Wireless
Wi-Fi

Dual Channel Wi-Fi OpenWrt Integration and Operations Guide

WR-GL-DCW-OpenWrt-V01-190513

RELEASED

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-OpenWrt-V01-190513
Document Title:	Dual Channel Wi-Fi OpenWrt Integration and Operations Guide
Revision History:	D01 – Released 03/13/19 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

Contents

1 SCOPE

1.1 Introduction and Overview

This document describes how to integrate and use the Dual Channel Wi-Fi (DCW) feature on hardware using the OpenWrt software distribution on a hardware device running a Linux kernel.

1.2 Purpose of Document

The purpose of this document is to explain the process for integrating the Dual Channel Wi-Fi feature into hardware running a Linux kernel. This document will describe the process generically, to allow for multiple hardware platforms using OpenWrt to be covered. In addition to the integration steps, a complete 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 OpenWrt.
OpenWrt	A Linux operating system targeting embedded devices. OpenWrt provides a fully writable file system with package management.
OSX	Proprietary operating system developed and owned by Apple.
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.

4 ABBREVIATIONS

This document uses the following abbreviations.

AP	access point
API	application programming interface
DCW	Dual Channel Wi-Fi
DCWAPD	Dual Channel Wi-Fi access point daemon
DHCP	dynamic host control protocol
GUI	graphical user interface

IP	Internet Protocol
LAN	local area network
MAC	media access control
RSSI	received signal strength indicator
SSID	service set identifier
UCI	Unified Configuration Interface
UDP	User Datagram Protocol
USB	Universal Serial Bus

5 ARCHITECTURE OVERVIEW

Specific hardware architecture will vary based on the hardware being used to implement the CableLabs OpenWrt Dual Channel Wi-Fi (DCW) distribution. Hardware platforms currently known to be able to load and run the software include the following:

- Edgewater Wireless EAP3033
- Netgear NightHawk
- TP Link Archer C7

Performance and implementation results may vary depending on the hardware platform implemented. The OpenWrt distribution of DCW is meant to provide a code baseline that can be modified for diverse hardware iterations.

6 HARDWARE COMPONENTS

Hardware componentry will vary based on platform chosen for implementation.

7 SOFTWARE COMPONENTS

7.1 Individual Components

Dual Channel Wi-Fi access point (AP) functionality comprises several individual software components.

7.1.1 OpenWrt

The *OpenWrt* component is the Linux-based operating system used for this implementation of Dual Channel Wi-Fi.

7.1.2 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.

7.1.3 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.

7.1.4 MAC Address Remapper

The *MAC Address Remapper* component provides functionality to the Linux kernel, which is responsible for the actual filtering and switching of Ethernet traffic onto the data channel.

For more information on this component, please refer to the MAC Address Remapper documentation.

7.1.5 DCWAPD

The *DCWAPD* component is the heart of the Dual Channel Wi-Fi business-logic implementation. The component is responsible for orchestrating all Dual Channel Wi-Fi operations. The component provides the "dcwapd." parameters to the UCI data model. All AP signaling logic is implemented in this component.

7.2 Code Repositories

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

Each individual software component can be found in its respective git repository.

Wireshark Dissectors	https://github.com/cablelabs/dcwwireshark
libdcwproto	https://github.com/cablelabs/libdcwproto
libdcwsocket	https://github.com/cablelabs/libdcwsocket
MAC Address Remapper	https://github.com/cablelabs/macremapper
DCWAPD	https://github.com/cablelabs/dcwapd
OpenWrt-dcw-feed	https://github.com/cablelabs/OpenWrt-dcw-feed

7.3 Building

These are the build instructions for compiling Dual Channel Wi-Fi access point functionality into an OpenWrt image.

1. Clone the OpenWrt source.

   ```
   $ git clone https://git.OpenWrt.org/OpenWrt/OpenWrt.git
   ```

2. Change directory to the cloned "OpenWrt" directory.

   ```
   $ cd OpenWrt
   ```

3. Using "vi" or another test editor, add the following line to the *feeds.conf.default* file.

   ```
   src-git cablelabs https://github.com/cablelabs/OpenWrt-dcw-feed.git
   ```

4. Update the package feeds and install the packages.

   ```
   $ ./scripts/feeds update -a
   $ ./scripts/feeds install -a
   ```

5. Make the default configuration.

   ```
   $ make defconfig
   ```

6. Run the menu configuration editor and select the desired platform, packages, etc.; be sure to save the changes when done.

