MUP Architecture for DMM

draft-ietf-dmm-mup-architecture-00

IETF123, DMM Working Group

Satoru Matsushima (SoftBank) on behalf of co-authors:

Katsuhiro Horiba
Yuya Kawakami (SoftBank)
Tetsuya Murakami, Keyur Patel (Arrcus)
Jakub Horn (Cisco)

Major Discussions During the WG Adoption Call

- MUP-C API related discussions

1. MUP-C API Text Syntax
2. MUP-C API Candidates

Discussion 1

- **MUP-C API Text Syntax**

  In §6.3

  > "A MUP Controller (MUP-C) provides an API. A consumer of the API inputs session information for a UE or a MN from mobile service system.

  > I am not sure that the MUP-C does 'provide' an API as much as
  > consumes an existing API from the mobile packet core, e.g. N4 to
  > provide GTP<->UE mapping.

  > Terminology-wise: MUP-C and mobile communication system control plane need to share
  > an API. So, IMO, 'consume' or 'leverage' is a good term.

Discussion 2

- **MUP-C API Candidate (1)**

  > But I don't think we should specify N4 as an implementation may
  > decide to directly integrate to the SBA over Nsmf or some future API.
  > My suggestion is:

  > "A MUP Controller (MUP-C) consumes an existing API from the 5G
  > Packet Core which provides it with UE and/or MN session
  > information."

Discussion 2 (Cont'd)

- **MUP-C API Candidate (2)**

  > Technically, the draft focuses on what happens southbound to the MUP-C. For the API between the
  > MUP-C and the mobile communication system control plane I see another draft applies here and
  > can complement the MUP architecture draft:

  https://datatracker.ietf.org/doc/draft-liebsch-dmm-mts/

  > The current revision defines a clear scope which is the API between (i) transport controller (or a
  > dedicated application built on top), such as the MUP-C or the Mobile Traffic Steering Controller the
  > mentioned draft, and the mobile communication system's control plane.

  > The same API may apply in between the a routing function and the mobile communication system's
  > user plane anchor.

  > The current API semantics (which is under development) can be enriched with session information
  > where the MUP-C derives routing information from.

Author’s reply to MUP-C API Discussions

- MUP-C API Text Syntax
  - The purpose of MUP-C API is to get/retrieve session information;
  - to achieve above that, both the providing/consuming or leveraging new/existing APIs should be available options to MUP-C;
  - the next revision clarifies that.

- MUP-C API Candidates
  - MUP Architecture is independent from specific mobile service architecture;
  - The MUP Architecture should be able to use any possible existing or new APIs;
  - another draft can describe a specific MUP-C API.

MUP Architecture

RECAP: What is the MUP Architecture?

- Independent from specific mobile service architecture
  - MUP is consumable by any mobile service architectures
- Pluggable User Plane part of mobile service architecture
  - MUP can be co-exist with genuine user plane defined in each mobile service architecture
- Independent from specific dataplane protocols
  - MUP can be implemented with any dataplane protocols
  - SRv6 MUP is a MUP architecture implementation by using SRv6
    - GTP-U/SRv6 translation behaviors set defined in RFC9433 is one of the options.

RECAP: Architecture Principles (1)

- “Segment” defined in MUP Architecture
  - Defined as the mobile user plane abstraction in MUP architecture
  - Two types of MUP Segment defined in the draft (Section 4):
    - Interwork Segment
    - Direct Segment

  - NOTE: “Segment” can be seen in many technologies, and is defined in each technology context:
    - “Segment” in Ethernet
    - “Segment” in (Multi-Segment) Pseudo-Wire (RFC5659)
    - “Segment” in EVPN (RFC7432)
    - “Segment” in Segment Routing Architecture (RFC8402)
    - “Segment” in Network Slicing (RFC9543)
    - “Segment” in MUP Architecture (This I-D)

RECAP: Architecture Principles (2)

- MUP Segment Auto-Discovery
  - MUP PE should be able to discover MUP Segments in the remote MUP PEs
    - MUP PE advertises auto-discovery route for hosted MUP Segments

  - MUP Segment Auto-Discovery Route (Section 5)
    - Interwork Segment Discovery (ISD) route
    - Direct Segment Discovery (DSD) route
    - Dataplane independent
    - Any dataplane specific attributes should be able to attach to the auto-discovery route
      - Service-SID in BGP Prefix-SID Attribute for SRv6MUP
      - BGP Tunnel-encap Attribute allows many dataplane options

