

# **IETF TE Topology Graphical Display**

IETF 121 Hackathon

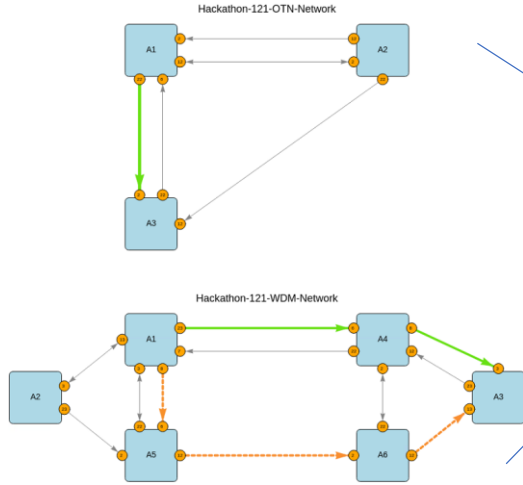
02-03 Nov 2024

Dublin, Ireland

# Hackathon Goals and Plan

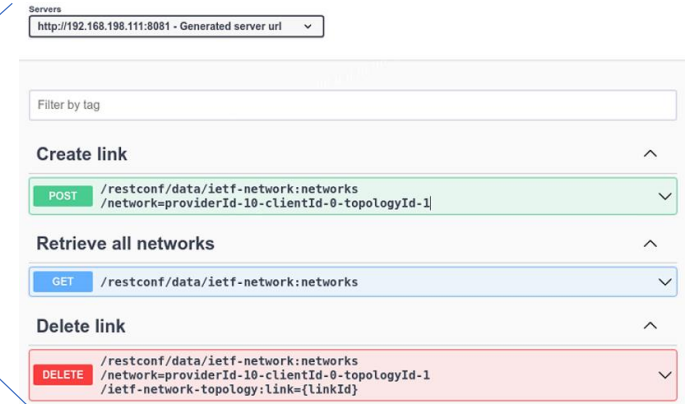
- We introduce a toolkit which auto-generates RESTCONF interfaces (e.g., URIs) for YANG models, and also simulates GRUD operations (e.g., GET, POST, DELETE, etc.) against data in the YANG datastore.
- The toolkit aims to help us understand YANG models, especially complex ones (e.g., IETF ACTN MPI models).
- As for the demonstration, we use this toolkit to show some key aspects of the IETF TE topology model (RFC8795), such as bidirectional links and link hierarchy.

# Overview of the Toolkit



IETF TE Topology GUI  
(RESTCONF Client)

Swagger Interface  
(RESTCONF Client)

