Qualcomm Technologies, Inc.

RmNet and QCMobileAP

Feature Specification

80-PP188-4 Rev. C

July 2, 2021

For additional information or to submit technical questions, go to https://createpoint.qti.qualcomm.com

Confidential – Qualcomm Technologies, Inc. and/or its affiliated companies – May Contain Trade Secrets

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

Revision history

Revision	Date	Description
A	January 2020	Initial release
B	June 2020	Minor editorial changes were made. No technical content was changed in this revision.
C	July 2021	Minor editorial changes were made. No technical content was changed in this revision.

Contents

Revision history 2

1 Introduction to RMNet and QCMobileAP 7

1.1 Conventions 7

1.2 Technical assistance 7

2 RmNet for modem IPAs 8

2.1 RmNet architecture 8

2.2 RmNet frame format 9

2.2.1 Data link layer (802.3) 10

2.2.2 Raw IP 10

2.2.3 QoS flow header 11

2.3 RmNet data session management 12

2.3.1 RmNet data session setup 12

2.3.2 TE IP after RmNet address assignment 13

2.3.3 RmNet data session teardown 15

2.3.4 RmNet Autoconnect session establishment and teardown 17

2.3.5 IP address change on the TE 18

2.4 Interconnection with different devices and interfaces 19

2.5 Multi-RmNet 19

2.6 RmNet configuration parameters 21

2.7 Non-IP data delivery 21

2.7.1 wqa1576839440414AP side call flow to implement AP embedded call 21

3 QCMobileAP software 23

3.1 QCMobileAP overview 23

3.2 QCMobileAP features 24

3.3 Modem architecture with QCMobileAP overview 25

3.3.1 QCMobileAP overview – IPv4 architecture 26

3.3.2 QCMobileAP overview – IPv6 architecture 27

3.3.3 USB tethering mechanisms with QCMobileAP 28

3.3.4 Concurrency for Hexagon or Cortex-A7 or tethered applications 28

3.4 Limitations with QCMobileAP 29
A References 30
A.1 Related documents 30
A.2 Acronyms and terms 30

Tables

Table 2-1: RmNet configuration parameters.....21

Figures

Figure 2-1: RmNet architecture 9
Figure 2-2: 802.3 frames 10
Figure 2-3: Set raw IP data format 10
Figure 2-4: QoS flow header 11
Figure 2-5: Set QoS flow header data format 11
Figure 2-6: RmNet data session setup 13
Figure 2-7: TE IPv4 address configuration using DHCP 14
Figure 2-8: TE IPv6 address configuration using stateless autoconfiguration 14
Figure 2-9: TE IPv4 address configuration using QMI 15
Figure 2-10: Data session teardown 16
Figure 2-11: Autoconnect mode 17
Figure 2-12: IP address change 18
Figure 2-13: Multi-RmNet 20
Figure 3-1: QCMobileAP overview 23
Figure 3-2: Modem system architecture 26

1 Introduction to RMNet and QCMobileAP

This document describes RM network interface for modem IP Accelerators (IPAs) and Qualcomm mobile access point (QCMobileAP) software functionalities.

1.1 Conventions

Function declarations, function names, type declarations, attributes, and code samples appear in a different font, for example, `cp armcc armcpp`.

Code variables appear in angle brackets, for example, `<number>`.

Commands to be entered appear in a different font, for example, **copy a:*.* b:**.

Button and key names appear in bold font, for example, click **Save** or press **Enter**.

1.2 Technical assistance

For assistance or clarification on information in this document, submit a case to Qualcomm Technologies, Inc. (QTI) at https://createpoint.qti.qualcomm.com/.

If you do not have access to the CDMATech Support website, register for access or send email to support.cdmatech@qti.qualcomm.com.

2 RmNet for modem IPAs

This chapter describes the features of RM network interface (RmNet). The RmNet feature in QTI MSM™ devices emulate a network interface for the connected terminal equipment (TE). The RmNet feature for the MSM behaves as a network adapter when attached to a TE and provides packet data connectivity to the attached TE. The term TE is inclusive of all form factors, including devices such as PCs, notebooks, or applications processor. The NET (RmNet) device coexists with other logical devices (for example, modem, diagnostic, NMEA), and so on).

RmNet relies on a control interface for any control signaling between the TE and mobile station (MS). For example, to initiate a data session on demand, to send any notification, and so on. Currently, RmNet supports only the Qualcomm MSM interface (QMI) as the control channel for signaling between the TE and MS. QMI defines the framework and messages for communication between the applications or drivers on the TE and the MS.

See *Qualcomm MSM Interface (QMI) Architecture* (80-VB816-1) for details on QMI.

Supported features

Supported features are as follows:

- IPv4 data connectivity
- Autoconnect mode
- Multi-RmNet
- Raw IP mode (not supported on all products)
- QoS (not supported on all products)
- Interconnects supported – USB and shared memory
- IPv6 data connectivity

2.1 RmNet architecture

The following figure shows the various modules involved in the RmNet control path and data path. Currently, RmNet supports only QMI as the control channel. QMI messages are used for control signaling between the TE and the MS.

See *Qualcomm MSM Interface (QMI) Architecture* (80-VB816-1) for details.

