Wireless

Wi-Fi

Dual Channel Wi-Fi Pace Xi5 RDK-V Integration and Operations Guide

WR-GL-DCW-PACE-Xi5-V01-190513

RELEASED

Notice

DISCLAIMER

This document is furnished on an “AS IS” basis and neither CableLabs nor its members provides any representation or warranty, express or implied, regarding the accuracy, completeness, noninfringement, or fitness for a particular purpose of this document, or any document referenced herein. Any use or reliance on the information or opinion in this document is at the risk of the user, and CableLabs and its members shall not be liable for any damage or injury incurred by any person arising out of the completeness, accuracy, or utility of any information or opinion contained in the document.

CableLabs reserves the right to revise this document for any reason including, but not limited to, changes in laws, regulations, or standards promulgated by various entities, technology advances, or changes in equipment design, manufacturing techniques, or operating procedures described, or referred to, herein.

This document is not to be construed to suggest that any company modify or change any of its products or procedures, nor does this document represent a commitment by CableLabs or any of its members to purchase any product whether or not it meets the characteristics described in the document. Unless granted in a separate written agreement from CableLabs, nothing contained herein shall be construed to confer any license or right to any intellectual property. This document is not to be construed as an endorsement of any product or company or as the adoption or promulgation of any guidelines, standards, or recommendations.

Document Status Sheet

Document Control Number:	WR-GL-DCW-PACE-Xi5-V01-190513			
Document Title:	Dual Channel Wi-Fi Pace Xi5 RDK-V Integration and Operations Guide			
Revision History:	D01 – Released 01/18/19 V01 – Released 05/13/19			
Date:	May 13, 2019			
Status:	~~Work in Progress~~	~~Draft~~	Released	~~Closed~~
Distribution Restrictions:	~~Author Only~~	~~CL/Member~~	~~CL/ Member/ Vendor~~	Public

Trademarks:

Contents

1 SCOPE

1.1 Introduction and Overview

This document describes how to integrate and use the Dual Channel Wi-Fi (DCW) feature on the Pace Xi5 RDK-V set-top box (STB).

1.2 Purpose of Document

The purpose of this document is to explain the process for integrating the Dual Channel Wi-Fi feature into the Pace Xi5 STB for RDK-V. In addition to the integration steps, a basic user manual and troubleshooting guide are included.

2 REFERENCES

2.1 Informative References

None

3 TERMS AND DEFINITIONS

This document uses the following terms.

data channel A downstream-only Wi-Fi connection used in Dual Channel Wi-Fi for offloading traffic from the primary channel connection.

Linux Open-source operating system created by Linus Torvalds that is the core kernel for RDK.

OSX Proprietary operating system developed and owned by Apple.

Pace Hardware vendor.

POSIX Portable Operating System Interface—a set of APIs and command-line utilities to help standardize across different flavors of UNIX and UNIX-like systems.

primary channel The main Wi-Fi connection used in Dual Channel Wi-Fi for both DCW signaling and upstream and downstream traffic.

Wi-Fi A technology enabling the wireless transmission and reception of LAN traffic.

Xi5 A model of set-top box manufactured by Pace.

4 ABBREVIATIONS AND ACRONYMS

This document uses the following abbreviations.

AP	access point
API	application programming interface
DCSTAD	Dual Channel Wi-Fi Station Daemon
DCW	Dual Channel Wi-Fi
LAN	local area network
MAC	media access control
RDK	Reference Design Kit
RDK-V	Reference Design Kit for Video
SSID	service set identifier
STB	set-top box
USB	Universal Serial Bus
WLAN	wireless local area network

5 ARCHITECTURE OVERVIEW

The Dual Channel Wi-Fi software architecture for the station consists of a daemon process that runs in the background to manage the state of Dual Channel Wi-Fi. When needed, the daemon invokes a helper script to handle the provisioning of the data channel interface(s). When connected to an access point (AP) that is not Dual Channel Wi-Fi capable, the daemon process should have no effect on the system aside from occasionally transmitting discovery frames, resulting in complete transparency for the user.

5.1 Demonstration Hardware

An Alfa AWUS036NHA (Atheros 9271) USB Wi-Fi adapter dongle is used in the development and demonstration of the Dual Channel Wi-Fi station functionality in the Pace Xi5 STB. It should be noted that this is not the final hardware intended for deployment. When the final production Wi-Fi hardware is ready, minimal integration work should be required.

When the USB dongle is connected to the set-top box, it functions as the “main” (primary channel) Wi-Fi adapter used by the video platform. The internal Broadcom 5 GHz adapter is reserved for the Dual Channel Wi-Fi data channel.