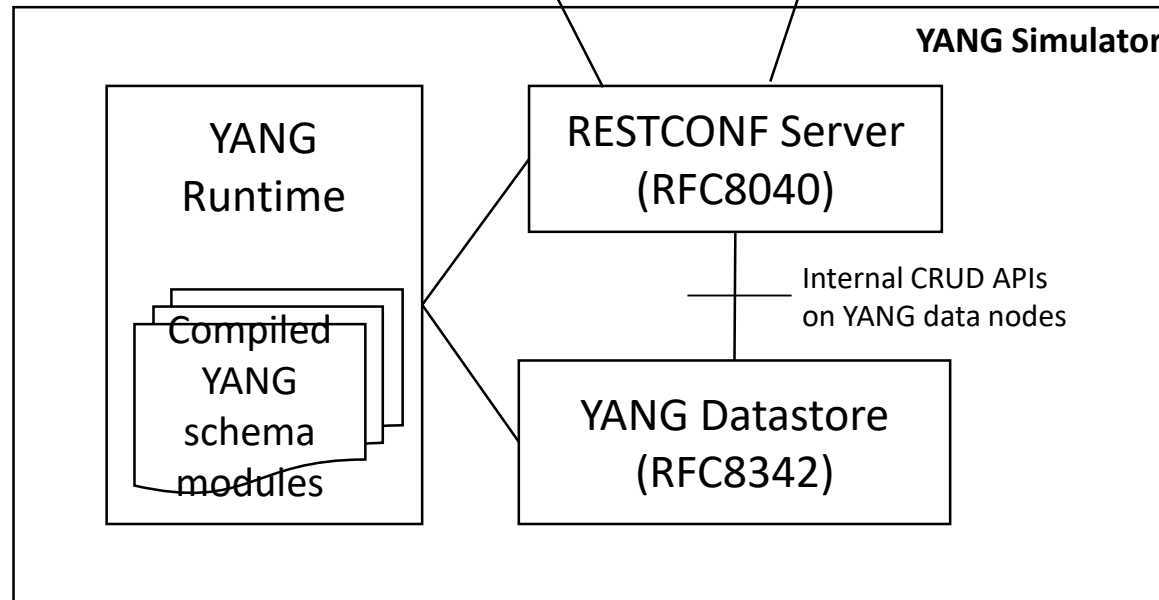


RESTCONF GET ietf-network:networks

RESTCONF GET, POST, DELETE, etc. on data resources

**YANG modules that are currently compiled/packaged into the toolkit (i.e., mainly ACTN models):**

- 'ietf-inet-types@2021-02-22.yang'
- 'ietf-layer1-types@2021-02-19.yang'
- 'ietf-network@2018-02-26.yang'
- 'ietf-network-state@2018-02-26.yang'
- 'ietf-network-topology@2018-02-26.yang'
- 'ietf-otn-topology@2021-02-22.yang'
- 'ietf-otn-tunnel@2021-06-25.yang'
- 'ietf-te@2021-02-20.yang'
- 'ietf-te-topology@2020-08-06.yang'
- 'ietf-te-topology-state@2020-08-06.yang'
- 'ietf-wson-topology@2018-02-27.yang'
- 'ietf-yang-types@2021-04-14.yang'
- ... ..



The YANG simulator can be packaged and run as a docker container


# Demo 1: Inject TE Topology JSON into YANG Datastore

- The YANG simulator offers a debug REST API, i.e., `pnccsimu/v1/reload-data`, to inject a JSON payload of the YANG datastore (i.e., a YANG data resource at root).
- We use this command to inject “`ietf-network:networks`” into the YANG datastore.
- Alternatively, we may also use RESTCONF POST or PUT command to create `ietf-network:network` instances. However, `pnccsimu/v1/reload-data` purges any existing data in the datastore before adding the new data, making it convenient to use.

```
# curl -X POST -H "content-type:application/json" http://127.0.0.1:8181/pnccsimu/v1/reload-data -d @./ietf121-hackathon-te-topo.json #
```

In addition, the simulator offers a API that purges the data in the YANG datastore:

