Introduction

Policy-based routing (PBR) provides a means to route particular packets to their destination via a specific next-hop.

Using policy-based routing you can control which packets follow which path through the network. The specific path that these packets will take can be based on configurable parameters such as priority, address, protocol, or VLAN membership.

The Allied Telesis SwitchBlade™ x8100, x908, x900, x600, and x610 series switches all support policy-based routing at wire speed, using hardware-based policy routing facilities. The AR-Series Firewalls support policy-based routing in software.

These products provide flexible options for configuring policy-based routing, and can support multiple different policy routes, with different nexthops, simultaneously. Possible benefits include both QoS and cost savings:

- QoS by using dedicated links for certain types of traffic.
- cost savings by splitting traffic between low-bandwidth, low-cost permanent paths and high-bandwidth, high-cost use-on-demand paths.

A data network is a significant investment. It is a tool that can add significant value to a business. In order to reap the maximum benefit from this investment, you need to have as much control as possible over the traffic that is flowing in the network.

Often straight-out destination-based IP routing does not provide the level of traffic-direction control that a network manager really needs. Policy-based routing provides a much finer-grained level of control over where packets are routed to.
Products and software version that apply to this guide

This guide applies to the following AlliedWare Plus™ managed Layer 3 switches, running version 5.4.5-2 or later:

- x900, x600, and x610 series switches
- SwitchBlade x908 switches
- SwitchBlade x8100 switches

and the following AR-Series Firewalls:

- AR2010
- AR2050
- AR3050
- AR4050

Feature support may change in later software versions. For the latest information, 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.

Related documents

The following documents give more information about the IP routing features on AlliedWare Plus products:

- Route Selection Feature Overview and Configuration Guide
- IP Addressing and Protocols Feature Overview and Configuration Guide
- the Command Reference for each product

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

Introduction ................................................................................................................................................1
  Products and software version that apply to this guide .................................................................2
  Related documents ...............................................................................................................................2
Configuring Policy-based Routing on the Switch Products .................................................................4
  Limitations of PBR on switch products .............................................................................................5
  Limitations on the main IPv4 hardware routing table when PBR is enabled on
  SBx8100 switches ..............................................................................................................................5
Configuring Policy-based Routing on an AR-Series Firewall..............................................................6
  Limitations ...........................................................................................................................................7
Show Commands on AR-Series Firewall Products .............................................................................8
  Debugging ..........................................................................................................................................10
Policy-based Routing Configuration Examples ..................................................................................12
  Example 1- managing VoIP ................................................................................................................12
  Example 2 - reducing data transmission costs ..............................................................................15
  Example 3 - routing TCP traffic .....................................................................................................20
Configuring Policy-based Routing on the Switch Products

On the switch products, policy-based routing is presented as an action in a QoS policy. It can be combined with other QoS actions performed in the same policy. Other QoS actions applied to the traffic will be carried out before the traffic is sent to its next-hop.

The selection of packets to be policy routed is carried out by the standard class-map packet matching rules. If the switch doesn’t have the configured next-hop in its ARP table, it will send an ARP request for it. If it does not receive a reply, the switch will send an ICMP destination unreachable message to the originating host. The switch does not use the configured default route if the policy-based routing next-hop is unavailable.

So, the steps in configuring PBR on the switches are:

**Step 1.** For the SwitchBlade x8100 only, the switch needs to arrange the hardware tables to allocate some space for policy-based routes, this is achieved by entering the command.

```
awplus(config)# platform pbr-enable
```

Then the switch must be **rebooted** for the command to take effect.

