TeraFlow SDN by ETSI

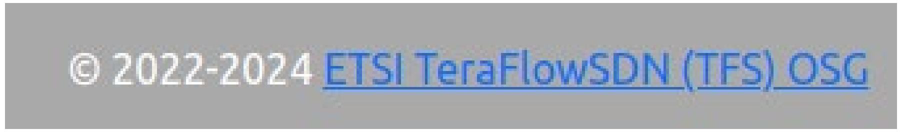
TFS Hackfest 4 Results Team ForFun

Waleed Akbar
Shayan Hajipour

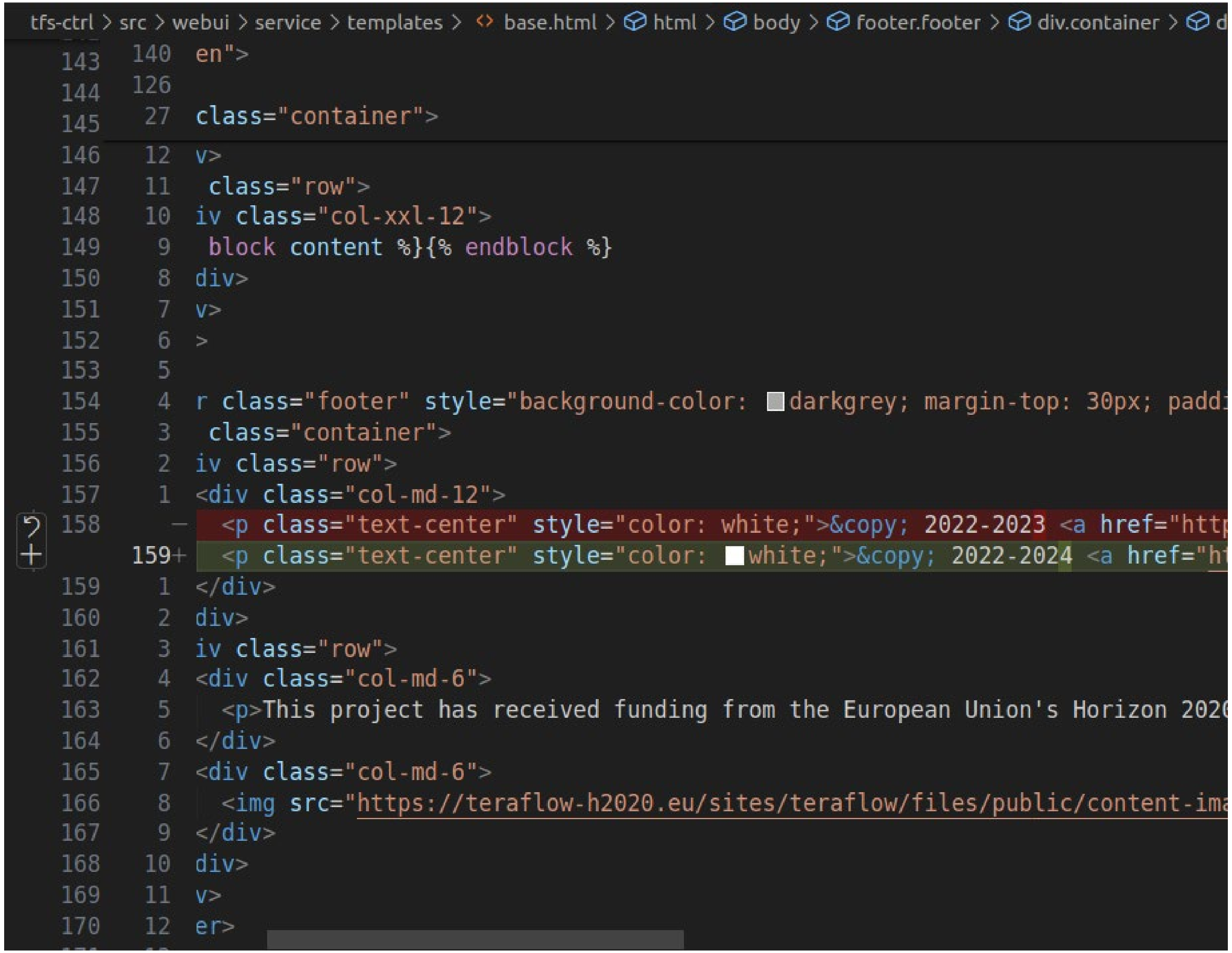
22/05/2024

1. Change WebUI License

Change the license message



© 2022-2024 ETSI TeraFlowSDN (TFS) OSG

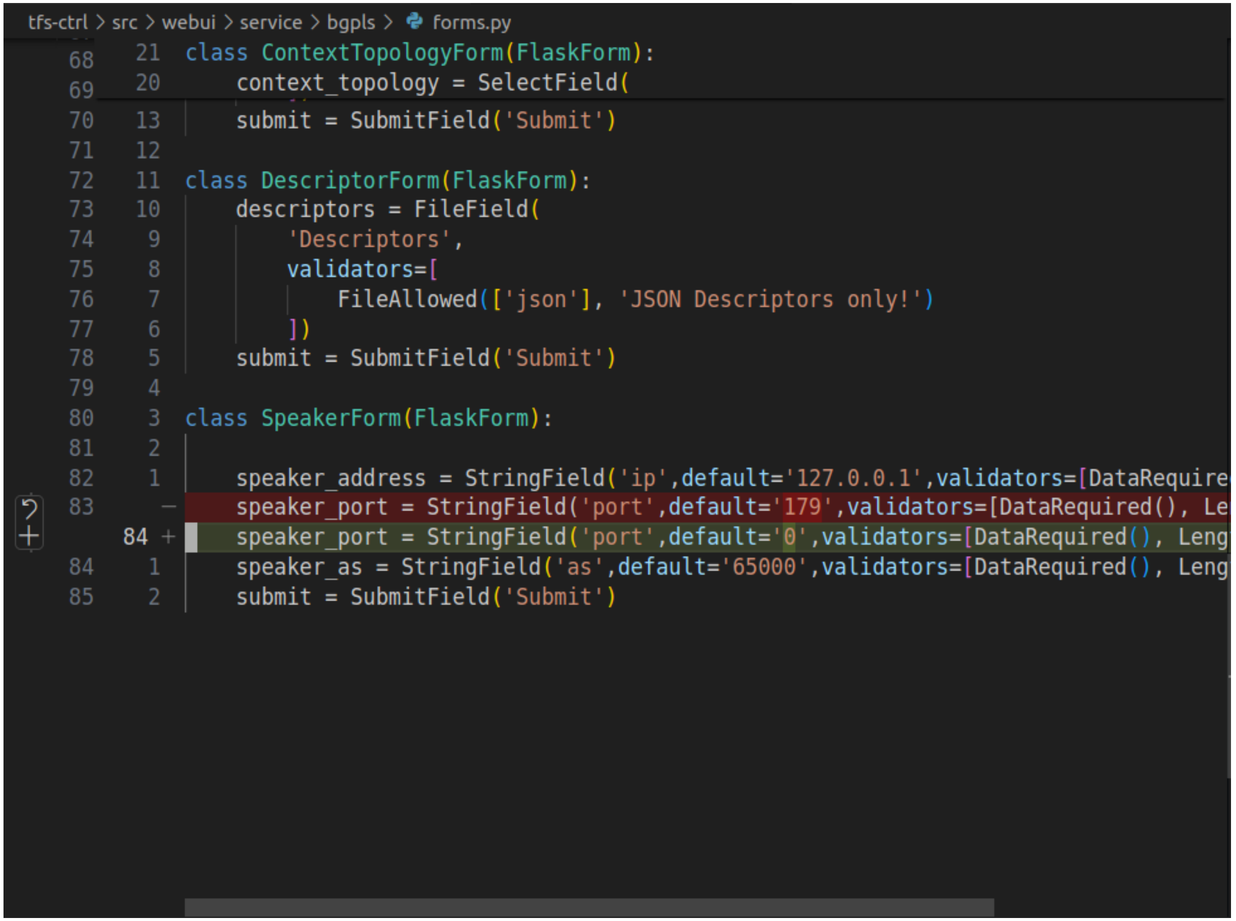


tfs-ctrl > src > webui > service > templates > <> base.html > html > body > footer.footer > div.container > d

```
en">

class="container">
v>
 class="row">
iv class="col-xxl-12">
 block content %}{% endblock %}
div>
v>
>

r class="footer" style="background-color: darkgrey; margin-top: 30px; paddi
 class="container">
iv class="row">
<div class="col-md-12">
-  <p class="text-center" style="color: white;">&copy; 2022-2023 <a href="http
+  <p class="text-center" style="color: white;">&copy; 2022-2024 <a href="ht
</div>
div>
iv class="row">
<div class="col-md-6">
  <p>This project has received funding from the European Union's Horizon 202
</div>
<div class="col-md-6">
  <img src="https://teraflow-h2020.eu/sites/teraflow/files/public/content-ima
</div>
div>
v>
er>
```

