Qualcomm Atheros, Inc.

QCA9531/QCA9558/QCA9563.ILQ.2.0 CSU5

Release Notes

80-Y9202-1 Rev. B

November 27, 2015

Confidential and Proprietary – Qualcomm Atheros, 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 Atheros, Inc. or its affiliated companies without the express approval of Qualcomm 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 Atheros, Inc.

Qualcomm ChipCode is a product of Qualcomm Technologies, Inc. Qualcomm StreamBoost, Qualcomm VIVE, and XSPAN are products of Qualcomm Atheros, Inc. Other Qualcomm products referenced herein are products of Qualcomm Atheros, Inc. or Qualcomm Technologies, Inc. or its other subsidiaries.

Questions or comments: https://createpoint.qti.qualcomm.com/

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 Atheros, Inc.
1700 Technology Drive
San Jose, CA 95110
U.S.A.

Revision history

Revision	Date	Description
A	November 2015	Initial release
B	November 2015	Updated Section 1.2

Contents

Figures

Tables

1 Introduction

1.1 Purpose

This document provides details on the QCA9558/QCA9531/QCA9563.ILQ.2.0 CSU5 release.

Despite being downloaded from the Qualcomm ChipCode™ portal, the Qualcomm Atheros Support site, or embedded on Equipment received from Qualcomm Atheros, Inc. ("QCA"), the Qualcomm Atheros QCA9558/QCA9531/QCA9563.ILQ.2.0 CSU5 software release ("SW Package") shall be considered Deliverables and is subject to the terms and conditions of the Qualcomm Atheros, Inc. Limited Use Agreement ("Agreement"). The applicable Use Period, as that term is defined in the Agreement, for the SW Package starts on the Effective Date of your Agreement or the date you received the SW Package, whichever is later, and expires on November 19, 2016 (unless a different Use Period for the SW Package is specified in the Agreement, in which case the Use Period in the Agreement shall prevail). By receiving and/or using the SW Package, you acknowledge and agree that your use of the SW Package is subject to the terms and conditions of the signed Agreement. If you do not agree to the terms of the Agreement, have not signed such Agreement, or have not received the written approval from QCA set forth below, you shall immediately delete the SW Package from all storage media and destroy any and all copies made.

If you wish to utilize the SW Package commercially, you must obtain written approval from QCA, which may be obtained by contacting QCA by email at QCA.Contracts@qualcomm.com or via facsimile to (408) 516-5825, that the SW Package is approved for use under the terms of an applicable definitive agreement in effect between you and QCA or you must enter into a new definitive license agreement relating to the SW Package.

Information published by QCA regarding any third-party information does not constitute a license to use such information or endorsement thereof. QCA provides any such third party information as-is, without any representation, warranty, or indemnity, either express or implied. Use of such information may require a license from a third party under the intellectual property rights of such third party, or a license from QCA or its affiliates under the intellectual property rights of QCA or its affiliates. Users assume all risk of any use of such third party information.

1.2 Identification

This release is:	QCA9531.ILQ.2.0.r5, QCA9558.ILQ.2.0.r5 and QCA9563.ILQ.2.0.r5
The Linux Foundation hosted open source label that corresponds to this release (caf_TAG) is:	**AU_LINUX_QSDK_RELEASE_BANANA_10.2.4_TARGET_ALL.2.9.592**

Instructions for using this release are described in section 5.

1.3 Related documentation

In addition to this release notes, refer to these documents for more information on using this release.

Document Number	Title
80-Y0805-3	*QCA9558 802.11n 3x3 Dual Band Features WLAN SoC Data Sheet*
80-Y6999-3	*QCA9563 802.11n 3x3 2.4 GHz Premium SoC for WLAN Platforms Data Sheet*
80-Y7991-1	*QCA9531 v2.0 802.11n 2x2 2.4 GHz Premium SoC for WLAN Platforms Data Sheet*
80-Y0732-1	*QCA9980 Dual Band 3x3 MIMO 802.11ac/abgn WLAN SoC Data Sheet*
80-Y1927-1	*AP135-02X Hardware Reference Guide*
80-Y1926-2	*CUS223-523 Hardware Reference Guide*
80-Y7207-1	*AP 10.2 Programmer's Guide – Wireless LAN*
80-Y7207-3	*AP 10.2 Command Line Interface (CLI) User Guide*
80-Y0861-1	*ART2 Reference Design for AR93xx/94xx/95xx & QCA98xx Reference Guide*
80-Y0852-1	*QCA98xx ART2 Non-Volatile Memory Structure Reference Guide*
80-Y7186-3	*Switch Software Development Kit Diagnostic Shell User Guide*

2 Features

2.1 Features

This release is made up of the following components supporting the QCA9558 SoC with XSPAN™ 802.11n technology and MIPS 74Kc processor, and QCA9880 with Qualcomm® VIVE™ 802.11ac technology:

- The QSDK package that provides basic open source Linux support, cross-compilers, boot loaders, file system images, and script files.
- Qualcomm Atheros Wi-Fi software driver, see Table 2-1 and Table 2-2.
- QCA9880 10.2.4 firmware source code (only for specific customers).
- WAPI source package for software requiring a third party license (only for specific customers).
- WLAN driver is Wi-Fi 802.11ac certification ready.

This release is tested with these platforms going through CST:

- AP135 (QCA9558 3x3) + CUS223 (QCA9880 3x3)

NOTE: For details on the features listed, see the Programmers Guide (xxx)

Table 2-1 Features unit-tested for QCA9531/QCA9558/QCA9563 2.4 GHz radios

Category	Description
Basic HT (802.11n) support	▪ 2.4 GHz ▪ AP mode (beaconing, association, disassociate, deauth) ▪ AP legacy power save ▪ 20/40 MHz channel width, 3 spatial streams, MCS 0–23 (basic and supported rate set) ▪ A-MPDU ▪ A-MSDU Rx ▪ Block ACK mechanism ▪ SGI ▪ Tx/Rx STBC, LDPC
Encryption/ Decryption	▪ WEP, TKIP, AES
Security modes	▪ Open ▪ Shared key, WPA/WPA2, WPA-PSK/WPA2-PSK ▪ Extended security (EAP methods)

