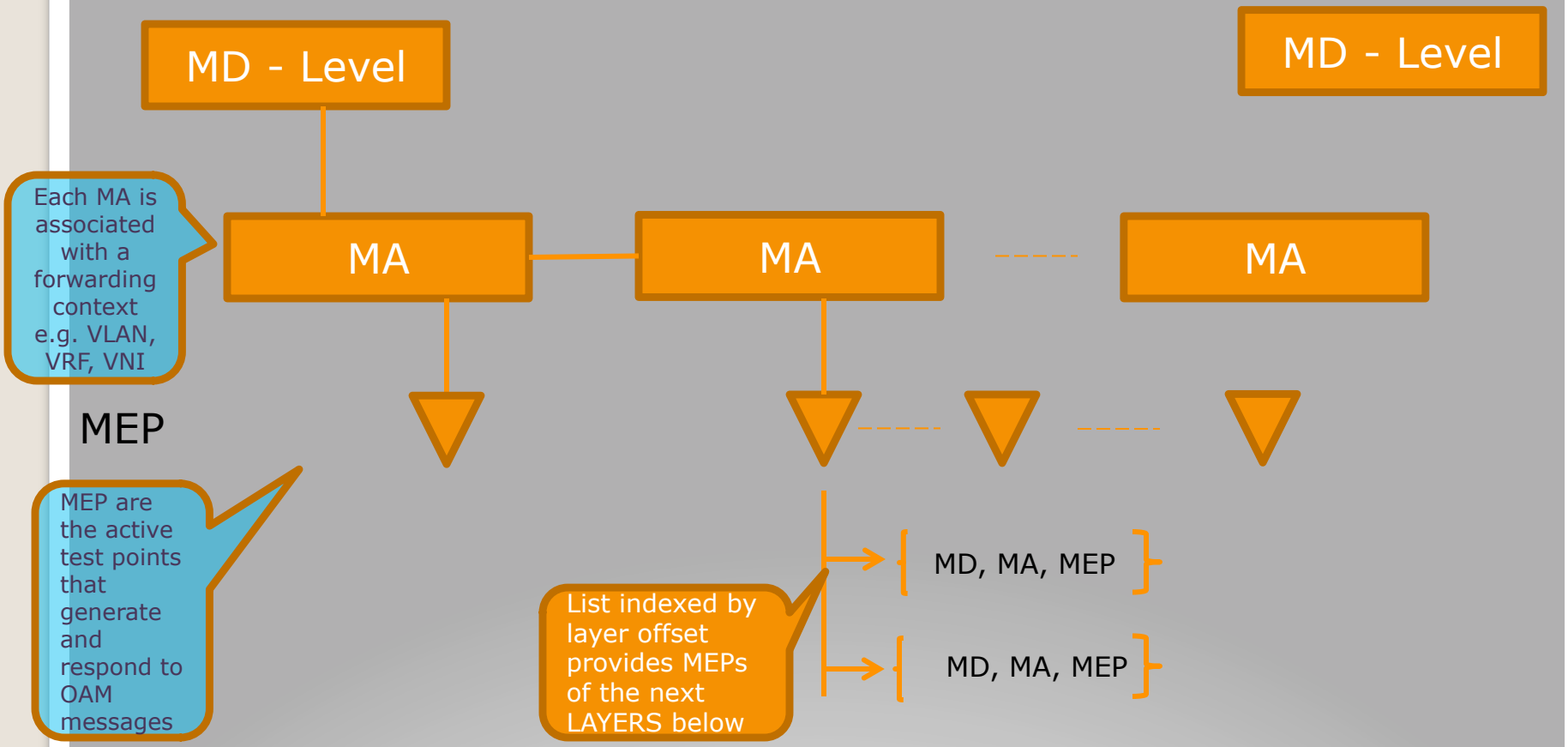
Choice of CFM Model

- CFM model is chosen as the base model for OAM
  - It perhaps is the most well defined OAM model
- Key Takeaway:
  - It is not the CFM protocol that is used; but the Framework of CFM,
- E.g. MPLS-TP (RFC 6371) use a Framework similar to CFM, yet underlying OAM protocol is different

