Technical Guide

USB Cellular Modem

Feature Overview and Configuration Guide

Introduction

This guide describes the AlliedWare Plus 3G and 4G/LTE USB Cellular Modem feature, and how to configure it. This feature offers an additional data connection for critical services that can automatically switch to a mobile network whenever a primary data connection becomes unavailable. This feature also provides tools for identifying and solving problems with USB Cellular Modems that are connected to your AR-Series Firewall or Secure VPN Router.

- **Cellular** refers to a wireless connection between the modem and a cellular service provider.
- **3G** refers to a low speed wireless connection that creates a **Serial** type connection between the modem and the router. A PPP link is established across the serial link.
- **4G/LTE** refers to a higher speed wireless connection that creates an **Ethernet** type connection between the modem and the router.

Products and software version that apply to this guide

This guide applies to AlliedWare Plus™ products that support USB Cellular Modems. USB Cellular Modems are supported from:

- 3G - version **5.4.5-2.3**
- Diagnostic tools - version **5.4.6-1**
- 4G/LTE - version **5.4.7-0.2**
- IPv4/IPv6 dual stack support for USB modem interfaces - version **5.5.0-0.1**

For further information regarding product support for this feature, see the following documents:

- The product's 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

What does a USB cellular modem do?

A USB cellular modem is a wireless mobile USB device that plugs directly into your router to provide Internet access via a mobile broadband connection.

3G cellular modem features

The 3G cellular **serial** interface '**cellular0**' supports the following features:

- Setting an Access Point Name (APN) to use to connect to a cellular network.
- Specifying a non-default chat-script to be executed when a USB cellular modem connects to a carrier network.
- Displaying information about inserted USB devices including cellular devices.
- Configurable mode-switching to allow additional USB cellular modems to be put into the correct state when inserted.

4G/LTE cellular modem features

The 4G cellular **Ethernet** interface '**wwan0**' supports the following features:

- Static or dynamic IPv4 addressing via DHCP.
- Configuration of the MTU (Maximum Transmission Unit).

IPv4/IPv6 dual stack features

- IPv6 support on USB modem interfaces is not available from all carriers. Only 3G or 4G modems using carriers that support dual stack (IPv4/IPv6) are able to use this feature.
- Modems supporting dual stack functionality are configured as cellular interfaces with PPP encapsulation.

Supported modems and products

For a list of 3G and 4G cellular USB modems known to be compatible with AlliedWare Plus products see USB Modem compatibility.

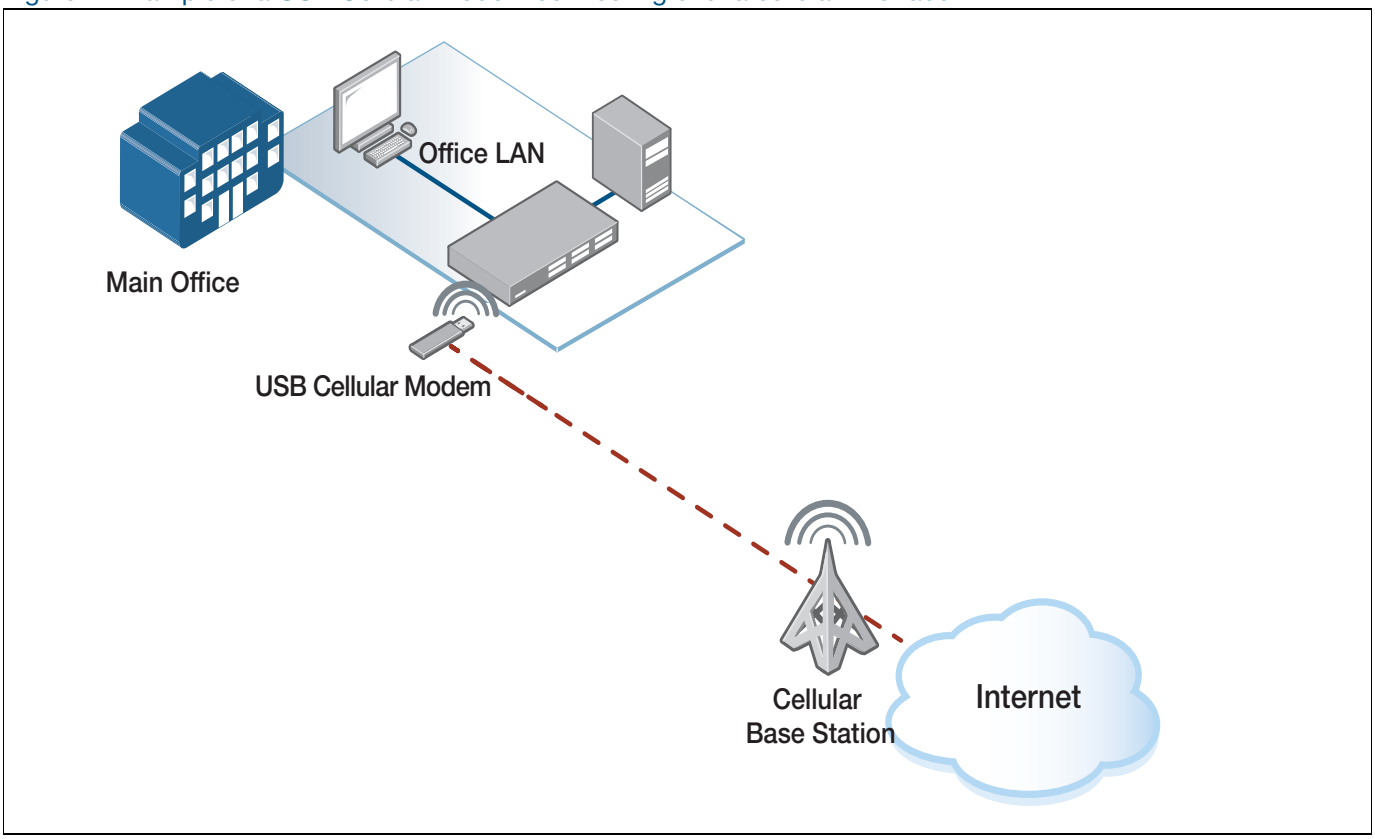
Unsupported modems

- Modems are unsupported if they do not support operation in **Serial mode,** or mode-switch to Wireless WAN (WWAN) **Ethernet mode**.
- Modems are unsupported if they have carrier-specific firmware loaded onto them to automatically pre-load settings specific to the network of the carrier. Such modems reject attempts to mode-switch the modem, for example, into serial mode for 3G. This can potentially be corrected by attempting to re-flash the modem with firmware supplied directly by the cellular USB device manufacturer (ZTE, Huawei, etc). Re-flashing the modem firmware is not possible from the router, but there are computer packages available that can do this.
- Modems are unsupported if they have been re-badged/re-branded and carrier-locked to a specific cellular carrier, so do not accept requests to be re-configured for use on other networks. This can potentially be corrected by using an external software tool (e.g. DC-unlocker) to unlock the modem. Unlocking your modem is an operation performed at your own risk.
- Modem hardware may support 3G or 4G functionality but the software loaded may be fixed to one service. It is important when purchasing a modem that confirmation is obtained to verify what the software will support, rather than accepting what the hardware is capable of.

How to use a USB cellular modem

In this example the main office LAN is connecting to the Internet via a USB cellular interface to access the cellular network.

Figure 1: Example of a USB Cellular Modem connecting over a cellular interface



3G cellular modem options

The router at the Main Office has a 3G USB cellular modem plugged in.

An Access Point Name (APN) is supplied by the carrier that the modem (with its inserted cellular SIM card) connects to.

The APN is configured on the router as part of the configuration of the cellular interface.

This information is used by the carrier to form a valid Internet connection via its cellular network and the public Internet. The APN allows the cellular carrier to ensure the correct WAN IP address is assigned to the serial PPP interface over the 3G USB cellular modem.

Additionally, when the serial PPP is established, a default route is dynamically created and added to the IP route table with a default administrative distance of 100, thereby enabling full Internet connectivity via the cellular connection.

4G/LTE cellular modem options

The router at the Main Office has a 4G/LTE USB cellular modem plugged in.

An interface '**wwan0**' is created and an IPv4 address is assigned from a DHCP server operating internally within the 4G modem. A default route via the 4G interface is automatically created providing Internet access.

