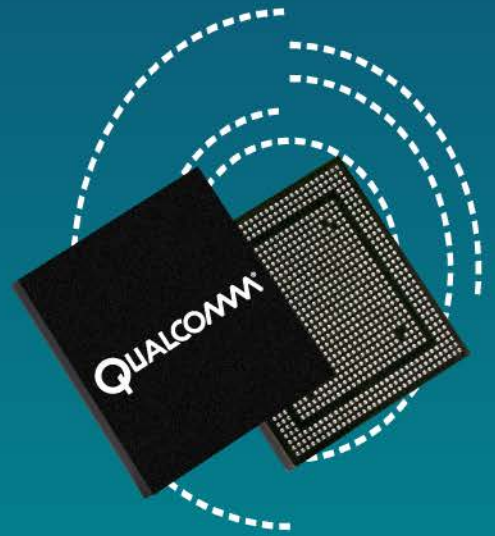
Qualcomm
TECHNOLOGIES, INC



QMAP Overview

80-NJ737-1 B

Confidential and Proprietary – Qualcomm Technologies, Inc.

NO PUBLIC DISCLOSURE PERMITTED: Please report postings of this document on public servers or websites to: DocCtrlAgent@qualcomm.com.

Restricted Distribution: Not to be distributed to anyone who is not an employee of either Qualcomm or its subsidiaries without the express approval of Qualcomm's Configuration Management.

Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm Technologies, Inc.

Qualcomm reserves the right to make changes to the product(s) or information contained herein without notice. No liability is assumed for any damages arising directly or indirectly by their use or application. The information provided in this document is provided on an "as is" basis.

This document contains confidential and proprietary information and must be shredded when discarded.

Qualcomm is a trademark of QUALCOMM Incorporated, registered in the United States and other countries. All QUALCOMM Incorporated trademarks are used with permission. Other product and brand names may be trademarks or registered trademarks of their respective owners.

This technical data may be subject to U.S. and international export, re-export, or transfer ("export") laws. Diversion contrary to U.S. and international law is strictly prohibited.

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

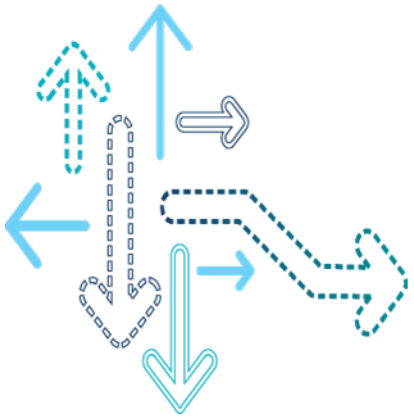
Revision History

Revision	Date	Description
A	Aug 2013	Initial release
B	Nov 2013	Updated QMAP Protocol Definition section and added Flow Control and QMAP Configuration sections

Contents

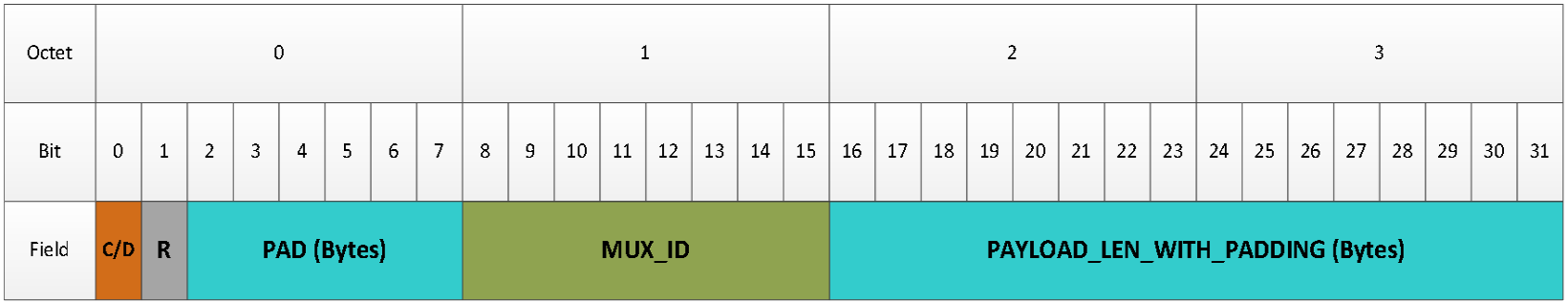
- QMAP Protocol Definition
- Flow Control
- QMAP Configuration
- References
- Questions?

