Bridging Introduction

This guide describes the bridge feature. Bridging can be used to connect two or more Layer 2 interfaces together to form a single broadcast domain. Bridging can also be used to connect two remote sites to the same broadcast domain. Bridge MAC filtering is a Layer 2 filter that is a collection of rules that are applied to a bridge. Each rule will match certain types of Layer 2 traffic, and will either discard it, or allow it to continue through the bridge.

Products and software version that apply to this guide

This guide applies to AlliedWare Plus products that support High Availability, running version 5.4.5 or later.

To see whether your product supports High Availability, see the following documents:

- The product's Datasheet
- The AlliedWare Plus Datasheet
- The product's Command Reference

These documents are available from the above links on our website at alliedtelesis.com.

Feature support may change in later software versions. For the latest information, see the above documents.
Contents

Bridging Introduction .......................................................................................................................................................... 1
  Products and software version that apply to this guide ................................................................................................. 1
What is Bridging? ............................................................................................................................................................... 3
Bridging Operation ............................................................................................................................................................ 4
Bridging Features ............................................................................................................................................................... 4
Bridge Configuration ......................................................................................................................................................... 7
  Show command examples ............................................................................................................................................. 8
Bridge Configuration Examples ........................................................................................................................................ 9
  Example 1: Simple bridge configuration ...................................................................................................................... 9
  Example 2: Bridging between multiple VLANS and Ethernet interfaces ................................................................. 10
  Example 3: Bridging an L2TPv3 tunnel sub-interface with MAC filtering ............................................................. 13
What is Bridging?

Bridging is a feature that can be used to connect two or more Layer 2 interfaces together to form a single broadcast domain. Bridging forwards packets in software, based on the Layer 2 header. This is similar forwarding logic to Layer 2 switching, which forwards packets in hardware.

There are two main use-cases for bridging:

- extending a broadcast domain across two or more physically separated sites.
- applying security processing to traffic transparently in a Layer 2 network.

This guide begins with a high-level description of the bridging feature. It goes on to provide some basic useful commands to create and add interfaces to a bridge, change the ageing timer, and verify the bridge configuration.

It concludes with an example configuration that adds filters to block or allow frames based on source MAC address, and Ethernet protocol type.

<table>
<thead>
<tr>
<th>LIST OF TERMS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual Tunnel Interface (VTI)</td>
<td>In order to apply higher-layer functions (like multicasting, routing protocols, filtering etc.) to a VPN tunnel, it is convenient to treat the tunnel as a virtual Layer 3 interface. The virtual IP interface that is overlaid on a VPN tunnel is called a Virtual Tunnel Interface.</td>
</tr>
<tr>
<td>Bridge Entity ID</td>
<td>A Layer 3 interface to allow the host to be managed over the bridged network.</td>
</tr>
<tr>
<td>L2TPv3 pseudo-wire</td>
<td>L2TPv3 is an IETF standard for the encapsulation of multi-protocol Layer 2 communications traffic over IP networks. A pseudo-wire is an emulated circuit. A pseudo-wire can extend Layer 2 circuits via intermediate packet switched networks, including the internet.</td>
</tr>
</tbody>
</table>

For example, you can connect to two physically separated VLANs such as a remote office and a main office network, via an L2TPv3 Ethernet Pseudowire. This is achieved by bridging each office VLAN to a virtual Tunnel Interface (VTI) terminating an L2TPv3 Ethernet pseudowire.
Bridging Operation

Bridging forwards packets in software, based on the Layer 2 header. This is similar to the forwarding logic in Layer 2 switching, which forwards packets in hardware. Switch ports cannot be bridged. Tunnels, physical Ethernet interfaces, and VLANs can be bridged. However, these interfaces can only be members of one bridge at one time. If a packet is bridged then it is not processed by the normal Layer 3 packet forwarding path, such as routing and firewall.