   ```
   $ make menuconfig
   ```

7. Build the image.

   ```
   $ make
   ```

   Or build the image with debugging.

   ```
   $ make -j1 V=s
   ```

DCW Package Feed Notes

The dcwifi packages can be found in the menuconfig in the following locations:

* dcstad: "Network -> Routing and Redirection"
* dcwapd: "Network -> Routing and Redirection"
* libdcwproto: "Libraries -> Networking"
* libdcwsocket: "Libraries -> Networking"
* macremapper: "Kernel modules -> Network Support" (listed as "kmod-macremapper")
* mrmctl: "Utilities"

7.4 OpenWrt Feed Directory Structure

The OpenWrt-dcw-feed repository contains the Makefiles and resources necessary to build and install the dcwifi components.

7.4.1 dcwifi/

The parent directory containing the dcwifi components.

7.4.1.1 dcstad/

Clones and builds the Dual Channel Wi-Fi client daemon.

7.4.1.2 dcwapd/

Clones and builds the Dual Channel Wi-Fi access point daemon. Contains the code and patches to integrate the DCWAPD code into the OpenWrt UCI data model.

7.4.1.3 libdcwproto/

Clones and builds the Dual Channel Wi-Fi protocol C library.

7.4.1.4 libdcwsocket/

Clones and builds the Dual Channel Wi-Fi socket C library.

7.4.1.5 macremapper/

Clones and builds the macremapper Linux kernel component.

7.4.1.6 mrmctl/

Clones and builds the user-land debug "mrmctl" utility.

7.4.2 luci-app-dcwapd/

Contains the code for the LuCI web interface application that can be used to configure and control Dual Channel Wi-Fi functionality.

8 PORTING TO OTHER PLATFORMS

Because the Dual Channel Wi-Fi feature may have dependencies on the hardware platform, it is not possible to provide an OpenWrt DCW code-base that will universally compile and run across all Wi-Fi hardware devices. However, the OpenWrt DCW code-base was architected to compile and run on many Wi-Fi hardware platforms, a few of which are listed in Section 5 of this document.

8.1 Software Component Portability/Reusability

All software components are virtually portable to any Wi-Fi platform that can run OpenWrt by using the Linux bridge and should be reusable with little to no modifications.

9 USER GUIDE

9.1 Graphical User Interface (GUI)

The OpenWrt version of DCW contains a graphical user interface (GUI) that can be used to configure the DCW feature within the Wi-Fi device. Instructions for configuration using the GUI are below. Depending on the device used, the appearance of the GUI may differ from what is shown in the screenshots in the configuration instructions, but the steps listed should be generic enough to allow a user to find the proper menus and settings on any device.