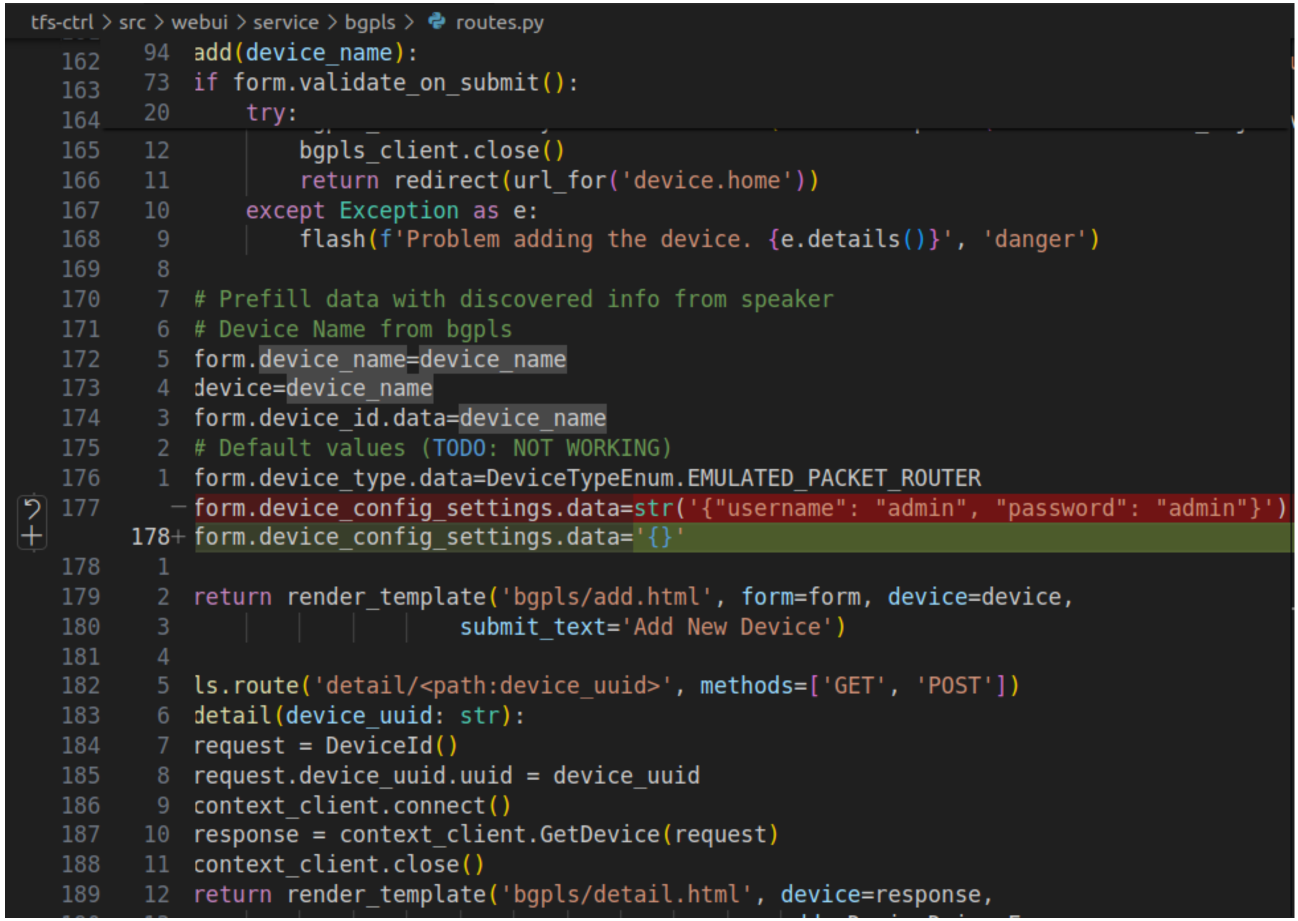
2. Modify Device Upload

Change default values in two files



tfs-ctrl > src > webui > service > bgpls > forms.py

```
class ContextTopologyForm(FlaskForm):
    context_topology = SelectField(
    submit = SubmitField('Submit')

class DescriptorForm(FlaskForm):
    descriptors = FileField(
        'Descriptors',
        validators=[
            FileAllowed(['json'], 'JSON Descriptors only!')
        ])
    submit = SubmitField('Submit')

class SpeakerForm(FlaskForm):

    speaker_address = StringField('ip',default='127.0.0.1',validators=[DataRequire
-   speaker_port = StringField('port',default='179',validators=[DataRequired(), Le
+   speaker_port = StringField('port',default='0',validators=[DataRequired(), Leng
    speaker_as = StringField('as',default='65000',validators=[DataRequired(), Leng
    submit = SubmitField('Submit')
```



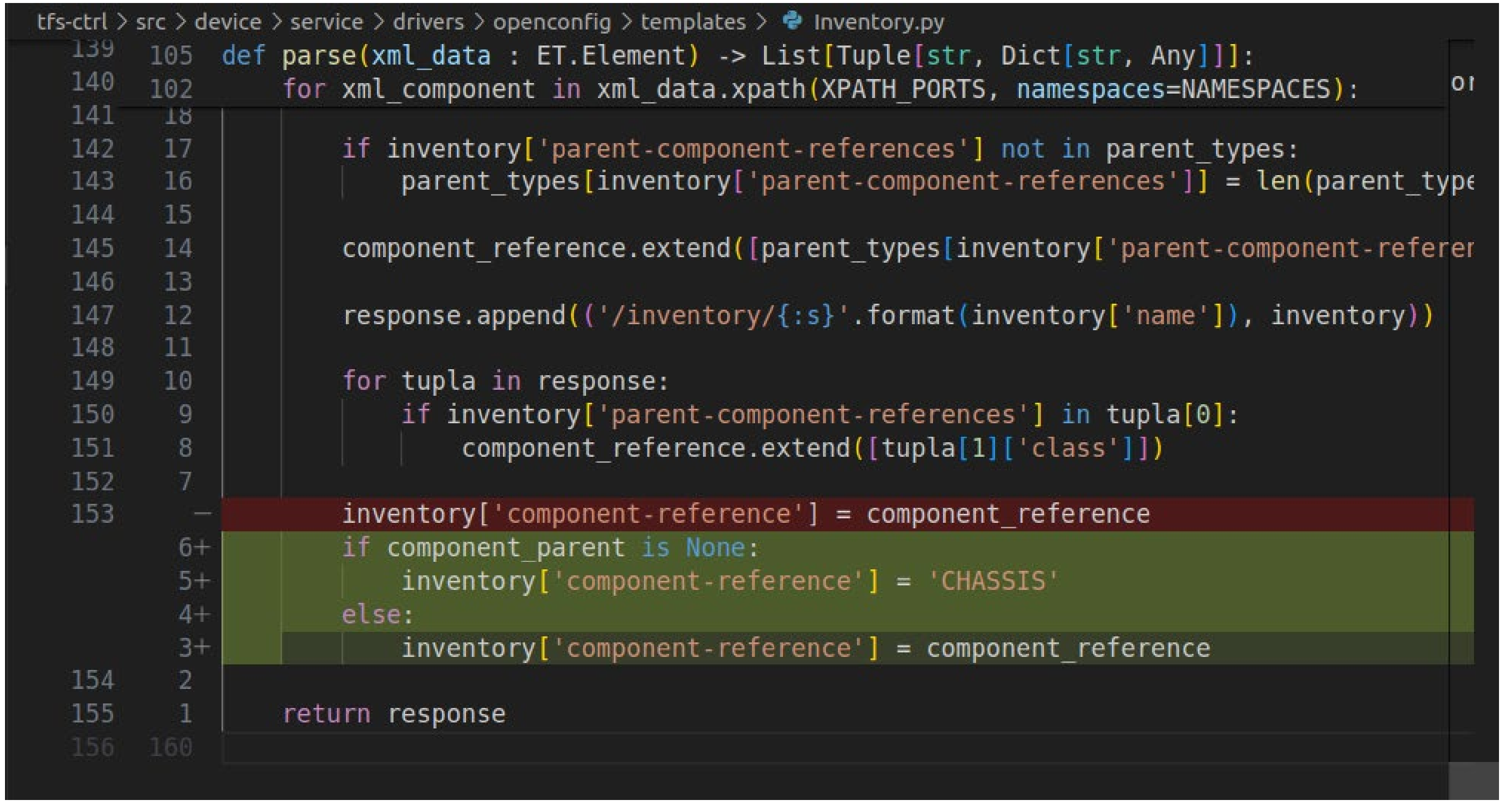
tfs-ctrl > src > webui > service > bgpls > routes.py

```
add(device_name):
if form.validate_on_submit():
    try:
        bgpls_client.close()
        return redirect(url_for('device.home'))
    except Exception as e:
        flash(f'Problem adding the device. {e.details()}', 'danger')

# Prefill data with discovered info from speaker
# Device Name from bgpls
form.device_name=device_name
device=device_name
form.device_id.data=device_name
# Default values (TODO: NOT WORKING)
form.device_type.data=DeviceTypeEnum.EMULATED_PACKET_ROUTER
- form.device_config_settings.data=str('{"username": "admin", "password": "admin"}')
+ form.device_config_settings.data='{}'

return render_template('bgpls/add.html', form=form, device=device,
                        submit_text='Add New Device')

ls.route('detail/<path:device_uuid>', methods=['GET', 'POST'])
detail(device_uuid: str):
request = DeviceId()
request.device_uuid.uuid = device_uuid
context_client.connect()
response = context_client.GetDevice(request)
context_client.close()
return render_template('bgpls/detail.html', device=response,
```

