New and Changed Information

The following table provides an overview of the significant changes up to this current release. The table does not provide an exhaustive list of all changes or the new features up to this release.

Table 1: New and Changed Information

<table>
<thead>
<tr>
<th>Cisco APIC Release Version</th>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.2(1)</td>
<td>This feature was introduced</td>
<td>--</td>
</tr>
</tbody>
</table>

About TWAMP

The Two-Way Active Measurement Protocol (TWAMP) defines a standard (RFC 5357) for measuring round-trip network performance between any two devices that support the TWAMP protocols. The TWAMP Server/Reflector is supported as part of the IP SLA responder in NX-OS. Cisco APIC configures the TWAMP support for switch groups and provides the monitoring of the test sessions and connections.

The devices must support the following TWAMP protocols:

- TWAMP-Control Protocol: Set up performance measurement sessions
- TWAMP-Test Protocol: Send and receive performance-measurement probes
Guidelines and Limitations for Configuring TWAMP

Follow these guidelines and limitations when configuring and using TWAMP:

- The TWAMP client resides on an open-source third-party TWAMP utility application that must be reachable by any of the switch nodes through a management port, in-band management port, or L3 Out interface.

TWAMP GUI Configuration

Creating TWAMP Policies Using the GUI

Creating a TWAMP Responder Policy

Procedure

Step 1 Choose Fabric > Fabric Policies.
Step 2 In the Navigation pane, choose Policies > Switch > TWAMP Responder Policies.
Step 3 Select the default responder policy or create a new one by performing the following actions:
   a) In the Navigation pane, right-click TWAMP Responder Policies, then choose Create TWAMP Responder. Alternatively, in the TWAMP Responder Policies summary table, click the Actions drop-down list and select Create TWAMP Responder Policy.
Step 4  Click Submit.

The new responder policy appears as a row in the **TWAMP Responder Policies** summary table and is now available to be included in leaf/spine policy groups.

### Creating a TWAMP Server Policy

**Procedure**

**Step 1**  In the **Navigation** pane, choose **TWAMP Server Policies**.

**Step 2**  Select the default server policy or create a new one by performing the following actions:

a)  In the **Navigation** pane, right-click **TWAMP Responder Policies**, then choose **Create TWAMP Server Policy**. Alternatively, in the **TWAMP Server Policies** work pane, click the **Actions** drop-down list and select **Create TWAMP Server Policy** Policy.

b)  In the **Name** field, enter a unique name for the server policy.

c)  For **Server State**, choose **Enabled**.

d)  In the **Inactivity Timer** field, set the value to the number of seconds after which the server checks the activity of a responder test session. If the test session is inactive, the server marks it for "clean up" at the next interval. The default is 900 seconds (15 minutes).

e)  In the **Server Port** field, enter a port number used to listen for test requests from TWAMP clients. The default port is 862.

**Step 3**  Click Submit.

The new server policy appears as a row in the **TWAMP Server Policies** summary table and is now available to be included in leaf/spine policy groups.

### Adding TWAMP Policies to Leaf/Spine Policy Groups Using the GUI

The following procedure associates TWAMP policies with a leaf/spine policy group.

**Before you begin**

This procedure requires an existing leaf or spine switch policy group or the creation of a new one. For more information regarding creating and configuring leaf/spine switch policy groups, see the *Cisco APIC Security Configuration Guide* for information about creating leaf switch policy groups and spine switch policy groups.

**Procedure**

**Step 1**  Choose **Fabric > Fabric Policies**.

**Step 2**  In the **Navigation** pane, choose **Switches > Leaf (or Spine) Switches > Policy Groups**.

**Step 3**  Select an existing policy group or create and select a new one.
Step 4 In the **Leaf/Spine Policy Group** work pane, select a policy from the **TWAMP Server Policy** drop-down list.
Step 5 Select a policy from the **TWAMP Responder Policy** drop-down list.
Step 6 Click **Submit**.

This leaf/spine policy group with the associated TWAMP policies can now be associated with a leaf/spine profile.