**Note:** No other switch products need this command to be entered. All the other switches support PBR by default.

**Step 2.** Enable QoS.

```
awplus(config)# mls qos enable
```

**Step 3.** Create a class-map that defines the match criteria for the traffic that is to be directed down the policy route.

```
awplus(config)# class-map <cmap name>
awplus(config-cmap)# match <match criteria>
```

**Step 4.** Create a QoS policy, and define a policy routing rule for the traffic that matches the class-map. A policy routing rule simply consists of defining the next-hop via which the traffic will be routed.

```
awplus(config)# policy-map <pmap name>
awplus(config-pmap)# class <cmap name>
awplus(config-pmap-c)# set ip next-hop <nexthop address>
```

**Step 5.** Apply the QoS policy to a port.

```
awplus(config)# interface port1.0.1
awplus(config-if)# service-policy input <pmap name>
```
Limitations of PBR on switch products

Limited to IPv4 only

On the switch products, policy-based routing is only available for IPv4 traffic. Policy based routing of IPv6 is not available on any of the switch products.

Limit on the number of rules

An all switch products, the total number of policy-based rules is limited to 128.

Limitations on the main IPv4 hardware routing table when PBR is enabled on SBx8100 switches

When the switch is booted up with the command `platform pbr-enable` in the startup script, then the silicon resources are allocated in such a way that the size of the hardware IPv4 unicast routing table is reduced by 512 entries.

This is not a severe limitation, as the hardware IPv4 routing table contains thousands of entries. The lowest number of entries is 5376, when the switch is configured with the default silicon profile and the platform routing ratio with IPv4 and IPv6 weighting balanced.
Configuring Policy-based Routing on an AR-Series Firewall

On the AR-Series Firewalls:

- Policy Based Routing is supported for both IPv4 and IPv6
- Multiple nexthops can be defined on each policy rule, with the first available nexthop being the one that is used
- If no nexthops are available, the traffic is not dropped, but instead is forwarded via the normal routing table

First, policy-based routing needs to be globally enabled;

```
AR4050S#configure terminal
AR4050S(config)#policy-based-routing
AR4050S(config-pbr)#policy-based-routing enable
AR4050S(config-pbr)#
```

Then a set of policy-routing rules are created.

The rules are of the form:

```
ip policy-route <ID> match <application> from <entity> to <entity> nexthop <list of nexthops>
```

or

```
ipv6 policy-route <ID> match <application> from <entity> to <entity> nexthop <list of nexthops>
```

Where:

- The `<ID>` is an identifier for the rule. Packets are matched against the rules in order of ascending ID. Whether you specify the ID number for a rule is optional. If an ID number is not specified, then they are automatically allocated in intervals of 10.

- The `<application>` is optional. It can be any predefined application, or application created by using the application command to enter application configuration mode. If the application parameter is not specified, then the rule matches any traffic type.

- The to and from `<entities>` are optional. They can be any entities that have been defined by the zone, network or host commands. If the ‘from’ entity is not specified, then the rule matches traffic from any source. Similarly, if the ‘to’ entity is not specified, the rule matches traffic to any destination.

- The list of nexthops can be up to eight nexthops. The nexthops can be any one of:
  - IPv4 addresses
  - IPv6 addresses
  - Interfaces
All the entries in the list must be of the same type - they can’t be a mixture of the three types. Of course, if the policy route is matching IPv4 traffic (i.e. is a rule configured with ‘ip policy-route…’), then the nexthops must be IPv4 addresses or interfaces. If the policy route is matching IPv6 traffic (i.e. is a rule configured with ‘IPv6 policy-route…’) then the next-hops must be IPv6 addresses or interfaces.

The option of using interfaces is for the case where the policy route is directed over a tunnel or a PPP link, where the nexthop is not relevant, or maybe not even known.

The nexthops in the list are chosen in turn until one is found to be available. If none of the nexthops in the list is available, then the traffic is just conventionally routed by entries in the device’s main IP route table.

Some example rules

```
AR4050S#configure terminal
AR4050S(config)#
AR4050S(config)#policy-based-routing
AR4050S(config-pbr)#ip policy-route match udp from inside to lan nexthop 10.1.1.2 10.1.2.2 10.1.3.2 10.1.4.2 10.1.5.2 10.1.6.2 10.1.7.2 10.1.8.2
AR4050S(config-pbr)#ipv6 policy-route match udp from inside to lan nexthop 2001:100::2
```

Limitations

Limit on the total number of rules

The limit on the total number of Policy rules is 128. This number is spread across IPv4 and IPv6 rules. So, the total number of IPv4 rules, plus the total number of IPv6 rules, cannot exceed 128.