DNS address information is also supplied via DHCP. The modem performs NAT between the router and the public assigned Internet IP address from the cellular provider.

Additionally, an AR-Series Firewall can have a firewall configured, and NAT to support devices on the Office LAN configured with RFC1918 private IP address.

Some 4G modems use PPP encapsulation instead of DHCP.

Configuring a 3G modem using a PPP interface

A minimum configuration required to use a supported 3G modem to connect to a cellular network contains a configuration of the APN for the cellular network, and a static PPP configuration associated with the cellular interface.

Step 1. **Configure the cellular interface**

```
awplus# configure terminal
awplus(config)# interface cellular0
awplus(conf-if)# encapsulation ppp 0
awplus(conf-if)# apn www.example.com
```

The APN is the name of the gateway used to form a connection between a carrier's cellular network and the public Internet. The mobile carrier reads the APN settings and then determines the correct IP address to apply, and connects to the appropriate secure gateway.

An APN must be set in order for the router to connect to the cellular network. APNs can usually be found by searching on-line, or by contacting the carrier directly. Some mobile network operators do not require a specific APN to be configured, in which case, any string can be used as the APN.

In a minimum configuration, an internal default chat-script is used to connect. The default chat-script uses the APN configured for a cellular interface. Setting or changing the APN or any PPP settings will restart any currently running PPP session for that cellular interface. Unsetting the APN will stop the PPP session.

Step 2. **Configure the static PPP interface**

```
awplus# configure terminal
awplus(config)# interface ppp0
awplus(conf-if)# ppp ipcp dns request
awplus(conf-if)# keepalive
awplus(conf-if)# ip address negotiated
awplus(conf-if)# ip tcp adjust-mss pmtu
```

Step 3. **Plug your modem into the router USB slot**

When a supported 3G modem (with an appropriate SIM card) is inserted into a router with a cellular interface configured, the following operations are performed:

- The USB device is identified as a 3G cellular modem.
- The modem is switched to USB serial mode, which internally creates a serial interface.
- A PPP session is started, which uses the serial interface to communicate with the network.
- The APN and modem vendor-specific settings are applied via a chat-script.

Default route over 3G

When a PPP link is established over 3G, it automatically creates a default route with Administrative Distance (AD) of 100. This route was previously invisible, but from version 5.4.7-0.1 you can view it by using the **show ip route** command, as shown in the following example:

```
Client#show ip route
Codes: C - connected, S - static, R - RIP, B - BGP
       O - OSPF, D - DHCP, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       * - candidate default

Gateway of last resort is 172.16.1.1 to network 0.0.0.0

S*      0.0.0.0/0 [100/0] via 172.16.1.1, ppp0
C       172.16.1.1/32 is directly connected, ppp0
C       172.16.2.1/32 is directly connected, ppp0
C       192.168.2.0/24 is directly connected, vlan1
```

- Internet connectivity should now be available.

If you need to use another default route via a different interface, configure the default route via the other interface with a lower AD.

Use the command:

```
awplus# ip route <subnet&mask> {<gateway-ip>|<interface>} [<distance>]
```

Configuring IPv4/IPv6 dual stack

When configured for dual IPv4/IPv6 stack operation, both IPv4 and IPv6 addressing is applied. This allows both IPv4 and IPv6 hosts to be able to communicate directly to the Internet via the cellular WAN to providers supporting this capability, without having to use IPv6 transition tunneling.

Cellular interface configuration

APN and PPP encapsulation are configured on the cellular interface.

Configure the command **ipv6 enable** on the cellular interface. This ensures the modem is configured via the internal default chat-script for dual IPv4/IPv6 mode.

Step 1. **Configure the cellular interface**

```
awplus# configure terminal
awplus(config)# interface cellular0
awplus(conf-if)# encapsulation ppp 0
awplus(conf-if)# apn www.example.com
awplus(conf-if)# ipv6 enable
```

PPP interface configuration

Configure the PPP interface with both IPv4 and IPv6 options for dual stack operation

The PPP options used to ensure the PPP link can be negotiated successfully with the service provider, and provide reliable Internet connectivity to websites.

When dual stack is used, the PPP Link Control Protocol (PPP LCP) options are first negotiated, followed by PPP IPv4 Network Control Protocol options (PPP IPCP), and then PPP IPv6 Network Control Protocol (PPP IPV6CP) is negotiated.

IPv4 DNS server information can be requested and learned dynamically via PPP negotiation or statically configured.

IPv4 addresses can be dynamically allocated by the service provider or statically configured.

Some options, such as <***username***>, and <***password***> are provided by the service provider.

IPv4 and IPv6 TCP MSS clamping can be used to adjust MSS based on the interface MTU. This ensures clients can access websites, and avoid potential fragmentation issues if path MTU discovery is possible.

Configure the command **ipv6 enable** on the PPP interface to negotiate IPv6 via the PPP link.

Configuring **ipv6 enable** negotiates IPv6 interface identifiers (local and remote peer IPv6 link-local addresses) via PPP messages only.

Subsequently, after successful negotiation of the IPv6 PPP NCP (Protocol field 0x0057 within the PPP header frame), Router Solicitations (RS) and Router Advertisements (RA) are used, as are other functions for IPv6 neighbor discovery. Global Scoped IPv6 addresses are obtained via SLAAC. These various IPv6 messages are all encapsulated within PPP frames.

IPv6 RA messages can also be used by your device to dynamically populate the IPv6 default routers list (RFC 4861), with a default IPv6 route via the link-local address of the peer.

Step 2. **Configure the ppp interface**

```
awplus# configure terminal
awplus(config)# interface ppp0
awplus(conf-if)# ppp ipcp dns request
awplus(conf-if)# ip address negotiated
awplus(conf-if)# ppp username <username>
awplus(conf-if)# ppp password <password>
awplus(conf-if)# ipv6 enable
awplus(conf-if)# ip tcp adjust-mss pmtu
awplus(conf-if)# ipv6 tcp adjust-mss pmtu
```

To check if IPv4 is configured, use the **show ip interface brief** command:

```
awplus#show ip interface brief
Interface              IP-Address          Status          Protocol
eth1                   unassigned          admin up        down
eth2                   unassigned          admin up        down
lo                     unassigned          admin up        running
vlan1                  192.168.1.1/24      admin up        down
cellular0              unassigned          admin up        down
ppp0                   192.168.100.1/32    admin up        running
```

To check if IPv6 is configured and is working, use the **show ipv6 interface brief** command:

```
awplus#show ipv6 interface brief
  * = Autoconfigured Address
Interface     IPv6-Address                      State        Status      Protocol
eth1          unassigned                        N/A          admin up    down
eth2          unassigned                        N/A          admin up    down
lo            unassigned                        N/A          admin up    running
vlan1         unassigned                        N/A          admin up    down
cellular0     unassigned                        N/A          admin up    down
ppp0          2001:db8::1/64                    preferred *  admin up    running
              fe80::3d53:f322:3eca:d5d2/10      preferred
```

Use the **Ping** command to the Internet via the cellular PPP interface to check your connection:

```
awplus#ping ipv6 2001:db8:1::2
PING 2001:db8:1::2(2001:db8:1::2) 56 data bytes
64 bytes from 2001:db8:1::2: icmp_seq=1 ttl=51 time=495 ms
64 bytes from 2001:db8:1::2: icmp_seq=2 ttl=51 time=69.0 ms
64 bytes from 2001:db8:1::2: icmp_seq=3 ttl=51 time=68.3 ms
64 bytes from 2001:db8:1::2: icmp_seq=4 ttl=51 time=67.8 ms
64 bytes from 2001:db8:1::2: icmp_seq=5 ttl=51 time=65.9 ms
```

You can also check the interface with the **show interface ppp0** command.

The global command **ipv6 forwarding** may be required to turn on IPv6 unicast routing for packet forwarding if not already enabled.