Category	Description
Modes of operation	▪ AP mode ▪ Extender AP mode (3-address) ▪ WDS (Root AP and Repeater AP) (4-address) ▪ Repeater independent mode ▪ Enhanced independent repeater mode
Retail features	▪ mBSSID (16 VAPs) ▪ Aggregation limits, WMM, WDS, repeater, Rx defrag, ACS, multicast enhancements, WPS/WPS2.0, extender AP, ANI, DCS, ACS, QWRAP, Band Steering, Airtime fairness **NOTE:** The ATF feature works only with the Premium profile. For profile information see section 5.4.
Miscellaneous features	▪ ANI, RIFS
CoreBSP	▪ QCA955x and QCA956x SoC BSP patches for 3.3.8 Linux kernel ▪ SoC GMAC driver with Ethernet switch driver ▪ SPI controller driver for NOR flash
Tools	▪ Basic CFG commands, pktlog, basic athstats, regdump

Table 2-2 Features unit-tested for QCA9880 5 GHz radios

Category	Description
Basic 802.11a support	▪ 5 GHz ▪ AP mode (beaconing, association, disassociate, deauth) ▪ AP legacy power save ▪ STA modes including scanning ▪ OFDM rates (6−54 Mbps) (basic and supported rate set) ▪ RTS/CTS ▪ DFS
Basic HT (802.11n) support	▪ 5 GHz ▪ AP mode (beaconing, association, disassociate, deauth) ▪ AP legacy power save ▪ STA modes including scanning ▪ 20/40 MHz channel width ▪ 3 Spatial stream ▪ MCS 0−23 (basic and supported rate set) ▪ A-MPDU, A-MSDU Rx ▪ Block ACK mechanism ▪ SGI, TX/RX STBC, LDPC, DFS

Category	Description
Basic VHT (802.11ac) support	▪ 5 GHz ▪ AP mode (beaconing, association, disassociate, deauth) ▪ AP legacy power save ▪ STA modes including scanning ▪ 20/40/80 MHz channel width ▪ MCS 0−9 (basic and supported rate set) ▪ 3 spatial streams ▪ VHT A-MPDU delimiter for Rx/Tx for single MPDU ▪ Rx/Tx A-MPDU, A-MSDU RX ▪ CCA on secondary ▪ Short guard interval ▪ LDPC, Tx/Rx STBC ▪ CTS with BW signaling in response to RTS with BW signaling (dynamic bandwidth switching) ▪ UAPSD, DFS
Encryption/Decryption	WEP, TKIP, AES
Security modes	▪ Open ▪ Shared Key, WPA/WPA2, WPA-PSK/WPA2-PSK ▪ Extended security (EAP methods) ▪ WAPI (AP only)
Retail feature set	▪ MBSSID ▪ Aggregation limits ▪ WMM ▪ WDS ▪ Repeater ▪ Rx defrag ▪ ACS, DCS ▪ Multicast enhancements ▪ WPS/WPS2.0 ▪ 200 clients support ▪ Airtime fairness, Band steering
Basic interop	Interop with various 802.11ac APs put in STA mode and various 802.11n STA cards available in the market today
Miscellaneous features	Management over HTT
Tools	Basic CFG commands, PKTLOG, Basic athstats, Regdump, Firmware debug log

2.2 Changes for this release

Table 2-3 Closed issues for the CSU5 release

S. No.	Description
1.	Fix for crash at ath_calibrate (Merged)
2.	In ATF, adding Print for invalid combinations of strict/fair within/across groups (Merged)
3.	Reverting the copyright changes made by vimrc script (Merged)
4.	Peregrine on GRX350 stops beacon transmission under full system (Merged)
5.	Restrict the command string (get_staDFSEnable) to 15 (Merged)
6.	Added changes to block/unblock a node (Merged)

S. No.	Description
7.	AP135, saw 300M at iwconfig ath0 list if set the AP as mode HT20 (Merged)
8.	endianess conversion is added for HTT and HTC header and removed for htt_mgmt_t (Merged)
9.	Kernel crash detected due to NULL descriptor (Merged)
10.	Adding a missing spin unlock in data path (Merged)
11.	ATF Inter-group policy not working without clients connected (Merged)
12.	iwpriv command to enable/disable tr069 params. (Merged)
13.	ATF table not displaying complete SSID (Merged)
14.	Memory leak in monitor mode.
15.	Kernel Panic observed in WDS Repeater AP while running
16.	Adding 'primary_vap' option to acfg vap structure
17.	Enabling bursted beacons by default
18.	Neighbor solicitation packets re-broadcasted by APUT
19.	Add extended capabilities IE whenever it is
20.	Improper Noise Floor values in spectral raw data
21.	Change the Bootp flags in DHCP payload to BROADCAST if it is set to UNICAST
22.	In quick stakickOut WDS Repeater mode in 2.4Ghz failed to kick out the station from the AP
23.	Target asserted while running one of HS2.0 sigma-dut test
24.	ATF: atfmaxclient is supported only on DA
25.	ATF feature failure due to regression (Merged)
26.	Weather radar logic for channel selection (Merged)
27.	Multicast enhancement change (Merged)
28.	Target assert in _rcRateFind_2D 1) (Merged)
29.	Vaps does not wait for cac_timeout after channel switch in MBSSID case (Merged)
30.	Low throughput at 20MHz SGI N-MCS7 & MCS15 Clients are operating at Standard Guard Interval (LGI) only
31.	Ignore error value returned for STA vap stats initialization (Merged)
32.	Deauth not seen in 2.4 GHz (Merged)
33.	Scheduled work for periodic scan instead of doing in timer (Merged)
34.	Fair-queue scheduling across SSID groups (Merged)
35.	In multivap scenerio with proxyarp enabled, (ni->ni_vap ! vap) check missing (Merged)
36.	Iwpriv mcast_rate panic (Merged)
37.	Fixing an issue where VAPs do not come up while scan in progress.(Merged)
38.	Avoid Dead loop (recursively calling dev_xmit_queue) when proxyarp enabled ... (Merged)
39.	Multicast enhancement support for MLD (Merged)
40.	Multicast enhance support for MLD (Merged)
41.	AVM timeout fixes 1. Removing WDS entries on node table reset (Merged)
42.	Fix NSS not reflecting in beacon. (Merged)
43.	Not setting the correct status in CE_recv_buf_enqueue. This results in ... (Merged)
44.	Fix for Klockwork issues in athtestcmd.c and ol_if_rtt.c files (Merged)
45.	In the case (as we are not able to convert multicast to unicast), instead of send it multicast instead of dropping it (Merged)
46.	dev_put mismatch in ald_nl_receive can lead to mismatch Fixed IR: IR-079366 ... (Merged)