Gen YANG Model

- Designed with extensibility in mind, so that every node is extensible by corresponding technologies
- Gen YANG models will serve as an umbrella that ensure consistency across different OAM technologies

Flow of the Model



More details

```
module: gen-oam
  +--rw domains
    +--rw domain* [technology MD-name-string]
      +--rw technology          identityref
      +--rw MD-name-string      MD-name-string

      +--rw MD-name-format?     identityref
      +--rw (MD-name)?
      |    +--:(MD-name-null)
      |     +--rw MD-name-null?      empty
      +--rw md-level            MD-level
      +--rw MAs
        +--rw MA* [MA-name-string]
          +--rw MA-name-string        MA-name-string

          +--rw MA-name-format?       identityref
          +--rw (MA-name)?
          |  +--:(MA-name-null)
          |     +--rw MA-name-null?         empty

          +--rw (connectivity-context)?
          |  +--:(context-null)
          |     +--rw context-null?         Empty
    .
    .
          +--rw (flow-entropy)?
          |  +--:(flow-entropy-null)
          |     +--rw flow-entropy-null?    empty
      .
      .
```

Designed with extensibility; allow different technologies add customization

MD Name string, generic and always present

Technology specific extensions
To accommodate different formats

MA Name string, generic and always present

Technology specific extensions
To accommodate different formats

Where forwarding paradigm define
e.g. VLAN, VRF etc..

Flow entropy to support multipathing, e.g. TRIL, nvo3, MPLS

More details

```
module: gen-oam
  +--rw domains
    +--rw domain* [technology MD-name-string]
      +--rw technology        identityref
      +--rw MD-name-string    MD-name-string
.
    .
    .

            +--rw MEP* [mep-id]
            |  +--rw mep-id                    MEP-id
            |  +--rw mep-name?                 string
            |  +--rw mep-direction             MEP-direction
            |  +--rw (mep-address)?
            |  |  +--:(mac-address)
            |  |  |  +--rw mac-address?       yang:mac-address
            |  |  +--:(ipv4-address)
            |  |  |  +--rw ipv4-address?      inet:ipv4-address
            |  |  +--:(ipv6-address)
            |  |     +--rw ipv6-address?      inet:ipv6-address
```