Static IPv4 and IPv6 routes via the PPP interface can be optionally configured as follows:

```
awplus# configure terminal
awplus(config)# ip route 0.0.0.0/0 ppp0 <distance>
awplus(config)# ipv6 route 0::/0 ppp0 <distvalue>
```

IPv6 via PPP as a backup interface

If IPv6 over PPP WAN is to be used as a backup WAN, not primary, then the following can be configured to ensure the IPv6 routing path via the PPP interface is a higher cost whilst the default IPv6 routing path via the primary WAN remains available.

```
awplus# configure terminal
awplus(config)# interface ppp0
awplus(conf-if)# no ipv6 nd accept-ra default-routes
```

This prevents the default route being created dynamically based on RA, and therefore prevent unwanted routing of IPv6 traffic via the backup WAN, whilst default route via the primary WAN is available.

You can globally configure the backup route static IPv6 default route with a higher Administrative Distance cost, via the PPP interface:

```
awplus# configure terminal
awplus(config)# ipv6 route ::/0 ppp0 55
```

Additionally, if the PPP interface 'IPv6 address' is to be statically configured (so not dynamically acquired via SLAAC), then the following is configured on the PPP interface:

```
awplus# configure terminal
awplus(config)# interface ppp0
awplus(conf-if)# no ipv6 nd accept-ra-pinfo
```

Configuring a 4G/LTE modem using a WWAN interface

4G/LTE USB modems generally have a built in DHCP server that can serve an IP address to the WWAN interface of the router. The IP address that is dynamically assigned can vary between makes and models of 4G modems, however it is commonly a 192.168.x.x/24 IP address and mask that is allocated from within the RFC 1918 private address range.

If an IP address is configured on another interface, and it happens to conflict with the IP address that is dynamically allocated via DHCP (to the WWAN interface), then an error message is generated, such as **'%192.168.8.1/24 overlaps with WWAN0'**.

To avoid this conflict, the IP address used on the other interface will need to be changed to be within a different subnet. No message is generated if the modem is inserted and the dynamically assigned IP address (via DHCP) to the '**wwan0**' interface overlaps with an existing address.

DNS address information and a default gateway IP route are also automatically assigned via DHCP from the 4G modem to the WWAN, providing Internet connectivity.

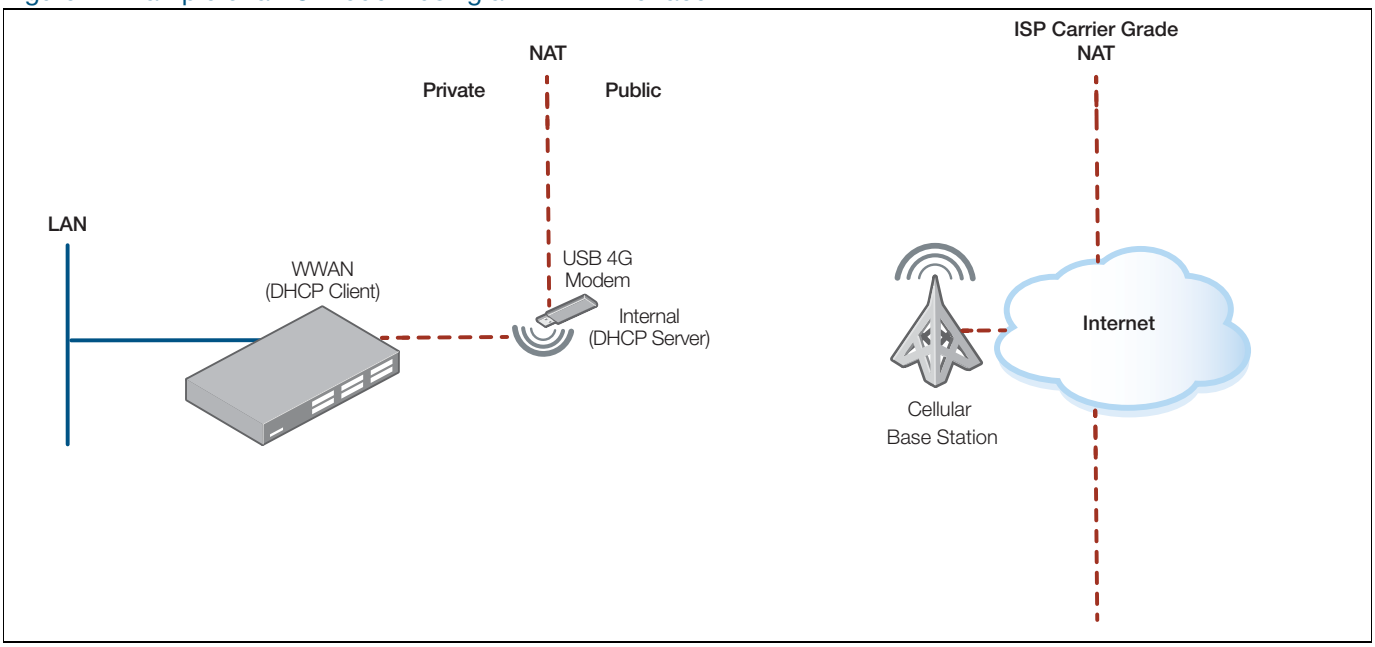
Note: This DHCP learned route is added to the IP route table with a default AD of 1. However, AD for this default route can be optionally altered. Changing the DHCP client default route AD ensures any pre-existing Internet traffic continues to be routed via other interfaces, even when the modem is plugged in. This is useful if the modem is to be used for backup Internet WAN connectivity.

A minimum configuration required to use a supported 4G modem to connect to a cellular 4G network, contains a DHCP client configuration associated with the cellular interface.

This allows the DHCP server (operating internally within the modem), to allocate an internal (private) IPv4 address to the WWAN interface of the router. The 4G modem is allocated an external (public) address from the carrier. The 4G modem performs its own internal IPv4 NAT.

Additionally, the carrier may perform its own Carrier Grade NAT between its internal network and the wider Internet. This may prevent connections initiated from devices located on the Internet from accessing the Cellular WAN IP address directly.

Figure 2: Example of a 4G modem using a WWAN interface



Step 1. **Configure the cellular interface**

```
awplus# configure terminal
awplus(config)# interface wwan0
awplus(conf-if)# ip address dhcp
```

Both 3G cellular and 4G WWAN interfaces can be simultaneously pre-configured (provisioned), allowing for an existing 3G modem to be swapped out and seamlessly upgraded to a 4G modem.

Step 2. **Plug your 4G modem into the router USB slot**

When a supported modem (with an appropriate cellular SIM card) is inserted into a router with a cellular interface configured, the following operations are performed:

- The USB device is identified as a 4G cellular modem and '**wwan0**' interface is enabled.
- A DHCP session is started internally between the modem and the router, and an inside (private) IP address, and gateway address (default route) is allocated from the modem to the WWAN interface.
- The modem registers itself with the cellular provider.
- An outside address is allocated to the modem, providing connectivity, and the modem performs its own internal NAT between the inside and outside interfaces.
- Internet connectivity should now be available.

