Autonomic Networking Infrastructure

Autonomic Networking makes network devices intelligent by introducing self-management concepts that simplify network management for the network operator.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Autonomic Networking

- The Autonomic Control Plane (ACP) is built automatically only across Ethernet ports. It utilizes only IPv6 addressing.
- If the device has no start-up configuration, all interfaces are up by default, to exchange the adjacency discovery (AD) messages.
- The ACP is automatically built between two adjacent devices supporting the autonomic networking infrastructure. The interfaces on both devices need to be up (and be Ethernet interfaces). The device either needs to be unconfigured (greenfield rollout) or have an autonomic networking configured explicitly.
- The ACP can also automatically be built between two adjacent devices if there is an intervening non-autonomic layer 2 cloud such as a Metro-Ethernet service. This is achieved by the Channel Discovery protocol (CD) on the autonomic devices, which probes for working VLAN encapsulations.
To build the ACP across intervening non-autonomic L3 devices, you need to explicitly configure a tunnel between the autonomic devices and enable autonomic adjacency-discovery on this tunnel.

Autonomic Registrar, commonly known as registrar, is required for the Autonomic Networking Infrastructure (ANI) feature to work. At least one device in the network must be configured as a registrar to enroll new devices into the autonomic domain. In a network where all required devices are already enrolled into the autonomic domain, a registrar is not required.

Each registrar supports only one autonomic domain. The registrar is needed only when new autonomic devices join the domain.

All new devices must have a physical connectivity to at least one autonomic device to contact the registrar for authentication and authorization.

A device can only be enrolled into one autonomic domain. Two devices enrolled into different domains will not build the autonomic control plane between each other.

For autonomic intent, the registrar must be configured with domain ID.

For Zero Touch Bootstrap to happen, there must be no startup-config file and the config-register must remain default i.e, 0x2102.

Restrictions for Autonomic Networking

General Restrictions

- Autonomic networking only supports unique device identifier (UDI)-based devices.
- Autonomic networking and Zero Touch Provisioning (ZTP) are different zero touch solutions. It is recommended that you do not test or use autonomic networking and ZTP at the same time.

Information About Autonomic Networking

Overview of Autonomic Networking

The aim of autonomic networking is to create self-managing networks to overcome the rapidly growing complexity of the Internet and other networks to enable their further growth. In a self-managing autonomic system, network management takes on a new role: instead of controlling the network elements individually and directly, the administrator defines network-wide policies and rules that guide the self-management process.
The following diagram provides a high-level architecture of an autonomic network.

**Figure 1: High-Level Architecture of an Autonomic Network**

Autonomic Networking is controlled by a separate software entity running on top of traditional operating systems that include networking components, such as IP, Open Shortest Path First (OSPF), and so forth. Traditional networking components are unchanged and unaware of the presence of the autonomic process. The autonomic components use normal interfaces that are exposed by the traditional networking components and interact with different devices in the network. The autonomic components securely cooperate to add more intelligence to devices so that the devices in an autonomic network can autonomically configure, manage, protect, and heal themselves with minimal operator intervention. They can also securely consolidate their operations to present a simplified and abstracted view of the network to the operator.

**Autonomic Networking Infrastructure**

The Autonomic Networking Infrastructure (ANI) feature simplifies the network bootstrap functionality by removing the need for any kind of prestaging, thereby allowing devices to join a domain securely, after which devices can be configured. The goal of the Autonomic Networking Infrastructure feature is to make new and unconfigured devices securely reachable by an operator or network management system. This is carried out in the following steps:

1. One device is defined and configured as the registrar. The registrar is the first autonomic domain device.
2. The network administrator collects a list of legitimate device identifiers of the devices to be added to the network. This list controls the devices that are added to the autonomic domain. Devices are identified by their unique device identifier (UDI). The list is compiled as a simple text file, one UDI per line. This step is optional because in the absence of a whitelist, all devices are allowed to join the domain. A whitelist is an approved list of entities that is provided a particular privilege, service, mobility, access, or recognition. Whitelisting means to grant access.
3. The whitelist of known devices is uploaded to the registrar as part of its configuration. This step is optional.
4. Any new autonomic device that is directly connected to the registrar, or another already enrolled domain device, will automatically receive a domain certificate from the registrar.
5 The autonomic control plane is automatically established across the autonomic domain to make new devices reachable.

The benefits of Autonomic Networking Infrastructure are as follows:

- Autonomic discovery of Layer 2 topology and connectivity by discovering how to reach autonomic neighbors.
- Secure and zero touch identity of new devices by using the device name and domain certificate.
- A virtual autonomic control plane that enables communications between autonomic nodes.

Autonomic behavior is enabled by default on new devices. To enable autonomic behavior on existing devices, use the autonomic connect command. To disable, use the no form of this command.