Bridging Features

The bridge combines its constituent interfaces into a virtual Layer 2 switch. A range of interface types can be attached to a bridge - Ethernet, tagged Ethernet, VLAN, and tunnel interfaces. By default, there are no limitations on the types of Ethernet traffic that the bridge will forward. Tagged or untagged traffic can be forwarded by the bridge. The software will check the validity of the Ethernet frame to be bridged, which includes checking Layer 3 protocol fields. Invalid frames will be dropped, and the ingress port (not the bridge, but the underlying port) discard counter will be incremented.

The bridge also implements the same forwarding rules as a switch. Broadcast and multicast traffic is flooded to all interfaces attached to the bridge. The source MAC addresses of packets ingressing each interface are stored in a forwarding table, just as with a switch, so that unicast packets will only be sent to an interface that is known to provide a path to the packet’s destination MAC.

Destination lookup failures (failure to find a packet’s destination MAC in the forwarding table) will result in the packet being flooded to all but the ingress interface, just as with a switch.

As with a switch, a MAC address will age out of the MAC forwarding table if packets with that particular source MAC address have not been received on the bridge interface for a certain length of time. The length of time (ageing time) can be configured using the `ageing-time` command.

The bridge is treated as a Layer 3 interface into the Layer 2 network to which its constituent interfaces are connected. As such, the bridge can have higher-layer configuration applied to it – i.e. IPv4 and/or IPv6 addresses can be attached to the bridge, the bridge can be a PIM interface, an OSPF interface, a destination interface for static IP routes, etc. A bridge can even be configured to learn an IP address by DHCP. If there is a DHCP server on one of the Ethernet segments attached on one of the bridge’s constituent interfaces, then the bridge can obtain a DHCP lease from that server.
For example, if a host attached to eth1 of the device, in subnet A, wishes to connect to a host attached to one of the interfaces of a bridge, in subnet B, then the device will route the packets between eth2 and the Bridge entity.

Multiple separate bridges can exist within the same physical device. However, any given interface can only be attached to one bridge at a time.
It is even possible to route packets between bridge entities.

Figure 3: Bridge route between two bridges example

Bridge entities can have UP and DOWN events. If all the interfaces within a bridge go down, then the bridge itself is deemed to have gone down. If any one of its constituent interfaces comes up, then the bridge is deemed to have an UP event. Triggers can be configured on bridge UP or DOWN events.

The maximum number of bridge entities that can exist within one physical device is 16.

Show commands are available to provide information like the:
- content of a bridge's MAC forwarding table - `show bridge macaddr`
- state of the bridge's Layer 3 interface - `show interface <entity ID>`
- number of octets/packets that have been sent/received by the interfaces attached to the bridge. This displays the interface counters for the specific interfaces that are part of the bridge, for example, `show interface eth1` (if eth1 is part of the bridge).
- counters, which represent the number of octets/packets that have been exchanged between the bridge entity and the rest of the device. This includes management traffic to/from the management IP address configured on the bridge, and data routed between the bridge entity and other Layer 3 interfaces of the device - `show interface <entity ID>`.
Bridge Configuration

From configuration mode, create your bridge. If required, you can then assign an IP address to the bridge. This step is optional and is carried out from interface configuration mode.

Note: At this point, you can also set the bridge MAC address table ageing time, if required.

Exit back to configuration mode to assign an interface to your bridge group. After assigning the interface to the bridge group, you can verify your configuration using the `show bridge` command.