5.1.1 Linux Kernel Patches

The included Linux kernel patches for the station code are not required for the Dual Channel Wi-Fi functionality. The purpose of these patches are for bring-up of the demo hardware. Dual Channel Wi-Fi itself has no dependency on cfg80211/mac80211 features being present in the Linux kernel. The only requirement is that the DCSTAD script be able to configure the data channel adapter’s Wi-Fi parameters.

5.2 Interface Management

Dual Channel Wi-Fi is not intended to change how the primary channel Wi-Fi network interface is managed. The station daemon process is designed to cooperate with an existing Wi-Fi network manager. The existing Wi-Fi manager is responsible for setting up and joining the primary channel to the required network. However, the existing Wi-Fi network manager must ignore any Wi-Fi adapters designated for use by Dual Channel Wi-Fi. The station daemon process will manage all data channel interfaces once an AP capable of Dual Channel Wi-Fi is detected.

5.3 Bootup Logic Flow

- The “dcstad-boot.sh” file is invoked from a systemd.service unit: “dcstad.service”.
- The boot script waits indefinitely for “wlan1” (USB dongle) to become available.

  Note: If the USB dongle is never plugged in, the device will wait forever, and Dual Channel Wi-Fi will never start. The device will function normally but without Dual Channel Wi-Fi.
- The internal 5 GHz adapter is renamed from “wlan0” to “wlan5”.
- The 2.4 GHz USB dongle adapter is renamed from “wlan1” to “wlan0” so that the video platform uses the USB dongle as its “main” adapter.
- The video platform and several other network and platform services are closed.
- The station daemon (DCSTAD) is started.
- The station daemon manages the data channel (“wlan5”) interface as necessary.

6 HARDWARE COMPONENTS

Two hardware components are required for this integration and operations guide.

- Pace model Xi5 STB
- Alfa AWUS036NHA (Atheros 9271) USB Wi-Fi adapter.

7 SOFTWARE COMPONENTS

7.1 Individual Components

Dual Channel Wi-Fi station functionality comprises several software components.

7.1.1 libdcwproto

The *libdcwproto* component is a platform-independent C library responsible for marshalling and serializing the Dual Channel Wi-Fi signaling messages. The library models every Dual Channel Wi-Fi signaling message as a C struct and provides conversion to/from a raw byte-buffer ready for transmission/reception.

This component is usable for both AP and station code.

7.1.2 libdcwsocket

The *libdcwsocket* component is a Linux- and OSX-specific C library that simplifies transmission and reception of Ethernet frames by using the CableLabs Ethertype code of 0xB4E3 and the CL3 protocol type of 0x00DC. More information regarding the specific details of the protocol can be found in the protocol specification document.

This component is usable for both AP and station code.

7.1.3 DCSTAD

The *DCSTAD* daemon component is the heart of the Dual Channel Wi-Fi business-logic implementation. The component handles all network protocol signaling for Dual Channel Wi-Fi. A “DCSTAD script” is required to grant this component the ability to configure the data channel Wi-Fi interfaces.

7.1.4 DCSTAD Script

The *DCSTAD Script* component acts like a hardware-abstraction layer to the DCSTAD component. The component is called upon by the station daemon (DCSTAD) only when there is an event. The station daemon itself has no knowledge of how to set up a service set identifier (SSID) on a specific adapter, but it does know which adapter needs to be joined to which SSID. For example, when the station daemon has discovered a new AP capable of Dual Channel Wi-Fi and has received the parameters for the new data channel(s), it invokes the DCSTAD script to perform the actual join (channel-bond) operation of the data channel.

The DCSTAD script is responsible for the following:

- setting up the data channel Wi-Fi adapter parameters such as SSID, WPA, etc.;
- ensuring that traffic received on the data channel is routed into the system correctly;
- reporting to the DCSTAD process with the result of the operation—success or failure; and
- cleaning up after itself when requested.

The provided DCSTAD script (dcstad-xi5.sh) uses the “iwconfig/cfg80211” tools to set up the Wi-Fi interface, but it could easily be adapted to use something else, such as the Broadcom proprietary “wl” utility. The “dcstad-xi5.sh” script is currently implemented in the Bash scripting language; it could be implemented in any of several programming languages that respect the application programming interface (API).