3. Correct Component-reference inventory parameter

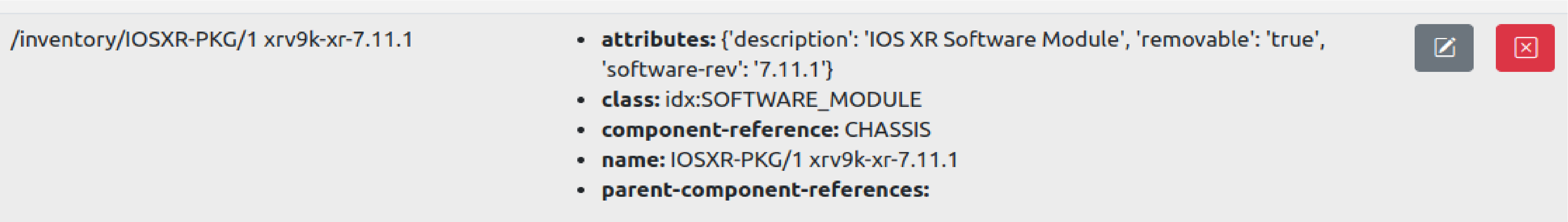
Change default values in two files

Add “CHASSIS” as value in case, “component_parent” == None



tfs-ctrl > src > device > service > drivers > openconfig > templates > Inventory.py

```
def parse(xml_data : ET.Element) -> List[Tuple[str, Dict[str, Any]]]:
    for xml_component in xml_data.xpath(XPATH_PORTS, namespaces=NAMESPACES):                    or

        if inventory['parent-component-references'] not in parent_types:
            parent_types[inventory['parent-component-references']] = len(parent_type

        component_reference.extend([parent_types[inventory['parent-component-referer

        response.append(('/inventory/{:s}'.format(inventory['name']), inventory))

        for tupla in response:
            if inventory['parent-component-references'] in tupla[0]:
                component_reference.extend([tupla[1]['class']])

        inventory['component-reference'] = component_reference
        if component_parent is None:
            inventory['component-reference'] = 'CHASSIS'
        else:
            inventory['component-reference'] = component_reference

    return response
```