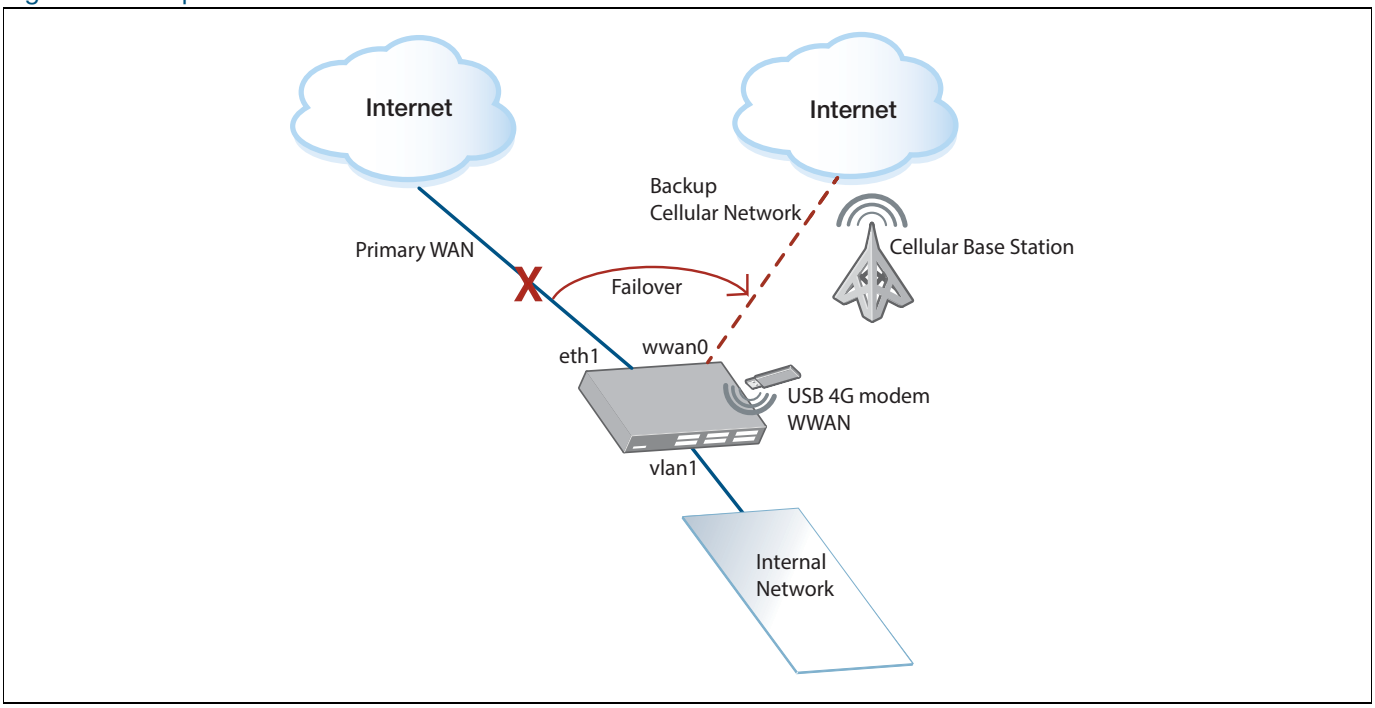
Additionally, the ISP carrier may perform its own Carrier Grade NAT, which (by default) may block sessions initiated externally from the Internet from reaching the modem.

Failover to a backup 4G modem

In this example, default routes are available via both the primary WAN and the backup 4G WWAN. The default route automatically assigned via DHCP (via the 4G cellular interface) has its AD set to a high cost value of 150. This ensures that traffic is routed via the primary WAN default route (which has a lower cost AD value).

If routing connectivity via the primary WAN interface fails, then the Internet traffic is automatically routed via the higher cost default routing path over the backup 4G WWAN. When lower cost routing connectivity is restored via the primary WAN, traffic is automatically routed via the primary WAN link again.

Figure 3: Example of a 4G modem failover



4G modem failover configuration example

In this example, an AR-Series Firewall is configured with a private zone protecting the Internal network associated with '**vlan1**'. A public zone is associated with '**eth1**'. The cellular interface '**wwan0**' provides redundant Internet WAN connectivity. NAT is also applied to traffic flows from the private zone to the public Internet.

Both '**eth1**' and '**wwan0**' are configured as DHCP clients, and receive their respective IP addressing, default routes, DNS, etc via DHCP.

The DHCP default route learned via the '**eth1**' interface is automatically added to the IP route database with a default AD of 1.

The '**wwan0**' interface is configured to alter the AD of its associated DHCP default route using the command **ip dhcp-client default-route distance**.

The higher AD cost is applied to the DHCP default route via the '**wwan0**' interface. This ensures traffic is routed via the primary '**eth1**' WAN interface, when both primary and backup WAN links are available.

4G failover configuration example

```
!
zone internet
 network wan
  ip subnet 0.0.0.0/0 interface eth1
 network wwan0
  ip subnet 0.0.0.0/0 interface wwan0
!
zone private
 network network
  ip subnet 192.168.2.0/24 interface vlan1
!
firewall
 rule 10 permit any from private.network to internet.wan
 rule 20 permit any from private.network to internet.wwan0
 rule 30 permit any from private.network to private.network
 protect
!
nat
 rule 10 masq any from private.network to internet.wan
 rule 20 masq any from private.network to internet.wwan0
 enable
!
interface eth1
 ip address dhcp
!
interface vlan1
 ip address 192.168.2.1/24
!
interface wwan0
 ip address dhcp
 ip dhcp-client default-route distance 150
!
```

The following show commands display output **before** a fail-over event, when both primary and backup WAN links are connected.

- The interface commands show that '**eth1'**, '**wwan0'** and '**vlan1'** interfaces are all connected and running.
- The **show ip route** command shows the default routing path used to forward traffic to the Internet that has been selected via the primary **'eth1**' interface.
- The **show ip route database** command shows that default routing paths via **both** the primary '**eth1'**and '**wwan0'** interfaces are available, and that the route via '**eth1'** is selected for forwarding traffic to the Internet as it has a lower cost AD of 1, compared to the default route via **'wwan0'**, which has the altered AD of 150.

4G show commands before a failover

```
gateway#show interface brief
Interface             Status          Protocol
port1.0.1             admin down      down
port1.0.2             admin up        running
port1.0.3             admin up        down
port1.0.4             admin up        down
eth1                  admin up        running
lo                    admin up        running
vlan1                 admin up        running
wwan0                 admin up        running
gateway#show ip interface
Interface             IP-Address          Status          Protocol
eth1                  10.34.199.102/24    admin up        running
lo                    unassigned          admin up        running
vlan1                 192.168.2.1/24      admin up        running
wwan0                 192.168.9.136/24    admin up        running
gateway#show ip route
Codes: C - connected, S - static, R - RIP, B - BGP
       O - OSPF, D - DHCP, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       * - candidate default

Gateway of last resort is 10.34.199.254 to network 0.0.0.0

S*      0.0.0.0/0 [1/0] via 10.34.199.254, eth1
C       10.34.199.0/24 is directly connected, eth1
C       192.168.2.0/24 is directly connected, vlan1
C       192.168.9.0/24 is directly connected, wwan0
gateway#show ip route database
Codes: C - connected, S - static, R - RIP, B - BGP
       O - OSPF, D - DHCP, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       > - selected route, * - FIB route, p - stale info

S       0.0.0.0/0 [150/0] via 192.168.9.1, wwan0
S    *> 0.0.0.0/0 [1/0] via 10.34.199.254, eth1
C    *> 10.34.199.0/24 is directly connected, eth1
C    *> 192.168.2.0/24 is directly connected, vlan1
C    *> 192.168.9.0/24 is directly connected, wwan0

Gateway of last resort is not set
gateway#
```

The following commands show interface state and default routing connectivity available via the backup '**wwan0'** interface **after** a fail-over event has occurred.

- In this case, '**eth1**' interface is down and the associated routing path via **'eth1'** is now inactive.
- The **show ip route** command output shows that the default route via '**wwan0'** is now being used for forwarding traffic to the Internet.
- The **show ip route database** command output shows the default route as **'wwan0'** and **'eth1'** now inactive as the default route.

4G show commands after a failover

```
gateway#show interface brief
Interface             Status          Protocol
port1.0.1             admin down      down
port1.0.2             admin up        running
port1.0.3             admin up        down
port1.0.4             admin up        down
eth1                  admin up        down
lo                    admin up        running
vlan1                 admin up        running
wwan0                 admin up        running
gateway#show ip interface
Interface             IP-Address          Status          Protocol
eth1                  10.34.199.102/24    admin up        down
lo                    unassigned          admin up        running
vlan1                 192.168.2.1/24      admin up        running
wwan0                 192.168.9.136/24    admin up        running
gateway#show ip route
Codes: C - connected, S - static, R - RIP, B - BGP
       O - OSPF, D - DHCP, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       * - candidate default

Gateway of last resort is 192.168.9.1 to network 0.0.0.0

S*      0.0.0.0/0 [150/0] via 192.168.9.1, wwan0
C       192.168.2.0/24 is directly connected, vlan1
C       192.168.9.0/24 is directly connected, wwan0
gateway#show ip route database
Codes: C - connected, S - static, R - RIP, B - BGP
       O - OSPF, D - DHCP, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       > - selected route, * - FIB route, p - stale info

S     *> 0.0.0.0/0 [150/0] via 192.168.9.1, wwan0
S        0.0.0.0/0 [1/0] via 10.34.199.254, eth1 inactive
C     *> 192.168.2.0/24 is directly connected, vlan1
C     *> 192.168.9.0/24 is directly connected, wwan0

Gateway of last resort is not set
gateway#
```

Advanced cellular options

Diagnostic tools

A number of internal parameters need to be applied to a modem in order to get it to connect. Occasionally, the default parameters applied by the router will not be appropriate for a given modem and it will fail to connect.