QMAP Protocol Definition



QMAP Header Format

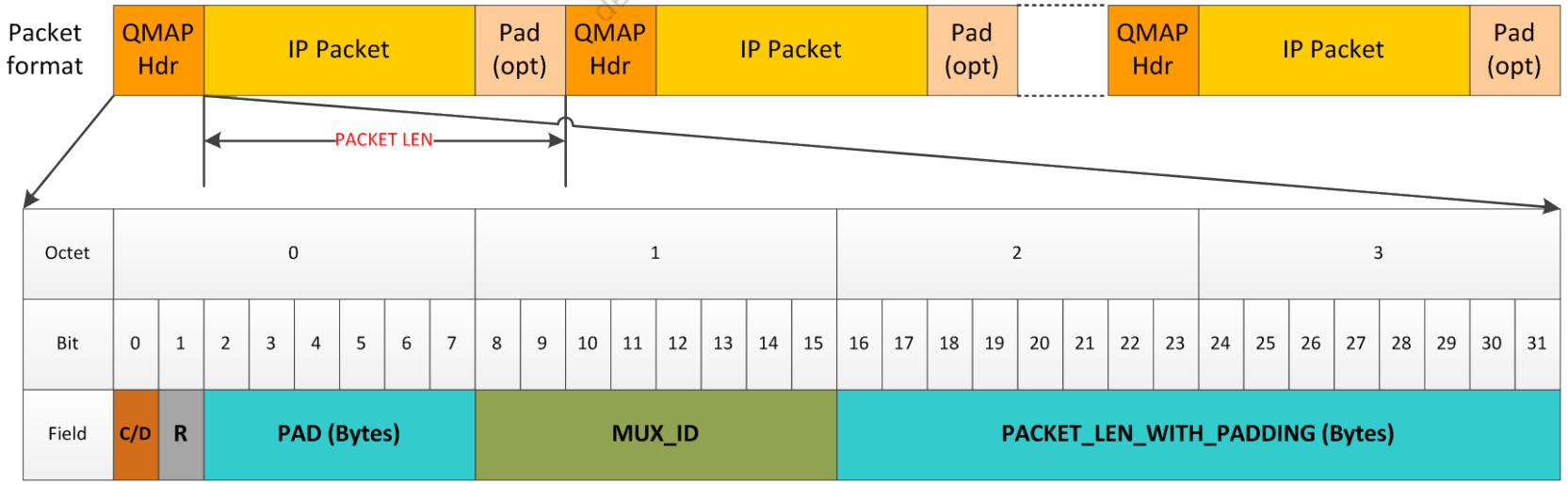
- QMAP header is 4 bytes long and has the following fields:
  - C/D bit – 1 bit; indicates if it is a QMAP control command or a data packet
    - 1 – QMAP control command
    - 0 – Data packet
  - RESERVED – 1 bit; set to 0; must be ignored by the receiver
  - PAD – 6 bits; indicates number of bytes padded to achieve a minimum of 4 byte alignment
    - Padded bytes may or may not be set to 0; receiver must ignore these bytes regardless
  - MUX_ID – 8 bits; indicates mux channel ID
    - If mux is not negotiated, this field is set to 0 and must be ignored by the receiver
    - If mux is negotiated, MUX_ID can be in the range of 1 to 0x7F; MUX_IDs 0 and 0x80 – 0xFF are reserved
  - PAYLOAD_LEN_WITH_PADDING – 16 bits; total payload length in bytes including padding
    - Does not include QMAP header
    - Receiver must ignore a packet with this field set to 0; QMAP packet will not carry any payload in this case