BGP Tunnel Encapsulation Attribute Tunnel Types

Registration Procedure(s)
First Come First Served
Reference
[RFC9012]
Available Formats

csv

Value	Description	Reference
0	Reserved	[RFC9012]
1	L2TPv3 over IP	[RFC9012]
2	GRE	[RFC9012]
3	Transmit tunnel endpoint (DEPRECATED)	[RFC9012]
4	IPsec in Tunnel-mode (DEPRECATED)	[RFC9012]
5	IP in IP tunnel with IPsec Transport Mode (DEPRECATED)	[RFC9012]
6	MPLS-in-IP tunnel with IPsec Transport Mode (DEPRECATED)	[RFC9012]
7	IP in IP	[RFC9012]
8	VXLAN Encapsulation	[RFC8365]
9	NVGRE Encapsulation	[RFC8365]
10	MPLS Encapsulation	[RFC8365]
11	MPLS in GRE Encapsulation	[RFC8365]
12	VXLAN GPE Encapsulation	[RFC8365]
13	MPLS in UDP Encapsulation	[RFC7510][RFC Errata 4350]
14	IPv6 Tunnel	[Martin_Djernaes]
15	SR TE Policy Type	[draft-ietf-idr-segment-routing-te-policy]
16	Bare	[Nischal_Sheth]
17	SR Tunnel (DEPRECATED)	[RFC9125]
18	Cloud Security	[Ramesh_Babu_Yakkala]
19	Geneve Encapsulation	[RFC8926]
20	Any-Encapsulation	[draft-ietf-bess-bgp-multicast-controller-06]
21	GTP Tunnel Type	[Keyur_Patel][Tetsuya_Murakami]
22	Dynamic Path Selection (DPS) Tunnel Encapsulation	[Venkit_Kasiviswanathan]
23	Originating PE (OPE)	[draft-heitz-bess-evpn-option-b-01]
24	Dynamic Path Selection (DPS) Policy	[https://eos.arista.com/eos-4-26-2f/dps-vpn-scaling-using-bgp][Sarah_Chen]
25	SDWAN-Hybrid	[draft-ietf-idr-sdwan-edge-discovery-04]
26	X-over-UDP	[Jeffrey_Haas]
27-65535	Unassigned	

https://www.iana.org/assignments/bgp-tunnel-encapsulation/bgp-tunnel-encapsulation.xhtml

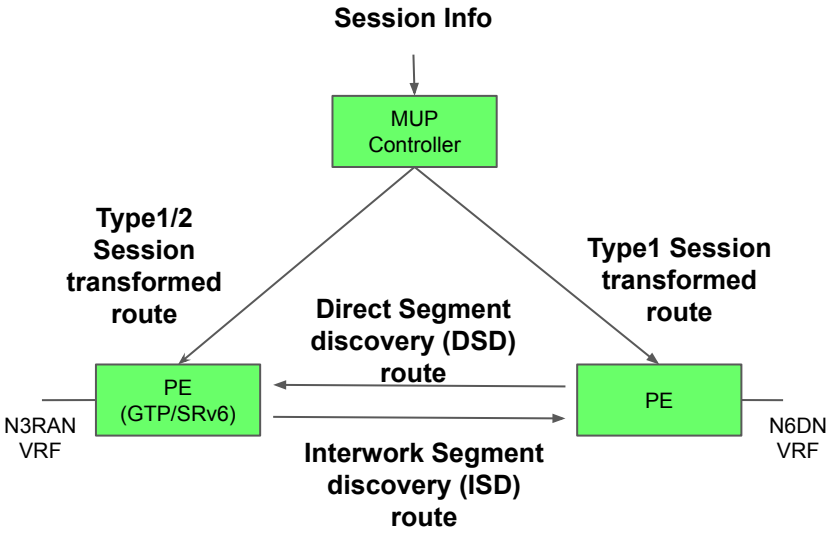
RECAP: Architecture Principles (3)

- Transforms Session to Routing
  - To leverage dataplane protocol to optimize the datapath for DMM