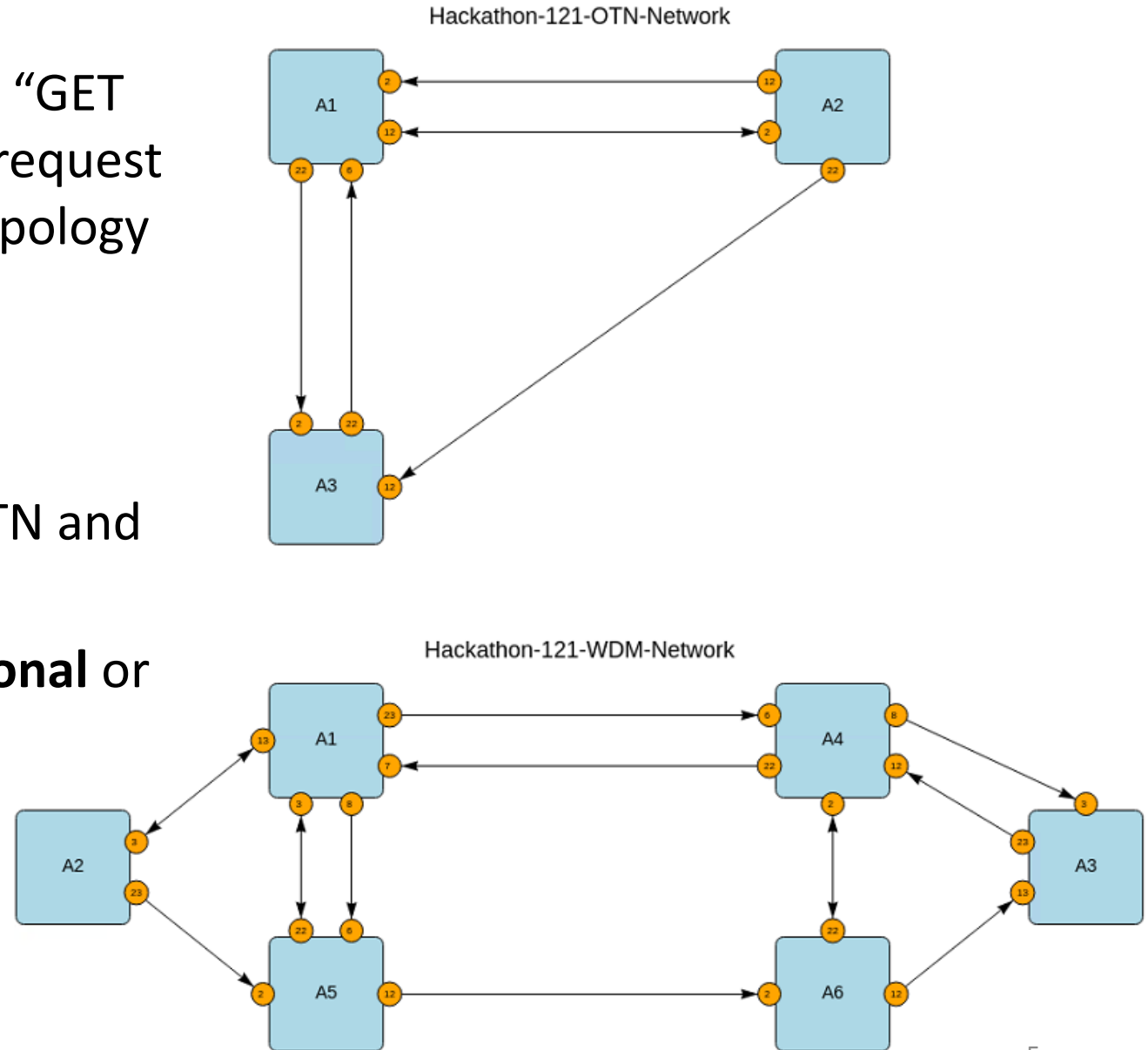
```
# curl -X POST -H "content-type:application/json" http://127.0.0.1:8181/pnccsimu/v1/clear-data -d '{}'
```



```
ietf121-hackathon-te-topo.json
1  {
2      "ietf-network:networks": {
3          "network": [
4              {
5                  "network-id": "providerId-10-clientId-0-topologyId-1",
6                  "ietf-te-topology:te": {
7                      "name": "Hackathon-121-OTN-Network"
8                  },
9                  "ietf-te-topology:te-topology-identifier": {
14                     "network-types": {
15                         "ietf-te-topology:te-topology": {
16                             "ietf-otn-topology:otn-topology": {}
17                         }
18                     },
19                     "node": [
152                    "ietf-network-topology:link": [
324                ],
325                {
326                    "network-id": "providerId-10-clientId-0-topologyId-2",
327                    "ietf-te-topology:te": {
328                        "name": "Hackathon-121-WDM-Network"
329                    },
330                    "ietf-te-topology:te-topology-identifier": {
335                     "network-types": {
336                         "ietf-te-topology:te-topology": {
337                             "ietf-wson-topology:wson-topology": {}
338                         }
339                     },
340                     "node": [
621                    "ietf-network-topology:link": [
777                ]
778            ]
779        }
780    }
781 }
```

# Demo 2: Graphical Display of TE Topology

- The GUI is a RESTCONF client that sends “GET /restconf/data/ietf-network:networks” request to the YANG simulator to retrieve the topology JSON
- The GUI visualizes key aspects of the TE topology
  - Shows multi-layer topology (e.g., OTN and WDM)
  - shows whether a TE link is **bidirectional** or **unidirectional**



