

# Extensions to the Access Control Lists (ACLs) YANG Model

[draft-ietf-netmod-acl-02](#)

O. Gonzalez de Dios (**Telefonica**) , S. Barguil (**Nokia**), M. Boucadair (**Orange**), Qin Wu (**Huawei**)

NETMOD WG Meeting

24<sup>th</sup> July 2023, San Francisco IETF#117

# Summary of changes and open issues

- The draft was adopted after IETF 115
- **Issues are tracked in <https://github.com/boucadair/enhanced-acl-netmod>**
- Editorial review
  - Fixed json examples with RFC 7951 encoding rules
- Included the yang code for the **use of aliases**
- A set of additional gaps have been identified and addressed in -02 version:
  - ISID Filter
  - VLAN filter
  - MPLS match headers
- An IANA-maintained module for ICMP types has been added to the draft
- Open issues:
  - IPv6 extended header fields matches  
<https://github.com/boucadair/enhanced-acl-netmod/issues/9>
  - Redirect action  
<https://github.com/boucadair/enhanced-acl-netmod/issues/5>
  - Identify commonly used actions:  
<https://github.com/boucadair/enhanced-acl-netmod/issues/24>

# Use of aliases

- **Motivation:** facilitate management by having an alias to refer to commonly used values of prefixes, ports, protocols...

```
+--rw aliases
  +--rw alias* [name]
    +--rw name      string
    +--rw prefix*   inet:ip-prefix
    +--rw port-range* [lower-port]
      | +--rw lower-port  inet:port-number
      | +--rw upper-port? inet:port-number
    +--rw protocol* uint8
    +--rw fqdn*     inet:domain-name
    +--rw uri*      inet:uri
```

ALIAS  
DEFIINTION



```
augment /acl:acls/acl:acl/acl:aces/acl:ace/acl:matches:
  +--rw (alias)?
    | +--rw alias-name*  alias-ref
```

MATCH THE  
ALIAS  
CONTENT



# ISID Filter

- Motivation:
  - EVPN-PBB service configuration requires the ability of filter by instance service identifier (**I-SID**).
  - Reuses ranges definition for other parameters.

```
{
  "ietf-access-control-list:acls": {
    "acl": [
      {
        "name": "test",
        "aces": {
          "ace": [
            {
              "name": "1",
              "matches": {
                "ietf-acl-enh:isid-filter": {
                  "lower-isid": 100,
                  "upper-isid": 200
                }
              },
              "actions": {
                "forwarding": "ietf-access-control-list:accept"
              }
            }
          ]
        }
      }
    ]
  }
}
```

# VLAN Filter

- Motivation:
  - To filter all packets that are bridged within a VLAN or that are routed into or out of a bridge domain is part of the VPN control requirements derived of the EVPN definition done in [RFC7209].
  - Ranges definition for other parameters are reused.

```
{
  "ietf-access-control-list:acls": {
    "acl": [
      {
        "name": "VLAN_FILTER",
        "aces": {
          "ace": [
            {
              "name": "1",
              "matches": {
                "ietf-acl-enh:vlan-filter": {
                  "lower-vlan": 10,
                  "upper-vlan": 20
                }
              },
              "actions": {
                "forwarding": "ietf-access-control-list:accept"
              }
            }
          ]
        }
      }
    ]
  }
}
```