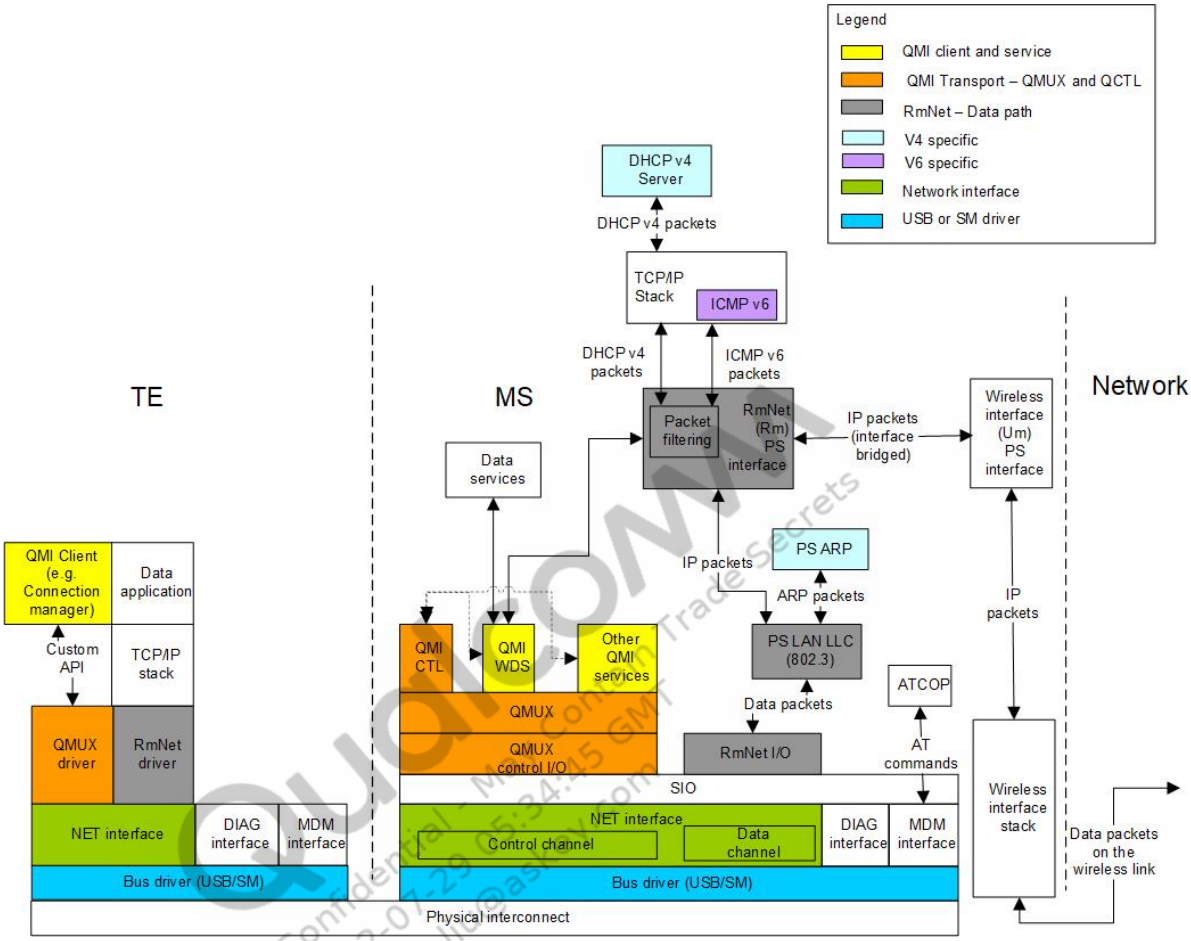

Figure 2-1 RmNet architecture

In the data path, RmNet does not use PPP and hence eliminates the high-level data link control (HDLC) framing or unframing overhead associated with PPP on the link between the TE and the MS. By default, RmNet uses 802.3 framing. RmNet also supports transferring raw IP packets (see Section Raw IP). When raw IP mode is used, the LAN LLC and ARP modules are not used.

DHCP v4 server and PS ARP modules are only used for IPv4. If the TE is using DHCP for IP address configuration on the TE, the DHCP v4 server is used (see Section TE IP after RmNet address assignment).

ICMPv6 module is used only for IPv6, for neighbor discovery, DAD, and stateless address autoconfiguration.

A single RmNet interface (instance) supports only one of IPv4 or IPv6 at a time. To have both IPv4 and IPv6 data sessions concurrently, each must go over a separate RmNet interface.

2.2 RmNet frame format

This section discusses the frame formats supported for the data packets transmitted over the RmNet interconnect (USB, shared memory).

2.2.1 Data link layer (802.3)

The data link layer format is shown in the following figure. In this format, IP packets are sent within 802.3 frames. This is the default data format.

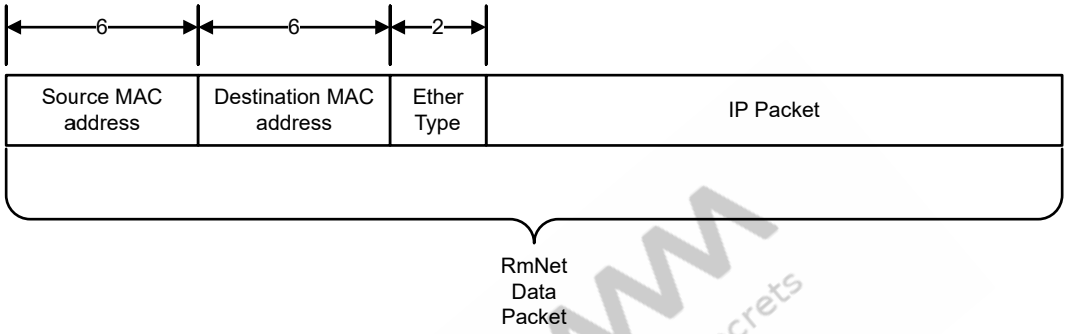

Figure 2-2 802.3 frames

2.2.2 Raw IP

In this format, raw IP packets are exchanged without any 802.3 header. The QMUX driver on the TE sends a QMI_CTL request to the MS to use the raw IP data format. This must be performed during power-up initialization before any data session is set up. The TE should assume raw IP data format only after the MS sends a response indicating that raw IP mode is used; otherwise, the default is 802.3 mode. See *QMI Control Service (QMI_CTL)* (80-VB816-3) for details.

This raw IP data format is shown in the following figure:

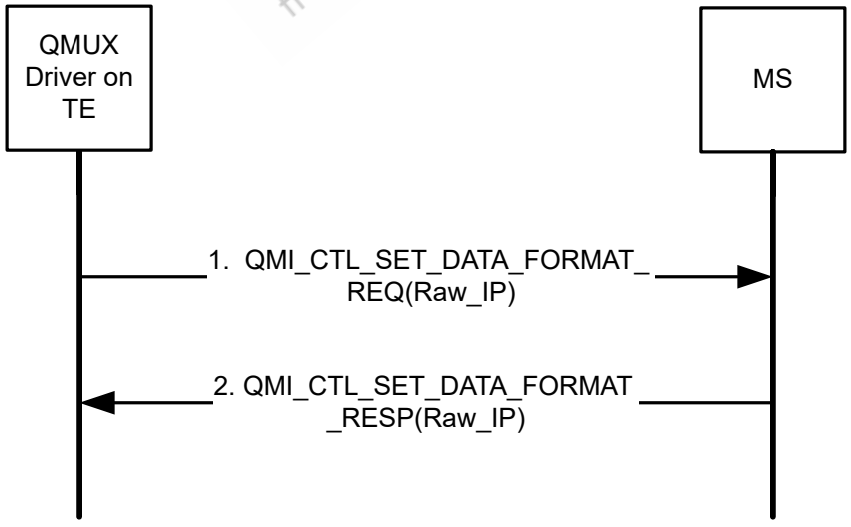

Figure 2-3 Set raw IP data format

2.2.3 QoS flow header

The QoS flow header data format, should be used only when the QMI QoS feature is used. The QMI QoS feature provides access to QoS over the wireless network to the applications on the TE. In this data format, the QoS flow header is added in front of the data packet (802.3 frame or raw IP packet). See *Qualcomm Interface Quality of Service (QMI QoS) Feature Description Document* (80-VF536-2) for details on QMI QoS.

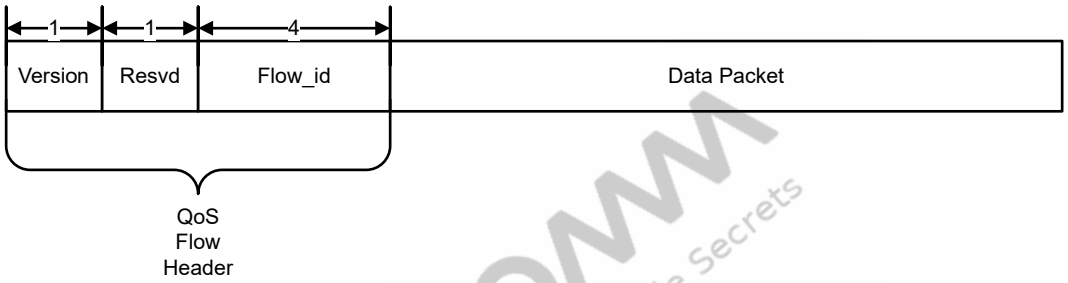

Figure 2-4 QoS flow header

The client side informs the MS about this data format using a QMI_CTL message, as shown in the following figure. This must be performed during power-up initialization before any data session is set up. The TE should assume this data format only after it receives a successful response from the MS. See *QMI Control Service (QMI_CTL)* (80-VB816-3) for details.