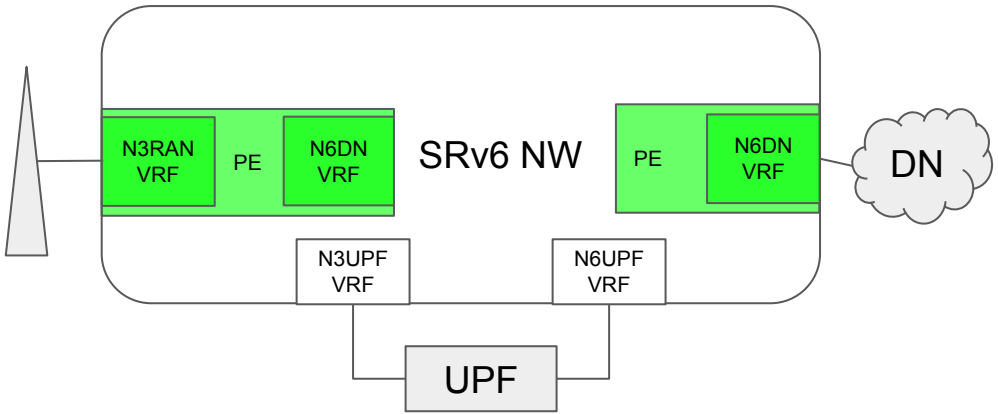
- Session Transformed (ST) Route (Section 6)
  - MUP Controller (MUP-C) advertises following ST routes:
    - Type1 ST (ST1) Route
    - Type2 ST (ST2) Route
  - Auto-discovery route should be used to resolve reachability for the ST routes
  - ST routes should be dataplane independent as well

MUP architecture with 3GPP 5G over SRv6 implementation

Control Plane (BGP)



Data Plane (in 3GPP 5G case)



Reminder: DMM Requirements (RFC7333)

MUP Architecture Principles guide DMM design and operation to fulfill the Reqs

- **REQ1: Distributed mobility management**
  - *IP mobility, network access solutions, and forwarding solutions provided by **DMM MUST enable traffic to avoid traversing a single mobility anchor far from the optimal route**. It is noted that the requirement on distribution **applies to the data plane only**.*
- **REQ3: IPv6 deployment**
  - *DMM solutions **SHOULD target IPv6 as the primary deployment environment** and **SHOULD NOT be tailored specifically to support IPv4**, particularly in situations where private IPv4 addresses and/or NATs are used.*
- **REQ4: Existing mobility protocols**
  - *A DMM solution **MUST first consider reusing and extending IETF standard protocols** before specifying new protocols.*
- **REQ5: Coexistence with deployed networks/hosts and operability across different networks**
  - *A DMM solution may require loose, tight, or no integration into existing mobility protocols and host IP stacks. Regardless of the integration level, DMM implementations **MUST be able to coexist with existing network deployments, end hosts, and routers** that may or may not implement existing mobility protocols. Furthermore, **a DMM solution SHOULD work across different networks, possibly operated as separate administrative domains**, when the needed mobility management signaling, forwarding, and network access are **allowed by the trust relationship between them**.*

FAQ

Q. Isn't RFC9433 SRv6 MUP itself?

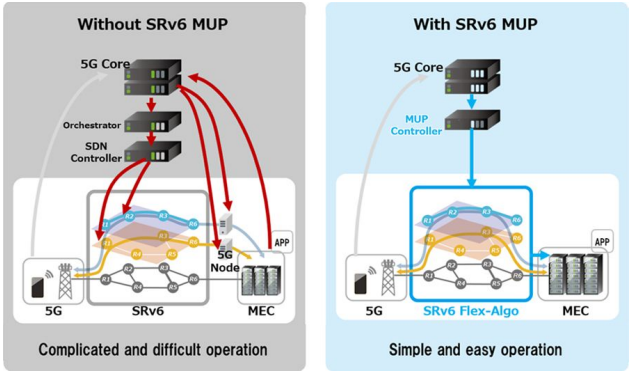
A. RFC9433 defines SRv6 Mobile User Plane behaviors ***under the 3GPP 5G architecture,*** which is not adopted by 3GPP.

Q. What does RFC9433 contribute MUP Architecture?

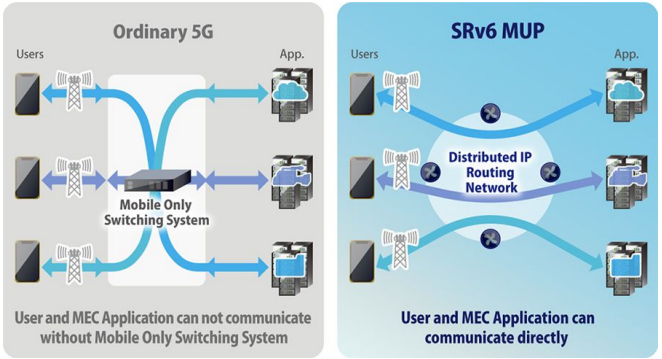
A. As MUP Architecture designed as pluggable User Plane for any mobile service systems, RFC9433 provides a dataplane option in SRv6 case ***under the MUP architecture.***

What the MUP architecture are providing for?