If your USB Cellular Modem does not work, you can go through the processes below to determine its characteristics and to apply the right parameters to it.

Step 1. **Identify whether a modem has mode-switched correctly**

Mode-switching USB devices

Most USB devices are detected as USB Mass Storage devices when they are first inserted, and will be subsequently **mode-switched** to the correct mode. Some devices may need to be manually configured to switch to the correct mode to be recognized as modems.

Turn on logging at the terminal by using the command:

```
awplus# terminal monitor
```

Then plug in the modem and examine the messages. In success or failure you can expect to see the following messages:

```
18:49:03 awplus kernel: usb 1-1: new high-speed USB device number 2 using xhci-hcd
18:49:03 awplus kernel: usb 1-1: New USB device found, idVendor=19d2, idProduct=2000
18:49:03 awplus kernel: usb 1-1: New USB device strings: Mfr=2, Product=1, SerialNumber=3
18:49:03 awplus kernel: usb 1-1: Product: ZTE CDMA Technologies MSM
18:49:03 awplus kernel: usb 1-1: Manufacturer: ZTE,Incorporated
18:49:03 awplus kernel: usb 1-1: SerialNumber: P673A2VDF_MS
18:49:03 awplus kernel: usb-storage 1-1:1.0: USB Mass Storage device detected
18:49:03 awplus kernel: usb-storage 1-1:1.0: device ignored
18:49:05 awplus usb_modeswitch: switch device 19d2:2000 on 001/002
```

You can also use the **show system usb detail** command:

```
Bus 001 Device 002: ID 19d2:2000 ZTE WCDMA Technologies MSM MF627/MF628/MF628+/M
F636+ HSDPA/HSUPA
Device Descriptor:
  ...
  idVendor           0x19d2 ZTE WCDMA Technologies MSM
  idProduct          0x2000 MF627/MF628/MF628+/MF636+ HSDPA/HSUPA
  iManufacturer           2 ZTE,Incorporated
  iProduct                1 ZTE CDMA Technologies MSM
  iSerial                 3 P673A2VDF_MS
  Configuration Descriptor:
    ...
    Interface Descriptor:
      ...
      bInterfaceClass         8 Mass Storage
      bInterfaceSubClass      6 SCSI
      bInterfaceProtocol     80 Bulk-Only
      iInterface              0
...
```

The output above shows that the USB vendor and product IDs of the device (prior to being mode-switched) are **0x19d2** and **0x2000**, respectively. The output also displays the USB Product, Manufacturer, and Serial Descriptors.

Note: The output also shows the **Interface Class** of the device as **Mass Storage**.

The following is an example of a successful mode-switched device:

```
19:31:43 awplus usb_modeswitch: switch device 19d2:2000 on 001/003
19:31:45 awplus kernel: usb 1-1: USB disconnect, device number 3
19:31:51 awplus kernel: usb 1-1: new high-speed USB device number 4 using xhci-hcd
19:31:51 awplus kernel: usb 1-1: New USB device found, idVendor=19d2, idProduct=0063
19:31:51 awplus kernel: usb 1-1: New USB device strings: Mfr=2, Product=1, SerialNumber=0
19:31:51 awplus kernel: usb 1-1: Product: ZTE CDMA Technologies MSM
19:31:51 awplus kernel: usb 1-1: Manufacturer: ZTE,Incorporated
19:31:51 awplus kernel: option 1-1:1.0: GSM modem (1-port) converter detected
19:31:51 awplus kernel: usb 1-1: GSM modem (1-port) converter now attached to ttyUSB0
19:31:51 awplus kernel: option 1-1:1.1: GSM modem (1-port) converter detected
19:31:51 awplus kernel: usb 1-1: GSM modem (1-port) converter now attached to ttyUSB1
19:31:51 awplus kernel: option 1-1:1.2: GSM modem (1-port) converter detected
19:31:51 awplus kernel: usb 1-1: GSM modem (1-port) converter now attached to ttyUSB2
19:31:51 awplus kernel: option 1-1:1.3: GSM modem (1-port) converter detected
19:31:51 awplus kernel: usb 1-1: GSM modem (1-port) converter now attached to ttyUSB3
19:31:51 awplus kernel: qmi_wwan 1-1:1.4: cdc-wdm0: USB WDM device
19:31:51 awplus kernel: qmi_wwan 1-1:1.4 wwan0: register 'qmi_wwan' at usb-xhci-hcd.0.auto-
1, WWAN/QMI device, a6:c6:58:7c:25:3e
19:31:51 awplus kernel: usb-storage 1-1:1.5: USB Mass Storage device detected
19:31:51 awplus kernel: scsi host2: usb-storage 1-1:1.5
19:31:51 awplus root: usb_modeswitch: switched to 19d2:0063 on 001/004
```

If the mode-switch output is logged, then the device has successfully mode-switched. You can then go to step 4 **Check carrier network connection** to check your modem configuration.

If nothing is logged after **usb_modeswitch** says that it is switching the device, then the device most likely wasn't found in the database. The database contains the devices with their default mode-switch configurations.

In this case, more diagnostic work is required. Continue to step 2: **Get USB identification information**, below.

Step 2. **Get USB identification information**

As mentioned above, the **show system usb detail** command outputs detail of the Modem ID, Manufacturer, etc. This information can also be obtained in more compact form from the command:

```
awplus# show system usb
```

Use the **show system usb** command to obtain identification information about a connected USB device. Use this information to find out the USB mode-switch parameters required to switch the modem to the correct mode.

The following example is output from the **show system usb** command:

```
awplus#show system usb
Bus 001 Device 002: ID 12d1:1001 Huawei Technologies Co., Ltd. E169/E620/E800 HS DPA Modem
```

Table 1: **show system usb** command output descriptions

PARAMETER	DESCRIPTION
Bus	USB bus number (001)
Device	Device number (002)
ID Vendor Product	 Vendor ID (12d1) Product ID (1001)
Vendor string descriptor	ID of the vendor (Huawei Technologies Co., Ltd)
Product string descriptor	ID of the Product (E169/E620/E800 HS DPA Modem)

The **show system usb detail** command can be used to obtain more identification information about the modem, for example, Serial ID, Device Type, and other USB related configuration information.

```
awplus# show system usb detail
```