**Associating a TWAMP-Enabled Leaf/Spine Profile with a Policy Group Using the GUI**

**Before you begin**

This procedure requires an existing leaf/spine profile or the creation of a new one. For more information regarding creating and configuring leaf/spine profiles, see the *Cisco APIC Security Configuration Guide*. This procedure also requires an existing leaf or spine policy group enabled for TWAMP. For more information, see *Adding TWAMP Policies to Leaf/Spine Policy Groups Using the GUI*, on page 4.

You can use the Cisco APIC GUI to associate TWAMP enabled leaf or spine profiles with policy groups.

**Procedure**

**Step 1** Choose **Fabric > Fabric Policies**.
**Step 2** In the Navigation pane, choose **Switches > Leaf (or Spine) Switches > Profiles**.
**Step 3** Select an existing profile or create and select a new one.
**Step 4** In the **Leaf/Spine Switch Profile** work pane, click + in the **Switch Associations** summary table and perform the following actions:
   a) In the **Name** field, enter a unique name for the switch association.
   b) In the **Blocks** field, select a range of leaf/spine nodes to create the block.
   c) In the **Policy Group** field, select the policy group you created with the selected TWAMP policies.
   d) Click **Update**.

After completing this procedure, a third-party, an open-source TWAMP tool that generates test sessions can now be used. Once the test sessions are initiated, follow the instructions in *Viewing TWAMP Connections and Sessions Using the GUI*, on page 5.

**Viewing TWAMP Connections and Sessions Using the GUI**

Once TWAMP test sessions are initiated, perform the following procedure to view the resulting connections and sessions.

**Procedure**

**Step 1** Choose **Fabric > Inventory**.
**Step 2** In the **Navigation** pane, choose **Pod > leaf/spine_name > Protocols > TWAMP**.
**Step 3** Select **TWAMP Connections** to view a summary list of TWAMP connections. Select a specific TWAMP connection to view details.
**Step 4** Select **TWAMP Sessions** to view a summary list of TWAMP sessions. Select a specific TWAMP session to view details.
TWAMP NX-OS-Style CLI Configuration

Creating TWAMP Responder and Server Policies Using the NX-OS-Style CLI

You can use the NX-OS-style CLI to configure TWAMP responder and server policies.

Procedure

**Step 1**  Enter the global configuration mode.

**Example:**
```
apic1# config
```

**Step 2**  Create a new TWAMP responder policy.

**Example:**
In the following command, a TWAMP responder policy named "resp-1" is created:
```
apic1(config)# template twamp responder-policy resp-1
```

**Step 3**  Enable the newly created TWAMP responder policy

**Example:**
In the following command, the new TWAMP responder policy is enabled:
```
apic1(config-twamp-responder-policy)# enable
```

**Step 4**  (Optional) Configure a timeout for the responder policy.

**Example:**
In the following command example, a timeout of 600 seconds (10 minutes) is configured for the responder policy:
```
apic1(config-twamp-responder-policy)# timeout 600
```

**Step 5**  Exit from the TWAMP responder policy mode.

**Example:**
```
apic1(config-twamp-responder-policy)# exit
```

**Step 6**  Create a new TWAMP server policy.

**Example:**
In the following command, a TWAMP server policy named "serv-1" is created:
```
apic1(config)# template twamp server-policy serv-1
```

**Step 7**  Enable the newly created TWAMP responder policy

**Example:**
In the following command, the new TWAMP responder policy is enabled:
```
apic1(config-twamp-server-policy)# enable
```

**Step 8**  (Optional) Configure an inactivity timer for the server policy.

Set the value to the number of seconds after which the server checks the activity of a responder test session. If the test session is inactive, the server marks it for "clean up" at the next interval.
Command characteristics:

- Type: Integer
- Unit: Seconds
- Range: 1 to 65535
- Default: 900 (15 minutes)

**Example:**
In the following command example, an inactivity timer of 600 seconds (10 minutes) is configured for the server policy:
```
apic1(config-twamp-responder-policy)# inactivity-timer 600
```

The new responder and server policies are now available to be included in leaf/spine policy groups.

**Adding TWAMP Policies to Leaf/Spine Policy Groups Using the NX-OS-Style CLI**

You can use the NX-OS-style CLI to add TWAMP policies to a leaf or spine policy group.

