

Network Resource Partitions (NRPs) YANG Modules

[draft-ietf-teas-nrp-yang-05](#)

TEAS WG

March. 2026

Bo Wu (Presenting), Dhruv Dhody, Vishnu Pavan Beeram, Tarek Saad, Shaofu Peng

Contributors : Xufeng Liu, Mohamed Boucadair, Daniele Ceccarelli, Bin Wen ,
Ran Chen, Luis M. Contreras, Ying Cheng, Liyan Gong

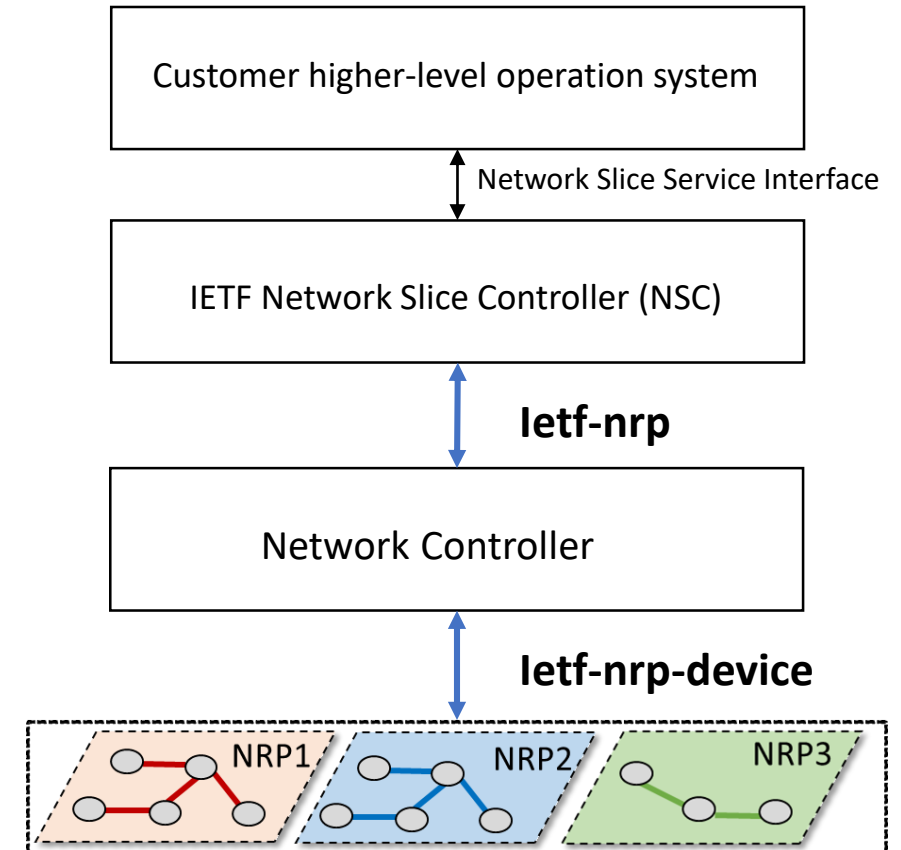


I E T F

TEAS 125

Recap

- Two YANG data models for configuring and managing NRPs:
 - **ietf-nrp** (Network-level model): For network-wide policy configuration by a NSC
 - **ietf-nrp-device** (Device-level model): For device-specific configuration by a Network Controller on individual network elements
- Foundation Framework
 - Based on RFC 9543 (IETF Network Slice Framework).
 - An NRP is a collection of network resources allocated from the underlay network to support one or more network slices.



-05 Major Updates Summary

- Thanks to Tom Petch, Adrian, Med, Lius, Xuesong, Italo, Joel Halpern for the valuable comments.
- Resolved all open issues from -04 (Appendix A), no remaining open issues

7 Open issues	-05 Change
MPLS Selector ID	Align with draft-mpls-mna-nrp-selector
Scalability of multiple NRPs	Operational guidance
Reference	Updated to RFC 9543, Topology filter dependency
NSC/controller/device usage, PHB scope, device model	Section 3.1 & 3.2, Section 3.4
NRP policy definition, operation vs mode clarification	Detailed in Section 3.2
Technology-agnostic vs IP-specific, IP/MPLS scope, SR inclusion	Updated as IP/MPLS/SR realization of generic RFC 9543
Examples	Added Appendix A with two JSON examples (APs, distributed NEs)