7.1.4.1 DCSTAD Script API

The API for this script is maintained in the DCSTAD (https://github.com/cablelabs/dcstad) repository as “dcstad-script-api.txt”. Below is a reference version of the script API.

```
The "DCSTAD Script" is responsible for executing on the following events (reasons):
  . Join response from AP   -- "JOIN"
  . Unjoin ACK from AP      -- "UNJOIN"
  . DCSTAD Process Startup  -- "STARTUP"
  . DCSTAD Process Shutdown -- "SHUTDOWN"
```

7.1.4.1.1 AP Join Response (JOIN)

```
  The script is invoked as such:
    $ /path/to/dcstad-script
  Environment Variables:
    REASON                   -- "JOIN"
    REPLY_FD                 -- The FD# for which to reply back to the daemon on.
    PRIMARY_INTF             -- The interface name used for the WiFi primary channel.
    PRIMARY_INTF_MACADDR     -- The MAC address of the primary channel interface.
    DATACHAN_INTF_COUNT      -- The count of data channel interface names used for this channel
bond. (must be >=1)
    DATACHAN_INTF_N          -- The interface name used for a data channel (example:
DATACHAN_INTF_0=wlan1, DATACHAN_INTF_1=wlan2, ...)
    DATACHAN_INTF_N_MACADDR  -- The MAC address of a data channel.
    DATACHAN_SSID_COUNT      -- The count of data channel SSID names used for this channel bond.
(must be >=1)
    DATACHAN_SSID_N          -- The SSID name available to be used used for a data channel bond
(example: DATACHAN_SSID_0=some_dcw_datachan1, DATACHAN_SSID_1=some_dcw_datachan2, ...)

  Exit Codes:
     = 0 -- Success (DCSTAD will ACK Server)
    >= 1 -- Failure (DCSTAD will NACK Server)

  Reply:
    For the "JOIN" reason, the script must tell the daemon which interfaces were
    joined to which SSIDs. The "REPLY_FD" environment variable is provided for
    this purpose. The file descriptor number in which to write the reply is
    provided in this variable.

    Example, if there is only one data channel interface ('wlan1') and one data
    channel SSID ('dcssid'), then the response to the "REPLY_FD" would be:
      "wlan1 dcssid"
```

7.1.4.1.2 AP Unjoin (UNJOIN)

```
  The script is invoked as such:
    $ /path/to/dcstad-script

  Environment Variables:
    REASON                   -- "JOIN"
    PRIMARY_INTF             -- The interface name used for the WiFi primary channel.
    PRIMARY_INTF_MACADDR     -- The MAC address of the primary channel interface.
    DATACHAN_INTF_COUNT      -- The count of data channel interface names used for this channel
bond. (must be >=1)
    DATACHAN_INTF_N          -- The interface name used for a data channel (example:
DATACHAN_INTF_0=wlan1, DATACHAN_INTF_1=wlan2, ...)
    DATACHAN_INTF_N_MACADDR  -- The MAC address of a data channel.

  Exit Codes:
     = 0 -- Success
    >= 1 -- Failure (not sure what to do here... log an error i guess?)
```

7.1.4.1.3 Daemon Startup (STARTUP)

```
  The script is invoked as such:
    $ /path/to/dcstad-script

  Environment Variables:
    REASON               -- "STARTUP"
    PRIMARY_INTF         -- The interface name used for the WiFi primary channel.
    DATACHAN_INTF_COUNT  -- The count of data channel interface names used for this channel bond.
(must be >=1)
    DATACHAN_INTF_N      -- The interface name used for a data channel (example:
DATACHAN_INTF_0=wlan1, DATACHAN_INTF_1=wlan2, ...)

  Exit Codes:
     = 0 -- Success (DCSTAD daemon will continue to startup and run)
    >= 1 -- Failure (DCSTAD daemon will abort and exit)
```

7.1.4.1.4 Daemon Shutdown (SHUTDOWN)

```
  The script is invoked as such:
    $ /path/to/dcstad-script

  Environment Variables:
    REASON               -- "SHUTDOWN"
    PRIMARY_INTF         -- The interface name used for the WiFi primary channel.
    DATACHAN_INTF_COUNT  -- The count of data channel interface names used for this channel bond.
(must be >=1)
    DATACHAN_INTF_N      -- The interface name used for a data channel (example:
DATACHAN_INTF_0=wlan1, DATACHAN_INTF_1=wlan2, ...)

  Exit Codes:
     = 0 -- Success (DCSTAD will continue to shut down)
    >= 1 -- Failure (DCSTAD will log an error and continue to shut down)
```

7.2 Code Repositories

All Dual Channel Wi-Fi code is maintained and stored in the CableLabs GitHub team “DCW,” located at https://github.com/orgs/cablelabs/teams/dcw/repositories.

For the Xi5 specifically, the RDK-V Yocto layer repository for Dual Channel Wi-Fi can be found at https://github.com/cablelabs/meta-cablelabs-dcwsta-xi5. This repository contains all of the Yocto recipes for cloning and building all software components necessary for building an Xi5 image with Dual Channel Wi-Fi functionality.

7.3 Building

To compile the Dual Channel Wi-Fi functionality into the Pace Xi5 image, follow the standard build procedure; deviations for Dual Channel Wi-Fi are shown in red, below.

For this example, we will use the example path “/build/pacexi5” (substitute for your build path).