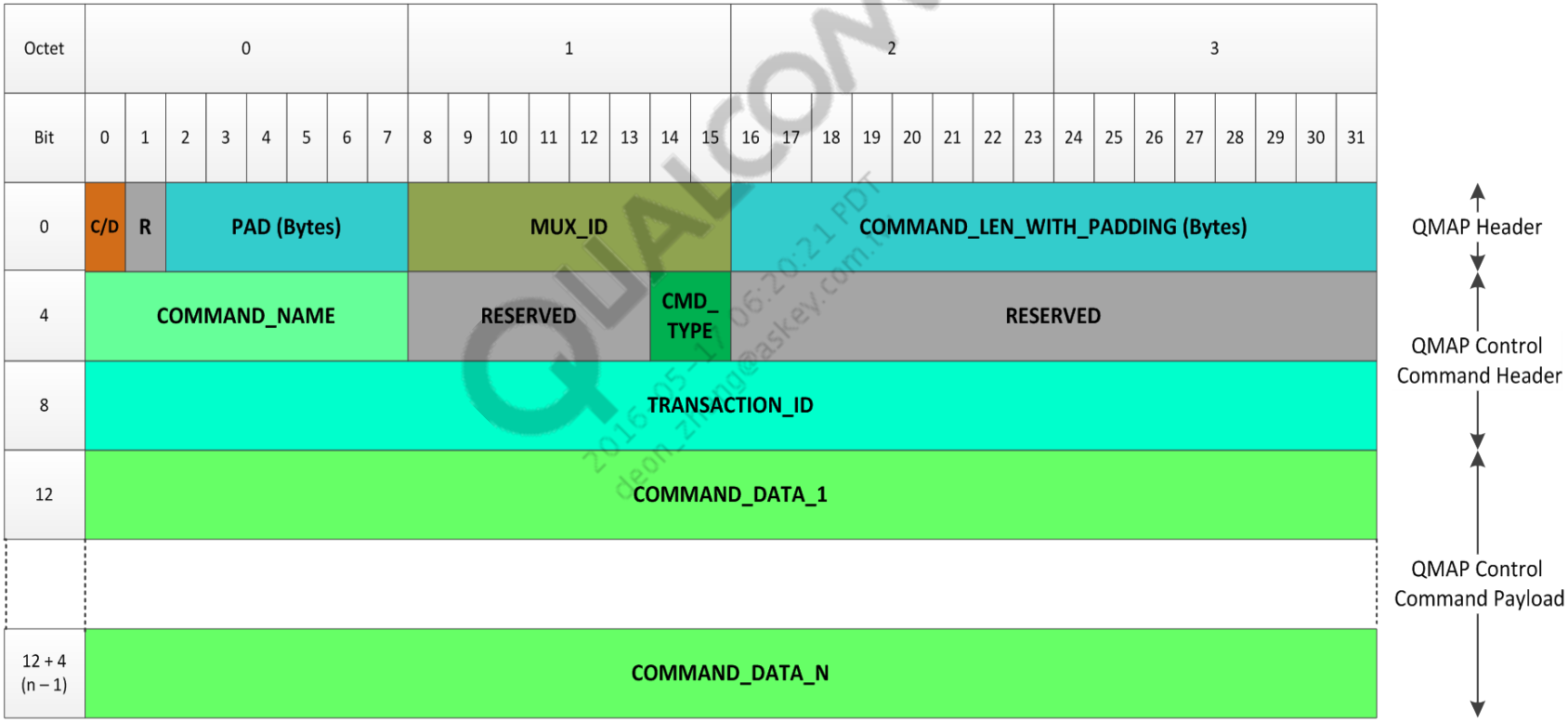
Note: QMAP header's endianness is the same as that of IP packet's.

QMAP Data Packets

- QMAP header is prepended to an IP packet
  - Even when multiple IP packets are aggregated, each IP packet has its own QMAP header
- Following QMAP header is used for data packets:
  - C/D bit – 1 bit; set to 0
  - PACKET_LEN_WITH_PADDING – 16 bits; total packet length in bytes including padding
    - Length is from the start of the IP header
  - Other fields are same as the standard QMAP header



QMAP Control Commands



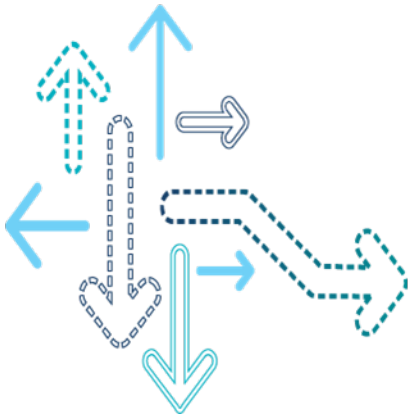
QMAP Control Commands (cont.)

- QMAP header
  - C/D bit – 1 bit; set to 1
  - COMMAND_LEN_WITH_PADDING – 16 bits; total command length in bytes including padding
    - Does not include QMAP header
    - Includes QMAP control command header and QMAP control command payload
    - Must be a multiple of 4
  - Other fields are same as the standard QMAP header
- QMAP control command header
  - COMMAND_NAME – 8 bits; indicates what QMAP control command it is
  - COMMAND_TYPE – 2 bits; indicates what type of QMAP control command it is
    - 0 – Request, i.e., sender is sending a QMAP control command to the receiver
    - 1 – Ack, i.e., receiver is acknowledging that it received a QMAP control command and that it successfully processed the command
    - 2 – Unsupported command, i.e., receiver does not support this QMAP control command
    - 3 – Invalid command, i.e., receiver encountered an error while processing the QMAP control command, probably because QMAP control command is malformed
  - RESERVED – 6 bits, 16 bits; must be ignored by the receiver
  - TRANSACTION_ID – 32 bits; each QMAP control command must be sent with a monotonically increasing transaction ID
    - Useful for debugging
    - System time could be used as transaction ID

QMAP Control Commands (cont.)