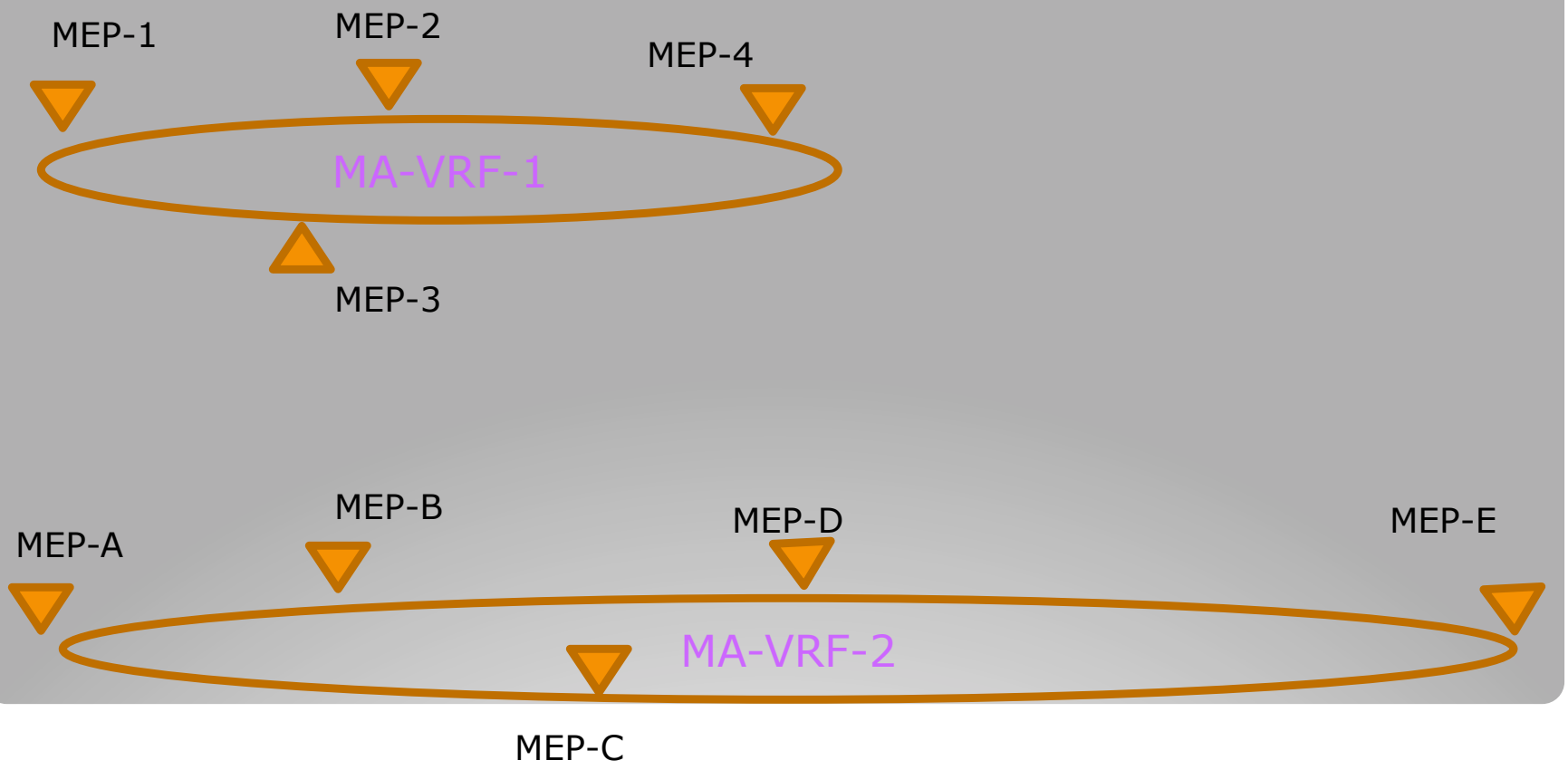
Application to IP

- Many overlay protocols (nvo3, LISP, SFC etc..) use IP as the primary underlay transport.
- IP Lacks a proper fault model.
- This leads to challenges and inefficiencies in network operations and troubleshooting of overlay protocols
  - Unable to map a fault to the underlay
  - Unable to create programmatic interfaces to undelay to automate fault isolation, fault verification and performance measurement.