# Demo 2: Graphical Display of TE Topology (cont'd)

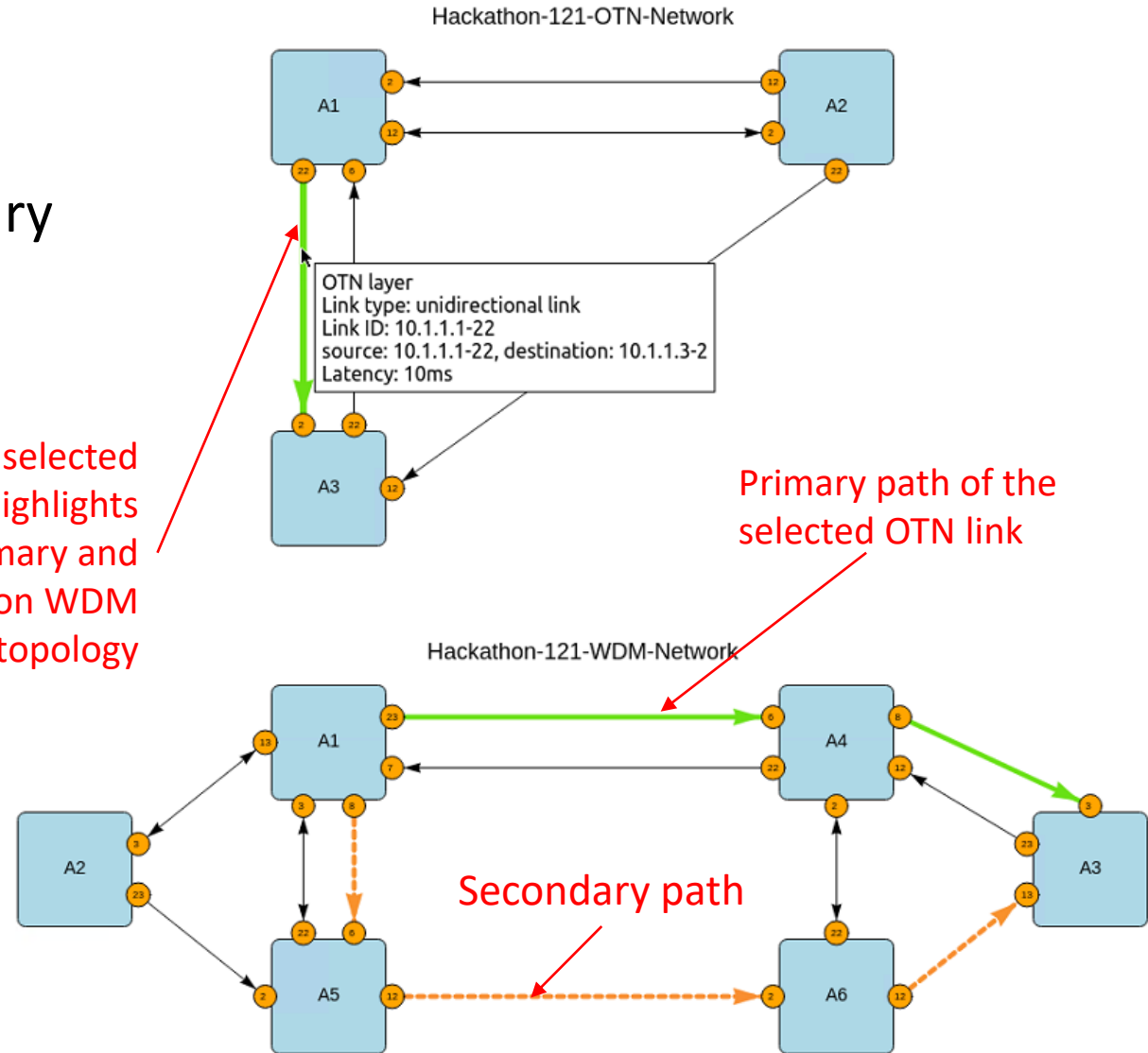
The GUI also displays other TE features:

- TE link attributes (e.g., link latency)
- Link hierarchy (e.g., primary and secondary WDM path supporting the OTN link)

```
IETF Network Link Model Instance
{
  "ietf-network-topology:link": [ 1 item
  {
    "link-id": "10.1.1.1-22"
    "source": { 2 items
      "source-node": "10.1.1.1"
      "source-tp": "22"
    }
    "ietf-te-topology:te": { 1 item
      "te-link-attributes": { 2 items
        "te-delay-metric": 10
        "underlay": { 2 items
          "backup-path": [ 1 item
            {
              "index": 1
              "network-ref":
                "providerId-10-clientId-0-topologyId-2"
              "path-element": [... ] 3 items
            }
          ]
          "primary-path": { 2 items
            {
              "network-ref":
                "providerId-10-clientId-0-topologyId-2"
              "path-element": [... ] 2 items
            }
          }
        }
      }
      "destination": { 2 items
        "dest-node": "10.1.1.3"
        "dest-tp": "2"
      }
    }
  }
  ]
}
```

Double-clicking a link shows the link's specification in JSON

When an OTN link is selected (single-clicked), GUI highlights its underlay primary and secondary path on WDM topology



# Demo 3: Swagger as RESTCONF Client

- Swagger may help user become familiar with RESTCONF interfaces of the YANG models
- We demonstrate how to create and delete network links via Swagger

The image shows a Swagger UI interface for the IETF PNC Simulator Management NBI. The main page lists several endpoints, including 'Create link', 'Retrieve all networks', and 'Delete link'. The 'Delete link' endpoint is highlighted in red.

**Create link**

**POST** /restconf/data/ietf-network:networks/network=providerId-10-clientId-0-topologyId-1

Parameters: No parameters

Request body **required** application/json

```
{
  "ietf-network-topology:link": [
    {
      "source": {
        "source-node": "10.1.1.1",
        "source-tp": "12"
      },
      "destination": {
        "dest-node": "10.1.1.2",
        "dest-tp": "2"
      },
      "link-id": "10.1.1.1-12"
    }
  ]
}
```

Execute Clear

**Responses**

```
curl -X 'POST' \
  http://192.168.198.111:8081/restconf/data/ietf-network:networks/network=providerId-10-clientId-0-topologyId-1 \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "ietf-network-topology:link": [
      {
        "source": {
          "source-node": "10.1.1.1",
          "source-tp": "12"
        },
        "destination": {
          "dest-node": "10.1.1.2",
          "dest-tp": "2"
        },
        "link-id": "10.1.1.1-12"
      }
    ]
  }'
```

Request URL: http://192.168.198.111:8081/restconf/data/ietf-network:networks/network=providerId-10-clientId-0-topologyId-1

Server response

Code	Details
200	<b>Response headers</b> connection: keep-alive content-length: 0 date: Tue, 15 Oct 2024 19:52:04 GMT keep-alive: timeout=60 vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers

**Delete link**

**DELETE** /restconf/data/ietf-network:networks/network=providerId-10-clientId-0-topologyId-1/ietf-network-topology:link={linkId}

Parameters

Name	Description
linkId <b>required</b>	

string (path)

10.1.1.1-12

Execute Clear

**Responses**

```
curl -X 'DELETE' \
  http://192.168.198.111:8081/restconf/data/ietf-network:networks/network=providerId-10-clientId-0-topologyId-1/ietf-network-topology:link=10.1.1.1-12 \
  -H 'accept: */*'
```

Request URL: http://192.168.198.111:8081/restconf/data/ietf-network:networks/network=providerId-10-clientId-0-topologyId-1/ietf-network-topology:link=10.1.1.1-12

Server response

Code	Details
200	<b>Response headers</b> connection: keep-alive content-length: 0 date: Tue, 15 Oct 2024 19:51:56 GMT keep-alive: timeout=60 vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers

# Conclusion and Future Work

- In this hackathon, we use a YANG toolkit to demonstrate some key aspects of TE topology:
  - **Bidirectional links** supported by RFC8345 (No issues were found in correlating two unidirectional links between the same two TPs as bidirectional links)
  - Navigation between **multiple layers of topologies**
  - The **underlay path** constructed from RFC8795 providing orders of the links in the underlay path, as well as the role of primary and secondary underlay paths.
- The implementation was based on an ad-hoc **profile of the TE topology** model
  - We found this approach quite practical to be used for applications, such as digital-map, which need a subset of the TE Topology model.
  - The profile can also be extended at any time to include other attributes defined in TE topology (e.g. link latency, etc.).
- For **future work**, a generic tool or a formal language to define specific profiles for any YANG models would be beneficial and should be considered.