S. No.	Description
47.	Checking return value of iv_stopping function and taking action. Sometimes ... (Merged)
48.	Checking return value of wlan_mlme_start_bss in osif_acs_start_bss and ... (Merged)
49.	IR106754: Peregrine on GRX350 stops beacon transmission under full system ... (Merged)
50.	Fix for Radartool command throws error wifi0: Bad address.This was caused ... (Merged)
51.	Manually propagating the change from 10.2 mainline to expose airtime ... (Merged)
52.	Revert "expose Air Time Fairness types to user space" (Merged)
53.	Kernel panic due to NULL pointer dereference during node cleanup (Merged)
54.	Low 2.4 GHz throughput issue with auto channel selection (ACS) (Merged)
55.	Watchdog rebooting issue. (Merged)
56.	Found IOT issue between MTK smart phone and LeTV router. Root ... (Merged)
57.	This reverts commit f3b611ba6e4d48caf58e0059bf84f1b0ef912abc(fix Nss for ... (Merged)
58.	Modification to load atf fw from file and also support for ... (Merged)
59.	Kernel Panic "iwpriv ath1 get_signal_dbm" for 5G peregrine radio (Merged)
60.	Klockwork issues (P1) in multiple files. (Merged)
61.	The fix is with reference to the issue QCA9882 skb_over_panic with pmf when ... (Merged)
62.	Fixed WNM notification bit not set in the beacon frame in Hotspot ... (Merged)
63.	HS2-R2 QOS map set element is not set in Association Response. (Merged)
64.	Fix for the issue different minimum txpower in 2.4 GHz and 5 GHz. (Merged)
65.	Keycache entry reset failure at powerup. (Merged)
66.	Fix NSS not reflecting in beacon. (Merged)
67.	Fix for abnormal DFS behavior when radar is found during the cac time. (Merged)
68.	Fix for the issue - channel load value more than 100% (Merged)
69.	Reduce the reset duration at the end of scan. (Merged)
70.	Doing mesgq and scan handling in workqueue. (Merged)
71.	Tune fair-queue algorithm parameters (Merged)
72.	Sending WMI commands to the firmware when the grouping policy is set as well ... (Merged)
73.	Change in the wifitool_mac_aton function. Integrating the fix from 10.4 to ... (Merged)
74.	Support for Dynamic ATF enable/disable on 5G (Merged)
75.	Klockwork issues in radartool.c file (Merged)
76.	Fix for Klockwork issues in multiple files. Integrating the fixes from 10.2 ... (Merged)
77.	Integrating the changes related to the limitation on loading FW binary with ... (Merged)
78.	Applying a patch which adds protection while scanning by using new flags for ... (Merged)
79.	Target assert observed in WDS Root AP
80.	Dynamic ATF disable/enable feature
81.	ATF SSID Grouping feature
82.	ATF UDP feature
83.	Add support for peer mcast2ucast drop empty.
84.	TR 69 throughput regression fix
85.	IGMPV3 mcast2ucast conversion fix
86.	Target assert fix with SA enabled
87.	Fix for Peregrine synchronization issue
88.	Fix for target assert observed in WDS STA AP

S. No.	Description
89.	Refuse RX BA request if reorder state buffer is not available
90.	HNAT support in QSDK. Verified on AP136, not tested together with Wi-Fi. This feature is released as engineering drop (ED) quality.

3 Supported Hardware

3.1 AP135 reference design

Table 3-1 AP135 reference design

Hardware Board	Supported	Comments
AP135-xxx	Yes	Includes CUS223-xxx 802.11ac module and AR8327 GigE chip

3.2 CUS223: QCA9880, 5 GHz 802.11ac-based reference design

Table 3-2 CUS223: QCA9880, 5GHz 802.11ac-based reference design

Hardware Board	Supported	Comments
CUS223-523	Yes	3x3 configuration, high power

3.3 AP147: QCA9531 reference design

Table 3-3 AP147: QCA9531-based reference design

Hardware Board	Supported	Comments
AP147-031	Yes	–
XB143-010	Yes	–

3.4 AP147 configuration

Table 3-4 Basic AP147 configuration details

CPU	650 MHz
DDR	600 MHz
AHB	216 MHz
DDR	16-bit DDR2
WLAN	QCA9531 2x2 in 2.4 GHz
SWITCH	Integrated 10/100 switch
Boot	Boot from NOR flash
USB	1x USB 2.0 in host mode

3.5 AP152: QCA9563 reference design

Table 3-5 AP152: QCA9563-based reference design

Hardware Board	Supported	Comments
AP152-030	Yes	–

3.6 AP152 configuration

Table 3-6 Basic AP1 configuration details

CPU	775 MHz
DDR	650 MHz
AHB	258
DDR	16-bits DDR2
WLAN	QCA9563 3x3 in 2.4GHz
Switch	SGMII, QCA8337N
Boot	Boot from NOR flash
USB	2xUSB 2.0 in host mode
PCIE	PCIE 1.1

3.7 CUS223: QCA9880, 5 GHz 802.11ac-based reference design

Table 3-7 CUS223: QCA9880, 5 GHz 802.11ac-based reference design

Hardware Board	Supported	Comments
CUS223-523	Yes	3x3 configuration, high power

3.8 Supported features

- QSDK 2.0 support on MIPS platform (QCA9558)
- Dual band dual concurrent (QCA9558 + QCA9880)
- Routing performance: 850-950 Mbps (optimized SNAT)
- Support for multiple PPPoE sessions
- IGMP snooping (AR8327)
- Qualcomm® StreamBoost™ premium

4 Performance Data and Known Issues

4.1 Performance data

4.1.1 Cabled shield box throughput

Setup Details

Cabled Setup, Channel: 149 (5 GHz), Build Used: NHSS.ILQ.2.9.r5-00520-P-1 (unified NART – 4.9.852)