Limit on the number of combinations of nexthops

There is a limitation on the number of nexthops that can be shared among the rules. The limit is a total of 14 different combinations of nexthops, spread over the IPv4 rules and the IPv6 rules.

The same set of nexthops, but arranged in a different order, is considered a different combination.

So, for example the following lists of nexthops would count as two out of the possible 14 combinations:

- 128.34.56.182, 176.73.231.93
- 176.73.231.93, 128.34.56.182
Variation of DSCP within a flow will not be taken account of

One of the parameters that can be used to define an application is the DSCP value in the packets. The definition of an application can include a specification that packets belonging to the application must have a particular DSCP value, or one of a list of DSCP values. Once the first packet of a stream has been matched against a rule, then any subsequent packets in the same stream are not matched against the rules again (for performance reasons).

If the first packet in a stream has a particular DSCP value, and so matches a given application, and thereby, a particular rule, then ALL packets in the same stream will be deemed to match that same rule, even if they have different DSCP values that don’t match the definition of the application.

Show Commands on AR-Series Firewall Products

show pbr rules

This command displays a list of the rules that have been configured:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Match</th>
<th>From</th>
<th>To</th>
<th>Valid</th>
<th>Nexthop</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>tcp1</td>
<td>any</td>
<td>entities.outside</td>
<td>Yes</td>
<td>172.16.2.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>172.16.3.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>172.16.4.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>172.16.5.2</td>
</tr>
<tr>
<td>40</td>
<td>tcp2</td>
<td>any</td>
<td>entities.outside</td>
<td>Yes</td>
<td>172.16.7.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>172.16.8.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>172.16.9.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>172.16.10.2</td>
</tr>
</tbody>
</table>

show ip pbr route <1-255>

This will show the routes that are created by the rules. You can also narrow the display by specifying the PBR rule ID. The output shows the main IP route table (the standard table that does not contain policy routes), followed by the sub-tables created for the policy rules. Each rule creates it own sub-table.
Similarly the command `show ipv6 pbr route` outputs the main IPv6 route table, and the policy-related sub-tables.
Debugging

The command `debug policy-based-routing` enables a debug that will output a message each time a rule is matched.

The debug will also output detailed information when you attempt to create an invalid rule:

```
23:51:42 AR3050-R IMISH[24833]: ip policy-route match test1 from test2 to test3 nexthop 5.5.5.5
AR3050-R(config-pbr)#23:51:42 AR3050-R pbrd: PBR: Can't find application test1
```

```
23:51:42 AR3050-R pbrd: PBR: Failed to add policy route 40
```

```
23:51:42 AR3050-R PBR: -4 route append default via 5.5.5.5 table 2040
```
The debug outputs information when an interface state change results in an active
nexthop being added to or deleted from a rule.

AR3050-R(config-if)#show debugging policy-based-routing
Policy Based Routing Debugging Status: on
Policy-based Routing Configuration Examples

The following three examples will provide you with a good understanding of how policy-based routing can be implemented. Each example has step-by-step instructions and some suggested configuration commands.

Example 1- managing VoIP

VoIP is being used to provide voice communications within an organization. To ensure high voice quality, dedicated data circuits are leased between remote sites, and a central site. All VoIP data is routed via these dedicated circuits from the remote sites to a central site. The VoIP is then distributed from the central site to its eventual destinations. By using this hub-and-spoke arrangement of low-bandwidth circuits for transporting VoIP, the organization is provided with good-quality voice communications in a cost-effective manner. All other data communication between the sites is transported over the Internet. The VoIP traffic is marked with a DSCP value of 46.