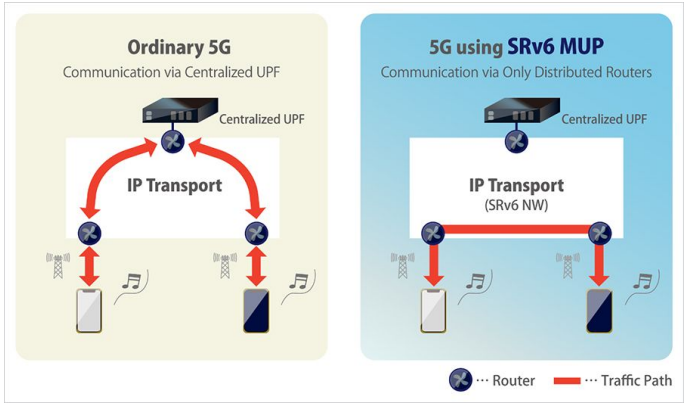
E2E Network Slicing w/o hussles



Optimal datapath for MEC



Optimal datapath for D2D



SRv6 MUP Field Trial Launched in SoftBank's Commercial 5G

SoftBank | About Us

News | Company Info | Vision and Strategy | Investor Relations | Sustainability

SoftBank Corp. Starts SRv6 MUP Field Trial in Commercial 5G Network

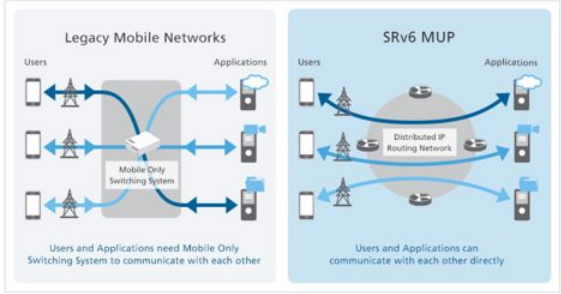
February 24, 2023
SoftBank Corp.

SoftBank Corp. ("SoftBank") today announced that it has started a field trial of Segment Routing IPv6 Mobile User Plane (SRv6 MUP[*1]) in its commercial 5G Network. The trial started on February 13, 2023. SoftBank-led SRv6 MUP can deliver 5G Multi-access Edge Computing (MEC) and network slicing with unprecedented cost-efficiency and operational advantages over legacy technologies. SoftBank is pursuing an accelerated introduction of SRv6 MUP in its commercial services.

SoftBank will exhibit an advanced demonstration of MEC and network slicing on SRv6 MUP, proving the operational ease of delivering network slicing to MEC when SRv6 MUP is used instead of conventional mobile user plane (U-plane) technologies. The demo will be hosted in the booth[*2] of Arrcus Inc. at Mobile World Congress Barcelona (MWC Barcelona) in Spain from February 27 to March 2, 2023. MWC Barcelona is the largest and most influential event for the connectivity ecosystem.

SoftBank will continue the development of SRv6 MUP to deliver highly efficient MEC and network slicing in collaboration with other operators and different industry players around the world.

Image of SRv6 MUP



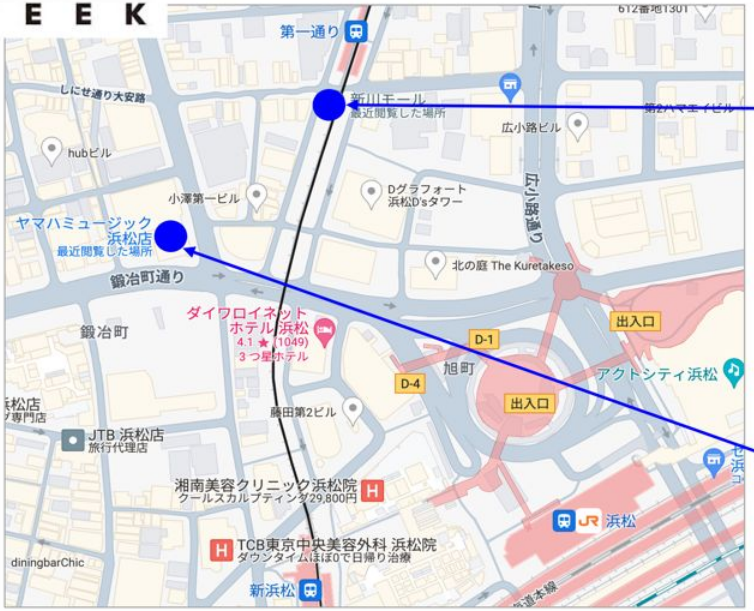
https://www.softbank.jp/en/corp/news/press/sbkk/2023/20230224_01/