Root AP: AP135-023-Q2715 (N10JBYJJ9) + CUS223-523-Q8226 (N10J3D722 (4.9.852) 3X3

Client AP: AP135-023-Q2715 (N10JBYLPX) + CUS223-523-Q8226 (N10J3D9GD) (4.9.852) 3X3

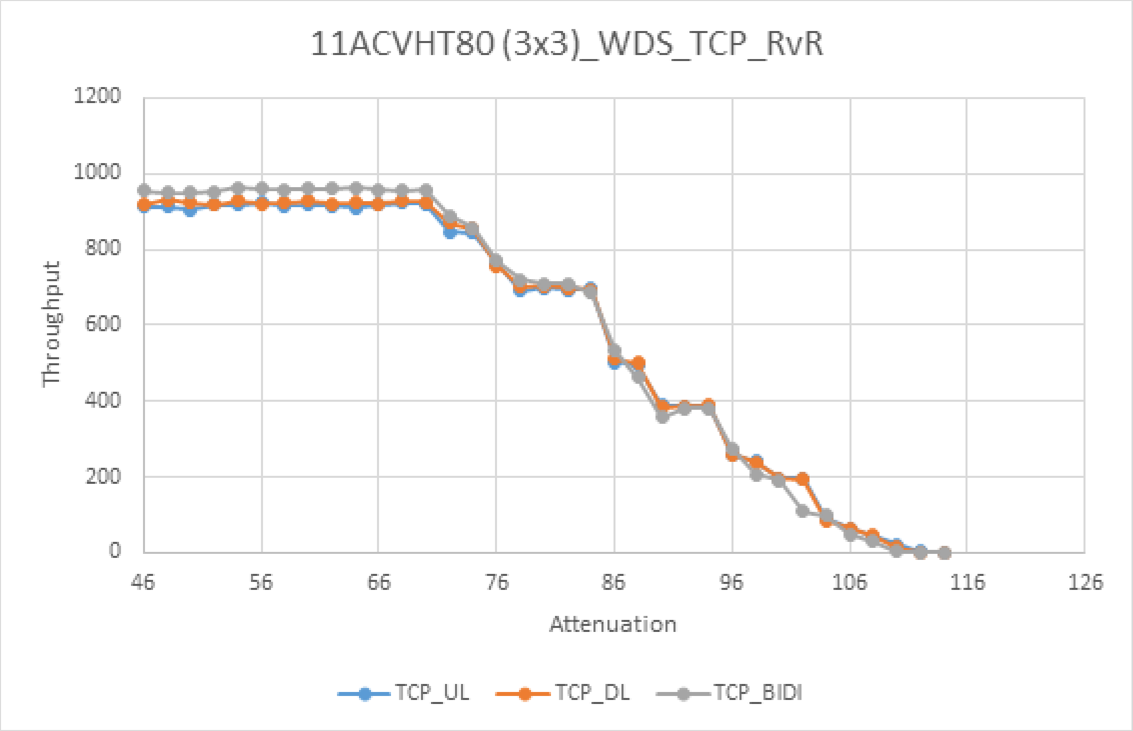

Figure 4-1 WDS 802.11ACVHT80 throughput: TCP - rate (in Mbps) vs. range with CUS223-523 (3x3)

Setup Details

Cabled Setup, Channel: 6 (2 GHz),

Build Used: NHSS.ILQ.2.9.r5-00520-P-1 (unified NART – 4.9.852)

Root AP: AP135-023-Q2715 (N10JBYJJ9) + CUS223-523-Q8226(N10J3D722 (4.9.852) 3X3

Client AP: AP135-023-Q2715 (N10JBYLPX) + CUS223-523-Q8226(N10J3D9GD) (4.9.852) 3X3

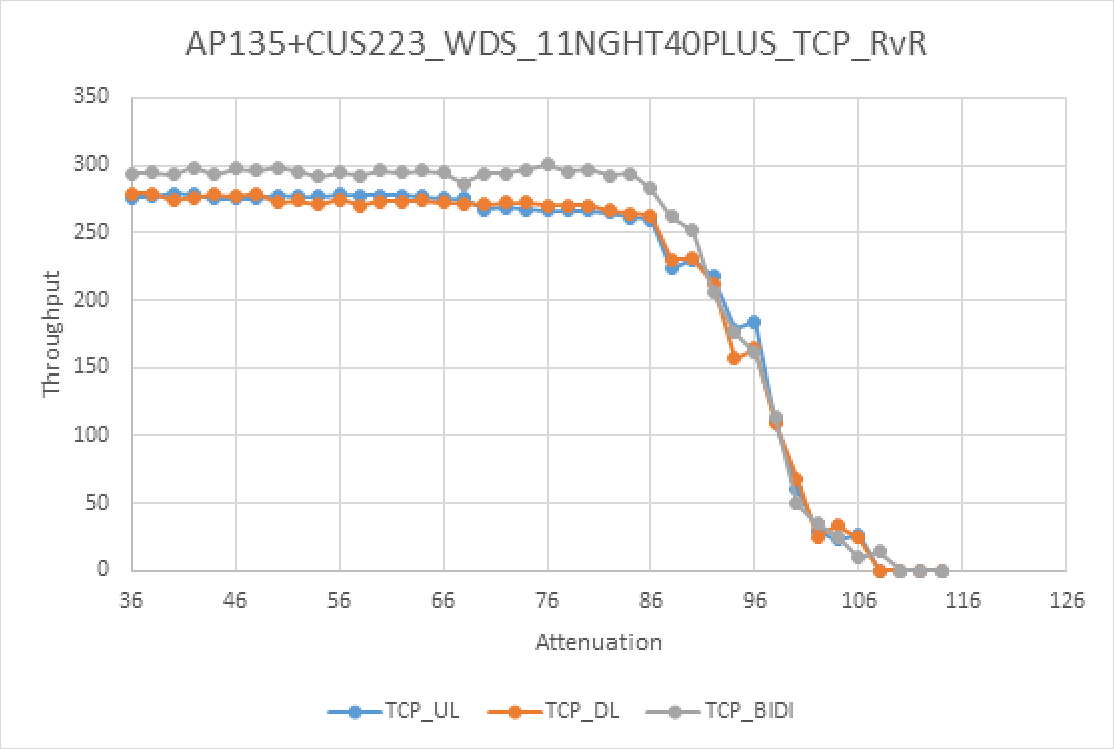

Figure 4-2 WDS 802.11NGHT40 throughput: TCP- rate (in Mbps) vs. range with CUS223-523 (3x3)