The components of autonomic networking are as follows:

- Registrar—A domain-specific registration authority in a given enterprise that validates new devices in the domain, provides them with domain-wide credentials, and makes policy decisions. Policy decisions can include whether a new device can join a given domain based on a preloaded whitelist. The registrar also has a database of devices that join a given domain and the device details.
- Channel Discovery—Used to discover reachability between autonomic nodes across nonautonomic Layer 2 networks.
- Adjacency Discovery—Used to discover autonomic neighbors. Adjacency discovery is done on Layer 3. It is also possible to discover autonomic neighbors across pre-established Layer 3 generic routed encapsulation (GRE) tunnels.

New Device Joining the Autonomic Network

The figure below illustrates how a new device joins an autonomic network.

Figure 2: New Device Joining the Autonomic Network

1 The new device sends out a hello message to the neighbor. In this case, the neighbor is part of an autonomic network domain.
2 The hello message includes the unique device identifier (UDI) of the new device.
3 The autonomic device acts as a proxy and allows the new device to join this autonomic network domain. The autonomic network device advertises itself with the domain information to its Layer 3 neighbors.
4 The new device is validated with the autonomic registrar on receiving the autonomic network hello message from the neighbor and thereby detecting the UDI information.
The new device advertises its domain certificate in its *hello* message with all neighbors. The neighbor information is exchanged every 10 seconds.

If the neighbor information changes, the entry is deleted and neighbor discovery is restarted. In the absence of a domain certificate and devices working with UDI, UDI is exchanged at a 10-second interval.

**Working of the Registrar**

One device in the network is configured as the registrar to validate all new devices locally; and to accept all devices that try to join the domain managed by the registrar. The registrar issues a domain certificate to devices using the domain name provided by the administrator.

The domain name is a group of devices in the network managed by the same set of rules. Any neighbor communicating with the registrar using the UDI certificate is sent an invitation to join the domain by registrar.

Currently, the registrar validates the new devices based on an optional list of UDIs that is specified when configuring the registrar. The list specifies a set of devices that are allowed to join the network.

**Figure 3: Provisioning Registrar in an Autonomic Network**

The registrar maintains a database of all devices that join or fail to join the domain. The failed devices can try to join the domain again. The registrar database tracks all devices in the domain and associates each device with the following states:

- **Accepted**—Devices that successfully join the domain.
• **Pending**—Devices that are in the process of joining the domain.

• **Quarantine**—Devices that failed to join the domain.

### Channel Discovery in Autonomic Networking

### Service Discovery in Autonomic Networking

Autonomic networking uses the multicast Domain Name System (mDNS) infrastructure to discover various services required by the devices in the autonomic networking domain. Few of the services discovered by the network using the mDNS infrastructure are: the AAA server, the configuration server, the syslog server, and the autonomic networking registrar. Autonomic networking listens to the mDNS advertisements on all the devices in the domain. Autonomic networking initiates the mDNS advertisements from the devices hosting the services.

### Autonomic Control Plane

When a new device in the domain receives a domain certificate, it exchanges the domain certificate in the hello messages with its neighbors. This creates an autonomic control plane between two autonomic devices of the same domain. There are different types of autonomic control planes that can be created based on the different capabilities of the devices. The autonomic control plane is established by using the following mechanisms:

- Configuring a loopback interface.
- Dynamically assigning an IPv6 address to the loopback interface.
- Configuring an autonomic VPN routing and forwarding (VRF).

### IPsec on Link Local GRE IPv6 Tunnels in Autonomic Networking

Autonomic control plane (ACP) is created by building hop-by-hop IPv6 generic routing encapsulation (GRE) tunnels between the pairs of autonomic neighbors. The GRE tunnels require a valid source address and a destination address on the physical interfaces. As the GRE tunnels are hop-by-hop, autonomic networking uses the link local addresses of the physical interfaces as the source or destination addresses of the tunnels. As the physical interfaces have no global IPv6 addresses, IPsec and IKE work with the link local addresses of the physical interfaces.
# How to Configure Autonomic Networking

## Configuring the Registrar

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Step 1    | enable           | Enables privileged EXEC mode.  
**Example:**  
Device> enable |
|           |                  | • Enter your password if prompted. |
| Step 2    | configure terminal | Enters global configuration mode. |
|           |                  | **Example:**  
Device# configure terminal |
| Step 3    | autonomic connect | (Optional) Connects a nonautonomic management device, such as an authentication, authorization, and accounting (AAA) server, or a syslog server, to the autonomic network. |
|           |                  | **Example:**  
Device(config)# autonomic connect |
| Step 4    | autonomic registrar | Enables a device as a registrar and enters registrar configuration mode. |
|           |                  | **Example:**  
Device(config)# autonomic registrar |
| Step 5    | domain-id domain-name | Represents a common group of all devices registering with the registrar. |
|           |                  | **Example:**  
Device(config-anra)# domain-id abc.com |
| Step 6    | device-accept "udi" | (Optional) Specifies the Unique Device Identifier (UDI) of a quarantined device to be accepted in the autonomic domain.  
**Note** This command is not required when configuring the registrar. It is required only after the registrar is enabled to accept previously quarantined devices.  
• udi—Must be entered in double quotes to ensure that spaces and special characters are inclusive in the argument.  
**Example:**  
Device(config-anra)# device-accept "FID:A901-12C-FT-D SN:CAT1902U88Y" |
| Step 7    | whitelist filename | (Optional) Allows loading a file on the local device that contains a list of devices to be accepted in a given domain.  
• The file must contain one UDI entry per line.  
**Example:**  
Device(config-anra)# whitelist flash:whitelist.txt |
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>no shut</td>
<td>Enables the autonomic registrar.</td>
</tr>
</tbody>
</table>