Policy routing is used to ensure that the VoIP packets are sent via the dedicated circuit, whilst all other data is sent over the Internet. The VoIP traffic is also put into the highest queue on the egress port, to ensure minimum packet loss and delay.
Configuration on a switch

**Step 1. Create the VLANs and assign IP addresses to them.**

```
awplus(config)# vlan database
awplus(config-vlan)# vlan 2-3 state enable

#VLAN1 connects to the local LAN
awplus(config)# interface vlan1
awplus(config-if)# ip address 50.0.0.10/8

#VLAN2 connects to the Internet
awplus(config)# interface vlan2
awplus(config-if)# ip address 200.0.0.1/30

#VLAN3 is directed towards the dedicated circuit
awplus(config)# interface vlan3
awplus(config-if)# ip address 172.16.1.1/30
```

**Step 2. Assign the VLANs to the ports (port 1.0.1 is already in VLAN 1 by default)**

```
awplus(config)# interface port1.0.1
awplus(config-if)# switchport
awplus(config-if)# switchport mode access

awplus(config)# interface port1.0.2
awplus(config-if)# switchport
awplus(config-if)# switchport mode access
awplus(config-if)# switchport access vlan2

awplus(config)# interface port1.0.3
awplus(config-if)# switchport
awplus(config-if)# switchport mode access
awplus(config-if)# switchport access vlan3
```

**Step 3. Configure the default route. This is the route to the Internet**

```
awplus(config)# ip route 0.0.0.0/0 200.0.0.2
```

**Step 4. Enable QoS globally**

```
awplus(config)# mls qos enable
```
**Step 5.** Create the class-map which will match on traffic with a DSCP value of 46

awplus(config)# class-map test
awplus(config-cmap)# match dscp 46

**Step 6.** Create the policy-map which will specify the actions to be taken on the classified traffic. The first action instructs the switch to send this packet to queue 7 on the egress port. The second action sets the next-hop for this traffic to 172.16.1.2, so that it will be directed towards the dedicated circuit.

awplus(config)# policy-map pbr
awplus(config-pmap)# class test
awplus(config-pmap-c)# set queue 7
awplus(config-pmap-c)# set ip next-hop 172.16.1.2

**Step 7.** Finally, the policy-map must be attached to the ingress port

awplus(config)# interface port1.0.1
awplus(config-if)# service-policy input pbr

**Configuration on an AR-series firewall**

**Step 1.** Assign IP addresses to interfaces

awplus(config)# interface vlan1
awplus(config-if)# ip address 50.0.0.10/8
#eth1 connects to the Internet
awplus(config)# interface eth1
awplus(config-if)# ip address 200.0.0.1/30
#eth2 is directed towards the dedicated circuit
awplus(config)# interface eth2
awplus(config-if)# ip address 172.16.1.1/30

**Step 2.** Configure the default route. This is the route to the Internet

awplus(config)# ip route 0.0.0.0/0 200.0.0.0

**Step 3.** Create the application which will match on traffic with a DSCP value of 46

awplus(config)# application voice
awplus(config-application)# protocol udp
awplus(config-application)# dscp 46
Step 4. Create zones define the sources and destinations of the traffic

awplus(config)# zone private
awplus(config-zone)# network internal
awplus(config-zone-network)# ip subnet 50.0.0.10/8
awplus(config)# zone WAN
awplus(config-zone)# network dedicated
awplus(config-network)# ip subnet 172.16.1.1/30
awplus(config)# zone internet
awplus(config-zone)# network public
awplus(config-network)# ip subnet 0.0.0.0/0

Step 5. Enable policy-based routing

awplus(config)# policy-based-routing
awplus(config-PBR)# policy-based-routing enable

Step 6. Add the rule to direct the voice traffic via the dedicated link

awplus(config-PBR)# ip policy-route 10 voice from private nexthop 172.16.1.2

Example 2 - reducing data transmission costs

In this scenario (using the same diagram as in example 1 above) we have a faster Internet connection on port1.0.2, (eth1 in the firewall case) which costs the company more to use. The ISP providing this faster connection charges on the basis of the amount of data sent over the connection.