4.2 CST Test Results

Table 4-1 CST Test Results

Name	# of Test Cases Planned to Execute	Coverage %	(#)	Fail %	(#)	Pass %	(#)	Block %	(#)	Not Run %	(#)
(F) AP135+CUS223	1,319	100%	1,319	0.53%	7	99.47%	1,312	0%	0	0%	0
(F) WLAN_Functionality	54	100%	54	0%	0	100%	54	0%	0	0%	0
(R) ATF	49	100%	49	0%	0	100%	49	0%	0	0%	0
(R) BS	5	100%	5	0%	0	100%	5	0%	0	0%	0
(F) WLAN_Performance	356	100%	352	0%	0	98%	352	0%	0	0%	0
(R) KPIs	30	100%	30	0%	0	100%	30	0%	0	0%	0
(R) ATF Performance	314	100%	314	0%	0	100%	314	0%	0	0%	0
(R) SGI/LGI Comparison	8	100%	8	0%	0	100%	8	0%	0	0%	0
(F) WLAN_Functionality_Regression	159	100%	159	0%	0	100%	159	0%	0	0%	0
(R) WLAN_Functionality_Regression	159	100%	159	0%	0	100%	159	0%	0	0%	0
(F) WLAN_Certification	27	100%	27	0%	0	100%	27	0%	0	0%	0
(R) Passpoint Rel 2	27	100%	27	0%	0	100%	27	0%	0	0%	0
(F) WLAN Charter Tests	30	100%	30	23%	7	77%	23	0%	0	0%	0
(R) Charter Regression	30	100%	30	23%	7	77%	23	0%	0	0%	0
(F) WLAN_Regulatory	674	100%	674	0%	0	100%	674	0%	0	0%	0
(R) DFS	271	100%	271	0%	0	100%	271	0%	0	0%	0
(R) Regulatory	403	100%	403	0%	0	100%	403	0%	0	0%	0
(F) Competitor Comparison	23	100%	23	0%	0	100%	23	0%	0	0%	0
(R) Competitor Data	23	100%	23	0%	0	100%	23	0%	0	0%	0
(F) AP147 (HB + Peregrine)	68	100%	68	0%	0	100%	68	0%	0	0%	0
(R) KPIs	18	100%	18	0%	0	100%	18	0%	0	0%	0
(R) WLAN Functional	50	100%	50	0%	0	100%	50	0%	0	0%	0
(F) AP152 (DF + Peregrine)	68	100%	68	0%	0	100%	68	0%	0	0%	0

Name	# of Test Cases Planned to Execute	Coverage %	(#)	Fail %	(#)	Pass %	(#)	Block %	(#)	Not Run %	(#)
(R) KPIs	18	100%	18	0%	0	100%	18	0%	0	0%	0
(R) WLAN Functional	50	100%	50	0%	0	100%	50	0%	0	0%	0
Overall	1,455	100.00%	1455	0.48%	7	99.52%	1448	0.00%	0	0.00%	0

4.3 Performance KPI

MODE	Throughput	CPU
WDS_11ACVHT80_3x3_TCP_DL	929	78%
WDS_11ACVHT80_3x3_TCP_UL	928	93%
WDS_11ACVHT80_3x3_TCP_BIDI	944	78%
WDS_11ACVHT80_3x3_UDP_DL	951	84%
WDS_11ACVHT80_3x3_UDP_UL	950	88%
WDS_11ACVHT80_3x3_UDP_BIDI	1001	86%
WDS_11NGHT40PLUS_3x3_TCP_DL	277	59%
WDS_11NGHT40PLUS_3x3_TCP_UL	277	54%
WDS_11NGHT40PLUS_3x3_TCP_BIDI	306	57%
WDS_11NGHT40PLUS_3x3_UDP_DL	338	78%
WDS_11NGHT40PLUS_3x3_UDP_UL	337	50%
WDS_11NGHT40PLUS_3x3_UDP_BIDI	323	61%
DBDC_11ACVHT80+11NGHT40PLUS_TCP_DL	1020	99%
DBDC_11ACVHT80+11NGHT40PLUS_TCP_UL	949	100%
DBDC_11ACVHT80+11NGHT40PLUS_TCP_BIDI	1047	99%
DBDC_11ACVHT80+11NGHT40PLUS_UDP_DL	1061	100%
DBDC_11ACVHT80+11NGHT40PLUS_UDP_UL	1036	100%
DBDC_11ACVHT80+11NGHT40PLUS_UDP_BIDI	1046	100%

4.4 Known Issues

Table 4-2 shows the open issues for this release.

Table 4-2 Open Issues in this release

No.	Issue	Description
1	Aggregate throughput degradation seen on 11ACVHT80 when ATF allocation for two clients is something other than 50:50 for TCP	This is as expected. This is also in line with the previous CS numbers with ATF. When there is variance in allocation %, then during that interval TCP protocol tries to scale its stream up for one client and stream down for the other client. This causes transactional loss and drop in aggregated throughput
2	UDP throughput	Issue is observed when the test is run with Veriwave LAN port connected directly to the WAN port. Issue is related to Ethernet switch driver or Linux networking.

5 Download and Build Instructions

Released software will be distributed via ChipCode for proprietary based code, and through the Linux Foundation for open source.

5.1 Download packages available through ChipCode

Qualcomm Atheros proprietary code is available from ChipCode.

NOTE: A web/GUI interface and a secure git server both allow access to this code. Browse available packages and obtain the download URL at https://chipcode.qti.qualcomm.com/ (see https://chipcode.qti.qualcomm.com/helpki/cloning-code-from-a-repository for more information).

See https://chipcode.qti.qualcomm.com/helpki for more information on installation and configuration of the correct version of git and OpenSSL on both Windows and Linux platforms that is required to support the authentication methods used by ChipCode.

5.1.1 Packages downloaded from external websites

These components are downloaded by QSDK as needed while building the various profile configurations. QSDK may be further customized to download additional components; this list only contains the components that are necessary for at least one of the QSDK 2.0 default profiles.

Most packages listed in are downloaded from external web sites, but several Qualcomm Atheros proprietary packages must be downloaded from a private access customer support account.