**Example:**
```
Device(config-anra)# no shut
```

**Step 9**
Exits registrar configuration mode and returns to global configuration mode.

**Example:**
```
Device(config-anra)# exit
```

**Step 10**
Exits global configuration mode and returns to privileged EXEC mode.

**Example:**
```
Device(config)# exit
```

---

### Verifying and Monitoring Autonomic Networking Configuration

**Procedure**

**Step 1**
**enable**

**Example:**
```
Device> enable
```
Enables privileged EXEC mode.

- Enter your password if prompted.

**Step 2**
**show autonomic device**

**Example:**
```
Device# show autonomic device
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDI</td>
<td>PID:A901-12C-FT-D SN:CAT1602U00U</td>
</tr>
<tr>
<td>Device ID</td>
<td>0006.f6ac.3be0-2</td>
</tr>
<tr>
<td>Domain ID</td>
<td>manual-cisco</td>
</tr>
<tr>
<td>Domain Certificate</td>
<td>(sub:&lt;ou=manual-cisco+serialNumber=PID:A901-12C-FT-DSN:CAT1602U00U,cn=0006.f6ac.3be0-2&gt;)</td>
</tr>
</tbody>
</table>

Displays the current state of an autonomic device including the global details.

**Step 3**
**show autonomic neighbors [detail]**

**Example:**
```
Device# show autonomic neighbors detail
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDI</td>
<td>&quot;PID:A901-12C-FT-D SN:CAT1602U0C3&quot;</td>
</tr>
<tr>
<td>Device ID</td>
<td>0006.f6ac.3be0-4</td>
</tr>
</tbody>
</table>

---
How to Configure Autonomic Networking

Displays information about the discovered neighbors.

**Step 4**

```
show autonomic registrar devices [accepted | quarantine | whitelist]
```

**Example:**

```
Device# show autonomic registrar

Domain ID manual-cisco
Whitelist
Database URL nvramp
Status Autonomic Registrar Live
Address FD8F:E354:CCF9:0:6:F6AC:3BE0:1
Certificate (sub:) cn=ANRA-CS
```

Displays information about the autonomic registrar.

**Step 5**

```
show autonomic control plane [detail]
```

**Example:**

```
Device# show autonomic control-plane

VRF Name cisco_autonomic
Device Address FD08:2EEF:C2EE::D253:5185:547A
RPL Type = Node, Inst-Id = 0, OCP = 0, Mode = Storing
Neighbor Name = 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
ACL Channel ACP Security
---------------------------------------------------------------------
PID:A901-4C-F-D SN:CAT1611U085 Tunnel100000
```

Displays information about the autonomic control plane.

**Step 6**

```
show autonomic interfaces
```

Displays information about the interfaces in the autonomic domain.

**Step 7**

```
debug autonomic {bootstrap | neighbor-discovery database | registrar | services} {aaa | all | ntp | events | packets} {info | moderate | severe}
```

Enables debugging autonomic network.

**Step 8**

```
clear autonomic {device | neighbor UDI}
```

Clears or resets autonomic information.

- The **clear autonomic device** command clears or resets all the device specific AN information, including the information obtained in bootstrapping process.

- The **clear autonomic neighbor** command clears the neighbor-related information learned in the neighbor discovery. If no neighbor is specified, it clears the entire neighbor database.
Additional References for Autonomic Networking

### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Command List, All Releases</td>
</tr>
<tr>
<td>Autonomic Networking commands</td>
<td>Cisco IOS Autonomic Networking Command Reference</td>
</tr>
</tbody>
</table>

### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support and Documentation website provides online resources to</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
<tr>
<td>download documentation, software, and tools. Use these resources to install</td>
<td></td>
</tr>
<tr>
<td>and configure the software and to troubleshoot and resolve technical issues</td>
<td></td>
</tr>
<tr>
<td>with Cisco products and technologies. Access to most tools on the Cisco</td>
<td></td>
</tr>
<tr>
<td>Support and Documentation website requires a Cisco.com user ID and password.</td>
<td></td>
</tr>
</tbody>
</table>

### Feature Information for Autonomic Networking

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.