The following example shows output from the **show system usb detail** command:

```
awplus#show system usb detail

Bus 001 Device 002: ID 12d1:1001 Huawei Technologies Co., Ltd. E169/E620/E800 HS
DPA Modem
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB               2.00
  bDeviceClass            0 (Defined at Interface level)
  bDeviceSubClass         0
  bDeviceProtocol         0
  bMaxPacketSize0        64
  idVendor           0x12d1 Huawei Technologies Co., Ltd.
  idProduct          0x1001 E169/E620/E800 HSDPA Modem
  bcdDevice            0.00
  iManufacturer           3 HUAWEI Technology
  iProduct                2 HUAWEI Mobile
  iSerial                 0
  bNumConfigurations      1
Configuration Descriptor:
    bLength                 9
    bDescriptorType         2
    wTotalLength           85
    bNumInterfaces          3
    bConfigurationValue     1
    iConfiguration          1 Huawei   Configuration
    bmAttributes         0xe0
      Self Powered
      Remote Wakeup
    MaxPower              500mA
    Interface Descriptor:
      bLength                 9
      bDescriptorType         4
      bInterfaceNumber        0
      bAlternateSetting       0
      bNumEndpoints           3
      bInterfaceClass       255 Vendor Specific Class
      bInterfaceSubClass    255 Vendor Specific Subclass
      bInterfaceProtocol    255 Vendor Specific Protocol
      iInterface              0
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x81  EP 1 IN
        bmAttributes            3
          Transfer Type            Interrupt
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0040  1x 64 bytes
        bInterval               5
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x82  EP 2 IN
        bmAttributes            2
          Transfer Type            Bulk
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0200  1x 512 bytes
        bInterval              32
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x01  EP 1 OUT
        bmAttributes            2
          Transfer Type            Bulk
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0200  1x 512 bytes
        bInterval              32
```

```
    Interface Descriptor:
      bLength                 9
      bDescriptorType         4
      bInterfaceNumber        1
      bAlternateSetting       0
      bNumEndpoints           2
      bInterfaceClass       255 Vendor Specific Class
      bInterfaceSubClass    255 Vendor Specific Subclass
      bInterfaceProtocol    255 Vendor Specific Protocol
      iInterface              0
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x83  EP 3 IN
        bmAttributes            2
          Transfer Type            Bulk
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0200  1x 512 bytes
        bInterval               32
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x02  EP 2 OUT
        bmAttributes            2
          Transfer Type            Bulk
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0200  1x 512 bytes
        bInterval               32
    Interface Descriptor:
      bLength                 9
      bDescriptorType         4
      bInterfaceNumber        2
      bAlternateSetting       0
      bNumEndpoints           2
      bInterfaceClass       255 Vendor Specific Class
      bInterfaceSubClass    255 Vendor Specific Subclass
      bInterfaceProtocol    255 Vendor Specific Protocol
      iInterface              0
      Endpoint Descriptor:
        bDescriptorType         5
        bEndpointAddress     0x84  EP 4 IN
        bmAttributes            2
          Transfer Type            Bulk
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0200  1x 512 bytes
        bInterval               32
      Endpoint Descriptor:
        bLength                 7
        bDescriptorType         5
        bEndpointAddress     0x03  EP 3 OUT
        bmAttributes            2
          Transfer Type            Bulk
          Synch Type               None
          Usage Type               Data
        wMaxPacketSize     0x0200  1x 512 bytes
        bInterval               32
Device Qualifier (for other device speed):
  bLength                10
  bDescriptorType         6
  bcdUSB               2.00
  bDeviceClass            0 (Defined at Interface level)
  bDeviceSubClass         0
  bDeviceProtocol         0
  bMaxPacketSize0        64
  bNumConfigurations      1
Device Status:     0x0001
  Self Powered
```

The following information is useful to troubleshoot the mode status:

Table 2: **show system usb detail** output descriptions for mode

PARAMETER	DESCRIPTION
Mode	**255** indicates device type/mode. In the case of a failed mode-switch, this will be: **8 Mass Storage.** After a successful mode-switch, most devices will display: **255 Vendor Specific Class**.

IMPORTANT NOTE: When finding out the identity of your modem, ensure that the Product and Vendor IDs are captured when the modem is in the state before any mode-switch happens. This is important because a device may be switched to an incompatible mode, and the ID numbers are changed after the switch has been performed.

Having found the vendor information for your modem, it is now possible to configure a custom mode-switch configuration. A mode-switch configuration file contains the information that must be written to a modem to put it into **Serial mode**. Different models of modem require different information to be written to them. The key is to find the right configuration file for your modem.

Mode-switch configuration files for various models of modem are usually shared by the community on the USB mode-switch forum. For more information, see ModeSwitchForum.

The following is an example of the contents of a mode-switch configuration file:

```
# ZTE devices, some Onda devices
TargetVendor= 0x19d2
TargetProductList="0001,0002,0015,0016,0017,0019,0024,0031,0033,0037,0042,0052,0055,0061,006
3,0064,0066,0091,0108,0
117,0128,0151,0157,0177,1402,2002,2003"
StandardEject=1
MessageContent="555342431234567020000000080000c8501010118010101010000000000000"
```

Step 3. **Map USB mode-switch configuration file to a USB Cellular Modem**

Syntax Having found the right file, save it, and then configure the router to associate the file with the right type of modem. Use the following commands:

```
awplus# configure terminal
awplus(config)# usb mode-switch {id <1-16>|vendor-id <vendor-id>|product-id <product-id>} [manufacturer <manufacturer>|product <product>|serial <serial>|vendor <vendor>|model <model>|revision <revision>] {file <file-name>}
```

A mode-switch configuration must specify an **ID number**, the USB **Vendor-ID** and **Product-ID** of the target device, and a mode-switch configuration **file** which must have the **.conf** extension.

Additional parameters can be specified, including USB descriptors (Manufacturer, Product, Serial), and SCSI descriptors (Vendor, Model, Revision). These are useful if there are multiple devices with the same vendor or product IDs but requiring different mode-switch configuration files.

A mode-switch configuration cannot be added if it has the same ID, or all the same parameters as another configuration. The value of the Vendor-ID, Product-ID, and USB Descriptors are included in the output of the **show system usb detail** command.

When specifying the descriptors, spaces must be substituted with underscores ("_") due to the design of the underlying **usb_modeswitch** utility. We recommend you use the optional parameters only if absolutely necessary, because entering them incorrectly will result in the device not being matched. The SCSI options in particular are unsuitable for matching on USB modem devices because they will generally only appear after the device has successfully mode-switched.

Examples The following are examples of commands that associate a mode-switch file with a model of modem:

```
awplus# configure terminal
awplus(config)# usb mode-switch id 1 vendor-id 12d1 product-id 140c manufacturer HUAWEI file switch.conf
awplus(config)# usb mode-switch id 1 vendor-id 19d2 product-id 2000 manufacturer ZTE file flash:/zte_modem.conf
```

If no path is specified in front of the name of the file, the file is assumed to be located in the root directory of Flash.

If a message similar to the following appears when a device is inserted:

```
usb_modeswitch: use overriding config file /etc/usb_modeswitch.d/<vendor>:<product>; make sure this is intended
```

where **<vendor>** and **<product>** are substituted with the vendor and product IDs of the device respectively, it means that the device was successfully mapped to the specified configuration file. Subsequently, if messages like the following appear, it is an indication that the device was successfully mode-switched to serial modem mode:

```
19:31:43 awplus usb_modeswitch: switch device 19d2:2000 on 001/003
19:31:45 awplus kernel: usb 1-1: USB disconnect, device number 3
19:31:51 awplus kernel: usb 1-1: new high-speed USB device number 4 using xhci-hcd
19:31:51 awplus kernel: usb 1-1: New USB device found, idVendor=19d2, idProduct=0063
19:31:51 awplus kernel: usb 1-1: New USB device strings: Mfr=2, Product=1, SerialNumber=0
19:31:51 awplus kernel: usb 1-1: Product: ZTE CDMA Technologies MSM
19:31:51 awplus kernel: usb 1-1: Manufacturer: ZTE,Incorporated
19:31:51 awplus kernel: option 1-1:1.0: GSM modem (1-port) converter detected
19:31:51 awplus kernel: usb 1-1: GSM modem (1-port) converter now attached to ttyUSB0
19:31:51 awplus kernel: option 1-1:1.1: GSM modem (1-port) converter detected
19:31:51 awplus kernel: usb 1-1: GSM modem (1-port) converter now attached to ttyUSB1
19:31:51 awplus kernel: option 1-1:1.2: GSM modem (1-port) converter detected
19:31:51 awplus kernel: usb 1-1: GSM modem (1-port) converter now attached to ttyUSB2
19:31:51 awplus kernel: option 1-1:1.3: GSM modem (1-port) converter detected
19:31:51 awplus kernel: usb 1-1: GSM modem (1-port) converter now attached to ttyUSB3
19:31:51 awplus kernel: qmi_wwan 1-1:1.4: cdc-wdm0: USB WDM device
19:31:51 awplus kernel: qmi_wwan 1-1:1.4 wwan0: register 'qmi_wwan' at usb-xhci-hcd.0.auto-
1, WWAN/QMI device, a6:c6:58:7c:25:3e
19:31:51 awplus kernel: usb-storage 1-1:1.5: USB Mass Storage device detected
19:31:51 awplus kernel: scsi host2: usb-storage 1-1:1.5
19:31:51 awplus root: usb_modeswitch: switched to 19d2:0063 on 001/004
```

Note: The product ID is now different to before the mode-switch.

IMPORTANT: Some devices ignore mode-switch configurations. These devices will usually be switched to a compatible mode. Examples of modems that ignore mode-switch configurations are the Huawei e220 modems (vendor ID 0x12d1 and the product IDs 0x1001, 0x1003-4, 0x1401-3F).

The **usb_modeswitch** utility logs execution output to **debug:/usb-modeswitch-<bus>-<port>**, where **<bus>** and **<port>** are the USB bus and port that the device is on (usually bus 1, port 1). If a mode-switch does not appear to work, it may be helpful to view the contents of the log by running the command **show file debug:/usb-modeswitch-1-1**.