**Before you begin**

This example assumes you have created TWAMP responder and server policies. For more information, see Creating TWAMP Responder and Server Policies Using the NX-OS-Style CLI, on page 6.

**Procedure**

**Step 1** Enter the global configuration mode.

**Example:**
```
apic1# config
```

**Step 2** Enter the internal fabric configuration mode.

**Example:**
```
apic1(config)# fabric-internal
```

**Step 3** (Optional) Enter a leaf policy group configuration mode.

**Example:**
In the following command, the configuration mode of a leaf policy group named "lpg-1" is entered:
```
apic1(config-fabric-internal)# template leaf-policy-group lpg-1
```

**Step 4** (Optional) Enter a spine policy group configuration mode.

**Example:**
In the following command, the configuration mode of a spine policy group named "spg-1" is entered:
```
apic1(config-fabric-internal)# template spine-policy-group spg-1
```

**Step 5** Enter the TWAMP policy you want associated with the leaf or spine policy group.

**Example:**
In the following command, TWAMP server policy named "serv-1" is associated with the spine policy group:

```
apic1(config-spine-policy-group)# inherit twamp-server-policy serv-1
```

Note: If you are configuring leaf policy groups, spine policy groups, or both, both TWAMP responder and server policies must be associated with each.

A leaf and/or spine policy group is now associated with a TWAMP responder and server policy. Any leaf or spine profile that associates with a respective leaf or spine policy group, that is configured with a TWAMP responder and server policy, can be measured using a TWAMP client with access to the network. For more information, see Associating a Leaf Profile with a TWAMP-Enabled Leaf Policy Group Using the NX-OS-Style CLI, on page 8 and Associating a Spine Profile with a TWAMP-Enabled Spine Policy Group Using the NX-OS-Style CLI, on page 9.

**Associating a Leaf Profile with a TWAMP-Enabled Leaf Policy Group Using the NX-OS-Style CLI**

You can use the NX-OS-style CLI to associate a TWAMP-enabled leaf policy group with a leaf profile.

**Procedure**

**Step 1** Enter the global configuration mode.

Example:
```
apic1# config
```

**Step 2** Enter the internal fabric configuration mode.

Example:
```
apic1(config)# fabric-internal
```

**Step 3** Create or enter a leaf profile configuration mode.

Example:
In the following command, the configuration mode of a leaf profile named "leaf-prof-1" is entered:
```
apic1(config-fabric-internal)# leaf-profile leaf-prof-1
```

**Step 4** Enter a leaf group configuration mode.

Example:
In the following command, the configuration mode of a leaf group named "leaf-group-1" is entered:
```
apic1(config-leaf-profile)# leaf-group leaf-group-1
```

**Step 5** Enter the name of the leaf policy group containing the TWAMP policies.

Example:
In the following command, a leaf policy group named "lpg-1" is associated with the leaf profile:
```
apic1(config-leaf-group)# leaf-policy-group lpg-1
```

A leaf policy group containing TWAMP responder and server policies is now associated with the leaf profile. Any leaf device belonging to the leaf profile can be measured using a TWAMP client with access to the network.
**Associating a Spine Profile with a TWAMP-Enabled Spine Policy Group Using the NX-OS-Style CLI**

You can use the NX-OS-style CLI to associate a TWAMP-enabled spine policy group with a spine profile.

**Procedure**

**Step 1** Enter the global configuration mode.

*Example:*

```
apic1# config
```

**Step 2** Enter the internal fabric configuration mode.

*Example:*

```
apic1(config)# fabric-internal
```

**Step 3** Create or enter a spine profile configuration mode.

*Example:*

In the following command, the configuration mode of a spine profile named "spine-prof-1" is entered:

```
apic1(config-fabric-internal)# spine-profile spine-prof-1
```

**Step 4** Enter a spine group configuration mode.

*Example:*

In the following command, the configuration mode of a spine group named "spine-group-1" is entered:

```
apic1(config-spine-profile)# spine-group spine-group-1
```

**Step 5** Enter the name of the spine policy group containing the TWAMP policies.

*Example:*

In the following command, a spine policy group named "spg-1" is associated with the spine profile:

```
apic1(config-spine-group)# spine-policy-group spg-1
```

A spine policy group containing TWAMP responder and server policies is now associated with the spine profile. Any spine device belonging to the spine profile can be measured using a TWAMP client with access to the network.