SoftBank

Remote Ensembles:
Professional Musicians Played in a Public Music Event

HAMAMATSU JAZZ WEEK

5G SoftBank × SYNCROOM



SRv6MUP
Disabled/Enabled 5G

SoftBank 光

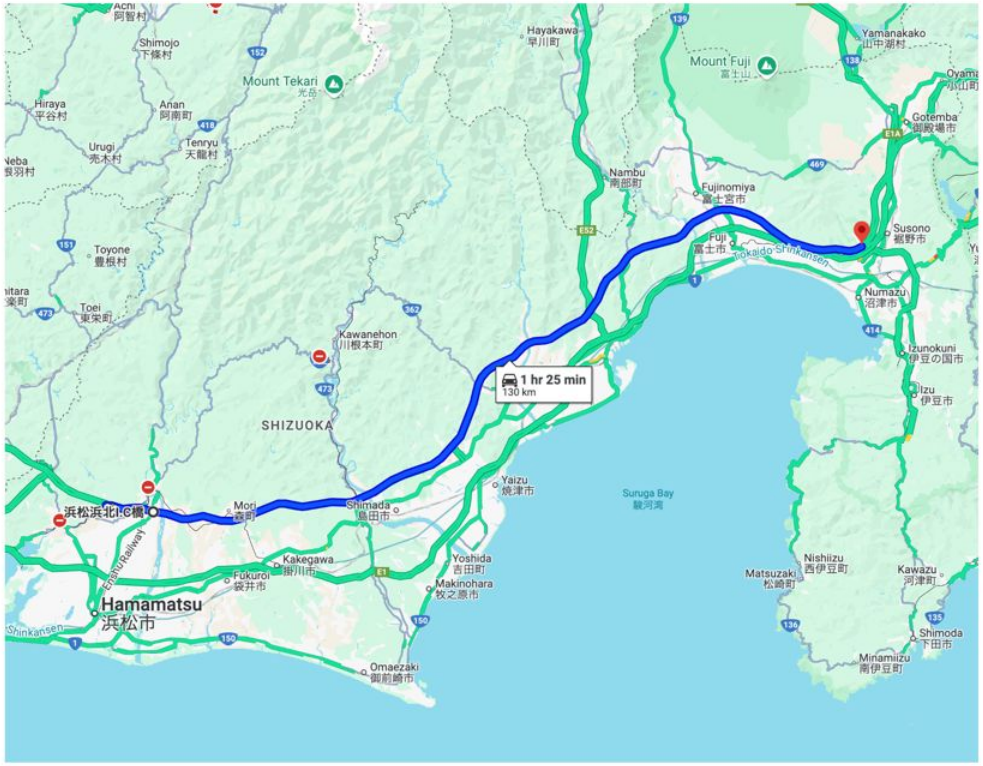
FTTH Internet

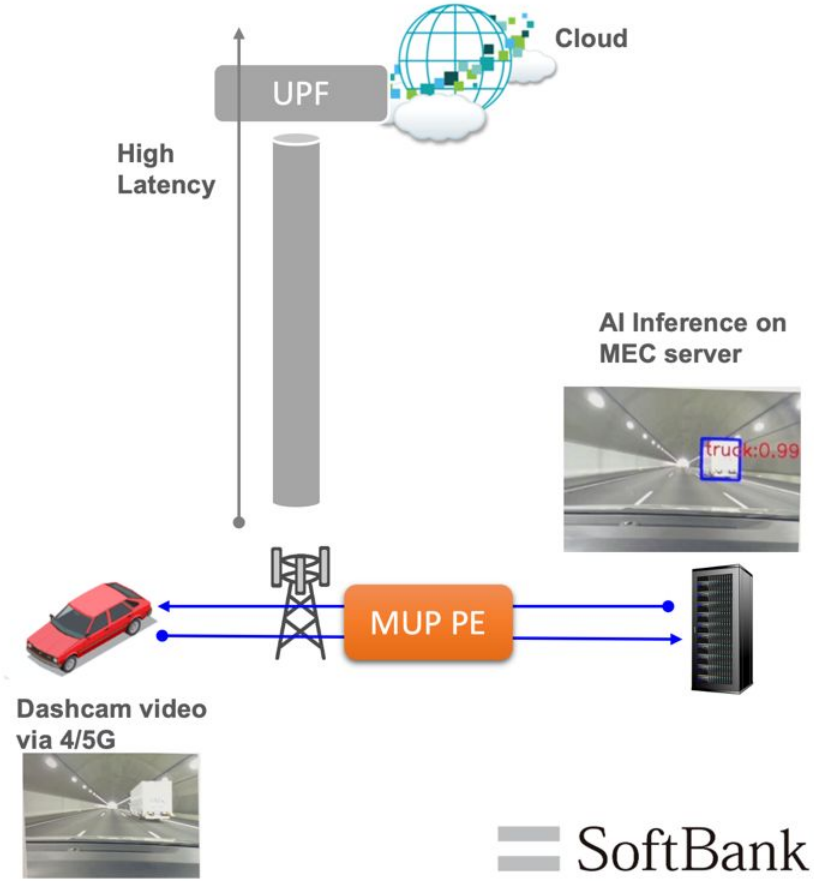
Instruments:
Piano, Saxophone

Instruments:
Drums, E.Guiter

SoftBank

V2N Trial: AI Dashcam w/ MEC servers
through SRv6 MUP enabled 4/5G





SoftBank

BGP-MUP Interop Hackathon at IETF116

ARRCUS NETWORK DIFFERENT

BBSakuraNetworks

cisco

古河電工 FNC

SoftBank

GoBGP

exa

IETF Hackathon

BGP-MUP SAFI Implementation and Interop

IETF 116

25-26 March 2023

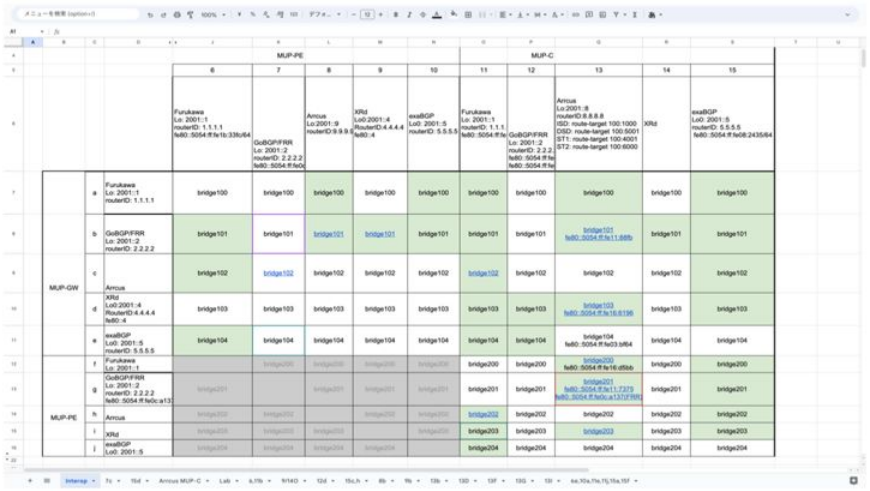
Yokohama

Hackathon Plan

Let's implement a new BGP SAFI and do the Interop togethe

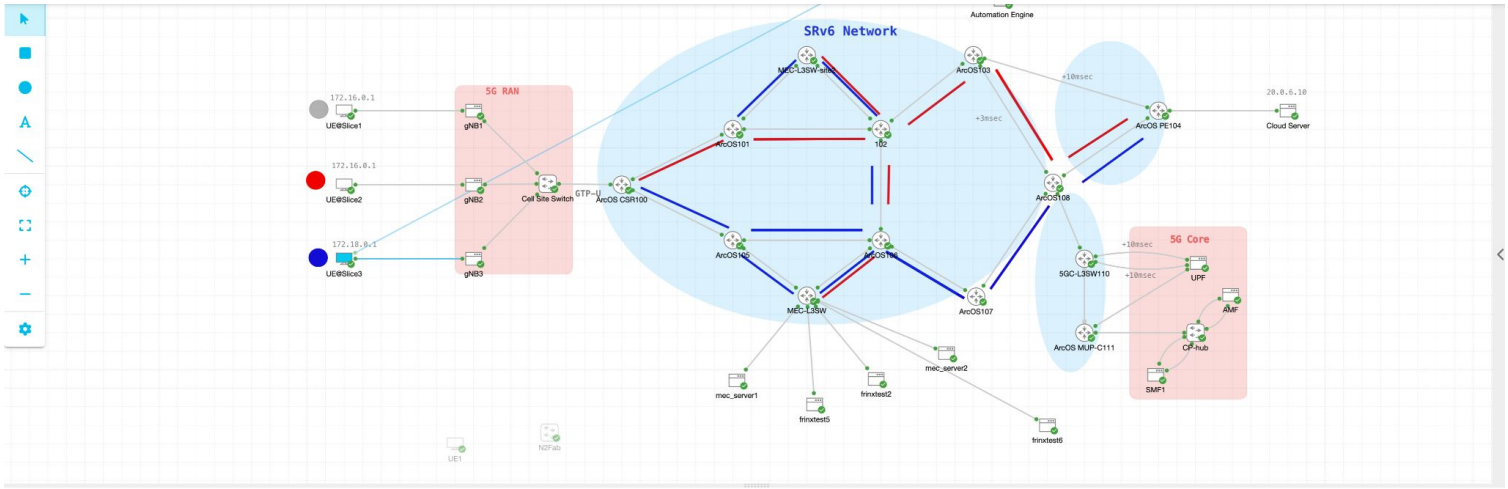
- MUP Architecture and BGP-MUP SAFI