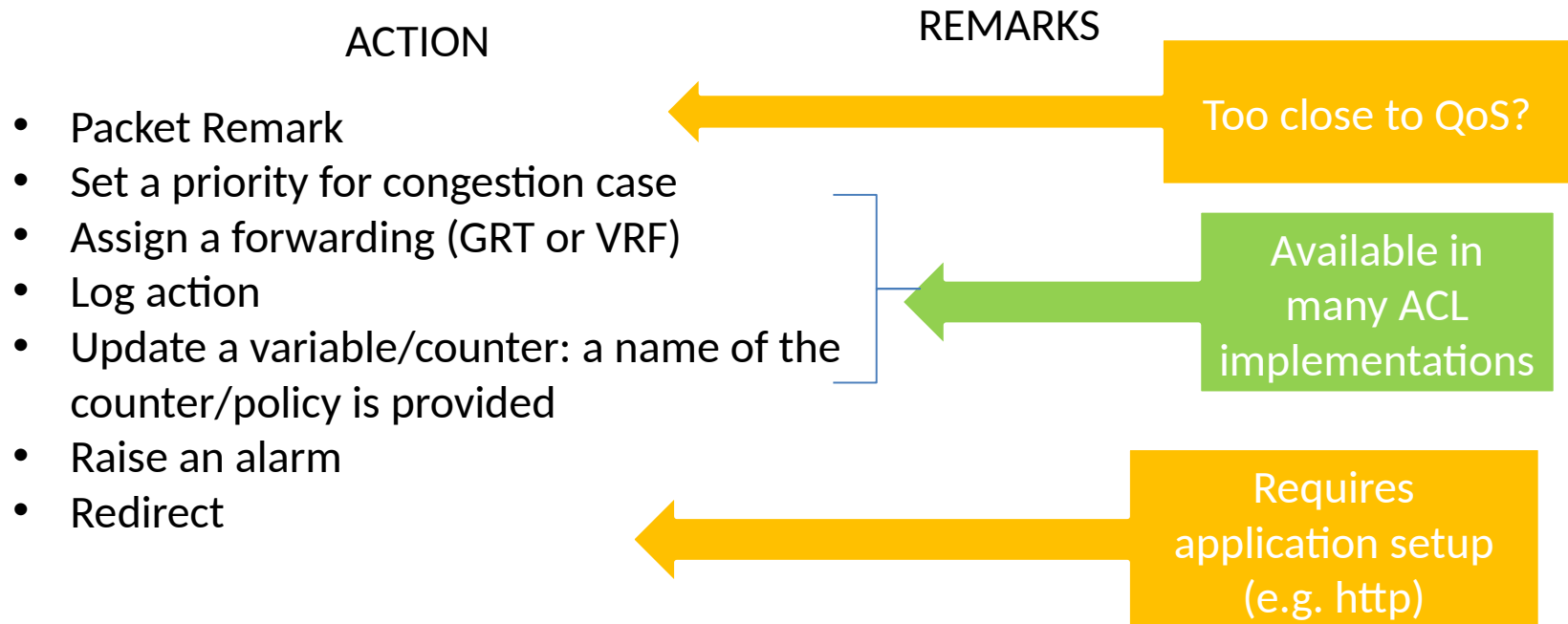
# MATCH MPLS headers

- The ACL models can be used to create rules to match MPLS fields on a packet.
- MPLS headers defined in [RFC3032] and [RFC5462] contains:
  - **Traffic Class:** 3 bits 'EXP' renamed to 'Traffic Class Field'
  - **Label Value:** A 20-bit field that carries the actual value of the MPLS Label.
  - **TTL:** Packet time-to-live value.

```
augment /acl:acls/acl:acl/acl:aces/acl:ace/acl:matches:
...
+--rw (mpls)?
  +--:(mpls-values)
    +--rw mpls-values {match-on-mpls}?
      +--rw traffic-class?    uint8
      +--rw label-position    identityref
      +--rw upper-label-range? uint32
      +--rw lower-label-range? uint32
      +--rw label-block-name  string
      +--rw ttl-value?        uint8
```

# Open issues: New Actions

- <https://github.com/boucadair/enhanced-acl-netmod/issues/24>
- Upon Match, current ACL model are ACCEPT, REJECT and DROP
- We have identified actions performed by different ACL implementations
- Which actions are common and should be part of the standard model and which ones are out of scope?



# Open issues: New Actions

- <https://github.com/boucadair/enhanced-acl-netmod/issues/24>
- Upon Match, current ACL model are ACCEPT, REJECT and DROP
- We have identified actions performed by different ACL implementations
- Which actions are common and should be part of the standard model and which ones are out of scope?

## ACTIONS PROPOSAL

- ~~Packet Remark~~
- ~~Set a priority for congestion case~~
- **Assign a forwarding (GRT or VRF)**
- **Log action**
- **Update a variable/counter:** a name of the counter/policy is provided
- ~~Raise an alarm~~
- ~~Redirect~~

# Next Steps

- Request WG to review the document, latest changes and provide feedback on open issues.
- Is the Working Group happy with current approach of augmenting **RFC 8519** in a new module?
- Request MPLS WG to review the extensions to cover MPLS header matches.
- Update draft with the proposed actions
- **Questions & Suggestions are welcome**