**Viewing TWAMP Connections and Sessions Using the NX-OS-Style CLI**

Use this example to view TWAMP connections and sessions for a current test.

**Before you begin**

This example assumes that you have initiated a TWAMP test session.

**Procedure**

**Step 1** View a TWAMP connection.

*Example:*

```
show ip sla twamp connection detail

Connection Id: 21
Client IP Address: 30.30.30.2
Client Port: 58316
Client VRF: default
Mode: Unauthenticated
Connection State: Connected
Control State: Active
Number of Test Requests - 0:1

Step 2  View a TWAMP session.

Example:
show ip sla twamp session source-ip 30.30.30.2 source-port 50790

IP SLAs Responder TWAMP is: Enabled
Recvr Addr: 30.30.30.1
Recvr Port: 7147
Sender Addr: 30.30.30.2
Sender Port: 50790
Sender VRF: default
Session Id: 30.30.30.1:15918249420668138422:DF55BE9
Connection Id: 21
Mode: UnAuthorized
DSCP: 0
Pad Length: 0
Number of Packets Received: 0

**TWAMP REST API Configuration**

**Creating TWAMP Policies Using the REST API**

**Procedure**

Create a TWAMP responder and server policy.

Example:
TWAMP Responder

POST https://10.197.144.60/api/node/mo/uni/fabric/twampRespP-twampResponder.json
{
   "totalCount": "1",
   "imdata": [
   {
      "twampResponderPol": {
         "attributes": {
            "adminSt": "enabled",
            "annotation": "",
            "descr": "",
            "dn": "uni/fabric/twampRespP-twampResponder",
            "name": "twampResponder",
            "nameAlias": "",
            "ownerKey": "",
            "ownerTag": "",
            "timeout": "60"
         }
      }
   }
}
TWAMP Server

POST https://10.197.144.60/api/node/mo/uni/fabric/twampServP-twampServer.json
{
  "totalCount": "1",
  "imdata": [
    {
      "twampServerPol": {
        "attributes": {
          "adminSt": "enabled",
          "annotation": "",
          "descr": "",
          "dn": "uni/fabric/twampServP-twampServer",
          "inactivityTimer": "60",
          "name": "twampServer",
          "nameAlias": "",
          "ownerKey": "",
          "ownerTag": "",
          "port": "862"
        }
      }
    }
  ]
}

Adding TWAMP Policies to Leaf/Spine Policy Groups Using the REST API

You can use the REST API to add TWAMP policies to a leaf or spine policy group.

Before you begin

This example assumes you have created TWAMP responder and server policies. For more information, see Creating TWAMP Policies Using the REST API, on page 10.

Procedure

Add a TWAMP responder and server policy to a leaf and spine policy group.

Example:

Add TWAMP responder and server policies to a leaf policy group.

POST https://10.197.144.60/api/node/mo/uni/fabric/funcprof/
  lenodepgrp-twampLeafPolGrp.json
{
  "totalCount": "1",
  "imdata": [
    {
      "fabricLeNodePGrp": {
        "attributes": {
          "annotation": "",
          "descr": "",
          "dn": "uni/fabric/funcprof/lenodepgrp-twampLeafPolGrp",
          "name": "twampLeafPolGrp",
          "nameAlias": ""
        }
      }
    }
  ]
}
Example:

Add TWAMP responder and server policies to a spine policy group.

POST https://10.197.144.60/api/node/mo/uni/fabric/funcprof/
spnodepgrp-twampSpinePolGrp.json