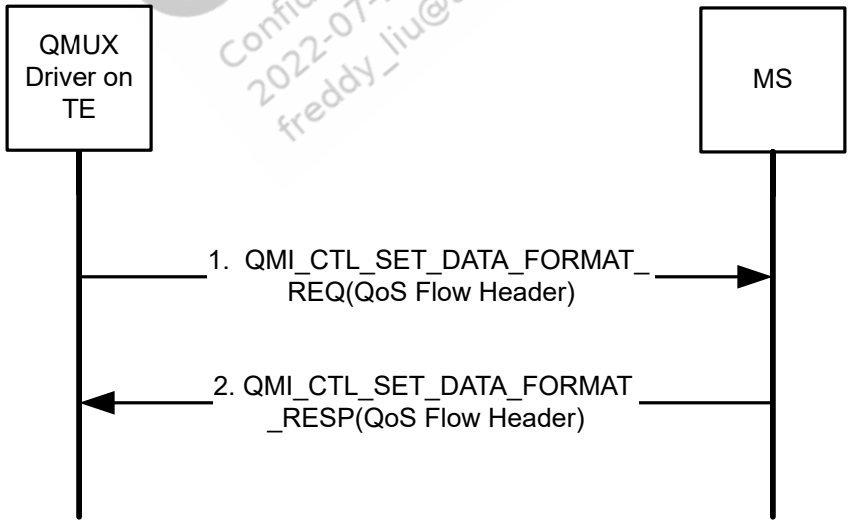

Figure 2-5 Set QoS flow header data format

2.3 RmNet data session management

This section discusses the various scenarios involving data session setup and teardown for RmNet as follows:

- Data session establishment – describes the steps involved when a TE requests the MS to set up a data session
- TE IP configuration – describes the procedure to assign an IP address and other IP configuration to the TE; these are two possible methods:
  - Using standard protocols
    - DHCP – TE uses DHCP procedure to get the IP configuration (for IPv4)
    - Stateless address autoconfiguration using ICMPv6 (for IPv6)
  - Using QMI – TE uses QMI messages to get the IP configuration (only for IPv4)
- Data session teardown – describes the steps involved when a TE requests the MS to tear down a data session.
- Autoconnect mode – describes the Autoconnect mode of operation where a data session is set up or torn down automatically, without an explicit request from the TE
- IP address change – describes the procedure to notify and convey the new IP configuration to the TE.

Components involved in data session establishment or teardown procedures are:

- TE OS stack – the native OS data stack running on the TE
- QMI WDS clients – QMI wireless data service clients (see *QMI Wireless Data Service Major Version 1, Minor Version 4 (ISOD)* (80-VB816-5) for details); sends and receives WDS messages; are responsible for notifying the TE OS stack of data connectivity and IP configuration using host OS-specific APIs
- QMUX driver – QMUX provides the transport for QMI clients and services (see *Qualcomm MSM Interface (QMI) Architecture* (80-VB816-1) for details). The client sends and receives QMI messages through an API exposed by the QMUX driver running on the TE.
- MS – the mobile station where the QMI services reside
- Network – the wireless network

2.3.1 RmNet data session setup

Preconditions and postconditions for RmNet data session are as follows:

- Precondition – the QMI WDS client has successfully obtained a client ID from the MS.
- Postcondition – the MS has obtained the IP configuration from the network.

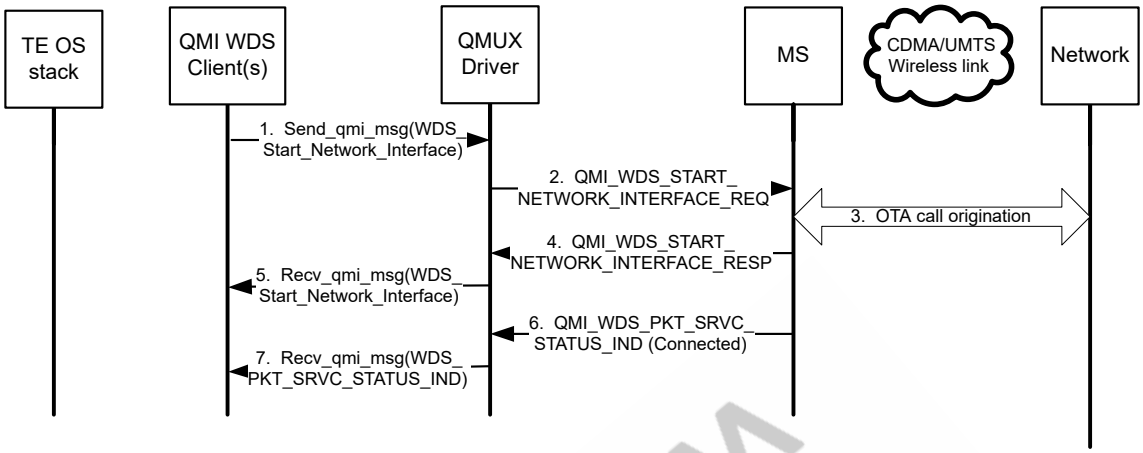

Figure 2-6 RmNet data session setup

The following steps correspond to the preceding figure:

1. The client on the TE sends a request to start the RmNet data session.
2. The QMUX driver adds the QMUX header to the request before sending it to the MS.
3. On receiving this request, the MS initiates the procedure to bring up the data session OTA.
4. Once the OTA data session is up, the MS sends a response indicating success destined to the client that initiated the request. In call origination failure, the MS sends a failure response and the following steps do not occur.
5. The QMUX driver removes the QMUX header and passes the response message to the appropriate client.
6. The MS also sends a broadcast indication to notify that the data session is now connected.
7. The QMUX driver on the TE forwards it to the QMI WDS clients.

NOTE A successful response means only that the data session originated successfully over the wireless network. The IP address configuration still must be completed on the TE (see *TE IP after RmNet address assignment* for details) before data transfer begins.

2.3.2 TE IP after RmNet address assignment

Using DHCP for IPv4 address

Preconditions and postconditions for using DHCPv4 are as follows:

- Precondition – The data session must be set up prior to this.
- Postcondition – The TE IP configuration is complete and data packets are transferred.

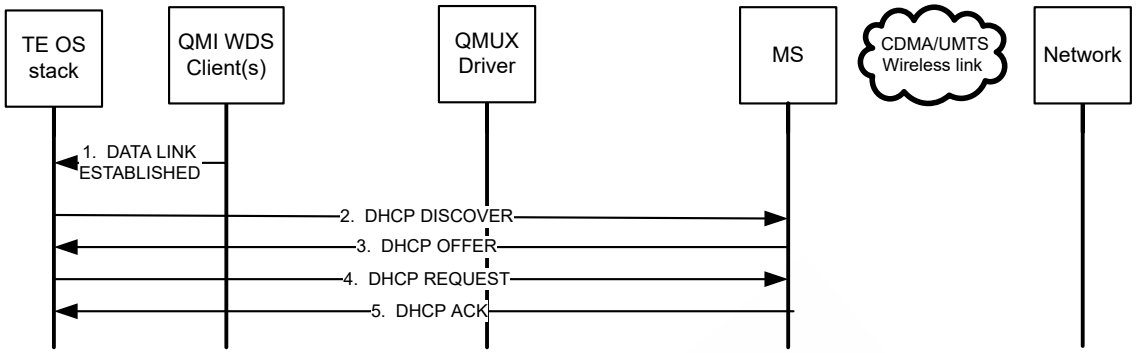

Figure 2-7 TE IPv4 address configuration using DHCP

The following steps correspond to the preceding figure:

1. The WDS client notifies the OS running on the TE that the data link is established.
2. Steps 2 to 5 DHCP negotiation occurs between the TE and the MSM to assign an IP address and other IP configuration to the TE. The IP address assigned to the TE is the same as the IP address assigned to the MS by the network.

Using stateless address autoconfiguration for IPv6 address

Preconditions and postconditions for stateless address autoconfiguration for IPv6 address are as follows:

- Precondition – the data session must be set up prior to this.
- Postcondition – the TE IP configuration is complete and data packets are transferred.