The company has decided that traffic from their web server will be sent to the Internet via this connection on Monday to Friday during business hours only (9.00am to 5.30 pm). This provides good web service during business hours, whilst keeping some limit on the total amount of data sent over the faster (more expensive) connection. Outside these times, traffic from the web server is sent via the default route. Any other traffic is always sent via the default route.

Additionally a ping poll can be configured that will regularly check that the Policy Route next-hop is reachable. If the ping poll fails to get a response from the next-hop, a trigger will be run which removes the policy from the ingress port. This will ensure that if the Policy Route next-hop fails, the configured default route will be used, ensuring no loss of connectivity. Once the next-hop is reachable again, another trigger adds the policy back onto the ingress port.
Configuration steps for the switch case

Step 1. Create the VLANs and assign IP addresses to them

awplus(config)# vlan database
awplus(config-vlan)# vlan 2-3 state enable

awplus(config)# interface vlan1
awplus(config-if)# ip address 50.0.0.10/8

awplus(config)# interface vlan2
awplus(config-if)# ip address 200.0.0.1/30

awplus(config)# interface vlan3
awplus(config-if)# ip address 172.16.1.1/30

Step 2. Assign the VLANs to the ports (port 1.0.1 is already in VLAN1 by default)

awplus(config)# interface port1.0.1
awplus(config-if)# switchport
awplus(config-if)# switchport mode access

awplus(config)# interface port1.0.2
awplus(config-if)# switchport
awplus(config-if)# switchport mode access
awplus(config-if)# switchport access vlan 2

awplus(config)# interface port1.0.3
awplus(config-if)# switchport
awplus(config-if)# switchport mode access
awplus(config-if)# switchport access vlan 3

Step 3. Configure the default route

awplus(config)# ip route 0.0.0.0/0 172.16.1.2

Step 4. Enable QoS globally

awplus(config)# mls qos enable

Step 5. Create the access-list which classifies on traffic from the Web server

awplus(config)# access-list 3001 permit tcp <web-server-IP address> eq 80 any
Step 6. Apply this access-list to a class-map

awplus(config)# class-map web-server
awplus(config-cmap)# match access-group 3001

Step 7. Apply this class-map to a Policy-map and configure the policy to send the traffic matching class web-server to a specific next-hop

awplus(config)# policy-map pbr
awplus(config-pmap)# class web-server
awplus(config-pmap-c)# set ip next-hop 200.0.0.2

Step 8. Create the script which will add the policy to the ingress port when trigger 1 is run

Edit policy-on.scp
   enable
   conf t
   int port1.0.1
   service-policy input pbr

Step 9. Create the script which will remove the policy from the ingress port when trigger 2 is run

Edit policy-off.scp
   enable
   conf t
   int port1.0.1
   no service-policy input pbr

Step 10. Configure the trigger which will add the policy to the ingress port at the required day/time

awplus(config)# trigger 1
awplus(config-trigger)# type time 09:00
awplus(config-trigger)# day monday tuesday wednesday thursday friday
awplus(config-trigger)# script 1 policy-on.scp

Step 11. Configure the trigger which will remove the policy from the ingress port at the required day/time

awplus(config)# trigger 2
awplus(config-trigger)# type time 17:30
awplus(config-trigger)# day monday tuesday wednesday thursday friday
awplus(config-trigger)# script 1 policy-off.scp
Step 12. Configure the ping poll which will regularly check that the next-hop is reachable

```bash
awplus(config)# ping-poll 1
awplus(config-ping-poll)# description "check policy route next hop"
awplus(config-ping-poll)# ip 200.0.0.2
awplus(config-ping-poll)# source-ip 200.0.0.1
awplus(config-ping-poll)# active
```