- QMAP control command payload
  - COMMAND_DATA_1 to COMMAND_DATA_N – Multiple of 4 bytes; optional payload for QMAP control commands
- Upon receiving a QMAP control command with COMMAND_TYPE 0, the receiver must echo the QMAP control command as is, except for changing the COMMAND_TYPE field appropriately to 1, 2, or 3
- The sender may optionally retransmit QMAP control commands if it did not receive the response (QMAP control command with COMMAND_TYPE as 1, 2, or 3) from the receiver

Flow Control



Flow Control

- If multiple PDNs are multiplexed over one physical endpoint (one USB/HSIC interface or one MHI channel), back pressure on the physical endpoint can no longer be used to indicate flow control of one PDN to the tethered device.
  - Hence, a new mechanism is needed to flow control each PDN independently.
- There are three options:
  - Out-of-band signaling using QMI message
    - QMI WDS-based flow control – Can be used if tethered device is not using QMI QoS
    - QMI QoS-based flow control – Can be used if tethered device is using QMI QoS
  - In-band signaling using QMAP control command
- Depending on the HLOS architecture, licensees may pick either QMI-based flow control or QMAP-based flow control or both.

Flow Control Requirements

- The table shows the peak throughput rates for a given technology and the Um Watermark (WM) levels.
- (Low WM level/uplink (UL) rate) gives an upper bound on the latency requirement for flow enable message.
  - If new data is not enqueued into Um WM within this limit, throughput will suffer as L1 will not be able to send any data to the network.
- (Do Not Exceed count ((DNE) – High WM level)/interconnect data rate) gives an upper bound on the latency requirement for the flow disable message.
  - If the tethered device does not stop sending data to the modem within this limit, throughput will suffer as Um DNE may get hit resulting in dropped data.
  - Technically, the ideal limit is lower than the upperbound to account for data in flight.

	DO Rev A	LTE (CAT4)
Peak throughput rates (Mbps)	1.8 UL/3.1 downlink (DL)	50 UL/150 DL
Um low WM (bytes)	4500	120,000
Um high WM (bytes)	7000	300,000
Um DNE (bytes)	64000	450,000
Latency requirement for flow enable (ms)	18	19

QMI-Based Flow Control (Out-of-Band Signaling)

- Any HLOS typically has the following architecture:
  - Components
    - QMUX driver processing QMI messages
    - Network driver processing data packets
    - QMI WDS/QoS clients managing PDN connections
  - When modem sends a flow control message, the QMUX driver processes it and relays it to the QMI WDS/QoS client, which in turn propagates the information to the network driver so that the network driver flow controls data transfer.
- Since QMI-based flow control messages take a couple of hops before reaching the network driver, there could be some propagation delay.
  - Some reasons for propagation delay are:
    - QMUX driver and/or QMI WDS/QoS clients running in user space while network driver is running in kernel space
    - Processes/threads involved in data path having higher priority than those in control path

QMAP-Based Flow Control (In-Band Signaling)

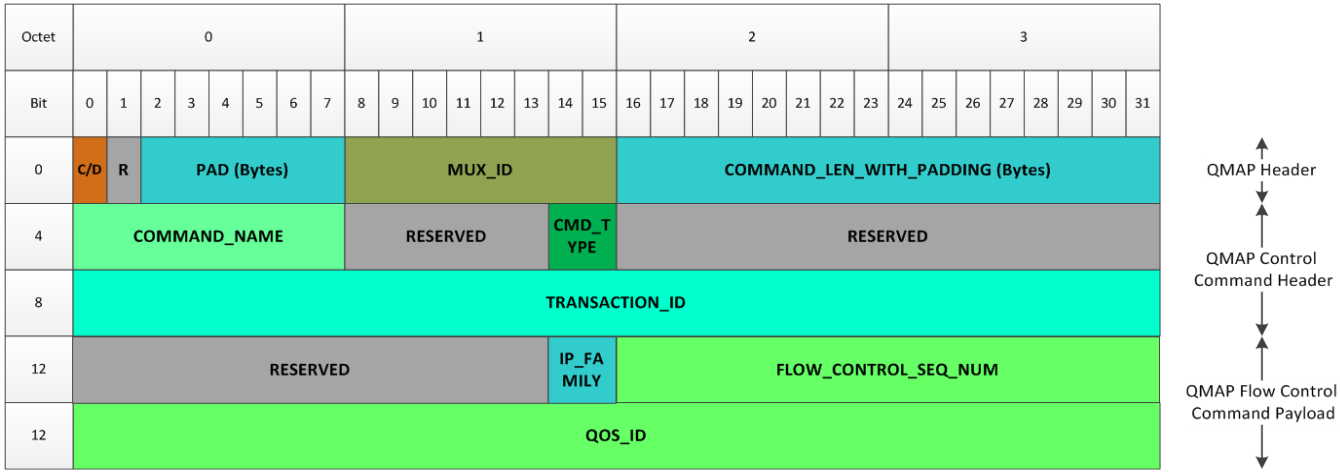
- Since the network driver itself processes QMAP commands as well, there will not be any propagation delay for flow control messages.
- Having said that, since QMAP commands are sent in the data path, they may be stuck behind a lot of DL data.
  - If the network driver does not prioritize processing QMAP commands over the data, which is in front, then depending on how much data is in front, there may be a lot of processing delay.

