

# Dynamic Network Adjustments for Cloud Service Scaling

draft-dunbar-neotec-net-adjust-cloud-scaling-02

[ldunbar@futurewei.com](mailto:ldunbar@futurewei.com)

[xiechf@chinatelecom.cn](mailto:xiechf@chinatelecom.cn)

[kausik.majumdar@oracle.com](mailto:kausik.majumdar@oracle.com)

[sunqiong@chinatelecom.com](mailto:sunqiong@chinatelecom.com)

IETF 121 Dublin

# Problem Statement

- **Key Challenges:**

- Lack of coordination between dynamic cloud service scaling and network configuration.
- Proprietary solutions limit interoperability across multi-vendor environments.
- Manual adjustments lead to delays and potential service disruptions.
- No standardized framework for automating network responses to cloud scaling.

- **Solution:**

- A framework that automates network adjustments triggered by cloud service changes using standardized YANG models.
- Exposing some network information to Cloud Controller for service orchestration.
- Extending RFC8969

# Dynamic-Load-Balancer YANG Model:

E.g., JSON code to request changing the load balance algorithm to optimize forwarding for a specific flow (like Flow X) dynamically across the network:

```
{
  "flow-optimization": {
    "flow-id": "Flow-X",
    "traffic-type": "ML",
    "priority": "high",
    "action": "optimize",
    "algorithm": "ml-optimized",
    "congestion-awareness": {
      "enabled": true,
      "state-feedback": "global"
    }
  }
}
```

```
module: dynamic-load-balancer
+--rw load-balancer
  +--rw router* [router-id]
    +--rw router-id      string
    +--rw algorithm      enumeration
                        | +-- round-robin      evenly across all paths.
                        | +-- least-connections select path with the fewest flows.
                        | +-- ip-hash          based on a hash of IP.
                        | +-- ml-optimized     Optimizes for ML flows.
                        | +-- packet-level     per-packet basis.
  +--rw paths* [path-id]
    +--rw path-id        string
    +--rw destination     inet:ipv4-prefix
    +--rw bandwidth       uint64
    +--rw latency         uint32
    +--rw congestion-awareness
      +--rw enabled       boolean
      +--rw state-feedback enumeration
                        +-- local      Only local path congestion state is used.
                        +-- global     Global network path congestion state is used.
```

The Network Controller sends directives to routers along the path between the source and destination of Flow X, instructing them to adjust their load balancing strategies in real-time, ensuring Flow X takes the highest bandwidth path while deprioritizing other flows as necessary. The network dynamically responds to congestion states to maintain optimal performance for Flow X.

# Dynamic-ACL Augment ietf-acl-enh YANG Model:

- E.g., JSON code to add a new rule (rule-3) to the ACL acl-123, allowing SSH traffic (port 22) from source IP 192.168.1.101 to destination IP 10.0.0.10. The existing rules, rule-1 and rule-2, control HTTPS (port 443) and block HTTP traffic (port 80), respectively

```
{
  "acl:acls": {
    "acl": [
      {
        "name": "dynamic-acl-001",
        "aces": {
          "ace": [
            {
              "name": "dynamic-rule-1",
              "actions": {
                "forwarding": "permit"
              },
              "matches": {
                "ipv4": {
                  "source-ipv4-network": "192.168.1.0/24",
                  "destination-ipv4-network": "10.0.0.0/24"
                },
                "protocol": "tcp",
                "source-port": 22
              },
              "cloud-service-trigger": "ml-service-scaling",
              "priority": 10
            }
          ]
        }
      }
    ]
  }
}
```

module: dynamic-acl

augment /acl:acls/acl:acl/acl:aces/acl:ace:

+++rw cloud-service-trigger? string

+++rw priority? uint32

- cloud-service-trigger: identifies the specific cloud service event that necessitates the ACL change. It is optional (?)
- Priority: sets the priority level of the Access Control Event (ACE), helping to determine the order in which the ACEs are evaluated. It is also optional.

# Dynamic-Bandwidth YANG Model:

E.g. when a cloud orchestration system detects increased traffic, it can dynamically request an increase in bandwidth to 1000 Mbps (1 Gbps) on network link link-123:

```
{
  "nt:networks": {
    "nt:network": [
      {
        "network-id": "cloud-network-1",
        "nt:link": [
          {
            "link-id": "link-123",
            "source-device": "device-1",
            "destination-device": "device-2",
            "requested-bandwidth": 1000 // 1000 Mbps bandwidth
          }
        ]
      }
    ]
  }
}
```

```
module dynamic-bandwidth {
  namespace "urn:ietf:params:xml:ns:yang:dynamic-bandwidth";
  prefix dbw;

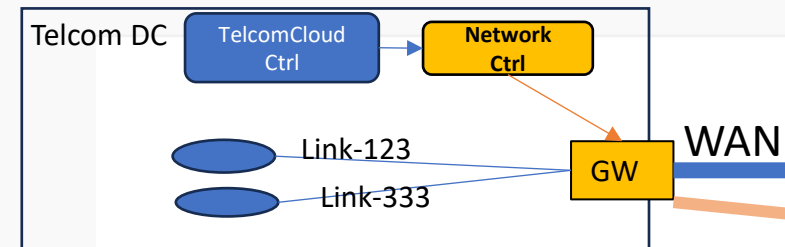
  import ietf-network-topology {
    prefix nt;
  }

  organization "IETF";
  contact "IETF Routing Area";
  description
    "YANG model for dynamically updating bandwidth on network links.";

  revision "2024-10-18" {
    description "Initial version.";
  }

  augment "/nt:networks/nt:network/nt:link" {
    description
      "Augment the network topology YANG model to update
      the bandwidth dynamically.";

    leaf requested-bandwidth {
      type uint64;
      description "Requested bandwidth in Mbps.";
    }
  }
}
```



More complicated scenario: The Blue WAN path is no longer enough for sudden surge of the Cloud service (blue), need GW to aggregate additional WAN paths to form a bigger pipe for the Blue service. Need a standard interface.

# Security Considerations

- **Authentication and Authorization:**

- Use mutual authentication methods such as TLS certificates to verify the identities of both the cloud orchestrator and the network controller before any configuration commands are accepted.
- OAuth or API Key-Based Access: For REST API-based communications, secure token-based authentication (e.g., OAuth 2.0) or unique API keys can be employed to validate requests from legitimate sources.

- **Data Integrity:**

- Use TLS to encrypt communication channels, protecting the integrity of the transmitted data.
- Employ checksums or hash functions on critical configuration messages to detect any tampering or unintended modifications during transit.

- **Monitoring and Auditing:**

- Maintain detailed logs of all configuration changes initiated by cloud scaling events, including timestamps, source entities, and specific parameters modified.
- Conduct periodic audits of the authorization policies, access logs, and configuration adjustments to ensure compliance with security policies and to detect any anomalies.