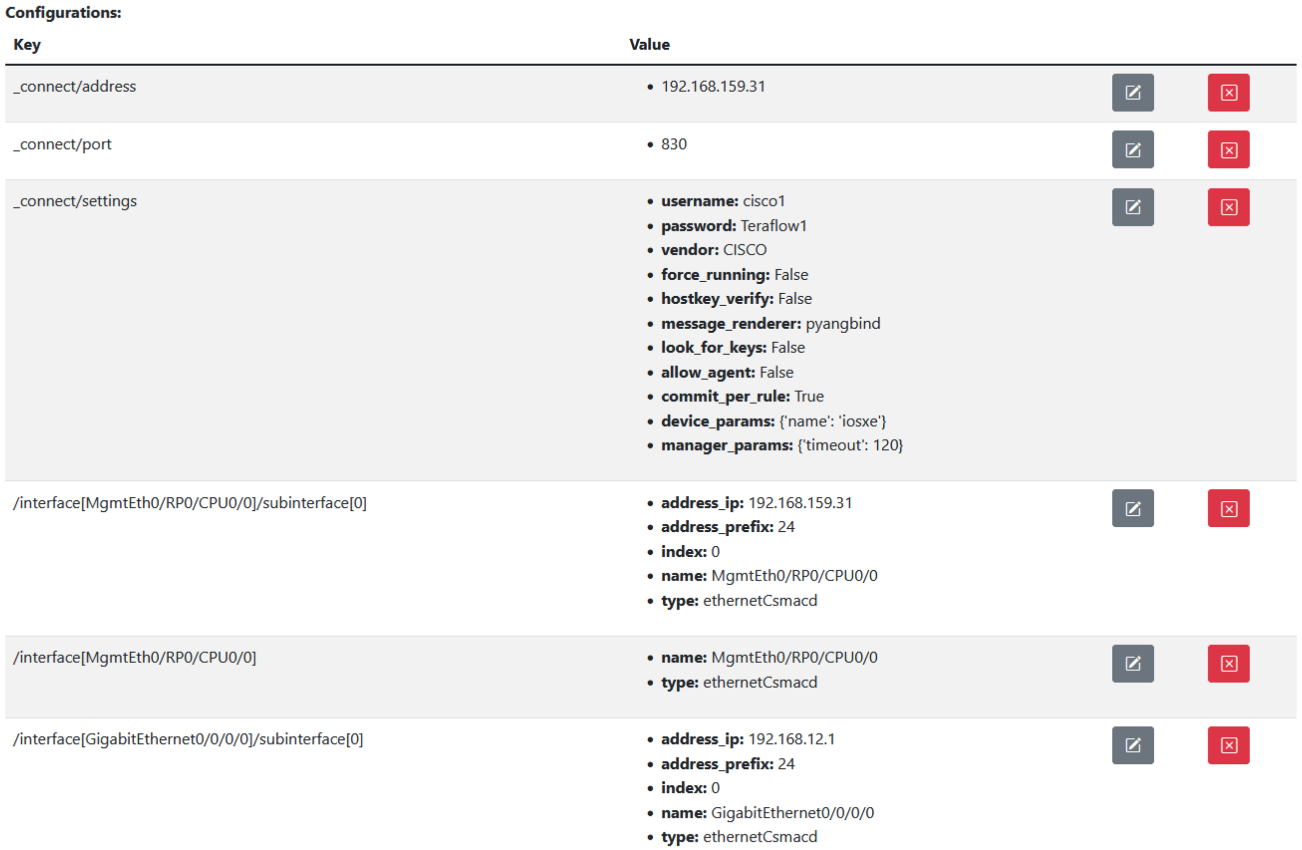
/inventory/IOSXR-PKG/1 xrv9k-xr-7.11.1

- **attributes:** {'description': 'IOS XR Software Module', 'removable': 'true', 'software-rev': '7.11.1'}
- **class:** idx:SOFTWARE_MODULE
- **component-reference:** CHASSIS
- **name:** IOSXR-PKG/1 xrv9k-xr-7.11.1
- **parent-component-references:**

4 and 5 - Remove inventory and endpoints config rules

TeraFlow SDN

Applied filter on the key of “config object” where “resource_key” is not equal to “inventory” and “endpoints”.

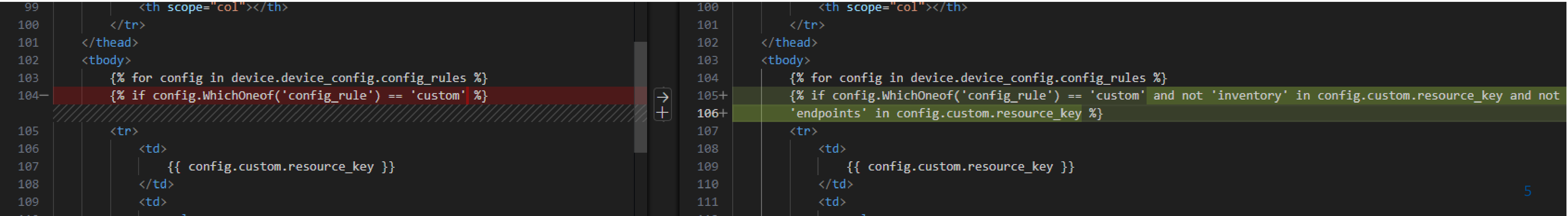


Configurations:

Key	Value		
_connect/address	• 192.168.159.31	✎	☒
_connect/port	• 830	✎	☒
_connect/settings	• **username:** cisco1 • **password:** Teraflow1 • **vendor:** CISCO • **force_running:** False • **hostkey_verify:** False • **message_renderer:** pyangbind • **look_for_keys:** False • **allow_agent:** False • **commit_per_rule:** True • **device_params:** {'name': 'iosxe'} • **manager_params:** {'timeout': 120}	✎	☒
/interface[MgmtEth0/RP0/CPU0/0]/subinterface[0]	• **address_ip:** 192.168.159.31 • **address_prefix:** 24 • **index:** 0 • **name:** MgmtEth0/RP0/CPU0/0 • **type:** ethernetCsmacd	✎	☒
/interface[MgmtEth0/RP0/CPU0/0]	• **name:** MgmtEth0/RP0/CPU0/0 • **type:** ethernetCsmacd	✎	☒
/interface[GigabitEthernet0/0/0/0]/subinterface[0]	• **address_ip:** 192.168.12.1 • **address_prefix:** 24 • **index:** 0 • **name:** GigabitEthernet0/0/0/0 • **type:** ethernetCsmacd	✎	☒

Updated statement:

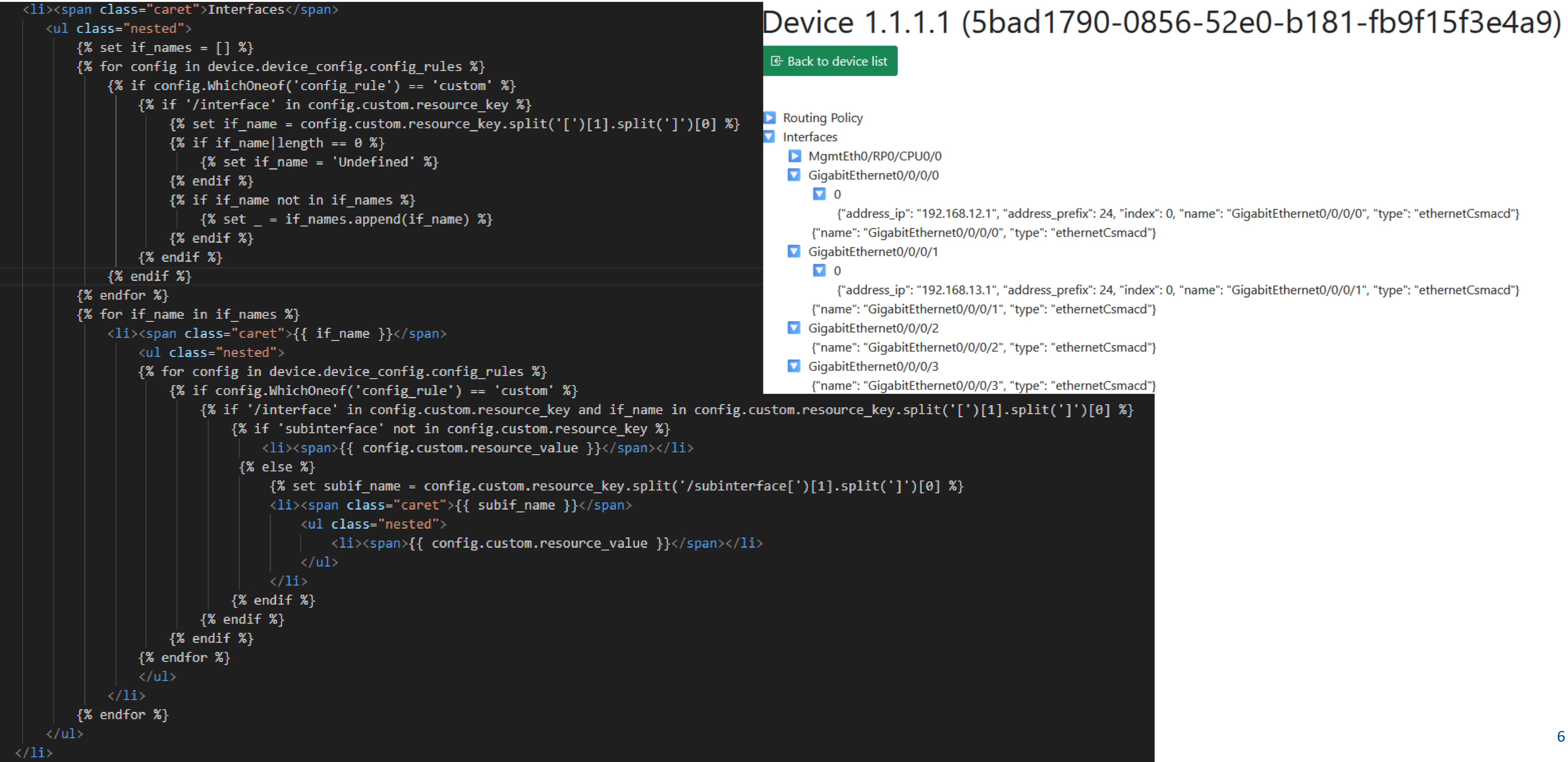
if config.WhichOneof('config_rule') == 'custom’ and not 'inventory' in config.custom.resource_key and not 'endpoints' in config.custom.resource_key



```
        <th scope="col"></th>
    </tr>
</thead>
<tbody>
    {% for config in device.device_config.config_rules %}
    {% if config.WhichOneof('config_rule') == 'custom' %}
    <tr>
        <td>
            {{ config.custom.resource_key }}
        </td>
        <td>
```

```
        <th scope="col"></th>
    </tr>
</thead>
<tbody>
    {% for config in device.device_config.config_rules %}
    {% if config.WhichOneof('config_rule') == 'custom' and not 'inventory' in config.custom.resource_key and not
    'endpoints' in config.custom.resource_key %}
    <tr>
        <td>
            {{ config.custom.resource_key }}
        </td>
        <td>
```

6 - Extend Logical Inventory: interfaces



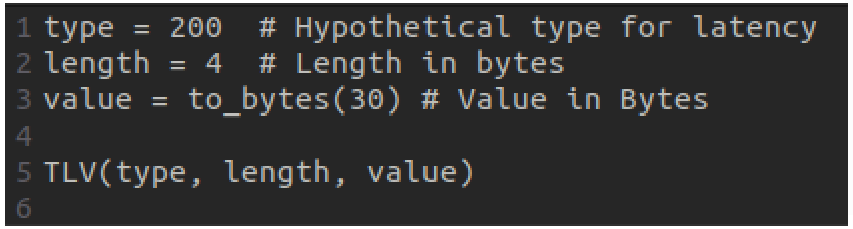
```
<li><span class="caret">Interfaces</span>
    <ul class="nested">
        {% set if_names = [] %}
        {% for config in device.device_config.config_rules %}
            {% if config.WhichOneof('config_rule') == 'custom' %}
                {% if '/interface' in config.custom.resource_key %}
                    {% set if_name = config.custom.resource_key.split('[')[1].split(']')[0] %}
                    {% if if_name|length == 0 %}
                        {% set if_name = 'Undefined' %}
                    {% endif %}
                    {% if if_name not in if_names %}
                        {% set _ = if_names.append(if_name) %}
                    {% endif %}
                {% endif %}
            {% endif %}
        {% endfor %}
        {% for if_name in if_names %}
            <li><span class="caret">{{ if_name }}</span>
                <ul class="nested">
                {% for config in device.device_config.config_rules %}
                    {% if config.WhichOneof('config_rule') == 'custom' %}
                        {% if '/interface' in config.custom.resource_key and if_name in config.custom.resource_key.split('[')[1].split(']')[0] %}
                            {% if 'subinterface' not in config.custom.resource_key %}
                                <li><span>{{ config.custom.resource_value }}</span></li>
                            {% else %}
                                {% set subif_name = config.custom.resource_key.split('/subinterface[')[1].split(']')[0] %}
                                <li><span class="caret">{{ subif_name }}</span>
                                    <ul class="nested">
                                        <li><span>{{ config.custom.resource_value }}</span></li>
                                    </ul>
                                </li>
                            {% endif %}
                        {% endif %}
                    {% endif %}
                {% endfor %}
                </ul>
            </li>
        {% endfor %}
    </ul>
</li>
```