Recommended Flow Control Strategy

- Since the performance of QMI-based flow control or QMAP-based flow control mechanisms are heavily dependent on the HLOS architecture and implementation, each licensee is recommended to evaluate their performance and use one or the other.
  - Design supports the licensee using both and acting on whichever comes faster. If the licensee wants to use both, it may choose to do so.

QMAP Flow Control Command

- C/D – Set to 1
- COMMAND_LEN_WITH_PADDING – Set to “16 + # of bytes used for padding”
- COMMAND_NAME – Indicates what QMAP control command it is
  - 1 – Command to enable flow control, i.e., flow disable
  - 2 – Command to disable flow control, i.e., flow enable
- IP_FAMILY – 2 bits; 1 – IPv4 2, – IPv6, 0 – Both
- FLOW_CONTROL_SEQ_NUM – 16 bits; sequence number for QMAP flow control command (see next slide for more details)
- QOS_ID – 32 bits; indicates which QoS flow needs to be flow controlled within a given PDN
  - QOS_ID is same as the “QoS Identifier” concept explained in [Q3]
  - QOS_ID of 0xFFFFFFFF indicates that all flows for a given PDN are flow controlled
  - When QMI QoS is not negotiated, this field is set to 0xFFFFFFFF
- Other fields are set as described previously



QMAP Flow Control Command – Sequence Number

- Motivation
  - Sequence number is useful in detecting duplicate flow control indications in the following scenarios:
    - The sender sends a flow control command, the receiver delays responding with ACK causing the sender to retransmit the flow control command. The receiver should be able to determine that the second flow control command is a duplicate of the first and ignore it.
    - If the licensee registers for both QMI flow control indications as well as QMAP flow control commands, the receiver would always get both of them. The receiver should be able to detect that whichever is received second is a duplicate and ignore it.
  - More importantly, the sequence number is useful in detecting stale flow control indications in the following scenarios:
    - If the licensee registers for both QMI flow control indications as well as QMAP flow control commands, and if QMI flow control indications are delayed and in the mean time the flow control state changes, e.g., consider the following case:
      - The sender sent a flow disable indication via both QMAP and QMI.
      - The receiver receives the flow disable QMAP command, but does not yet receive QMI indication.
      - The sender sends a flow enable indication via both QMAP and QMI.
      - The receiver receives the flow enable QMAP command.
      - The receiver finally receives flow disable QMI indication. If the receiver processes this blindly, the receiver will be flow controlled even though it is ready to receive more data.
    - If the licensee registers for both QMI flow control indications as well as QMAP flow control commands, and if QMAP flow control commands are delayed and in the meantime the flow control state changes

QMAP Flow Control Command – Sequence Number (cont.)

- Design
  - Each time a flow is disabled, the sequence number is incremented.
  - When flow is enabled, this sequence number is not incremented. As a consequence, flow disable/flow enable pair use the same sequence number for each incarnation.
  - Both the QMI flow control indication and QMAP flow control command are sent with the same sequence number for any given flow control incarnation.
  - The receiver must ignore the flow control message if its sequence number is less than or equal to the last sequence number processed by the receiver.
  - When retransmitting a flow control message, the sender uses the same sequence number as that of the first transmission.
  - The sequence number is unique per MUX_ID, QoS_ID, and the IP family, and is reset when the flow is deleted.