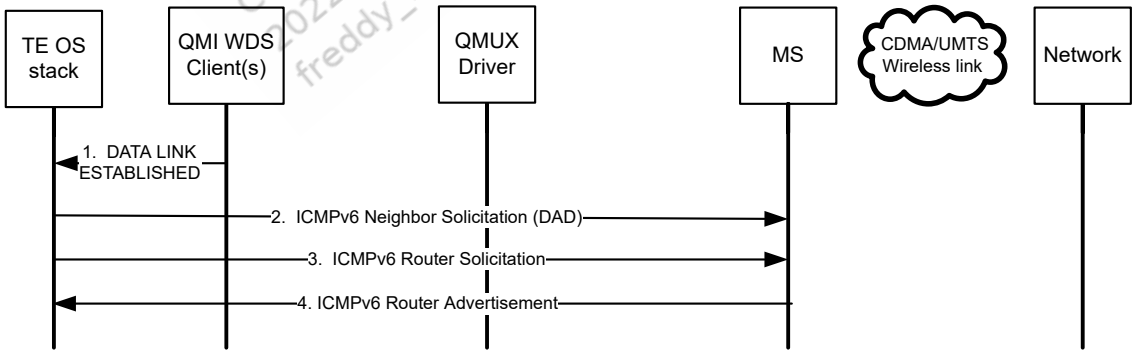

Figure 2-8 TE IPv6 address configuration using stateless autoconfiguration

The following steps correspond to the preceding figure:

1. The WDS client notifies the OS running on the TE that the data link is established.
2. Steps 2 to 4. The TE must do DAD and stateless address autoconfiguration to configure the address.

NOTE TE must do DAD for each ID it generates.

Using QMI (only IPv4)

Preconditions and postconditions for using QMI are as follows:

- Precondition – the data session must be set up prior to this.
- Postcondition – the TE IP configuration is complete and data packets are transferred.

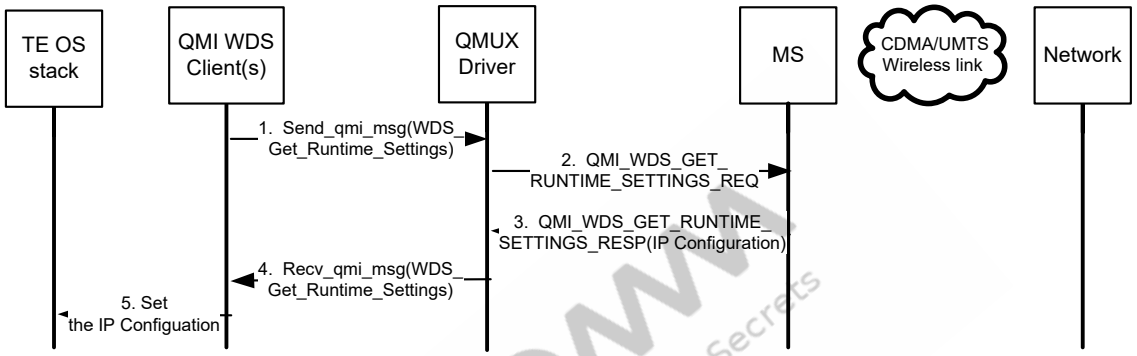

Figure 2-9 TE IPv4 address configuration using QMI

The following steps correspond to the preceding figure:

1. Steps 1 to 2, the WDS client sends a request to the MS to query the IP configuration to be set on the TE.
2. Steps 3 to 4, the MS responds with the IP address and other configuration. The IP address assigned to the TE is same as the network assigned IP address.
3. Step 5, the client then configures that IP address and other configuration parameters on the TE using host-specific APIs.

RmNet guard timer

When the data session is set up with the wireless network, a guard timer is started. The TE is expected to query its IP configuration (as shown in Sections *Using DHCP for IPv4 address* to *Using QMI (only IPv4)*) before the expiry of this timer. If the timer expires before the TE queries the IP configuration from the device, the data session is torn down. This is performed to save the wireless network resources. The guard timer value is set to 2 min.

2.3.3 RmNet data session teardown

Preconditions and postconditions for tearing down the RmNet data session are as follows:

- Precondition – The data session is set up earlier.
- Postcondition – The data session is torn down and the TE loses IP connectivity.

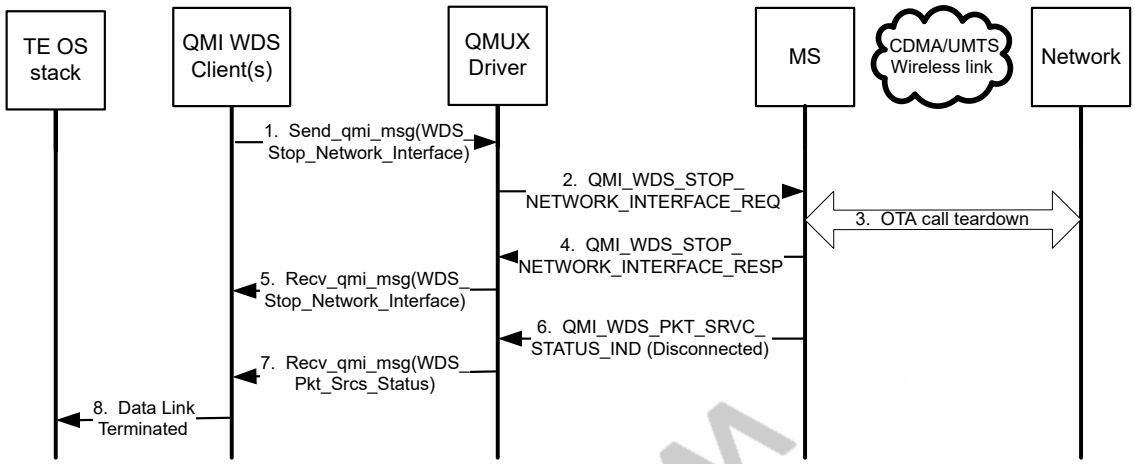

Figure 2-10 Data session teardown

The data session teardown occurs as follows:

1. The client sends a request to stop the RmNet data session.
2. The QMUX driver adds the QMUX header to the message and sends the request to the MS.
3. The MS initiates the call teardown OTA.
4. The MS sends a SUCCESS response destined for the client.
5. The QMUX driver removes the QMUX header and passes the response to the appropriate client.
6. The MS also sends a broadcast indication to notify that the data session is disconnected.
7. The QMUX driver on the TE forwards it to the WDS client.
8. The WDS client notifies the OS running on the TE that the data link was terminated.

2.3.4 RmNet Autoconnect session establishment and teardown

The preceding sections illustrate the mechanism manually to bring up and tear down a data session. To support always on operation, RmNet has an Autoconnect mode that, when enabled, brings up the data session automatically when the device is connected to the TE, and tears it down automatically when the device is disconnected from the TE. Autoconnect is available only for RmNet over plug-and-play interconnects, such as USB.