Step 13. Configure the trigger which will be activated when the ping poll fails

```bash
awplus(config)# trigger 3
awplus(config-trigger)# type ping-poll 1 down
awplus(config-trigger)# script 1 policy-off.scp
awplus(config-trigger)# active
awplus(config-trigger)# time after 09:00:00 before 17:30:00
awplus(config-trigger)# day monday tuesday wednesday thursday friday
```

Step 14. Configure the trigger which will be activated when the next-hop is reachable

```bash
awplus(config)# trigger 4
awplus(config-trigger)# type ping-poll 1 up
awplus(config-trigger)# script 1 policy-on.scp
awplus(config-trigger)# active
awplus(config-trigger)# time after 09:00:00 before 17:30:00
awplus(config-trigger)# day monday tuesday wednesday thursday friday
```

Configuration on an AR-Series Firewall

Step 1. Assign IP addresses to interfaces

#VLAN1 connects to the local LAN

```bash
awplus(config)# interface vlan1
awplus(config-if)# ip address 50.0.0.10/8
```

#eth1 connects to the Internet

```bash
awplus(config)# interface eth1
awplus(config-if)# ip address 200.0.0.1/30
```

#eth2 is directed towards the dedicated circuit

```bash
awplus(config)# interface eth2
awplus(config-if)# ip address 172.16.1.1/30
```
Step 2. **Configure the default route. This is the route to the Internet**

awplus(config)# ip route 0.0.0.0/0 200.0.0.0

Step 3. **Create Zones define the sources and destinations of the traffic**

awplus(config)# zone private
awplus(config-zone)# network internal
awplus(config-network)# ip subnet 50.0.0.10/8
awplus(config-network)# host webserver
awplus(config-host)# ip address <server address>

awplus(config)# zone internet
awplus(config-zone)# network public
awplus(config-network)# ip subnet 0.0.0.0/0

Step 4. **Enable Policy-based routing**

awplus(config)# policy-based-routing
awplus(config-PBR)# policy-based-routing enable

Step 5. **Add the rule to direct the voice traffic via the dedicated link**

awplus(config-PBR)# ip policy-route 10 http from private.internal.webserver nexthop 200.0.0.2

Step 6. **Create the script which will enable PBR when trigger 1 is run**

Edit policy-on.scp
enable
cnf t
policy-based-routing
policy-based-routing enable

Step 7. **Create the script which will disable PBR when trigger 2 is run**

Edit policy-on.scp
enable
cnf t
policy-based-routing
no policy-based-routing enable

From this point on, the configuration of the triggers and the ping polling is exactly the same as the switch case.
Example 3 - routing TCP traffic

When using policy-based routing, there is a distinct difference between the behavior of the switches and that of the AR-series firewalls with regard to traffic destined to the device itself. In the case of the switches, traffic that’s destined for the switch itself can be unexpectedly routed to the PBR next-hop. This includes TCP traffic such as telnet. In the example below, all TCP traffic is to be Policy Routed. In the case of a switch, this will also affect Telnet to the switch itself. To avoid this, extra configuration is required.

In the case of the AR-series firewalls products, the traffic destined to the device itself is not subject to Policy Based Routing, so no extra config is required to avoid that occurring.

Configuration steps for the switch case:

**Step 1. Create two IP interfaces - VLAN1 and VLAN2**

```bash
awplus(config)# interface vlan1
awplus(config-if)# ip address 50.0.0.10/8

awplus(config)# interface vlan2
awplus(config-if)# ip address 200.0.0.1/30
```

We do not want traffic destined to the addresses of either of these interfaces to be Policy Routed.

**Step 2. Create ACLs for any TCP traffic destined for these networks, as well as the PBR ACL**

```bash
awplus(config)# mls qos enable
awplus(config)# access-list 3001 permit tcp any 50.0.0.0/8
awplus(config)# access-list 3002 permit tcp any 200.0.0.1/32
awplus(config)# access-list 3003 permit tcp any any
```
Note: In the diagram above, any traffic matching the Policy-based routing access-list 3003, which classifies on TCP traffic ingressing port1.0.1, will be sent to the next-hop address of 200.0.0.2 on VLAN2. Even L2 traffic in the same VLAN through the switch will be sent to the PBR next-hop if it matches the PBR access-list.