<table>
<thead>
<tr>
<th>Step 1. Creating your bridge</th>
<th>awplus# configure terminal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enter Configuration mode.</td>
</tr>
<tr>
<td>awplus(config)# bridge &lt;id&gt;</td>
<td>Enter your bridge entity ID.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2. Configuring your bridge</th>
<th>awplus(config)# interface br&lt;id&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enter interface configuration mode.</td>
</tr>
<tr>
<td>awplus(config-if)# ageing-time</td>
<td>(Optional), enter the time that an entry will stay in the MAC address table for the bridge before being deleted. Note: The default is 300 seconds (5 minutes).</td>
</tr>
<tr>
<td>&lt;ageing-timer&gt;</td>
<td></td>
</tr>
<tr>
<td>awplus(config-if)# ip address</td>
<td>(Optional), enter the ip address.</td>
</tr>
<tr>
<td>&lt;ipaddr&gt;</td>
<td></td>
</tr>
<tr>
<td>awplus(config-if)# exit</td>
<td>Exit back to Configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3. Run the show command to verify your configuration</th>
<th>awplus# interface &lt;interface-name&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enter the interface name that you want to add to the bridge.</td>
</tr>
<tr>
<td>awplus(config-if)# bridge-group &lt;id&gt;</td>
<td>Enter the bridge group ID.</td>
</tr>
<tr>
<td>awplus(config-if)# exit</td>
<td>Exit back to Configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4. Removing a bridge</th>
<th>awplus# configure terminal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enter Configuration mode</td>
</tr>
<tr>
<td>awplus# no bridge &lt;id&gt;</td>
<td>Enter the no variant of the bridge command</td>
</tr>
</tbody>
</table>
Show command examples

Use the `show bridge` command to check and verify your bridge configuration.

**Output 1: Example output from the show bridge command**

```
awplus#show bridge
Bridge Name  Aging Timer  Interfaces
---------------------
br5          300          eth1
br10         300          eth0
             300          eth2
br11         100
br15         300
```

Use the `show interface br<id>` command to display detailed information about the specified bridge.

**Output 2: Example output from the show bridge interface command**

```
awplus#show interface br1
Interface br1
 Link is UP, administrative state is UP
 Hardware is Bridge
 IPv4 address 192.168.1.13/24 broadcast 192.168.1.255
 index 33555969 metric 1
 MAC ageing time 300
 <UP,BROADCAST,RUNNING,MULTICAST>
 SNMP link-status traps: Disabled
 input packets 782, bytes 172480, dropped 0, multicast packets 0
 output packets 3, bytes 180, multicast packets 0 broadcast packets 0
 Time since last state change: 2 days 16:37:48
```

Use the `show bridge macaddr` command to display MAC addresses that a bridge knows about.

**Output 3: Example output from the show bridge macaddr command**

```
awplus#show bridge macaddr
Bridge Name  Interface  mac addr  is local?  ageing
---------------
br10          eth0      52:54:83:e2:8b:99  no         2
br10          eth0      52:54:c0:26:73:a4  yes        0
br10          eth0      96:58:3e:02:17:8f  no         211
br10          eth2      52:54:57:14:32:13  no         6
br10          eth2      52:54:9e:c4:7f:97  yes        0
br10          eth2      a6:d0:62:b8:d5:16  no         211
```

The `is local?` column refers to addresses that are associated with interfaces that are part of the bridge. The `ageing` column is a count of how many seconds it has been since the MAC address was last seen. Once this reaches the ageing timer value, the entry is removed from the MAC address table as the source address on a packet entering the associated interface.
Bridge Configuration Examples

Example 1: Simple bridge configuration

This example shows how to create a bridge with the ID of 2, and to assign the IP address 192.168.1.1/24. Interface vlan1 is added to bridge group 2 and Interface eth2 is added to the bridge group 2.

