



Configuring Autonomic Networking

- [Finding Feature Information, page 1](#)
- [Prerequisites for Autonomic Networking, page 1](#)
- [Restrictions for Autonomic Networking, page 2](#)
- [Information About Autonomic Networking, page 3](#)
- [How to Configure Autonomic Networking, page 6](#)
- [Verifying and Monitoring Autonomic Networking, page 8](#)
- [Configuration Examples for Autonomic Networking, page 10](#)
- [Additional References for Autonomic Networking, page 11](#)
- [Feature History and Information for Autonomic Networking, page 12](#)

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 at the end of this module.

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

Prerequisites for Autonomic Networking

- The Autonomic Networking Infrastructure feature supports only Ethernet ports and IPv6 addresses.
- All interfaces are up by default to exchange adjacency discovery messages if there is no startup configuration in the corresponding device.
- The Autonomic Control Plane is automatically built between two adjacent devices supporting the autonomic networking infrastructure. The Ethernet interfaces on both devices need to be up, and the

device should either be unconfigured (greenfield rollout) or have autonomic networking configured explicitly.

- The Autonomic Control Plane can also be automatically built between two adjacent devices if there is an intervening nonautonomic layer 2 cloud such as a Metro ethernet service. This is achieved by the Channel Discovery protocol on the autonomic devices. This protocol probes for working VLAN encapsulations.
- To build the ACP across intervening nonautonomic L3 devices, you should 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 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 the 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.
- To contact the registrar for enrolment to the autonomic domain, all new devices must have L2 reachability to at least one device that is already enrolled to the domain. If there is no L2 reachability, user needs to configure the tunnel between the devices and configure autonomic adjacency discovery on them.
- A device can be enrolled only into one autonomic domain. Two devices enrolled into different domains will not build the autonomic control plane between each other.
- Autonomic intent can be configured only on the registrar and from there it is propagated to all the devices in the domain.
- For Zero Touch Bootstrap to take place, there must be no startup-config file present and the config-register must remain default which is 0x2102.

Restrictions for Autonomic Networking

- Autonomic networking supports only unique device identifier (UDI) -based devices.
- Autonomic networking and Zero Touch Provisioning (ZTP) are different zero touch solutions. We recommend that you do not test or use autonomic networking and ZTP at the same time.
- All the devices in an autonomic network should be contiguously autonomic. If there is no continuity, manual configuration is required to configure a tunnel through a nonautonomic network.
- Only untagged probes and channel are supported.
- Devices running Cisco IOS XE Denali 16.3.x Release and later are not compatible with devices running releases earlier than IOS XE 3.18 or 15.6(01)T. To facilitate interwork between these devices, autonomic adjacency discovery should be configured on the interfaces.
- When autonomic networking is enabled, you must not disable IPv6 unicast routing manually.
- The autonomic Registrar functionality is not supported in Cisco Catalyst 9300 and Cisco Catalyst 9500 switches.

Information About Autonomic Networking

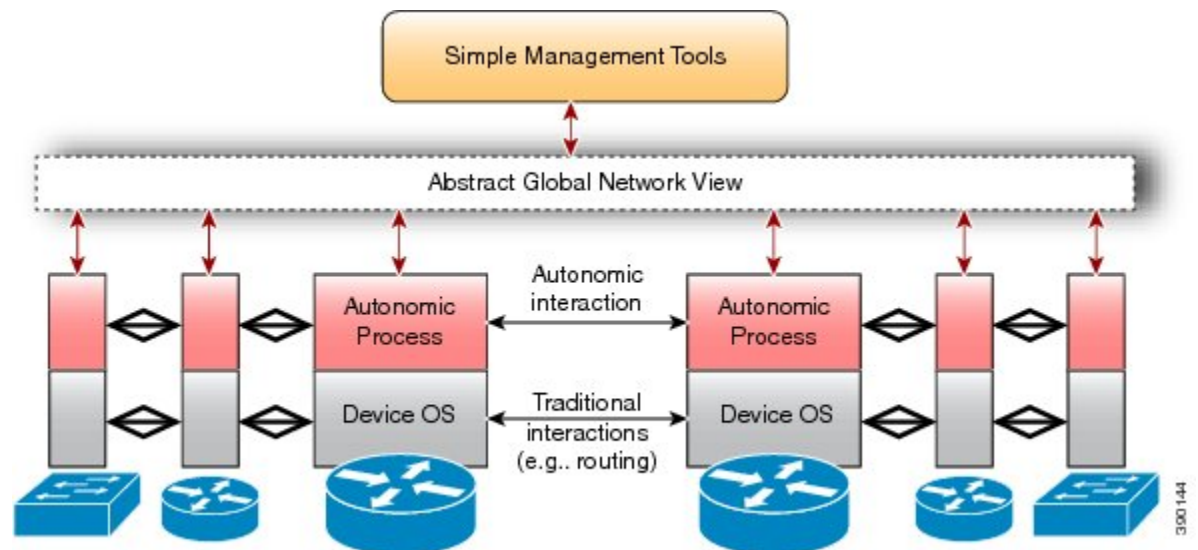
Autonomic networking makes network devices intelligent by introducing self-management concepts that simplify network management for network operators.