  - https://datatracker.ietf.org/doc/draft-mhkk-dmm-sr architecture/
  - https://datatracker.ietf.org/doc/draft-mpmz-bess-mu
- Participated BGP developers
  - Arrcus
  - Cisco
  - ExaBGP
  - FRR
  - Furukawa
  - GoBGP
  - (Open BMP)

The Interop Matrix



https://github.com/IETF-Hackathon/ietf116-project-presentations/blob/pdfs-from-html/ietf-116-hackathon-bgp-mup-safi-interop.pdf

MUP Working Code and Demo



NODE INFO | SIMULATE | CONNECTIVITY | CONSOLE | VNC | EDIT CONFIG | INTERFACES | SERIAL LINE 0 | DOWNLOAD HISTORY | SETTINGS

```
64 bytes from 20.0.6.10: seq=33133 ttl=42 time=47.785 ms
64 bytes from 20.0.6.10: seq=33134 ttl=42 time=48.331 ms
64 bytes from 20.0.6.10: seq=33135 ttl=42 time=48.332 ms
64 bytes from 20.0.6.10: seq=33136 ttl=42 time=48.370 ms
64 bytes from 20.0.6.10: seq=33137 ttl=42 time=48.110 ms
64 bytes from 20.0.6.10: seq=33138 ttl=42 time=48.041 ms