This would mean that any TCP traffic from PC-A destined for another device connected to the switch in the same VLAN (e.g. PC-B) would not be forwarded to its destination but sent to the next-hop of 200.0.0.2. So, we configure ACL 3001 to match on any TCP traffic destined for the entire subnet in use on VLAN1, not just the IP address of the switch itself.

Allied Telesis recommends that the PBR policy always includes a class to allow traffic. This class must precede the Policy Routing class.

**Step 3. Apply these access-lists to class-maps**

```bash
awplus(config)# class-map local1
awplus(config-cmap)# match access-group 3001

awplus(config)# class-map local2
awplus(config-cmap)# match access-group 3002

awplus(config)# class-map tcp
awplus(config-cmap)# match access-group 3003
```

**Step 4. Apply the class-maps to a Policy-map and configure the policy to send the matching TCP traffic to a specific next-hop**

The effect of the policy map is that:

- Any traffic matching classes local1 or local2 (i.e. ACLs 3001 or 3002) will simply pass through to the normal forwarding process.

- Any traffic matching class tcp will be policy routed.

```bash
awplus(config)# policy-map pbr
awplus(config-pmap)# class local1
awplus(config-pmap)# class local2
awplus(config-pmap)# class tcp
awplus(config-pmap-c)# set ip next-hop 200.0.0.2
```

**Step 5. Finally, the policy-map must be attached to the ingress port**

```bash
awplus(config)# interface port1.0.1
awplus(config-if)# switchport
awplus(config-if)# switchport mode access
awplus(config-if)# service-policy input pbr
```

Note: Further reading - more detail of the configuration of policy-based routing can be seen in the QoS chapter of the x900, x600, and x610 series software reference manuals.
Configuration steps for the AR-Series Firewall case:

**Step 1. Create the zone for the public side of the firewall**

```
awplus(config)# zone internet
awplus(config-zone)# network public
awplus(config-network)# ip subnet 0.0.0.0/0
awplus(config-network)# host RouterA
awplus(config-host)# ip address 200.0.0.1
awplus(config-host)# host RouterB
awplus(config-host)# ip address 200.0.0.4
```

**Step 2. Create the zone for the private side of the firewall**

```
awplus(config)# zone private
awplus(config-zone)# network internal
awplus(config-network)# ip subnet 50.0.0.0/8
awplus(config-network)# host A
awplus(config-host)# ip address 50.0.0.5
awplus(config-host)# host B
awplus(config-host)# ip address 50.0.0.6
```

**Step 3. Apply IP addresses to interfaces**

```
awplus(config)# interface eth1
awplus(config-if)# ip address 200.0.0.1/30

awplus(config)# interface vlan1
awplus(config-if)# ip address 50.0.0.10/8
```

**Step 4. Create the default route to the Internet**

```
awplus(config)# ip route 0.0.0.0/0 200.0.0.4
```

**Step 5. Set up policy based routing**

```
awplus(config)# policy-based-routing
awplus(config-PBR)# policy-based-routing enable
awplus(config-PBR)# ip policy-route 20 match TCP from private to internet nexthop 200.0.0.2
```
If PBR debug is enabled on the AR-series firewall, then the packets in a TCP session from a host on the private LAN, destined to the Internet, will show up in the debug.

```
awplus#09:26:39 awplus kernel: PBR: IN=lan1 OUT=
MAC=02:00:00:56:78:9c:ec:cd:6d:82:6c:7f:08:00 SRC=50.0.0.6
DST=200.0.0.4 LEN=60 TOS=0 PREC=0 TTL=64 ID=4799 DF PROTO=TCP
SPT=41272 DPT=23 WINDOW=14600 RES=0 SYN URGP=0 MARK=0xff000000
```