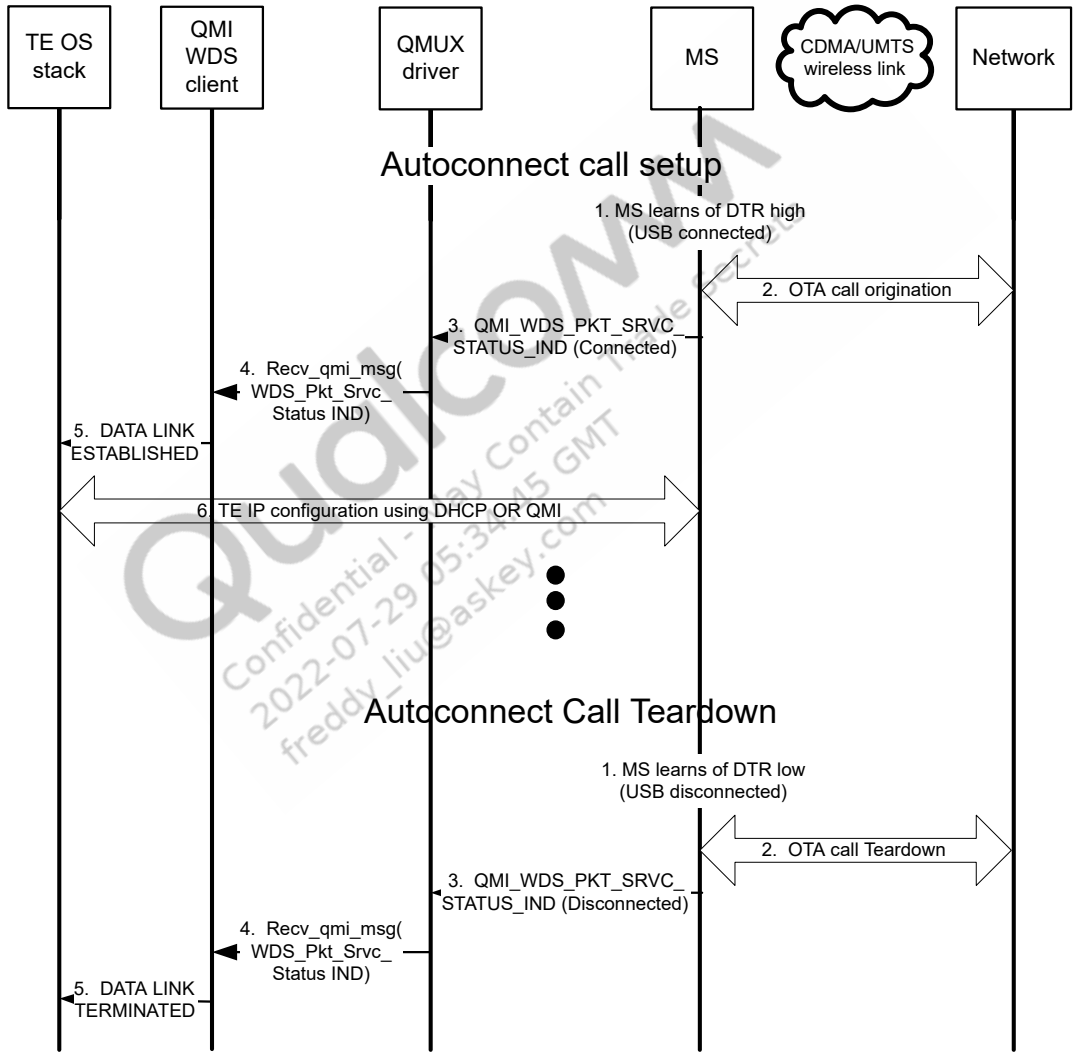

Figure 2-11 Autoconnect mode

To set up an Autoconnect data session:

1. The MS learns that the USB is connected. For example, through DTR high.
2. The MS brings up the data session OTA.
3. The MS sends a broadcast QMI wireless distribution service (WDS) indication to notify that the data session is connected.

4. The QMUX driver on the TE forwards it to the WDS client.
5. The WDS client then notifies the OS running on the TE that the data link is established.
6. The IP configuration of the TE is performed using DHCP or QMI, as shown in Section *TE IP after RmNet address assignment*.

To set up an Autoconnect data session teardown:

1. The MS learns that the USB is disconnected. For example, through DTR low.
2. The MS tears down the data session OTA.
3. The MS sends a broadcast QMI WDS indication to notify that the data session is disconnected.
4. The QMUX driver on the TE forwards it to the WDS client.
5. The WDS client then notifies the OS running on the TE that data connectivity is terminated.

During the Autoconnect mode operation, the data session becomes disconnected for reasons other than USB disconnection. For example, loss of signal, and so on. The RmNet device tries to re-establish the data session using a backoff mechanism. RmNet starts with a timer of one sec. Every time a data session establishment attempt fails (lower layers of the device or the network rejects), it backs off the timer by a factor of two and then the MS tries again when the timer expires. The timer backoff continues up to a maximum of two min, after which RmNet retries every two min. During this process, if any attempt is successful, the timer is reset back to the original value of one sec.

2.3.5 IP address change on the TE

Preconditions and postconditions for changing IPv4 or IPv6 address are as follows:

- Precondition – The data session is up and the IP address on the TE has been configured.
- Postcondition – The TE was notified of an IP address change and the TE IP address configuration is performed again.

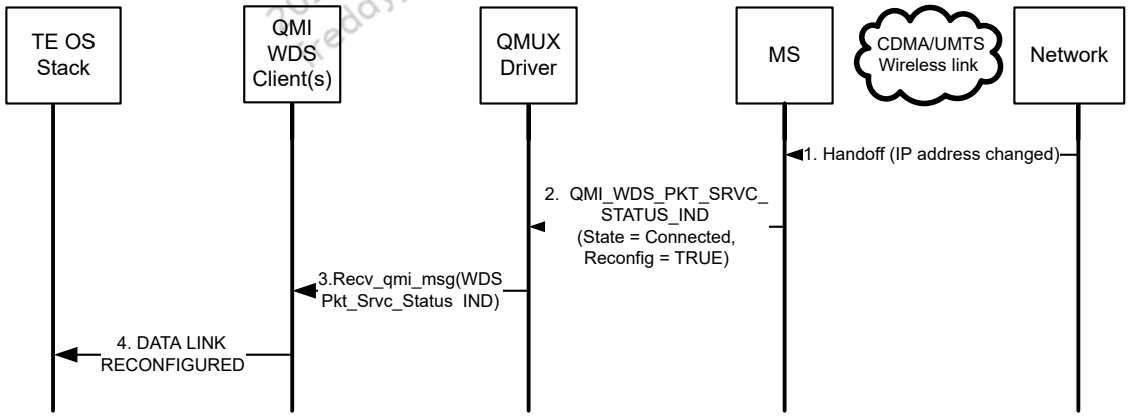

Figure 2-12 IP address change

To change an IP address:

1. Handoff takes place, causing the MS IP address to change.
2. The MS sends a broadcast WDS indication to notify that the data session is connected, but reconfiguration is required.

3. The QMUX driver on the TE forwards it to the QMI WDS client.
4. The WDS client notifies the OS running on the TE that the data link IP configuration must be reconfigured.

2.4 Interconnection with different devices and interfaces

Currently, RmNet is supported over USB and shared memory interconnects. One or more NET devices (RmNet logical devices) are supported.

Each logical device must have a separate data and control channel. Channel independence implies that each channel must act as if there were no physical coupling between the communication channels, including (but not limited to) separate Tx and Rx path queuing, independent flow control mechanisms, and independent data transmission scheduling. The interconnection must provide for framing of data packets exchanged, that is, delineating packet boundaries to the transport proto col. For example, 802.3.

To support multiple logical devices, the underlying interconnect must provide for a mechanism for multiplexing multiple logical devices over a single physical connection.

2.5 Multi-RmNet

Multi-RmNet refers to support for multiple logical devices, that is, NET interfaces. Multi-RmNet feature supports multiple IP data sessions (each session is either IPv4 or IPv6). A maximum of three RmNet instances are used simultaneously.

Each RmNet logical device appears as an independent network adapter on the TE, capable of getting its own IP address and transferring data independently. The following figure shows this for two RmNet instances. Each instance operates on a separate port and is independent of the other instance:

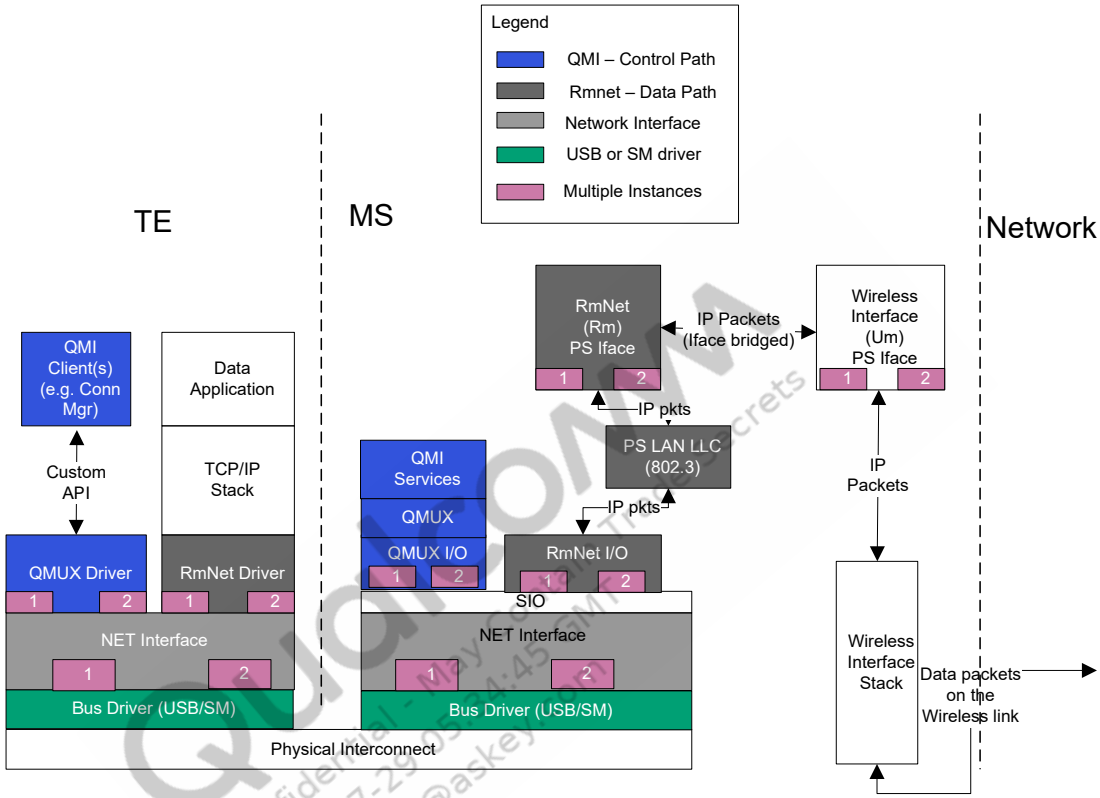

Figure 2-13 Multi-RmNet

2.6 RmNet configuration parameters

The following table lists the RmNet configuration parameters that are supported. These are applicable only for RmNet over USB.

Table 2-1 RmNet configuration parameters

Configuration parameter	Description	Default value
USB_CURRENT_DEVICE (FS-USB) – NV item 2782 HS_USB_CURRENT_COMPOSITION (HS_USB) - NV item 4526	Determines whether RmNet is part of the USB composition; must be set to a composition value that includes RmNet	0 (RmNet not present)
RMNET_AUTOCONNECT – NV item 3534	Determines whether RmNet autoconnect is enabled; applicable only for RmNet over USB; consists of two components. an index and an enable or disable setting; the index indicates RmNet instance (currently only one instance, that is, index 0 is supported); second component is set to 1 or 0 to enable or disable autoconnect on that particular index	{0, 0} (Autoconnect disabled)

2.7 Non-IP data delivery

The non-IP data delivery (NIDD) is a new feature introduced to NB.

Refer to Support for Non-IP Data Delivery (NIDD) in the following specification:

General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access (3GPP TS 23.401 Version 14.4.0)

Check if a platform or build supports it

Non-IP is implemented per FR38978 (CR2010416 and CR2046848). It is delivered by post CS of MPSS.JO3.0.2. Any build following this one should include these CRs.

For any other platform or build, check if these CRs are applicable.

Supported call flows

Supported: AP embedded and tethered RmNet calls only.

Not supported: Modem embedded call, DUN.

2.7.1 wqa1576839440414AP side call flow to implement AP embedded call

Perform the following to bring-up the non-IP data call:

Get a data service handle

```
qapi_Status_t qapi_DSS_Get_Data_Srvc_Hndl(qapi_DSS_Net_Ev_CB_t user_cb_fn,
void *user_data, qapi_DSS_Hndl_t *hndl);
```

Configure the data call parameters prior to starting the data call:

1. Set the call parameter `QAPI_DSS_CALL_INFO_MO_EXCEPTION_DATA_E to QAPI_DSS_EXT_RADIO_TECH_NONIP`.

   ```
   qapi_Status_t qapi_DSS_Set_Data_Call_Param(qapi_DSS_Hndl_t hndl,
   qapi_DSS_Call_Param_Identifier_t identifier, qapi_DSS_Call_Param_Value_t
   *info);
   ```

2. Start the data call.

   ```
   qapi_Status_t qapi_DSS_Start_Data_Call(qapi_DSS_Hndl_t hndl);
   ```

3. Send the non-IP data.

   ```
   qapi_Status_t qapi_DSS_Nipd_Send(qapi_DSS_Hndl_t hndl, uint8_t *data,
   uint32_t data_len, uint8_t ex_data);
   ```

   The modem side handler for SNI is triggered in this step. wds_start_network_interface_req_msg to see how these APIs are handled.

   - profile_index → Use QMItest pro tool to set the PDN type to non-IP
   - ip_family_preference → Set this to IPv4
   - ext_technology_preference → Set this to non IP

4. Stop the data call.

   ```
   qapi_Status_t qapi_DSS_Stop_Data_Call(qapi_DSS_Hndl_t hndl);
   ```

5. Release the service handle.

   ```
   qapi_Status_t qapi_DSS_Rel_Data_Srvc_Hndl(qapi_DSS_Hndl_t hndl);
   ```

Test

The non-IP data flow on AP and modem side is verified using the following testing applications:

1. CLI dss_netapp
2. QMI test pro

See *Non-IP Data Call Overview* (80-P2200-67) for detailed test steps.

3 QCMobileAP software

This chapter describes QCMobileAP architecture, features, USB tethering mechanisms, and QCMobileAP limitations.

3.1 QCMobileAP overview

The QCMobileAP user equipment acts as an access point (AP) which allows multiple clients to connect to the Internet over Wi-Fi.

MobileAP is also commonly known by other names. For example, QCMobileAP, mobile hotspot, Wi-Fi tethering, MiFi, and pocket router.

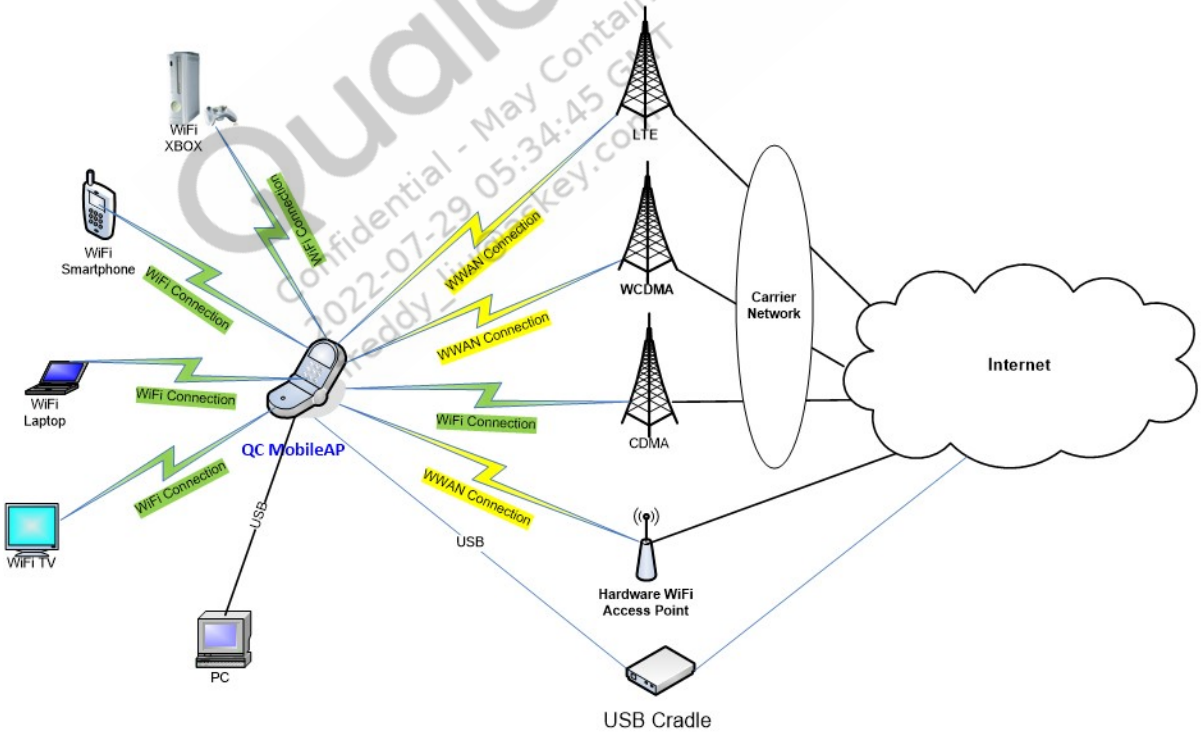

Figure 3-1 QCMobileAP overview

NOTE Only one USB port is supported, it can be configured either in WAN mode (connected to USB cradle) or LAN mode (connected to PC).