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 them to grow further. In a self-managing autonomic system, network management takes on a new role where, instead of controlling the network elements individually and directly, the administrator can define network-wide policies and rules to guide the self-management process.

The following figure 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 autonomously 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 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 reachable by an operator or network management system, securely. This is carried as described here:

- 1 A device is defined and configured as the registrar. This registrar is the first autonomic domain device.
- 2 This step is optional. 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 the 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 (Optional) The whitelist of known devices is uploaded to the registrar as part of its configuration.
- 4 Any new autonomic device that is directly connected to the registrar, or another 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** command. To disable, use the **no** form of this command.

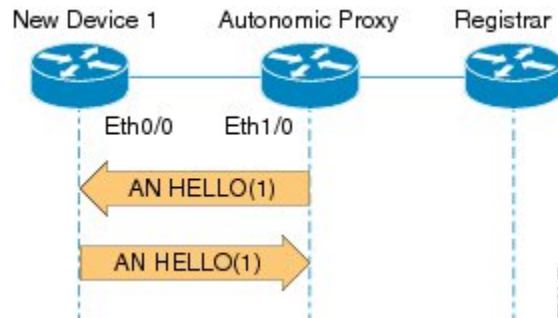
The components of autonomic networking are as follows:

- **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 a decision on 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 performed 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 following figure 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 On receiving the autonomic network hello message from the neighbor and detecting the UDI information, the new device is validated with the autonomic registrar.
- 5 The new device advertises its domain certificate in its hello message with all neighbors. The neighbor information is exchanged every 10 seconds.



Note

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.

Channel Discovery in Autonomic Networking

Channel Discovery occurs automatically on all the interfaces when Autonomic Networking is enabled on the device. Note that autonomic Networking is enabled by default on devices with no configuration (greenfield devices, and assuming they have AN functionality), but will be passive. They will only be able to receive and answer CD probes, which are L2 frames. Only a device with domain certificate or one that is already enrolled to a domain can send out CD probes on all of its Ethernet interfaces that are up. As a result of this, neighbors will be dynamically discovered. The probing will continue over time, so that newly added neighbors are discovered over time.

Adjacency Discovery in Autonomic Networking

After a channel is established, the proxy will send ND Hello messages to the new device, that is the one that is already enrolled in the domain and can act as a proxy for a new device joining the domain. The new device will send AN Hello messages in response back to the proxy. The Hello messages consist of an identification for the new device (UDI). On receiving AN Hello messages from the new device and detecting the UDI

information, the AN proxy will send the details to the Autonomic Networking Registrar (ANR) for validating this new device.

Service Discovery in Autonomic Networking

Autonomic networking uses the multicast Domain Name System (mDNS) infrastructure to locate the various services required by the devices in the autonomic networking domain. A 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. From the devices hosting the services, autonomic networking initiates the mDNS advertisements.

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 autonomic VPN routing and forwarding (VRF).

How to Configure Autonomic Networking

Configuring the Registrar

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	autonomic Example: Device# autonomic	Enables autonomic networking.

	Command or Action	Purpose
Step 4	domain-id <i>domain-name</i> Example: Device(config-registrar)# domain-id abc.com	Represents a common group of all devices registered with the registrar.
Step 5	device-accept <i>udi</i> Example: Device(config-registrar)# device-accept PID:A901-12C-FT-D SN:CAT1902U88Y	(Optional) Specifies the 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.
Step 6	whitelist <i>filename</i> Example: Device(config-registrar)# whitelist flash:whitelist.txt	(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. Note If this command is not configured, all the devices are accepted into the domain.
Step 7	no shut Example: Device(config-registrar)# no shut	Enables the autonomic registrar.
Step 8	exit Example: Device(config-registrar)# exit	Exits registrar configuration mode and returns to global configuration mode.
Step 9	exit Example: Device(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.