The following is an example of the log output after a failed mode-switch:

```
awplus#show file debug:usb_modeswitch_1-1

USB_ModeSwitch log from Mon Jun 20 20:29:46 UTC 2016

Use global config file: /etc/usb_modeswitch.conf

Use top device dir /sys/bus/usb/devices/1-1
Check class of first interface ...
 Interface 0 class is 08.

----------------
USB values from sysfs:
  manufacturer  ZTE,Incorporated
  product       ZTE CDMA Technologies MSM
  serial        P673A2VDF_MS
----------------
bNumConfigurations is 1 - don't check for active configuration
ConfigList: /etc/usb_modeswitch.d/19d2:2000 /usr/share/usb_modeswitch/19d2:2000
/usr/share/usb_modeswitch/19d2:#linux
SCSI attributes not needed, move on
Check config: /etc/usb_modeswitch.d/19d2:2000
! matched. Read config data
Use config file from override folder /etc/usb_modeswitch.d
Logger is /usr/bin/logger
Driver module is "option", ID path is /sys/bus/usb-serial/drivers/option1

Command to be run:
usb_modeswitch -W -D -u -1 -b 1 -g 5 -v 19d2 -p 2000 -f $flags(config)

Verbose debug output of usb_modeswitch and libusb follows
(Note that some USB errors are to be expected in the process)
--------------------------------
Read long config from command line

 * usb_modeswitch: handle USB devices with multiple modes
 * Version 2.3.0 (C) Josua Dietze 2015
 * Based on libusb1/libusbx
 ! PLEASE REPORT NEW CONFIGURATIONS !

DefaultVendor=  0x19d3
DefaultProduct= 0x2000
TargetVendor=   0x19d2
TargetProductList="0001,0002,0015,0016,0017,0019,0031,0033,0037,0052,0055,0061,0
063,0064,0066,0091,0108,0117,0128,0157,0177,1402,2002,2003"

StandardEject=1
MessageContent="555342431234567020000000080000c8501010118010101010100000000000"
System integration mode enabled
Use given bus/device number: 001/005 ...
Look for default devices ...
 bus/device number matched
  found USB ID 19d2:2000
 No devices in default mode found. Nothing to do. Bye!


--------------------------------
(end of usb_modeswitch output)


Core program reported switching failure. Exit
```

The log reports that **No devices in default mode found**. The value for **DefaultVendor** in the mode-switch configuration file is set to **0x19d3**, but the program reports finding a USB device with a vendor ID of **19d2**, which means that the configuration file is wrong. **DefaultVendor** should be set to **0x19d2**.

For more information about how to write mode-switch configuration files, see the man page for 'usb_modeswitch'.

Step 4. **Check carrier network connection**

If the mode-switch is successful, but the modem still has not connected, then more information is required.

```
awplus# show cellular <cellular-interface-name>
```

The **show cellular** command displays status information about USB modems currently plugged into the network. If a cellular interface name is entered, the command only shows information about the USB modem associated with that specified interface.

Different vendors and models of modems often provide different sets of information. Vendor specific information will not be displayed if the information is unable to be obtained from the modem. For information that is common to most USB modems, the text **(unknown)** will be displayed if the information was not obtained successfully.

Some modems use the same internal communications channel via the internal USB interface for both control plane diagnostics, and actual IP or PPP user data plane communications. Alternatively, some modems support separate internal communications channels for data plane data, and control plane data via the internal USB interface.

For modems that use a single internal communications channel, if the interface is in use (for example configured with PPP), then USB and cellular interface show commands (such as **show system usb, show system usb detail** and **show cellular**) will fail with a message displayed.

Note: For these modems with a single internal USB communications channel, the PPP encapsulation configuration in the cellular interface will first need to be unconfigured to allow the cellular commands to become available.

The **show cellular** command can be used to find out why a modem might not be connecting to a carrier network. Once you find the problem, you can write a **chat-script** to put the device into the desired configuration.

Example output using the **show cellular** command when a Huawei E1762 modem is plugged in:

```
awplus#show cellular
Interface cellular0
  Manufacturer: huawei
  Model ID: E1762
  Revision ID: 11.126.10.00.74
  Serial ID: 351553036840711
  IMSI: 530011104647258
  Signal Quality:
    RSSI: -71 dBm
    Bit Error Rate: (unknown)
  Active Service Class: Data mode
  Phone Activity Status: Ready
  Service Center Address:
    Phone Number: +6421600600
    Number Type: International
GPRS Mobile Station Class: Class A
  Serial Port Configuration:
    Baud rate: 115200
    Character Format: 8-N-1
    Parity: Space
  Terminal Equipment Character Set: IRA
  Cable interface DTE-DCE local flow control:
    To DTE: RTS
    To DCE: CTS
  System Time: 1980/01/06,03:37:39
  GPRS Network Registration Status: Registered, home network
  PIN Request Status: READY
  Functionality Level: Full functionality (power-saving disabled)  Facility Lock Status:
    SIM card lock: Not active
    SIM fixed dialling memory feature: Not active
    Network personalization: Not active
    Network subset personalization: Not active
    Service provider personalization: Not active
    Corporate personalization: Not active
    Lock phone to first SIM card: Not active
  Call Mode: Single mode
  Wireless Data Service: 3GPP systems (GERAN, UTRAN and E-UTRAN)
  GPRS Service Status: Mobile station is attached to a GPRS service
  Dialling Number Type: National
  Bearer Service Type:
    Autobauding: Enabled
    Service: Data circuit asynchronous (UDI or 3.1 kHz modem)
    Connection Element: Non-transparent
  Automatic time and time zone update via NITS: Not enabled
  PPP support between TE and MT: Supported
  Last Error Report: No cause information available
  PLMN selection method: User controlled PLMN selected from Access Technology
  PDP Contexts:
    Context ID: 1
      Type: IP
      APN: www.vodafone.com
      Address: 0.0.0.0
      Header Compression: Off
      Status: Not active
  Primary DNS: 0.0.0.0
  Secondary DNS: 0.0.0.0
  Diagnostic mode baud rate: 115200
  TE-DCE baud rate: 115200
  Tolerance to long delays in PDP call setup: Enabled
  Hardware Version: CD25TCPV
  System Info:
    System Service State: Valid service
    System Service Domain: CS and PS service
    Roaming Status: Not roaming
    System Mode: WCDMA mode
    SIM card state: Valid USIM card state
    System Sub-mode: WCDMA mode
    System Config:
    Supported System Mode: Auto-select
    Network Acquisition Order: WCDMA, then GSM
    Service Domain Support: CS and PS
```

```
Card-Lock:
    Lock Status: Unlock code does not need to be provided
    Remaining Unlock Attempts: 10
    PLMN ID of the operator who has locked this device: None
  Signal Strength:
    RSSI (dBm): -64
    ECIO (dBm): -5
    RSCP (dBm): -69
  ICCID: 984610411061462785F5
  Software Version: E1762 11.126.10.00.74,CD25TCPV,Ver.B
  HSUPA status: Enabled
  HSDPA status: Enabled
  Card Mode: USIM
  Device Mode:
    Mode ID: 20
    Port Modes:
      Port 0: MDM
      Port 1: NDIS
      Port 2: DIAG
      Port 3: PCUI
      Port 4: CDROM
  Data Service Traffic:
    Last Connection Time (s): 5134
    Last Bytes Transmitted: 0
    Last Bytes Received: 168
    Total Connection Time (s): 64354
    Total Bytes Transmitted: 910
    Total Bytes Received: 3168
  PIN Status:
    Status: READY
    Remaining input attempts:
      PUK: 10
      PIN: 3
      PUK2: 10
      PIN2: 3
```

The following information is useful for troubleshooting:

Table 3: **show cellular** command output descriptions

OUTPUT	DESCRIPTION
Interface cellular0 Manufacturer: Model ID: Revision ID: Serial ID: IMSI:	This information is good for searching for solutions online or providing to support staff.
Signal Quality:	If this is exceptionally low (less than -113dbm), the signal strength may be insufficient to establish a reliable link with the network.
Last Error Report:	If an AT command failed, this field ‘Last Error Report’ may provide additional information about the cause.
PDP Contexts: Context ID: APN: Address: Header Compression: Status:	This should be the APN configured in the cellular interface.
Card-Lock: Lock Status:	If the value is ‘Unlock code needs to be provided’, it means that the modem is ‘locked’ and rejects changes to its settings. Unlocking your device is performed at your own risk.