Packages
alljoyn-14.02.00-src.tar.gz
angular-0.1-gffab5c4.tar.gz
angular-loadcontent-g7a480b2.tar.gz
angular-mocks-0.1-gdeea515.tar.gz
angular-ozker-gc089f69.tar.gz
angular-route-0.1-ge548b5b.tar.gz
angular-translate-0.1-g5969f29.tar.gz
aperture-g9841cd0.tar.gz
app-flow-scripts-gf69f16b.tar.gz
LinuxART2CS10.2v4.9.592.tar.bz2
backgroundsize.min.htc
bridge-utils-1.5.tar.gz
busybox-1.19.4.tar.bz2
callhomeclient-gb313c8d.tar.gz
cape-g6c1ec51.tar.gz
celes-gefd708c.tar.gz
curl-7.29.0.tar.bz2
v3.1.6.tar.gz
dnsmasq-2.62.tar.gz
dosfstools-3.0.12.tar.gz
drflocs-gcda5d50.tar.gz
dropbear-2011.54.tar.bz2
e2fsprogs-1.42.4.tar.gz
ethtool-3.4.1.tar.xz
fast-classifier-g0cfe91b.tar.gz

Packages
fcgi-2.4.0.tar.gz
flowman-ga7f9daf.tar.gz
flowmark-gc491474.tar.gz
flux-g0ec2341.tar.gz
v0.11.0.tar.gz
hotplug2-201.tar.gz
igmpproxy-0.1.tar.gz
iperf-2.0.5.tar.gz
iproute2-3.3.0.tar.bz2
iputils-s20101006.tar.bz2
jansson-2.4.tar.gz
jigglyp0f-gf707a60.tar.gz
jquery-1.7.2.min.js
jquery-contextmenu-1.01-gecb2ce1.tar.gz
flot-0.8.0.zip
jquery-flot-axislabels-0.1-ga0d11e5.tar.gz
jquery-flot-gant-0.1-g90ec5b8.tar.gz
2.0.0beta10.tar.gz
jquery.sparkline.min.js
jquery-swapsies.js
jquery-ui-1.8.21-gde5bb86.tar.gz
libdaemon-0.14.tar.gz
libevent-2.0.19-stable.tar.gz
libhyficommon-master.144.tar.bz2

libieee1905-master.144.tar.bz2
json-c-0.9.tar.gz
libnetfilter_conntrack-0.9.1.tar.bz2
libnfnetlink-1.0.0.tar.bz2
libnl-bf-g7d47666eb3c414feb8901970d35b96461214c2bf.tar.gz
libpcap-1.1.1.tar.gz
readline-5.2.tar.gz
libsbdaemon-g83a34bc.tar.gz
libstorage-master.165.tar.bz2
libubox-2013-07-04-11e8afea0f7eb34f8c23a8e589ee659c46f3f8aa.tar.gz
libwpa2-master.144.tar.bz2
libxml2-2.7.8.tar.gz
logrotate_3.8.1.orig.tar.gz
lua-5.1.4.tar.gz
luci-app-streamboost-gd6f04f2.tar.gz
luci-theme-steelblue-g437387d.tar.gz
macoui-db-20140508110915-g1532ba2-generic.tar.gz
miniupnpd-1.8.20130426.tar.gz
monit-5.4.tar.gz
netifd-2013-05-13-bc4a4bb127622c76085ecec7fd20448aad7bafaf.tar.gz
nodedetectd-gbfefe1a.tar.gz
ntfs-3g_ntfsprogs-2011.4.12.tgz
openssl-1.0.1e.tar.gz
opkg-618.tar.gz
overlord-g303dac8.tar.gz
ozker-g287b18e.tar.gz
p0f-3.06b.tgz
p0f-db-20140626230340-g-generic.tar.gz
policy-redis-20140710141214-g725dc2f-generic.tar.gz
popt-1.7.tar.gz
qca-hostap-10.2.2.40.tar.bz2
qca-hyctl-master.155.tar.bz2
qca-hyd-master.172.tar.bz2
qca-hyfi-bridge-g700fc2b.tar.gz
qca-hyfi-iptv-helper-master.144.tar.bz2
qca-hyfi-qdisc-g2c15fe4.tar.gz
qca-libhyfi-bridge-master.144.tar.bz2
qca-plc-fw-master.126.tar.bz2
qca-plc-serv-master.151.tar.bz2
qca-spectral-10.2.2.40.tar.bz2
qca-ssdk-gf6fe95d.tar.gz
qca-ssdk-shell-gb847ddb.tar.gz
qca-vhyfid-master.177.tar.bz2
qca-wifi-10.2.2.40.tar.bz2
qca-wsplcd-master.163.tar.bz2
qdiscman-g6738a31.tar.gz
quagga-0.99.21.tar.gz
radvd-1.9.1.tar.gz

v2.1.0.tar.gz
redis-2.6.13.tar.gz
rp-pppoe-3.10.tar.gz
sbmacouid-gca9ff30.tar.gz
sbnamed-ge17d28c.tar.gz
sbnoded-gc26d169.tar.gz
sbsaved-ga651af7.tar.gz
sea-gb0cb51f.tar.gz
shortcut-fe-g0cfe91b.tar.gz
sigma-dut-2013-04-13-c0c4709b11f19478a411909aec1e05279294e9c8.tar.gz
sysstat-10.1.7.tar.bz2
tftp-hpa-0.48.tar.gz
tresbar-gdea08b4.tar.gz
u-boot-2012.04.01.tar.bz2
ubus-2013-01-13-bf566871bd6a633e4504c60c6fc55b2a97305a50.tar.gz
uci-2013-01-04.1.tar.gz
uClibc++-0.2.4.tar.bz2
uhttpd-2012-10-30-e57bf6d8bfa465a50eea2c30269acdfe751a46fd.tar.gz
urijs-0.1-g2bdf950.tar.gz
util-linux-2.21.2.tar.xz
wide-dhcpv6-20080615.tar.gz
wireless_tools.29.tar.gz
wopr-db-20140619165349-gba7dcd5-generic.tar.gz
xtables-addons-1.42.tar.xz
zlib-1.2.7.tar.bz2

5.2 Set up the build environment

This framework has been developed using Ubuntu (from version 10.04 to version 13.04), and Debian. However, the QSDK framework regenerates the critical tools required to compile the firmware, at build-time. In that sense, the framework is independent from the host environment. Though the framework is developed using the above distributions, it is expected to work fine on other distributions such as RedHat, Mint, or Fedora.

The command for Debian/Ubuntu is (needs to be customized for other distributions):

```
$ sudo apt-get install gcc g++ binutils patch bzip2 flex make gettext \
  pkg-config unzip zlib1g-dev libc6-dev subversion libncurses5-dev gawk \
  sharutils curl libxml-parser-perl
```

As the framework automatically downloads the open source components specified in section 5.1, make sure that an internet connection is active on the build host while creating the build.

Install repo and git

Repo is used to download and update packages from multiple repositories by reading an XML manifest; the repo script must be available somewhere in the path. Newer versions of Ubuntu create folder **~/bin**, add that folder to the PATH variable, and place repo there, while some older versions do not. It is recommended to create the **~/bin** folder manually, add it to PATH, and place repo there.

NOTE: ChipCode requires git version 1.8.1.2 or higher configured for openssl. Ubuntu 12.04 installs git 1.7.8 configured for gnuTLS. The recommended solution is to download the latest git version from github.com then configure, compile, and install it manually.