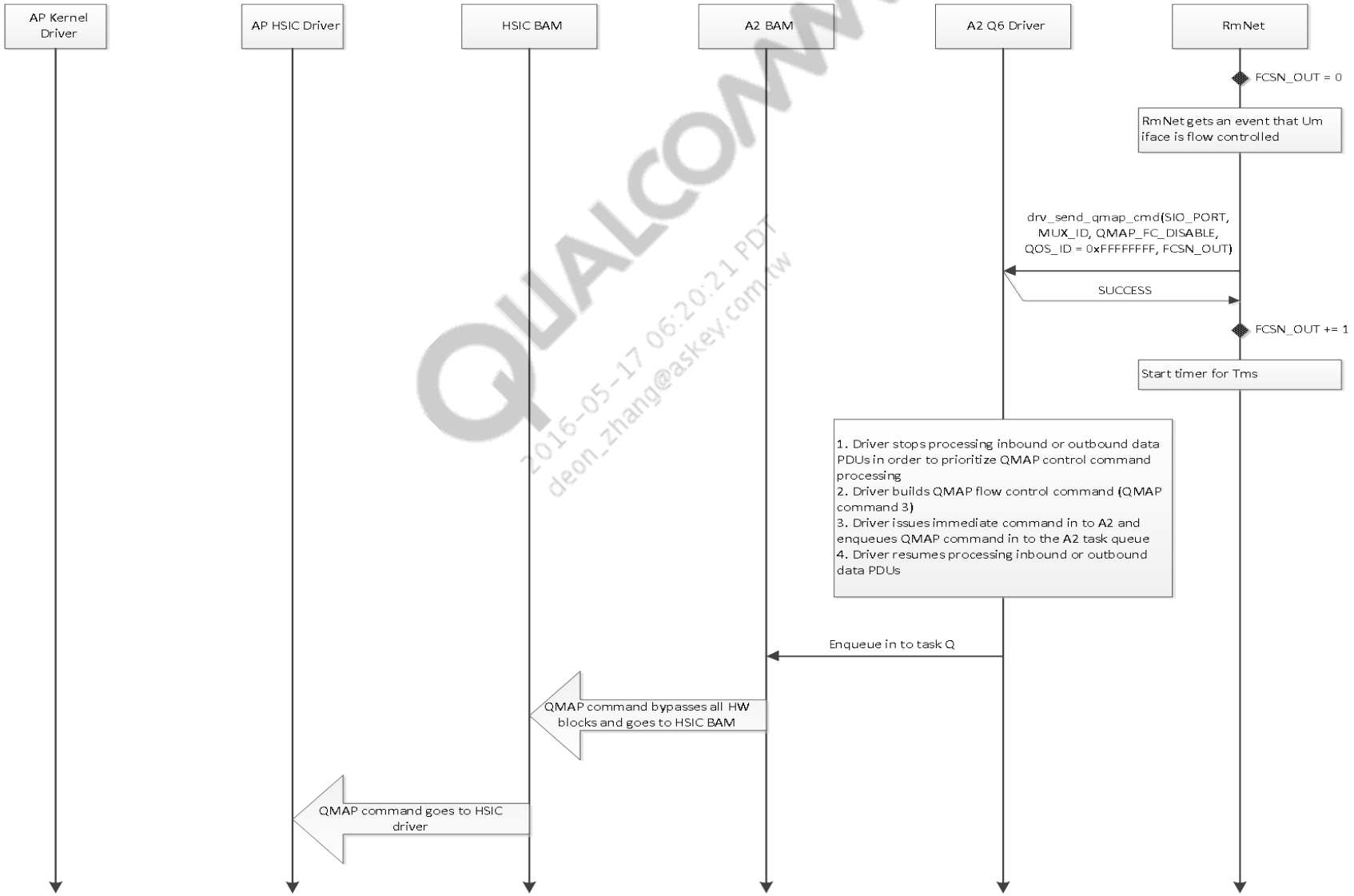
QMAP Flow Control Command – Retransmission Strategy

- Flow control commands are retransmitted by the sender until a response is received from the receiver.
  - The timer is NV-configurable and is set to 8 ms by default. It can be refined as needed.
    - Since flow enable has a latency upper bound of 18 ms, an 8 ms timer provides at least one retransmission for flow control commands corresponding to UL data traffic, before 18 ms elapses and adversely impacts throughput.
      - Refer to slide 13 to understand from where 18 ms is coming.