Configuring the Intent

Before You Begin

A registrar must be configured with a domain ID and must be up by executing the **no shutdown** command.



Note

The intent can only be configured on the registrar.

Procedure

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: Device> enable	<ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	autonomic intent Example: Device(config)# autonomic intent	Configures intent on the registrar and enters intent configuration mode.
Step 4	acp outer-vlans <i>vlan-id-range</i> Example: Device(config-intent)# acp outer-vlans 30,35,50-60,25,80	Configures the VLAN range.
Step 5	end Example: Device(config-intent)# end	Exits intent configuration mode and returns to privileged EXEC mode.

Verifying and Monitoring Autonomic Networking

Verifying and Monitoring Autonomic Networking Configuration

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	show autonomic device Example: Device# show autonomic device	Displays the current state of an autonomic device including the global details.
Step 3	show autonomic neighbors [detail] Example: Device# show autonomic neighbors detail	Displays information about the discovered neighbors.

	Command or Action	Purpose
Step 4	show autonomic control-plane [detail] Example: Device# show autonomic control-plane detail	Displays information about the autonomic control plane.
Step 5	show autonomic l2-channels [detail] Example: Device# show autonomic l2-channels	Displays the results of Channel Discovery.
Step 6	show autonomic interfaces Example: Device# show autonomic interfaces	Displays information about the interfaces in the autonomic domain.
Step 7	debug autonomic {Bootstrap Channel-Discovery Infra Intent Neighbor-Discovery Registrar Services } {aaa all ntp events packets} {info moderate severe}	Enables debugging of the autonomic network.
Step 8	clear autonomic {device neighbor UDI registrar accepted-device device UDI}	Clears or resets autonomic information. <ul style="list-style-type: none"> • The clear autonomic device command clears or resets all the device-specific autonomic networking information, including the information obtained during the bootstrapping process. • The clear autonomic neighbor command clears the neighbor-related information obtained during the neighbor discovery process. If no neighbor is specified, it clears the entire neighbor database. • The clear autonomic registrar accepted-device command clears the public key stored for each device enrolled by the registrar.

Verifying Autonomic Intent Configuration

Procedure

Step 1 enable

Example:

```
Device> enable
Enables privileged EXEC mode.
```

- Enter your password if prompted.

Step 2 show autonomic intent**Example:**

```
Device# show autonomic intent

Intent File : Available
Version Num : 1395908791 (Parsed)
Version Time : 2014-03-27 13:56:31 IST
Outer Vlans : 30,35,50-60,25,80
DB count : 15
Vlans in DB : 25, 30, 35, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 80,
Displays the status and details of the intent of the device.
```

Step 3 show autonomic intent neighbors**Example:**

```
Device# show autonomic intent neighbors

Neighbor                               Intent version
-----
PID:Unix SN:1652527195                 1395908791
Displays the status and details of the intent of the neighboring devices.
```

Configuration Examples for Autonomic Networking

Example: Configuring the Registrar

```
Device> enable
Device# configure terminal
Device(config)# autonomic registrar
Device(config-registrar)# domain-id abc.com
Device(config-registrar)# whitelist flash:anra-domain.txt
Device(config-registrar)# end
```

Example: Configuring and Verifying Autonomic Networking Intent

```
Device> enable
Device# configure terminal
Device(config)# autonomic
Device(config)# autonomic intent
Device(config-intent)# acp outer-vlans 30,35,50-60,25,80
Device(config-intent)# end
Device# show autonomic intent

Intent File : Available
```

```

Version Num : 1395908791 (Parsed)
Version Time : 2014-03-27 13:56:31 IST
Outer Vlans : 30,35,50-60,25,80
DB count    : 15
Vlans in DB :
25, 30, 35, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 80,

```

```
Device# show autonomic intent neighbors
```

```

Neighbor                               Intent version
-----                               -
PID:Unix SN:1652527195                 1395908791

```

Additional References for Autonomic Networking

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	

MIBs

MIB	MIBs Link
All supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/support

Feature History and 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.

Release	Modification
Cisco IOS XE Everest 16.5.1a	This feature was introduced.