Device 1.1.1.1 (5bad1790-0856-52e0-b181-fb9f15f3e4a9)

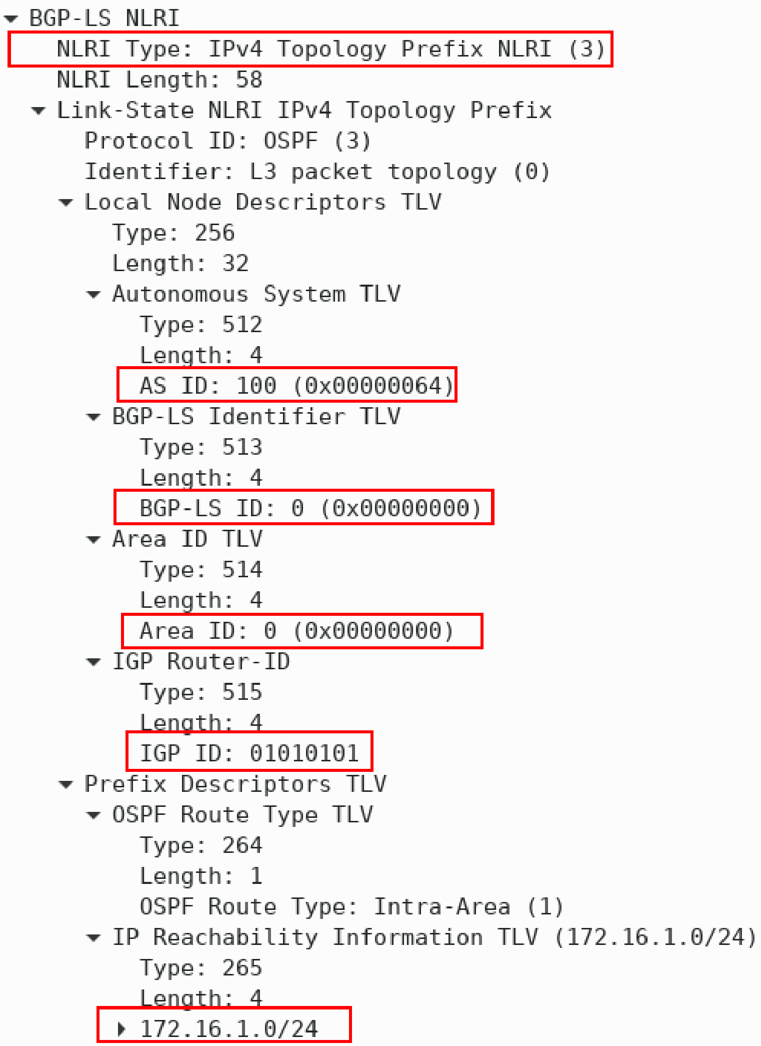
- Routing Policy
- Interfaces
  - MgmtEth0/RP0/CPU0/0
  - GigabitEthernet0/0/0/0
    - 0
      - {"address_ip": "192.168.12.1", "address_prefix": 24, "index": 0, "name": "GigabitEthernet0/0/0/0", "type": "ethernetCsmacd"}
    - {"name": "GigabitEthernet0/0/0/0", "type": "ethernetCsmacd"}
  - GigabitEthernet0/0/0/1
    - 0
      - {"address_ip": "192.168.13.1", "address_prefix": 24, "index": 0, "name": "GigabitEthernet0/0/0/1", "type": "ethernetCsmacd"}
    - {"name": "GigabitEthernet0/0/0/1", "type": "ethernetCsmacd"}
  - GigabitEthernet0/0/0/2
    - {"name": "GigabitEthernet0/0/0/2", "type": "ethernetCsmacd"}
  - GigabitEthernet0/0/0/3
    - {"name": "GigabitEthernet0/0/0/3", "type": "ethernetCsmacd"}

7. Add new BGP TLV

- Add delay-related TLV to BGP link-state update message
  - Activate it on router or ask vendor to add support
- Receive BGP update messages in TFS
  - Interrogate Routers periodically
  - Add listener to receive BGP update messages (Not Scalable)
- Parse the BGP update message in TFS
  - src/bgpls_speaker/service
  - Extract the delay message in TLV format
- Update the link stats and UI