First, clone the initial repos:

```
$ mkdir /build/pacexi5
$ cd /build/pacexi5
$ repo init -u ssh://gerrit.teamccp.com:29418/rdk/yocto_oe/manifests/pace-bcm-manifest -b stable2
-m pacexi5.xml --repo-url=ssh://gerrit.teamccp.com:29418/rdk/tools/git-repo --no-repo-verify -g
all
$ repo sync --no-tags
$ git clone -b release git@github.com:cablelabs/meta-cablelabs-dcwsta-xi5.git
```

To build the image:

```
$ source meta-rdk/setup-environment build-xi5
  -- choose option 9 (pacexi5.conf)
$ echo 'BBLAYERS =+ "${RDKROOT}/meta-cablelabs-dcwsta-xi5"' >> conf/bblayers.conf
$ bitbake comcast-mediaclient-vbn-image
```

7.4 Yocto Meta Layer Directory Structure

All build recipes and scripts are located in the “meta-cablelabs-dcwsta-xi5” repository. Recipes in this repository reference several software component repositories that are cloned during the bitbake build process.

7.4.1 recipes-dcw/

Recipes under this directory pertain to building components specific to Dual Channel Wi-Fi.

7.4.1.1 dcstad

Clones and builds the Dual Channel Wi-Fi station daemon. Provides the “DCSTAD script” in addition to the platform bootup scripts.

7.4.1.2 libdcwproto

Clones and builds the DCW protocol C library. The DCSTAD component is statically linked with this recipe.

7.4.1.3 libdcwsocket

Clones and builds the DCW socket C library. The DCSTAD component is statically linked with this recipe.

7.4.2 recipes-firmware/

Recipes under this directory contain Wi-Fi device firmware files.

7.4.2.1 dcwdevfw

Kits the Alfa USB Wi-Fi adapter dongle firmware into the RDK image. The provided binary firmware image was obtained from an Ubuntu Linux distribution on which this device was tested.

7.4.3 recipes-kernel/

Recipes under this directory pertain to extending the Linux kernel functionality for USB Wi-Fi support.

7.4.3.1 linux

Contains a Broadcom STB Linux patch file to include the device drivers for the Alfa USB Wi-Fi adapter dongle and cfg80211/mac80211 support to accompany said drivers.

Note: We are currently cheating a little bit by building the driver as a module to ensure that the STB's internal Broadcom chip always comes up first as wlan0 and the USB dongle comes up as wlan1. There are too many places in the image that are expecting that the system is going to boot up with the Broadcom adapter initially as wlan0. After initial bootup, the adapters can be renamed.

8 PORTING TO OTHER PLATFORMS

The Dual Channel Wi-Fi station functionality is designed to be portable across a wide variety of POSIX platforms, specifically OSX and Linux variants. The station daemon (DCSTAD) has little to no dependency on the platform itself and should be able to function on almost any Linux build. The responsibility of the "DCSTAD script" is to implement the hardware bindings so that the station daemon can control the hardware.

8.1 Software Component Portability/Reusability

The following software components are virtually portable to any RDK-V platform and should be reusable with little to no modifications.

- libdcwproto
- libdcwsocket
- DCSTAD

The following software component is highly likely to not be reusable for a new platform and will require a rewrite or another solution architecture.

- DCSTAD script

8.2 Considerations for Porting to Another RDK-V Platform

- Designate at least one interface to be reserved for the Dual Channel Wi-Fi data channel.
- Ensure the platform Wi-Fi network manager ignores all data channel interfaces.
- Implement a DCSTAD script that understands how to provision the data channel interfaces.
- Implement platform-specific bring-up scripts and build recipes.

9 USER GUIDE

Dual Channel Wi-Fi station functionality on the Xi5 does not require nor does it have any configuration parameters. Setting up Dual Channel Wi-Fi and joining an AP is fully automated and seamless during both device bootup and device operation.

10 TROUBLESHOOTING

Because the station/client code has no configuration, there is very little that can go wrong with respect to misconfiguration.

If the Dual Channel Wi-Fi functionality is not working as expected, validate the following.

- 2.4 GHz USB Wi-Fi dongle is securely connected and illuminated.
- Set-top box has been on for at least five minutes.
- Dual Channel Wi-Fi station daemon is operating.
  - `$ ps -fe | grep dcstad`
  - Log file location: /var/log/dcw.log
- Primary and data channels are both associated.
  - `$ iwconfig wlan0`
  - `$ iwconfig wlan5`

11 KNOWN ISSUES AND LIMITATIONS

The current Dual Channel Wi-Fi bootup script uses a complex method for closing and rebooting most platform services on system bring-up because there is limited understanding and documentation on how the Comcast network manager is intended to work on this platform. When the final hardware targeted for production use is manufactured, the recommendation is to explore how to properly initialize the network platform and Dual Channel Wi-Fi devices on bootup.

* * *