NRP YANG Model Overview

- **Architecture** — Dual-model hierarchy (network policy → device configuration) with grouping reuse across layers.
- **Data plane** — Selector agnosticism (IPv6, MPLS, ACL) supports flexible choice

Network Configuration Model

```
module: ietf-nrp
  augment /nw:networks:
    +--rw nrp-policies
      +--rw policy-profiles
        | +--rw phb-profile-identifier* [id]
        | ...
      +--rw nrp-policy* [name]
        +--rw name string
        +--rw nrp-id? uint32
        +--rw mode? identityref
        +--rw resource-reservation
        | ...
      +--rw selector
        | ...
        +--rw phb-profile? leafref
        +--rw topology
        ...

  augment /nw:networks/nw:network/nw:network-types:
    +--rw nrp!
  augment /nw:networks/nw:network/nw:node:
    +--rw (nrp-attachment-type)?
      +--:(single-nrp)
        | +--ro nrp-node-attributes
        | ...
      +--:(multi-nrp)
        +--ro node-nrp-list
        ...
```

Three partition modes
(Control-Only, Data-Only, Hybrid)

Monitoring: per-NRP and network-wide NRPs

Device Model

```
module: ietf-nrp-device
  +--rw nrp-policies
    +--rw qos-profiles
      | +--rw phb-profile-identifier* [id]
      | ...
    +--rw nrp-policy* [name]
      +--rw name string
      +--rw nrp-id? uint32
      +--rw resource-reservation
      | ...
    +--rw selector-id
      | ...
    +--rw phb-profile? leafref
    +--rw igp-congruent!
    | ...
    +--rw interfaces
      ...
```

Grouping

Complete MPLS NRP Selector Definition

- Move from placeholder to full In-Stack and Post-Stack support.

```
+--rw selector
| +--rw ipv4
| | +--rw destination-prefix*   inet:ipv4
| +--rw ipv6
| | +--rw (selector-type)?
| |   +--:(dedicated)
| |   | +--rw ipv6-hbh-eh?       uin
| |   +--:(srv6-sid-derived)
| |   | +--rw srv6-sid*         ine
| |   +--:(ipv6-destination-derived)
| |     +--rw destination-prefix* ine
| +--rw mpls
| +--rw acl-ref*   nrp-acl-ref
```

```
+--rw selector-id
| +--rw ipv4-derived           # Existing infrastructure reuse
| | +--rw destination-prefix*  inet:ipv4-prefix
| +--rw ipv6                   # 32-bit space, finest granularity
| | +--rw (selector-type)?
| |   +--:(dedicated)         # 32-bit NRP Selector ID
| |   | +--rw ipv6-hbh-eh?    uint32
| |   +--:(srv6-derived)     # SID-based identification
| |   | +--rw srv6-sid*      srv6-types:srv6-sid
| |   +--:(ipv6-destination-derived) # Prefix-based
| |     +--rw destination-prefix* inet:ipv6-prefix
| +--rw mpls                   # 20-bit (or 13-bit) or 32 space
| | +--rw (mpls-encapsulation)?
| |   +--:(mna-in-stack)      # NRPS13/NRPS20/ENRPS20
| |   | +--rw in-stack-identifier? uint32
| |   | +--rw identifier-format?  enumeration
| |   +--:(post-stack)        # 32-bit PSD option
| |     +--rw post-stack-identifier? uint32
| +--rw acl-ref*              acl-ref # Flexible encoding
```

Add Scalability Consideration (Section 6)

- Add three-phase operational framework covering planning, provisioning, and monitoring at scale.
- Create NRPs only when strict isolation is required; share topology and profiles when possible

01

Planning Phase

- ✓ Reusable **qos-profiles** defined once and applied to any NRP.
- ✓ **Topology sharing** via filter/selection allows underlay sharing among NRPs.

02

Provisioning Phase

- ✓ NSC conveys **intent** through **nrp-policy** at network level.
- ✓ Network controller handles device-level allocation through device model. Minimize per-NRP data-plane configuration.

03

Monitoring Phase

- ✓ Leverages dual-view approach selecting appropriate view based on operational need
- ✓ Supports both **per-NRP** and **network-wide** (native topology with NRP list).

Next Step

- Solicit WG review
- Request WG Last Call