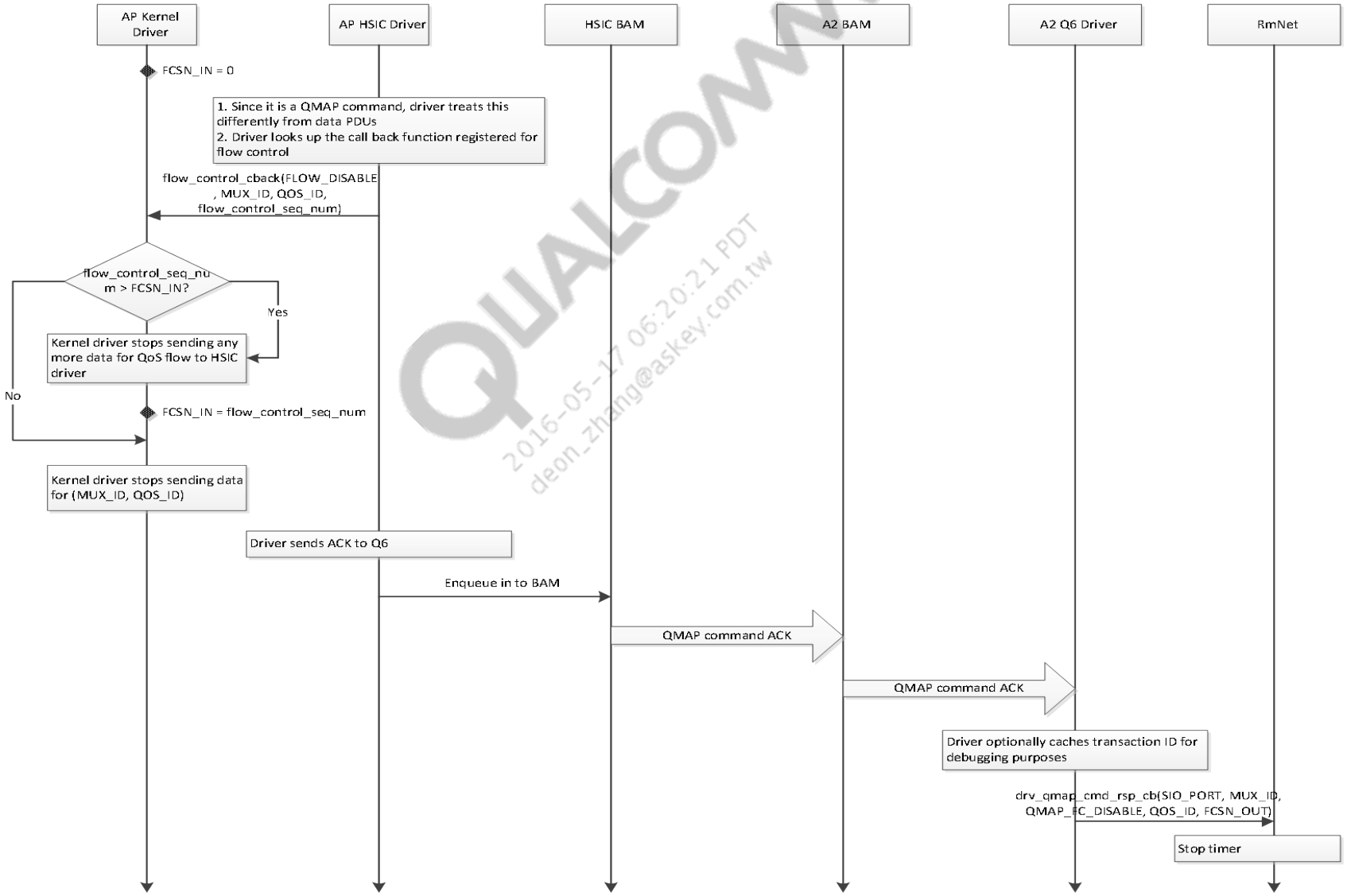
QMAP Flow Control Command – Implementation Guidelines

1. Both the sender and the receiver should prioritize flow control commands over other data.
   - For example, if there is a queue to buffer QMAP packets, a flow control command could be enqueued at the front so that it is dequeued first.
   - This ensures that a QMAP flow control command is not stuck behind a whole bunch of data (in either UL or DL) and reduces latency.
2. Receiver should send ACK for the flow disable command behind any data in flight for the disabled (MUX_ID, QoS_ID, IP_Family)
   - It means that flow control responses should not be prioritized
3. The receiver should not send any more data for the disabled (MUX_ID, QoS_ID, IP_Family) once ACK is sent.
4. The receiver should send ACK for the flow enable command before it enqueues new data packets for the enabled (MUX_ID, QoS_ID, IP_Family).
5. Following guidelines 2, 3, and 4 may enable the sender to estimate latency and dynamically tune WM levels

UL Flow Control



UL Flow Control (cont.)



QMI WDS Flow Control

- QoS NOT negotiated in QMI_CTL_SET_DATA_FORMAT or QMI_WDA_SET_DATA_FORMAT
- Register WDS flow control indication in QMI_WDS_SET_EVENT_REPORT

Type	0x1B	—	1	UL flow control indicator
Length	1	—	2	—
Value	→	report_uplink_flow_control	1	Values ▪ 0 – Do not report ▪ 1 – Report UL flow control change event

- WDS flow control indication is generated via QMI_WDS_EVENT_REPORT_IND

Type	0x27	—	1	UL flow control
Length	1	—	2	—
Value	→	uplink_flow_control	1	UL flow control status – Values ▪ 0 – Not flow controlled ▪ 1 – Flow controlled

- The indication is reported to clients bound to the same IP family of the data call

QMI QoS Flow Control

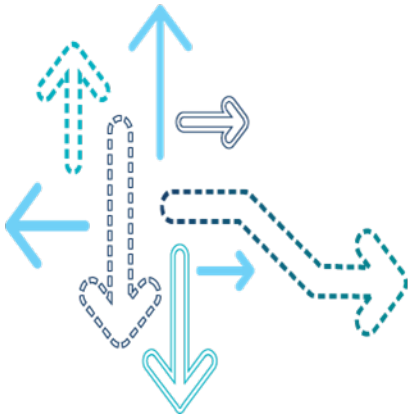
- QMI QoS negotiated through QMI_CTL_SET_DATA_FORMAT (TLV 0x01) or QMI_WDA_SET_DATA_FORMAT (TLV 0x10)
- Enable QoS event report – QMI_QOS_SET_EVENT_REPORT