5.3 Download the software

The first step is to generate the QSDK framework by re-assembling the code from ChipCode and The Linux Foundation. The example given in this section assumes that all packages listed in section 5.1 are downloaded and placed in the top-level directory:

Copy this package from ChipCode and download to the top-level directory:	Download from ChipCode at:
qsdk-qca-wifi-2.9.592.tar.bz2	apss_proc\out\proprietary\Wifi
qsdk-qca-wlan-2.9.592.tar.bz2	apss_proc\out\proprietary\Wifi
qsdk-qca-wifi-fwbin-2.9.592.tar.bz2	apss_proc\out\proprietary\Wifi
qsdk-qca-wifi-fwcommon-2.9.592.tar.bz2	apss_proc\out\proprietary\Wifi
qsdk-whc-2.9.592.tar.bz2	apss_proc\out\proprietary\Wifi
qsdk-qca-hostap-2.9.592.tar.bz2	apss_proc\out\proprietary\Wifi
qsdk-qca-wififw_lteu-2.9.592.tar.bz2	apss_proc\out\proprietary\Wifi
qsdk-qca-wififw_max-2.9.592.tar.bz2	apss_proc\out\proprietary\Wifi
qsdk-qca-wififw_atf-2.9.592.tar.bz2	apss_proc\out\proprietary\Wifi
qsdk-qca-wififw-AR9888-2.9.592.tar.bz2	apss_proc\out\proprietary\Wifi
qsdk-qca-wififw-AR9887-2.9.592.tar.bz2	apss_proc\out\proprietary\Wifi
qsdk-qca-shortcut-fe-2.9.592.tar.bz2	apss_proc\out\proprietary\Shortcut-fe

- For StreamBoost customers, these packages are downloaded from ChipCode and copied to the top-level directory:

Copy this package from ChipCode and download to the top-level directory:	Download from ChipCode at:
streamboost-ar71xx-2.9.592-qualcomm-ap135.tar.bz2	apss_proc\out\proprietary\Streamboost-qualcomm-ap135

```
$ repo init -u git://codeaurora.org/quic/qsdk/releases/manifest/qstak -b
   release -m
   caf_AU_LINUX_QSDK_RELEASE_BANANA_10.2.4_TARGET_ALL.2.9.592.xml
$ repo sync
$ mkdir -p qsdk/dl
$ tar xjvf qsdk-qca-wifi-2.9.592.tar.bz2 -C qsdk
$ tar xjvf qsdk-qca-wlan-2.9.592.tar.bz2 -C qsdk
$ tar xjvf qsdk-whc-2.9.592.tar.bz2 -C qsdk
$ tar xjvf qsdk-qca-hostap-2.9.592.tar.bz2 -C qsdk
$ tar xjvf qsdk-qca-wifi-fwbin-2.9.592.tar.bz2 -C qsdk/qca/src/qca-wifi/
$ tar xjvf qsdk-qca-wifi-fwcommon-2.9.592.tar.bz2-C qsdk/qca/src/qca-wifi/
$ mv qsdk/qca/src/qca-wifi/fwcommon qsdk/qca/src/qca-wifi/fwincludes
$ tar xjvf qsdk-qca-shortcut-fe-2.9.592.tar.bz2 -C qsdk
$ tar xjvf qsdk-qca-wififw_lteu-2.9.592.tar.bz2
$ mv qsdk-qca-wififw_lteu-2.9.592/qca-wifi-fw-AR9888_lteu.tar.bz2 qsdk/dl
$ tar xjvf qsdk-qca-wififw_max-2.9.592.tar.bz2
$ mv qsdk-qca-wififw_max-2.9.592/qca-wifi-fw-AR9888_max_clients.tar.bz2
qsdk/dl
$ tar xjvf qsdk-qca-wififw_atf-2.9.592.tar.bz2
$ mv qsdk-qca-wififw_atf-2.9.592/qca-wifi-fw-AR9888_atf.tar.bz2 qsdk/dl
$ tar xjvf qsdk-qca-wififw-AR9888-2.9.592.tar.bz2
$ mv qsdk-qca-wififw-AR9888-2.9.592/qca-wifi-fw-AR9888.tar.bz2 qsdk/dl
$ tar xjvf qsdk-qca-wififw-AR9887-2.9.592.tar.bz2
$ mv qsdk-qca-wififw-AR9887-2.9.592/qca-wifi-fw-AR9887.tar.bz2 qsdk/dl
```

- Additionally, customers with access to the StreamBoost release packages should run:

```
$ tar xjvf streamboost-ar71xx-2.9.592-qualcomm-ap135.tar.bz2 –C qsdk
$ sed "s:qca/feeds/customers-.*:qca/feeds/customers-qualcomm-ap135:g" -i
qsdk/feeds.conf
```

NOTE: The local directory **qsdk** is created by these repo steps as a sub-directory of the current working directory, from which repo is executed. This is the working QSDK top level directory.

5.4 Build the software

1. Install the different feeds in the build framework.
   ```
   $ cd qsdk
   $ make package/symlinks
   ```
2. Copy the base configuration to use for the build. Two profiles are available: Premium and Wireless. In this command, replace <profile> with the profile to generate.
   ```
   $ cp qca/configs/qca955x.ln/ar71xx_<profile>.config .config
   ```
3. Regenerate a complete configuration file and start the build:
   ```
   $ make defconfig
   $ make V=s
   ```
   To build ART alone:
   ```
   $ make V=s package/feeds/qca/art2/compile
   ```
4. These instructions download the packages required for the corresponding profile and create the image. Once the build is complete, the corresponding firmware should be available in the **qsdk/bin/ar71xx** directory (where <*board*> is the board supported by the generated profile):
   - kernel: openwrt-ar71xx-generic-<*board*>-kernel.bin
   - squashfs: openwrt-ar71xx-generic-<*board*>-rootfs-squashfs.bin
5. When the build is complete with no error, these files will be in the directories listed:

Files	Directory
ART	**build_dir/linux-ar71xx_generic/LinuxART2CS10.2v4.9.396/**
The file **nart.out** and other *.so files	**build_dir/linux-ar71xx_generic/LinuxART2CS10.2v4.9.396/nartbuild/**
art.ko	**build_dir/linux-ar71xx_generic/LinuxART2CS10.2v4.9.396/driver/linux/modules/**

5.5 Flashing instructions

5.5.1 Setup the flashing environment

1. As a preliminary step, ensure that the board console port is connected to the PC using the following RS232 parameters:
   - □ 115200 bps
   - □ 8N1
2. Ensure that the PC is connected to the board using one of the Ethernet ports. The PC should have a TFTP server launched and listening on the interface to which the board is connected. At this stage, power up the board, and after a few seconds, press any key during the countdown.