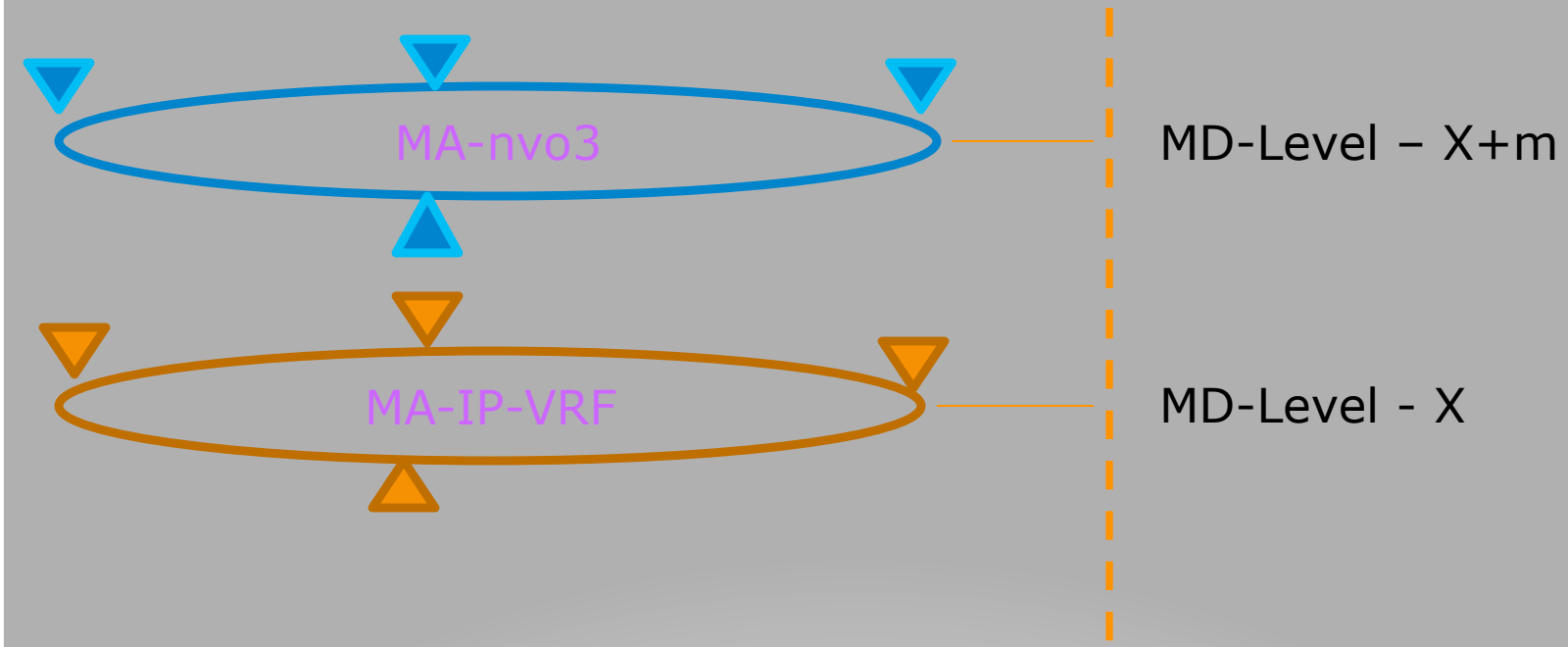
Application to IP

- Consider each VRF as a separate MA
- MA is defined as fault domain where protocol under consideration has reachability
  - E.g. VLAN or VRF
- Each device in the domain has a MEP at the logical interface of the VRF
  - One could even have at individual interface level.
- OAM messages are generated, received and processed through these MEP

Application to IP



Nested OAM
E.g. Interaction of nvo3 and IP



E.g. Application to NVO3

- We augment the nodes with NVO3 specifics
  - Flow-entropy , MEP addressing, context-id

Application to NVO3 continued

```
identity nvo3 {
      base  goam:technology-types;
      description "nvo3 type";
 }
```

```
augment "/goam:domains/goam:domain/goam:MAs/goam:MA" {
        leaf technology-sub-type {
              type identityref {
                base technology-sub-type;
              }
        }
    }
```

Augment based on technology

```
augment "/goam:domains/goam:domain/goam:MAs/goam:MA/goam:flow-
  entropy" {
      case flow-entropy-vxlan {
        leaf flags-vxlan {
          when "/goam:domains/goam:domain/goam:technology-sub-type='vxlan'";
          type flags-vxlan;
        }
        leaf flow-entropy-vxlan {
          when "/goam:domains/goam:domain/goam:technology-sub-type='vxlan'";
          type flow-entropy-vxlan;
        }
      }
    }
```

```
augment "/goam:domains/goam:domain/goam:MAs/goam:MA/goam: connectivity-context "
    {
      case context-id-nvo3 {
        leaf vni {
          when "/goam:domains/goam:domain/goam:technology-type='nvo3'";
          type vni;
        }
      }
    }
```

MEP Addressing for NVO3

Proposal - 1

MEP address is a list [Dst IP , VNI]

Proposal - 2

Within NVE multiple VNI

NVE can have multiple IP addresses, shared between multiple VNI

Overlay MEP: VNI

Underlay MEP:IP

(ref:draft-ietf-nvo3-arch)

Preferred

References

- High level YANG model
  - https://tools.ietf.org/html/draft-tissa-lime-yang-oam-model-02
- Technology dependent
  - TRILL: https://tools.ietf.org/html/draft-ietf-trill-yang-oam-00
  - NVO3: http://tools.ietf.org/html/draft-tissa-nvo3-yang-oam-00

Q&A and Next Steps