Figure 4: Example bridge configuration

The steps to configure this example are listed below:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>awplus# configure terminal</td>
<td>Enter Configuration mode.</td>
</tr>
<tr>
<td>awplus(config)# bridge 2</td>
<td>Enter your bridge ID.</td>
</tr>
<tr>
<td>awplus(config)# interface br2</td>
<td>Enter into Interface mode on the bridge.</td>
</tr>
<tr>
<td>ip address 192.168.1.1/24</td>
<td>Enter the IP address.</td>
</tr>
<tr>
<td>awplus(config-if)# exit</td>
<td>Exit back to Configuration mode.</td>
</tr>
<tr>
<td>awplus(config)# interface vlan1</td>
<td>Enter into Interface mode on vlan1.</td>
</tr>
<tr>
<td>awplus(config-if)# bridge-group 2</td>
<td>Enter the bridge group for vlan1.</td>
</tr>
<tr>
<td>awplus(config-if)# interface eth2</td>
<td>Enter into Interface mode on eth2.</td>
</tr>
</tbody>
</table>
Example 2: Bridging between multiple VLANS and Ethernet interfaces

This example shows how to bridge traffic between VLAN and Ethernet interfaces for multiple VLAN IDs.

Figure 5: Example bridge configuration with multiple VLANs

First, for each VLAN to be bridged, configure a bridge entity. In this example, two VLANs are to be bridged, so two bridge entities are configured.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bridge-group 2</code></td>
<td>Enter the bridge group for eth2.</td>
</tr>
<tr>
<td><code>exit</code></td>
<td>Exit back to Configuration mode.</td>
</tr>
<tr>
<td><code>exit</code></td>
<td>Exit back to Global Configuration mode.</td>
</tr>
</tbody>
</table>

```
! bridge 1
bridge 10
!
```

Traffic is bridged between VLAN111 and ethernet sub-interface 1.111
Traffic is bridged between VLAN112 and ethernet sub-interface 1.112

Port 1.0.1  VLAN111 and VLAN112

Port 1.0.2  VLAN111

eth1  eth 1.111

eth 1.112
Configure the VLAN IDs to be bridged in the VLAN database.

```plaintext
! vlan database
vlan 111-112 state enable
!
```

Associate the switch ports with the VLANs. In this example switch port 1.0.1 is 802.1q tagged member of VLANs 111 and 112, and switch port 1.0.2 is untagged member of VLAN111.

```plaintext
! interface port1.0.1
  switchport mode trunk
  switchport trunk allowed vlan add 111-112
  switchport trunk native vlan none
!
interface port1.0.2
  switchport access vlan 111
!
```

Configure Ethernet WAN interface with 802.1q tagged Ethernet sub interfaces associated each VLAN ID to be bridged.

```plaintext
interface eth1
  encapsulation dot1q 111
  encapsulation dot1q 112
!
```

Associate each VLAN and Ethernet sub interface with a bridge entity ID.

```plaintext
interface vlan111
  bridge-group 1
!
interface vlan112
  bridge-group 10
!
interface eth1.111
  bridge-group 1
!
interface eth1.112
  bridge-group 10
!
```

Any traffic associated with VLAN111 (Bridge 1) remains isolated from traffic associated with VLAN112 (Bridge 10). There is no Layer 2 traffic flows between bridge entities. There is no
Layer 2 traffic flow between interfaces associated with different VLAN IDs as each VLAN is associated with a different bridge entity.

- Ethernet frames via Ethernet sub-interface eth1.1.1.1 are tagged with VLAN ID 111
- Ethernet frames via Ethernet sub-interface eth1.1.1.2 are tagged with VLAN ID 112
- Ethernet frames via trunked port1.0.1 have appropriate 802.1q VLAN ID tag 1.1.1 or 1.1.2 applied.
- Ethernet frames via access port1.0.2 (VLAN 111) will remain untagged.

Use the `show bridge` command to display your configuration:

**Output 4: Example output from the show mac-filter command**

```
awplus# show bridge
Bridge Name Aging Timer Interfaces
------------------------------------------
br1 300 eth1.11
     vlan111
br10 300 eth1.112
     vlan112
awplus#
```

Use the `show interface` command to display detailed information about the specified bridge.