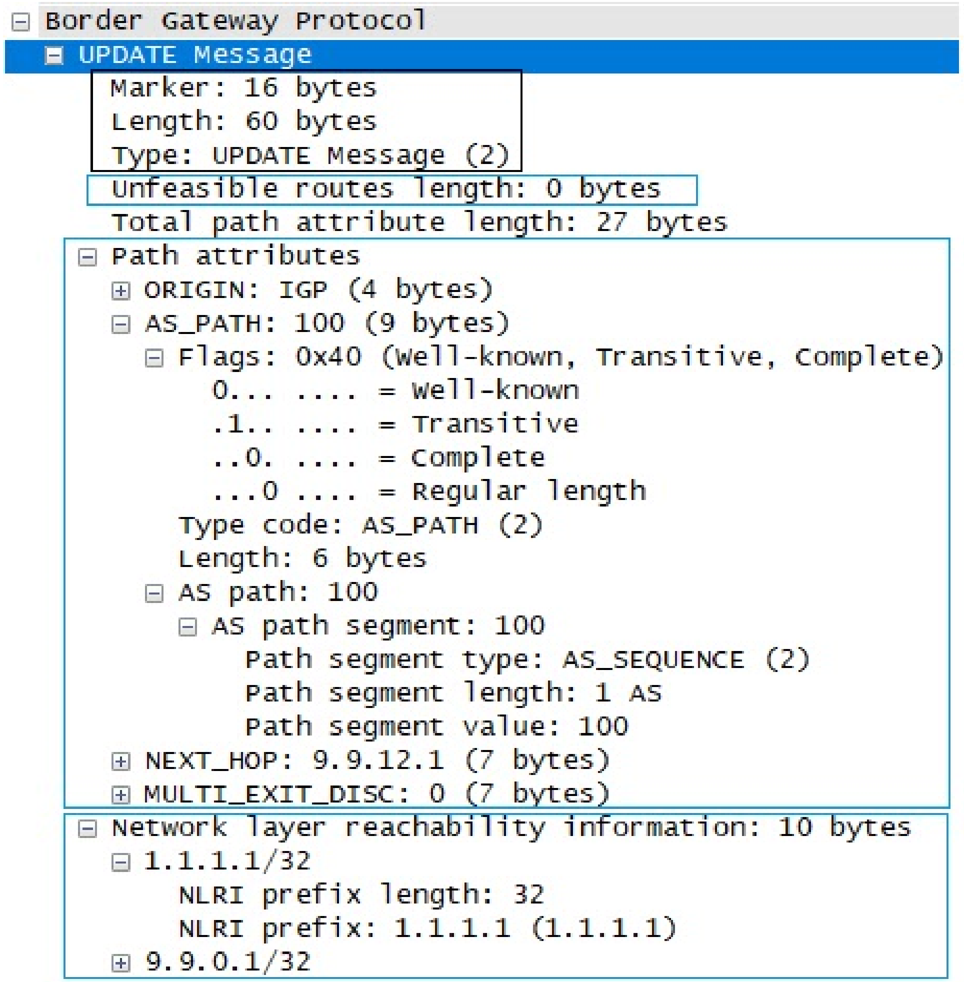
```
type = 200  # Hypothetical type for latency
length = 4  # Length in bytes
value = to_bytes(30) # Value in Bytes

TLV(type, length, value)

```



```
▾ BGP-LS NLRI
    NLRI Type: IPv4 Topology Prefix NLRI (3)
    NLRI Length: 58
  ▾ Link-State NLRI IPv4 Topology Prefix
      Protocol ID: OSPF (3)
      Identifier: L3 packet topology (0)
    ▾ Local Node Descriptors TLV
        Type: 256
        Length: 32
      ▾ Autonomous System TLV
          Type: 512
          Length: 4
          AS ID: 100 (0x00000064)
      ▾ BGP-LS Identifier TLV
          Type: 513
          Length: 4
          BGP-LS ID: 0 (0x00000000)
      ▾ Area ID TLV
          Type: 514
          Length: 4
          Area ID: 0 (0x00000000)
      ▾ IGP Router-ID
          Type: 515
          Length: 4
          IGP ID: 01010101
    ▾ Prefix Descriptors TLV
      ▾ OSPF Route Type TLV
          Type: 264
          Length: 1
          OSPF Route Type: Intra-Area (1)
      ▾ IP Reachability Information TLV (172.16.1.0/24)
          Type: 265
          Length: 4
        ▸ 172.16.1.0/24
```

8. Manage link deactivation

- Receive BGP update messages in TFS
  - Interrogate Routers periodically
  - Add listener to receive BGP update messages (Not Scalable)
- Parse the BGP update message in TFS
  - src/bgpls_speaker/service
  - Extract list of withdrawn routes
  - There is no withdrawn route in image
- Update the link state and UI



```
⊟ Border Gateway Protocol
  ⊟ UPDATE Message
      Marker: 16 bytes
      Length: 60 bytes
      Type: UPDATE Message (2)
      Unfeasible routes length: 0 bytes
      Total path attribute length: 27 bytes
    ⊟ Path attributes
      ⊞ ORIGIN: IGP (4 bytes)
      ⊟ AS_PATH: 100 (9 bytes)
        ⊟ Flags: 0x40 (Well-known, Transitive, Complete)
            0... .... = Well-known
            .1.. .... = Transitive
            ..0. .... = Complete
            ...0 .... = Regular length
          Type code: AS_PATH (2)
          Length: 6 bytes
        ⊟ AS path: 100
          ⊟ AS path segment: 100
              Path segment type: AS_SEQUENCE (2)
              Path segment length: 1 AS
              Path segment value: 100
      ⊞ NEXT_HOP: 9.9.12.1 (7 bytes)
      ⊞ MULTI_EXIT_DISC: 0 (7 bytes)
    ⊟ Network layer reachability information: 10 bytes
      ⊟ 1.1.1.1/32
          NLRI prefix length: 32
          NLRI prefix: 1.1.1.1 (1.1.1.1)
      ⊞ 9.9.0.1/32
```

TeraFlow SDN

by ETSI

Thank You!