3.2 QCMobileAP features

This section lists the complete QCMobileAP features in the table.

Table 3-1 Feature set

QCMobileAP features	Comments
Reference connection manager on application processor	Provides reference connection manager that handles QCMobileAP configuration and WWAN connectivity.
Data forwarding to and from 4G or 3G network	All RAT types supported: LTE, GW, and 1xEV-DO
IPv4 NAT (symmetric NAT, full cone NAT, address restricted cone NAT, port restricted cone NAT)	NAT support
DHCP server	Supports multiple clients
Proxy DNS	DNS proxy for NATed clients
IP firewall	Firewalling based on configured rules
VPN passthrough	IPSec, PPTP, and L2TP
Connection management	Automatic WWAN connection management
Port forwarding	Static NAT entries
Configuration API	API to configure NAT tables, firewalls, DHCP address range, and so on.
ALGs	FTP, PPTP, SIP, RTSP; work in progress for enabling more ALGs, that is, H.323, IRC, UDPLITE, AMANDA, NETBIOS, SNMP, and TFTP.
DMZ	Forwards all downlink packets to a preset client address if no NAT match found
Enable or disable roaming autoconnect	Provides configuration function that enables or disables autoconnect during roaming
QCMobileAP IPv6	Supports IPv6 over QCMobileAP
AP+STA mode	Operates as Wi-Fi AP as well as Wi-Fi client and makes external Wi-Fi hotspot as backhaul
AP+AP mode (dual SSID)	Supports guest AP mode and provides configuration to control access for the guest AP clients
RNDIS or Std-ECM and Wi-Fi AP concurrency	Enables USB tethered clients and QCMobileAP Wi-Fi clients to access the same WWAN concurrently
UPnP and DLNA	Enables UPnP IGD v1 device class and DLNA media server
Bonjour	Enables Bonjour (mDNS resolver) for device and service discovery
HTTP or HTTPS reference webserver	Enables reference webserver for web-based QCMobileAP configuration
Concurrent DUN+MobileAP	Enables USB connected DUN TE and QCMobileAP Wi-Fi clients to communicate to the same WWAN PDN
IMS voice over QCMobileAP	Supports IMS VoIP Wi-Fi clients to communicate over LTE network (IMS and Internet on the same PDN)
Connected devices display	Shows the LAN devices IP and MAC addresses connected over Wi-Fi or USB

QCMobileAP features	Comments
Configuration storage and factory reset	Provides support for saving the configuration file and to reset to default configuration
IPv6 prefix delegation	Allows granting LTE-delegated IPv6 prefixes to the LAN clients
USB cradle mode	Supports QCMobileAP device connection to external modem (cradle) over USB and provides data backhaul
AP+STA bridging mode	Enables bridging between LAN interfaces (Wi-Fi AP and USB interface) and the Wi-Fi STA interface in concurrent WLAN AP+STA mode.
AP+AP+STA mode	Combination of AP+AP and AP+STA; STA gets configured as the backhaul.
ODU device enablement	MDM device acts as an outdoor data unit (ODU) and provides connectivity over Ethernet to the home.
CPE device enablement	MDM device works as a low-cost home router and has Wi-Fi as well as Ethernet connections.
eMBMS over QCMobileAP and ODU	Supports eMBMS traffic management and forwarding to LAN eMBMS clients.
UPnP IGDv2	Enables IGD ver 2, IPv6 support, firewall pinholes, allows single application through restricted firewall, statistics for pinhole, actions for control of port forwarding ranges.
Dynamic DNS	Dynamically updates DNS records on the upstream name servers with active DNS configuration of hostnames, URLs, addresses of hosts connected to QCMobileAP.
Standalone STA mode	Enables standalone STA mode on the WLAN chip to enable embedded applications running on Cortex-A7 to set up connections through the external Wi-Fi hotspot.

3.3 Modem architecture with QCMobileAP overview

This section describes QCMobileAP architecture, features, and concurrency for Qualcomm® Hexagon™ DSP, Cortex-A, or tethered applications.

The following figure shows the modem system architecture:

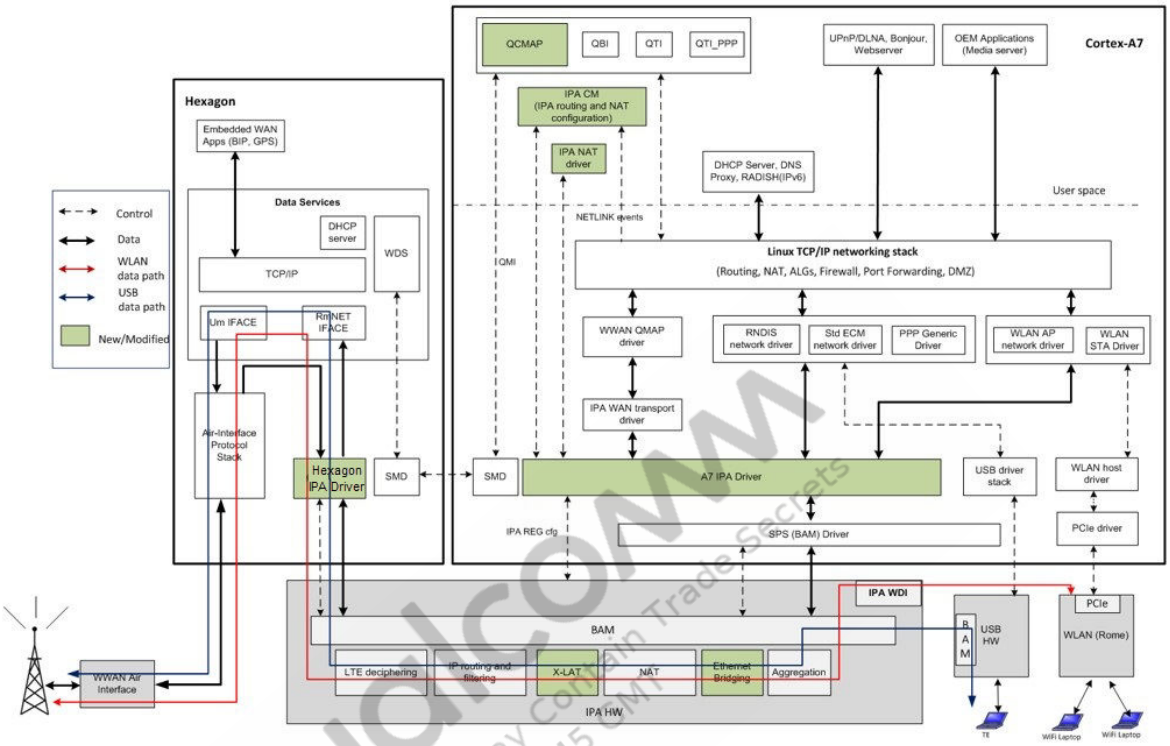

Figure 3-2 Modem system architecture

3.3.1 QCMobileAP overview – IPv4 architecture

The following are the QCMobileAP components and its features on applications side (ARM Cortex-A7):

- QCMobileAP provides API to connect with the connection manager daemon.
  - Provides an interface to turn QCMobileAP mode on and off.
  - Provides an interface to configure network policy for WWAN network selection.
  - Brings up and tears down the WWAN connection.
  - Configures the DHCP server.
  - Invokes Linux kernel utilities, that is, IP tables to configure Linux side NAT, routing, and firewall.
  - Provides NAT, routing, and firewall functionalities on Cortex-A7.
  - Provides an interface to turn concurrent AP+STA router or bridge mode on and off.
  - Provides an interface to turn USB WAN mode on and off, while connected to a cradle device, providing data backhaul connectivity.
  - Provides an interface to turn AP+AP mode (dual-SSID) on and off.
- Cortex-A7 embedded applications bring up a data call on extra PDNs using the data management APIs or on same PDN as for QCMobileAP data call.