```json
{
  "totalCount": "1",
  "imdata": [
    {
      "fabricSpNodePGrp": {
        "attributes": {
          "annotation": "",
          "descr": "",
          "dn": "uni/fabric/funcprof/spnodepgrp-twampSpinePolGrp",
          "name": "twampSpinePolGrp",
          "nameAlias": "",
          "ownerKey": "",
          "ownerTag": ""
        },
        "children": [
          {
            "fabricRsTwampResponderPol": {
              "attributes": {
                "annotation": "",
                "tnTwampResponderPolName": "twampResponder"
              }
            }
          },
          {
            "fabricRsTwampServerPol": {
              "attributes": {
                "annotation": "",
                "tnTwampServerPolName": "twampServer"
              }
            }
          }
        ]
      }
    }
  ]
}
```
Leaf and/or spine policy groups are associated with a TWAMP responder and server policy. Any leaf or spine profile that associates with a respective leaf or spine policy group, that is configured with a TWAMP responder and server policy, can be measured using a TWAMP client with access to the network. For more information see Associating a Leaf/Spine Profile with a TWAMP-Enabled Policy Group Using the REST API, on page 13.

**Associating a Leaf/Spine Profile with a TWAMP-Enabled Policy Group Using the REST API**

Use the example below to associate a leaf profile with a TWAMP-enabled leaf policy group and a spine profile with a TWAMP-enabled spine policy group.

**Before you begin**

This example assumes that a leaf or spine policy group has been created and includes support for TWAMP responder and server policies. See

**Procedure**

Associate a leaf profile with a TWAMP-enabled leaf policy group or a spine profile with a TWAMP-enabled spine policy group.

**Example:**

Associate a leaf profile with a leaf policy group.

```json
POST https://10.197.144.60/api/node/mo/uni/fabric/leprof-leafProfile.json
{
   "totalCount": "1",
   "imdata": [
      { "fabricLeafP": {
         "attributes": {
            "annotation": "",
            "descr": "",
            "dn": "uni/fabric/leprof-leafProfile",
            "name": "leafProfile",
            "nameAlias": "",
            "ownerKey": "",
            "ownerTag": ""
         },
         "children": [
            { "fabricLeafS": {
               "attributes": {
                  "annotation": "",
                  "descr": "",
                  "name": "leafSel",
                  "nameAlias": "",
                  "ownerKey": "",
                  "ownerTag": "",
                  "type": "ALL"
               },
               "children": [ ]
            },
            { "fabricRsLeNodePGrp": {
               "attributes": {
                  "annotation": "",
                  "tDn": "uni/fabric/funcprof/
               },
               "children": [ ]
            }
         ]
      }
   ]
}
```
Example:

Associate a spine profile with a spine policy group.

POST https://10.197.144.60/api/node/mo/uni/fabric/spprof-spineProfile.json
{
  "totalCount": "1",
  "imdata": [
    {
      "fabricSpineP": {
        "attributes": {
          "annotation": "",
          "descr": "",
          "dn": "uni/fabric/spprof-spineProfile",
          "name": "spineProfile",
          "nameAlias": "",
          "ownerKey": "",
          "ownerTag": ""
        },
        "children": ["fabricSpineS": {
          "attributes": {
            "annotation": "",
            "descr": "",
            "name": "spineSel",
            "nameAlias": "",
            "ownerKey": "",
            "ownerTag": "",
            "type": "ALL"
          },
          "children": ["fabricRsSpNodePGrp": {
            "attributes": {
              "annotation": "",
              "tDn": "uni/fabric/funcprof/spnodegrp-twampSpinePolGrp"
            }
          }
        ]
      }
    ]
  }
}
**Viewing TWAMP Connections and Sessions Using the REST API**

**Procedure**

View TWAMP connection or session information for a specific leaf or spine.

**Example:**

**TWAMP connection information.**

method: GET  
url: https://10.197.144.60/api/node/class/topology/pod-1/node-101/twampConnection.json  
response: 
{"totalCount":"0","subscriptionId":"72057679954247834","imdata":[]}  
timestamp: 12

**Example:**  
**TWAMP session information.**

method: GET  
url: https://10.197.144.60/api/node/class/topology/pod-1/node-101/twampSession.json  
response: 
{"totalCount":"1","subscriptionId":"72057679954247836","imdata":[]}  
timestamp: 12:54:47 DEBUG
Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.