Type	0x10	—	1	Global Flow Reporting
Length	1	—	2	—
Value	→	Global flow reporting	1	▪ 0x01 – Enable ▪ 0x00 – Disable

- QoS flow events are reported via MI_QOS_SET_EVENT_REPORT_IND

Type	0x10	—		QoS Flow state
Length	6	—	2	—
Value	→	QoS identifier	4	QoS identifier
		New_flow	1	▪ 1 – Newly added flow ▪ 0 – Existing flow
		Global flow reporting state change	1	This indicates that the flow that was added/modified/deleted: ▪ 0x01 – Flow activated ▪ 0x02 – Flow modified ▪ 0x03 – Flow deleted ▪ 0x04 – Flow suspended ▪ 0x05 – Flow enabled ▪ 0x06 – Flow disabled

QMAP Configuration



Negotiating QMAP

- The QMI_WDA_SET_DATA_FORMAT_REQ message allows a tethered device to negotiate QMAP per RmNet port.
  - It is part of the QMI WDA service; see [Q2].
- The message format is shown below.

Field	Field value	Parameter	Size (bytes)	Description
Type	0x12	—	1	UL data aggregation protocol
Length	4	—	2	—
Value	→	ul_data_aggregation_protocol	4	UL data aggregation protocol to be used for UL data transfer. Values:\n ▪ 0x00 – UL data aggregation is disabled (Default) ▪ 0x01 – UL TLP is enabled ▪ 0x02 – UL QC_NCM is enabled ▪ 0x03 – UL MBIM is enabled ▪ 0x04 – UL RNDIS is enabled ▪ 0x05 – UL QMAP is enabled
Type	0x13	—	1	DL data aggregation protocol
Length	4	—	2	—
Value	→	dl_data_aggregation_protocol	4	DL data aggregation protocol to be used for DL data transfer. Values:\n ▪ 0x00 – DL data aggregation is disabled (Default) ▪ 0x01 – DL TLP is enabled ▪ 0x02 – DL QC_NCM is enabled ▪ 0x03 – DL MBIM is enabled ▪ 0x04 – DL RNDIS is enabled ▪ 0x05 – DL QMAP is enabled

Negotiating QMAP (cont.)

- The QMI_WDA_SET_DATA_FORMAT_RESP message indicates to the tethered device if QMAP is successfully negotiated per the RmNet port.
  - It is part of the QMI WDA service; see [Q2].
- The message format is as shown below.

Field	Field value	Parameter	Size (bytes)	Description
Type	0x12		1	UL data aggregation protocol
Length	4		2	
Value	→	ul_data_aggregation_protocol	4	UL data aggregation protocol to be used for UL data transfer. Values:\n ▪ 0x00 – UL data aggregation is disabled (Default) ▪ 0x01 – UL TLP is enabled ▪ 0x02 – UL QC_NCM is enabled ▪ 0x03 – UL MBIM is enabled ▪ 0x04 – UL RNDIS is enabled ▪ 0x05 – UL QMAP is enabled
Type	0x13		1	DL data aggregation protocol
Length	4		2	
Value	→	dl_data_aggregation_protocol	4	DL data aggregation protocol to be used for DL data transfer. Values:\n ▪ 0x00 – DL data aggregation is disabled (Default) ▪ 0x01 – DL TLP is enabled ▪ 0x02 – DL QC_NCM is enabled ▪ 0x03 – DL MBIM is enabled ▪ 0x04 – DL RNDIS is enabled ▪ 0x05 – DL QMAP is enabled

Negotiating ‘QMAP Flow Control’

- A new QMI WDA message (0x2B – QMI_WDA_SET_QMAP_SETTINGS) is defined to support negotiation of QMAP flow control

Field	Field value	Parameter	Size (bytes)	Description
Type	0x10		1	QMAP in-band flow control
Length	1		2	
Value	→	in_band_flow_control	1	Configures the in-band flow control; values:\n ▪ 0 – Disables in-band flow control \n ▪ 1 – Enables in-band flow control

References

Ref.	Document	
Qualcomm Technologies		
Q1	*Application Note: Software Glossary for Customers*	CL93-V3077-1
Q2	*QMI WDA 1.12, QMI Wireless Data Administrative Service Spec*	80-VB816-26
Q3	*QMI QOS 1.5, QMI Quality of Service Spec*	80-VB816-7

Questions?

https://support.cdmatech.com