- QCMobileAP connection manager service interfaces with clients like QTI Tethering Interface to provide hostless tethering functionality.
- Linux kernel handles routing between Cortex-A7 embedded applications, WLAN clients, and USB terminal equipment (TE).
- IP accelerator (IPA) – enables hardware-accelerated data path for IP packets to provide high data rates and to save Cortex-A7 CPU use; key features are as follows:
  - Filtering – Once the IP packets are available, based on the filtering rules configured, IPA selectively filters out traffic to be sent to different endpoints, which helps to achieve functionalities like application level gateway (ALG) and firewall.
  - Network address translation (NAT) – the IPA does both source and destination NAT.
  - Routing – based on the NAT block output for WWAN data and on the source or destination address or port, the routing block routes the IP packets to either the modem address space or to one of the several BAM endpoints, each of which is dedicated for a specific destination.
  - Insert header – based on the rule that matches, the IPA prepends a transfer with a header; it also attaches a header on a per-IP packet basis.
  - Aggregates or deaggregates IP frames.
  - Deciphers LTE
  - IPv4 to IPv6 translation using XLAT – support RFC 6877 – 464XLAT: Combination of stateful and stateless translation.
  - Ethernet bridging in IPA – support IPA hardware accelerated data path for USB ↔ WLAN and WLAN ↔ WLAN (inter-BSSID) use cases.
- Initial NATing and routing are performed in the applications Cortex-A7 Linux stack. Once the dynamic Conntrack table entry is created, all subsequent NATing and routing gets performed in IPA (optimized hardware-accelerated path).
- WLAN data path offloads to IPA.

3.3.2 QCMobileAP overview – IPv6 architecture

The following are the QCMobileAP components and its features on Cortex-A7 side:

- On Cortex-A7 side, RADISH acts as a multicast forwarder. It forwards multicast packets (router advertisement (RA), router solicitation (RS), neighbor advertisement (NA), neighbor solicitation (NS)) between the IPv6 WWAN network interface and the LAN interfaces.
- When an IPv6 call is brought up, the modem sends RA, which is forwarded through the LAN interfaces to Wi-Fi and USB clients via RADISH.
- On the new Wi-Fi client connection, RADISH forwards RS to the modem (Hexagon) and receives RA, which is then propagated to the interfaces.

3.3.3 USB tethering mechanisms with QCMobileAP

The following are the supported mechanisms of USB tethering:

- RmNet
  - QTI-proprietary tethering
  - Network-assigned address handed to the TE
  - No concurrent support with QCMobileAP
- RNDIS
  - Microsoft-promoted tethering used on Windows XP/7/8 and some Linux distributions
  - Private IP address is assigned to the TE
  - Supported concurrently with QCMobileAP
- Std ECM
  - Used by MAC OS and many Linux distributions
  - Private IP address is assigned to the TE
  - Supported concurrently with QCMobileAP
- DUN
  - Used by all operating systems.
  - Private IP address is assigned to the TE
  - Supported concurrently with QCMobileAP
- MBIM
  - Microsoft-promoted tethering used on Windows 8 and later
  - Network-assigned address handed to the TE
  - No concurrent support with QCMobileAP

3.3.4 Concurrency for Hexagon or Cortex-A7 or tethered applications

The following are the QCMobileAP concurrencies:

- DHCP server single instance assigns IP addresses to all hosts including WLAN clients and USB TE. Therefore, all the clients get IP addresses allocated from the same subnet and address range.
- QCMobileAP brings up a WWAN call and enumerates the interface with a network assigned public IP address on Cortex-A7.
- Embedded applications on Cortex-A7 use the Internet PDN network interface brought up by QCMobileAP. Alternatively, applications running on the Cortex-A7 brings up data calls on different

PDNs. For example, admin PDN. And the corresponding network interface is used by setting up appropriate routing rules.

- Supported concurrencies are as follows:
  - Applications on Hexagon communicating over WWAN using the same PDN as QCMobileAP. For example, GPS using Internet PDN.
  - Applications on Hexagon communicating over WWAN using a PDN different from what is used by QCMobileAP. For example, BIP using admin PDN.
  - Applications on Cortex-A7 communicating over WWAN using the same PDN as QCMobileAP. For example, media client using Internet PDN.
  - Applications on Cortex-A7 communicating over WWAN using a PDN different from what is used by QCMobileAP. For example, OTADM using admin PDN.
  - Applications on Cortex-A7 communicating with LAN clients including USB (RNDIS or Std ECM or DUN) TE and WLAN clients. For example, Cortex-A7 applications like web server, media server, file storage.
  - USB (RNDIS or Std ECM or DUN) TE, WLAN clients, and embedded clients communicating with each other.
  - USB (RNDIS or Std ECM or DUN) TE, WLAN clients, and embedded clients communicating with WWAN.

3.4 Limitations with QCMobileAP

Seamless transition among WWAN network, WLAN external hotspot, and USB backhaul are not supported. When switching to concurrent AP+STA or USB Backhaul mode, the existing data sessions of LAN clients, and embedded application on the Cortex-A7 are no longer maintained and are disconnected abruptly. Fresh connections must be re-established.

A References

A.1 Related documents

Title	Number
Qualcomm Technologies, Inc.	
Qualcomm Interface Quality of Service (QMI QoS) Feature Description Document	80-VF536-2
QMI Control Service (QMI_CTL) Spec	80-VB816-3
Non-IP Data Call Overview	80-P2200-67
Qualcomm MSM Interface (QMI) Architecture	80-VB816-1
Modem IPA Configuration Manager User Guide	80-NC254-64
Standards	
Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall Description; Stage 2	3GPP TS 36.200
Nonaccess-Stratum (NAS) Protocol for Evolved Packet System (EPS); Stage 3	3GPP TS 24.301
Quality of Service Concept and Architecture	3GPP TS 23.107
General Packet Radio Service (GPRS) Enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Access	3GPP TS 23.401
Mobile Radio Interface Layer 3 Specification; Core Network Protocols	3GPP TS 24.008
Mobile Radio Interface Signalling Layer 3; General Aspects	3GPP TS 24.007
Non-Access Stratum (NAS) configuration Management Object (MO)	3GPP TS 24.368

A.2 Acronyms and terms

Acronym or term	Definition
ADB	Android debug bridge
BAM	Bus access manager
DL	Downlink
DSD	Data system determination
EFS	Embedded file system
eNB	Evolved node B
EPC	Evolved packet core
EPS	Evolved packet system (E-UTRAN + EPC)

Acronym or term	Definition
E-UTRAN	Evolved universal terrestrial radio access network
GERAN	GSM or EDGE radio access network
IPA	IP accelerator
IRAT	Inter radio access technology
LTE	Long term evolution
MBIM	Mobile broadband interface model
MHI	Modem host interface
MME	Mobility management element
NAS	Non access stratum
NAT	Network address translation
PDCP	Packet data convergence protocol
PDN	Packet data network
P-GW	Packet data network gateway
QMI	Qualcomm MSM interface
QoS	Quality of service
RAT	Radio access technology
RLC	Radio link control
RmNet	Remote wireless wide area network
RNDIS	Remote network driver interface specification
SAE	System architecture evolution
S-GW	Serving gateway
TCP	Transmission control protocol
UE	User equipment
UL	Uplink