Table 3: **show cellular** command output descriptions (continued)

OUTPUT	DESCRIPTION
Data Service Traffic: Last Connection Time (s): Last Bytes Transmitted: Total Connection Time (s): Total Bytes Transmitted: Total Bytes Received:	Useful to check whether traffic is being sent/received by the USB modem.

Some fields may be displayed in different formats for different devices, for example **System Time** is a text string printed exactly as given by the modem.

Some fields such as the **Status** field under **PIN Status** are displayed in an abbreviated format. Consult the AT (Hayes) command reference for the device for the exact meanings of the output.

Querying the modem for information is slow. Currently the **show cellular** command takes about 10 seconds to display for a single cellular interface.

Some of the meanings of the abbreviated outputs are given in the following tables:

Table 4: PIN Request Status

ABBREVIATION	DESCRIPTION
CS PS	circuit-switched packet-switched
READY	Not waiting for any password
SIM PIN	Waiting for SIM PIN to be given
SIM PUK	Waiting for SIM PUK to be given
PH-SIM PIN	Waiting for phone-to-SIM PIN to be given
PH-FSIM PIN	Waiting for phone-to-very-first-SIM PIN to be given
PH-FSIM PUK	Waiting for phone-to-very-first-SIM PUK to be given
SIM PIN2	Waiting for SIM PIN2 to be given
SIM PUK2	Waiting for SIM PUK2 to be given
PH-NET PIN	Waiting for network personalization PIN to be given
PH-NET PUK	Waiting for network personalization PUK to be given
PH-NETSUB PIN	Waiting for network subset personalization PIN to be given
PH-NETSUB PUK	Waiting for network subset personalization PUK to be given
PH-SP PIN	Waiting for service provider personalization PIN to be given
PH-SP PUK	Waiting for service provider personalization PUK to be given
PH-CORP PIN	Waiting for corporate personalization PIN to be given
PH-CORP PUK	Waiting for corporate personalization PUK to be given

Table 5: Network Type: Domain

ABBREVIATION	DESCRIPTION
CS_ONLY	CS domain service available
PS_ONLY	PS domain service available
CS_PS	CS and PS domain service available
CAMPED	Camped in a cell

Table 6: Device Mode: Port Modes

ABBREVIATION	DESCRIPTION
MDM	Modem
NDIS	Network card
DIAG	3G application interface
PCUI	PC User Interface
CDROM	CD image
SD	SD card
PCSC	Generic smart card

Table 7: PIN Status: Status

ABBREVIATION	DESCRIPTION
READY	No password requested
SIM PIN	SIM PIN requested
SIM PUK	SIM PUK requested
SIM PIN2	PIN2 requested
SIM PUK2	PUK2 requested

If the cellular connection is up, the show command will not show any information. This is because PPP locks the device, meaning that it can't be queried for information. In this case it will show the following output:

```
awplus#show cellular
Interface cellular0
% Status information unavailable. USB modem currently in use.
```

Step 5. **Assign chat-script**

A common reason for a modem to fail to connect is that it is not being sent the right commands in its chat script. A chat-script is a set of AT (Hayes) commands sent to a modem to cause it to connect.

Specifying a custom chat-script

Some 3G modems will require a non-default chat-script to be specified. This is accomplished by using the **chat-script** command, for example:

```
interface cellular0
 encapsulation ppp 0
 apn www.example.com
 chat-script flash:/modem.chat
```

The chat-script must have the file extension **.chat**. If the file system prefix ('flash' in the example above) is not specified, the file system defaults to flash. If the file does not exist, the default chat-script will be used instead. When a device is inserted and successfully recognized as a cellular modem, the chat-script will be run and its progress will be displayed in the terminal monitor.

The following message should confirm that the specified chat-script was run:

```
19:43:41 ARC pppd[11849]: [ppp0] [19:43:41.172] Script /usr/sbin/chat -E -v -t15 -f /flash/
modem.chat finished (pid 11860), status = 0x0
```

If PPP reports that **Connect script failed**, or **Modem hangup**, or **Connection terminated** then the modem failed to connect to the cellular network. If the progress of the chat-script execution reports the string **CONNECT** and PPP reports **Serial connection established**, then a connection to the cellular network has been established. The output of the **show ip route** command should confirm this.

Note: Note that setting or unsetting the chat-script will restart the PPP session for that cellular interface (provided that the device is inserted and the APN is configured).

The following is an example of assigning a default chat-script to a cellular interface:

```
awplus#configure terminal
awplus(config)#int cellular0
awplus(config-if)#chat-script connect.chat
```

Writing chat scripts

Sending commands to a device or configuring a chat-script requires some knowledge of AT (Hayes) commands. Chat-scripts consist of a list of **expect-send** pairs of messages. For each pair of messages, the chat program waits to receive the first message from the router, and once it has received it, it sends the second message to the modem. The messages sent to the modem are AT (Hayes) commands. Different vendors and models of modems support different sets of commands, but most modems have some commands in common. In the chat-script, anywhere that the text $APN is specified is substituted with the configured APN.

AT (Hayes) commands references are available for various models of modems that list the supported AT (Hayes) commands for that device, as well as their outputs and syntax. For further information, visit the modem manufacturers website to obtain the AT (Hayes) command reference for your specific modem.

The following is an example of the default chat-script contents:

```
ABORT 'BUSY'
ABORT 'NO CARRIER'
ABORT 'VOICE'
ABORT 'NO DIALTONE'
ABORT 'NO DIAL TONE'
ABORT 'NO ANSWER'
ABORT 'DELAYED'
REPORT CONNECT
TIMEOUT 6
'' 'ATH E1'
'' 'ATQ0'
'OK-AT-OK' 'ATZ'
TIMEOUT 3
'OK\d-AT-OK' 'ATI'
'OK' 'ATZ'
'OK' 'ATQ0 V1 E1 S0=0'
'OK' 'AT&C1 &D2'
'OK' 'AT+FCLASS=0'
'OK-AT-OK' 'AT+CGDCONT=1,"IP","$APN"'
'OK' 'ATDT*99***1#'
TIMEOUT 30
CONNECT ''
```

Table 8: AT (Hayes) commands

COMMAND	DESCRIPTION
AT+FCLASS=0	This message tells the modem to set the Active Service Class to ‘data mode’.
ABORT	This command causes the chat-script to stop if the following message is reported by the modem.
TIMEOUT	This command sets the maximum amount of time for the chat program to wait for a response from the modem.
$APN	Any occurrences of the text $APN in the script are substituted with the APN configured on the interface.
CONNECT	If the string CONNECT is received from the modem, a connection to the cellular network has been established.
'OK-AT-OK' 'AT+CGDCONT=1,"IP","$APN"'	Sets the APN.
'OK' 'ATDT*99***1#'	Dials the carrier network.

Allied Telesis NETWORK SMARTER

North America Headquarters | 19800 North Creek Parkway | Suite 100 | Bothell | WA 98011 | USA | T: +1 800 424 4284 | F: +1 425 481 3895
Asia-Pacific Headquarters | 11 Tai Seng Link | Singapore | 534182 | T: +65 6383 3832 | F: +65 6383 3830
EMEA & CSA Operations | Incheonweg 7 | 1437 EK Rozenburg | The Netherlands | T: +31 20 7950020 | F: +31 20 7950021

alliedtelesis.com

C613-22087-00 REV D

Allied Telesis

NETWORK SMARTER

North America Headquarters | 19800 North Creek Parkway | Suite 100 | Bothell | WA 98011 | USA | T: +1 800 424 4284 | F: +1 425 481 3895
Asia-Pacific Headquarters | 11 Tai Seng Link | Singapore | 534182 | T: +65 6383 3832 | F: +65 6383 3830
EMEA & CSA Operations | Incheonweg 7 | 1437 EK Rozenburg | The Netherlands | T: +31 20 7950020 | F: +31 20 7950021

alliedtelesis.com