64 bytes from 20.0.6.10: seq=33139 ttl=42 time=48.488 ms
64 bytes from 20.0.6.10: seq=33140 ttl=42 time=48.599 ms
64 bytes from 20.0.6.10: seq=33141 ttl=42 time=48.111 ms
64 bytes from 20.0.6.10: seq=33142 ttl=42 time=48.281 ms
64 bytes from 20.0.6.10: seq=33143 ttl=42 time=48.218 ms
64 bytes from 20.0.6.10: seq=33144 ttl=42 time=47.601 ms
64 bytes from 20.0.6.10: seq=33145 ttl=42 time=48.005 ms
64 bytes from 20.0.6.10: seq=33146 ttl=42 time=48.298 ms
64 bytes from 20.0.6.10: seq=33147 ttl=42 time=48.044 ms
64 bytes from 20.0.6.10: seq=33148 ttl=42 time=48.233 ms
64 bytes from 20.0.6.10: seq=33149 ttl=42 time=48.169 ms
64 bytes from 20.0.6.10: seq=33150 ttl=42 time=48.091 ms
64 bytes from 20.0.6.10: seq=33151 ttl=42 time=48.505 ms
```

CPU 3.92% | MEMORY 59.45% | DISK 8.86% | Notifications 0 | Status OK

Next Step

- Clarify MUP-C API text
- WG Last Call

EoF