**Output 5: Example output from the show bridge interface command**

```
awplus# show interface
Interface Status Protocol
port1.0.1 admin up running
port1.0.2 admin up running
port1.0.3 admin up down
port1.0.4 admin up down
port1.0.5 admin up down
port1.0.6 admin up down
port1.0.7 admin up down
port1.0.8 admin up running
eth2 admin up down
eth1 admin up running
lo admin up running
vlan1 admin up running
vlan111 admin up running
vlan112 admin up running
br1 admin up running
br10 admin up running
eth1.112 admin up running
eth1.111 admin up running
```
Example 3: Bridging an L2TPv3 tunnel sub-interface with MAC filtering

For example, you can connect to two physically separated networks, such as remote office and main office networks, via an L2TPv3 Ethernet pseudo-wire. This is achieved by bridging each office VLAN to a Virtual Tunnel Interface (VTI). In the example below, the VTI is named TUNNEL 11. Each VTI Interface is configured for tunnel mode L2TPv3.

This setup shows how to bridge VLAN10 and VLAN20 between the local office, across the Internet via the L2TPv3 Ethernet pseudo-wire and the remote office.

- Traffic transported via the L2TPv3 Ethernet pseudo-wire can be secured via the tunnel protection IPSEC configuration. For more information see the IPSEC Command Reference.

Figure 6: Bridge route between local office and remote office example

Figure 7: Encapsulation packet header
Output 6: Example AR4050S configuration

```conf
bridge 1
bridge 2

interface tunnel1
  encapsulation dot1q 10
  encapsulation dot1q 20
  ip address 10.10.10.1/24
  tunnel source eth2
  tunnel destination 172.16.1.2
  tunnel local id 2
  tunnel remote id 1
  tunnel mode l2tp v3
  tunnel protection ipsec

interface tunnel1.10
  bridge-group 1

interface tunnel1.20
  bridge-group 2

interface vlan10
  bridge-group 1

interface vlan20
  bridge-group 2
```

Output 7: Example AR3050S configuration

```conf
bridge 1
bridge 2

interface tunnel1
  encapsulation dot1q 10
  encapsulation dot1q 20
  ip address 10.10.10.2/24
  tunnel source eth2
  tunnel destination 172.16.1.6
  tunnel local id 1
  tunnel remote id 2
  tunnel mode l2tp v3
  tunnel protection ipsec

interface tunnel1.10
  bridge-group 1

interface tunnel1.20
  bridge-group 2

interface vlan10
  bridge-group 1

interface vlan20
  bridge-group 2
```
**Bridge Filtering**

Filtering can be configured on the bridge to block/allow frames based on destination and source MAC address, and Ethernet protocol type. In this example, the goal is to filter some frames from specific MAC addresses coming from SW1 going to SW2.

The initial configuration of the devices is as follows:

Rule ‘a’ configures a MAC-filter to filter traffic from 0000.0c00.0200 to any destination while allowing all other traffic on br2.

Rule ‘b’ ensures all other traffic within the bridge entity is not blocked by the implicit deny all filter that is created when the bridge filtering is used within a bridge entity. The following configuration is added:

**Output 8: Configuration for adding a MAC-filter**

```plaintext
mac-filter onBr2
  rule a deny dmac any smac 0000.0c00.0200 proto any vlan any
  rule b permit dmac any smac any proto any vlan any

interface br2
  mac-filter-group onBr2
```

Use the `show mac-filter` command to display current filters:

**Output 9: Example output from the show mac-filter command**

```
<table>
<thead>
<tr>
<th>Bridge</th>
<th>Rule</th>
<th>DMAC</th>
<th>SMAC</th>
<th>Pkt Count</th>
<th>Byte Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>br2</td>
<td>a</td>
<td>any</td>
<td>0000.0c00.0200</td>
<td>10254</td>
<td>471684</td>
</tr>
<tr>
<td>br2</td>
<td>b</td>
<td>any</td>
<td>any</td>
<td>82020</td>
<td>3772920</td>
</tr>
</tbody>
</table>
```