5.5.2 Execute flashing commands

All flashing commands start with the following U-Boot configuration. The IP address as well as the TFTP server IP@ must reflect the current network topology. To ensure this, execute these commands from U-Boot:

```
setenv ipaddr <CURRENT_IP@>
setenv serverip <CURRENT_TFTP_SERVER_IP@>
saveenv
```

Board	AP135 MIPS platform	CUS223 PCIE 802.11ac Wi-Fi module
Files	Kernel	openwrt-ar71xx-generic-ap135-kernel.bin
	Filesystem	openwrt-ar71xx-generic-ap135-rootfs-squashfs.bin

Reflash	TFTP reflash via U-Boot	`$ tftp 0x80060000 openwrt-ar71xx-generic-ap135-kernel.bin`
		`$ erase 0x9fe80000 +$filesize`
		`$ cp.b $fileaddr 0x9fe80000 0x160000`
		`$ tftp 0x80060000 openwrt-ar71xx-generic-ap135-rootfs-squashfs.bin`
		`$ erase 0x9f050000 +$filesize`
		`$ cp.b $fileaddr 0x9f050000 $filesize`
		`$ setenv bootcmd 'bootm 0x9fe80000'`
		`$ saveenv`

A Minimizing Build Time and Avoiding Multiple Downloads

Total QSDK build time can be reduced by eliminating all additional internet downloads normally performed during reruns of the install/make steps described in section 5.2, due to unpredictable bandwidth and server availability/loading of the remote repository host servers.

To reduce the time needed to obtain all files prior to **make** and to avoid repeated downloads of data, create local mirror repository that users at the same company can access via an internal network using these commands in the ChipCode repository (this example uses the distribution name *ipq_distro*; the distribution name will be something like *qualcomm /qca9558-ilq-2-0_qca_internal_integrationandtestwireless-wireless*).

- Create the local mirror repository:

```
cd /path/to/local/mirror/access/for/all
git clone --mirror https://chipcode.qti.qualcomm.com/ipq_distro.git
```

- Keep the needed files up-to-date by running:

```
cd /path/to/local/mirror/access/for/all/ipq_distro.git
git fetch
```

- For the CodeAurora repositories:

```
cd /path/to/local/mirror/access/for/all
repo init -u git://codeaurora.org/quic/qsdk/releases/manifest/qstak -b
   release -m caf_TAG --mirror
repo sync -j4
```

See section 1.2 for the caf_TAG.

Each user can create their own local copy of the QSDK code using these git and repo commands to access the mirror repositories in place of the corresponding commands in section 5.2:

- For the ChipCode repository:

```
git clone /path/to/local/mirror/access/for/all/ipq_version.git
```

- For the Code Aurora repositories:

```
repo init --reference /path/to/local/mirror/access/for/all/ -u
   git://codeaurora.org/quic/qsdk/releases/manifest/qstak -b release -m
   caf_TAG
repo sync -j4
```

See section 1.2 for the caf_TAG.

B ART2

B.1 Overview

This section describes the Qualcomm Atheros ART2 v4.9.854 package to support the chips and boards identified within this release note.

B.2 Supported functionalities

The following functionalities are supported in this build:

- Tx power calibration of a DBDC configuration (QCA9882, QCA9531)
- Saving calibration data to flash
- Saving Calibration data of QCA9882 device to EEPROM
- DBDC operation system
- Users can vary the interframe spacing to effectively vary the transmitted signal duty cycle
- Power accuracy measurement using LitePoint
- Frequency accuracy measurement using LitePoint
- Spectral mask testing using LitePoint
- EVM measurement using LitePoint
- Rx Sensitivity using LitePoint
- Calibration structure updates can be made via a **set** command with the **commit** command
- A GUI controlled test flow is included
- The ART Windows driver to support QCA9882

B.3 New `test_flow` operation

The structure and operation of the `test_flow` script has changed. This section describes these changes.

B.3.1 MAC addresses

The MAC addresses are no longer calculated from the Qualcomm Atheros label. MAC addresses are now generated from a text file that acts as a “cartridge” representing a block of MAC addresses. This file is **C:\ART2\station_files\macid.txt**, and it contains a starting address, the current address, and the final or limit address. The cart command `macid` reads the current MAC address from this file, and checks that it is not greater than the limit. If the address is less than the limit, cart increments the current address.

B.3.2 Initialization scripts

Most of the functionality that was previously in the start.art script is now in a set of scripts in **C:\ART2\station_files**. These scripts are as follows:

Script	Description
conn_load.art	Contains the connect and load commands for the board being processed
directories.art	Contains the paths to the directories used for the report and log files
equipment.art	Contains the commands describing the calibration equipment for this STA
pathloss.art	Contains the pathloss data for this STA
ref_devices.art	Describes the number of radios on each board being processed, as well as the Agile numbers for each radio

B.3.3 Shift operation

The `test_flow` script now runs in a loop that iterates over boards as well as radios per board. A “shift” consists of a number of boards to be processed. The `test_flow` script should be left running in cart for the entire “shift”. For each board, the script will prompt the user for the serial number of the board, and then run the `conn_load` script and run calibration and verification for each radio on the board, at the end of the “shift”, the user can exit and a summary of the pass/fails for the shift will be displayed.

For details on the new operation see the ART2 Reference Guide *ART2 Reference Design for AR93xx/94xx/95xx & QCA98xx Reference Guide (80-Y0861-1)*.

B.4 Known limitations

This build has these known limitations:

- Windows device drivers must be installed via the Windows device manager.
- Users can commit calibration data only a small number of times in case of OTP storage; there are 1024 bytes of OTP available, and typical CAL data takes up around 200 to 300 bytes.
- Composite EVM measurement is currently supported only for 802.11ac rates.
- The Wireless Network Management (WNM) is disabled by default and has to be enabled manually while testing Voice enterprise feature in 2.4 GHz VAPs. If WNM feature is enabled, then Quick Station kick out feature will not work for 2.4 GHz VAPs.

B.5 NART Commands

- After QSDK boots up, enter these commands on the DUT console:

```
cd /usr/sbin
/etc/init.d/art start
```

- There are no changes to the Windows-based commands.