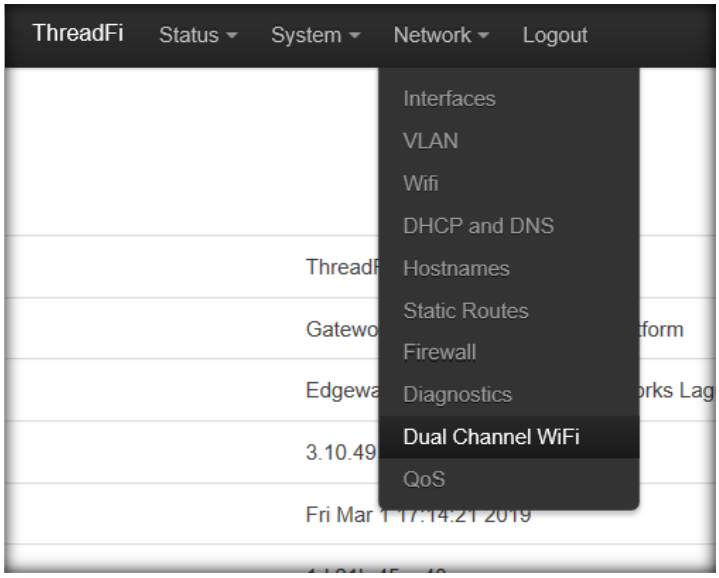
9.1.1 Logging into the GUI

1. Connect the computer's Ethernet port to an Ethernet port on the target Wi-Fi device.
2. Configure the laptop with an IP address in the same subnet as the Wi-Fi device.
   a. The default IP address and subnet of a newly built device using the software should be the following:
      i. IP Address: 192.168.1.1
      ii. Subnet Mask: 255.255.255.0
3. Open a browser on the laptop and point it to the address of the Wi-Fi device. A login screen should appear.
   a. Credentials for Edgewater Wireless OpenWrt devices are as follows:
      i. Username: admin
      ii. Password: password
   b. Credentials for most OpenWrt devices are as follows:
      i. Username: root
      ii. No password

9.1.2 Configuring DCW

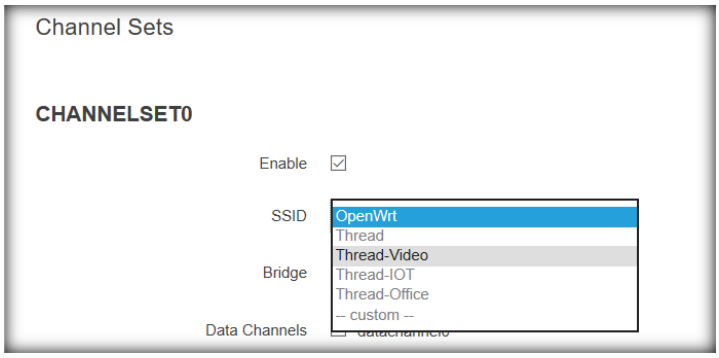
Once logged into the device, use the GUI to configure DCW.

Select Dual Channel Wi-Fi from the Network menu.

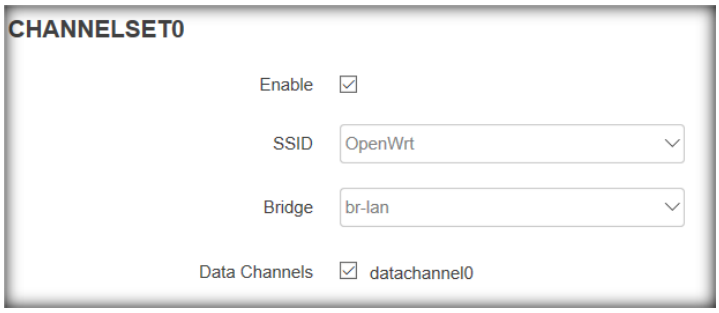


To Enable DCW, tick the Enable checkbox under General Settings.

To set the primary channel for the DCW feature, move to the CHANNELSET0 section under "Channel Sets." Tick the Enable checkbox, then select the desired SSID for the primary channel from the drop-down list. The default for the bridge is automatically selected and can be assumed to be correct. For an advanced setup, the bridge can be customized.

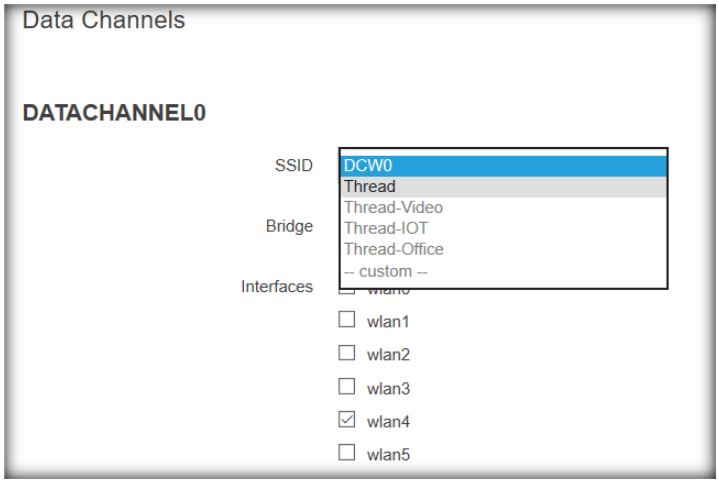


Next, tick the box next to the desired data channel. The default data channel selection can be assumed to be correct. For an advanced setup, additional data channels can be added and subsequently selected for use with the primary channel.

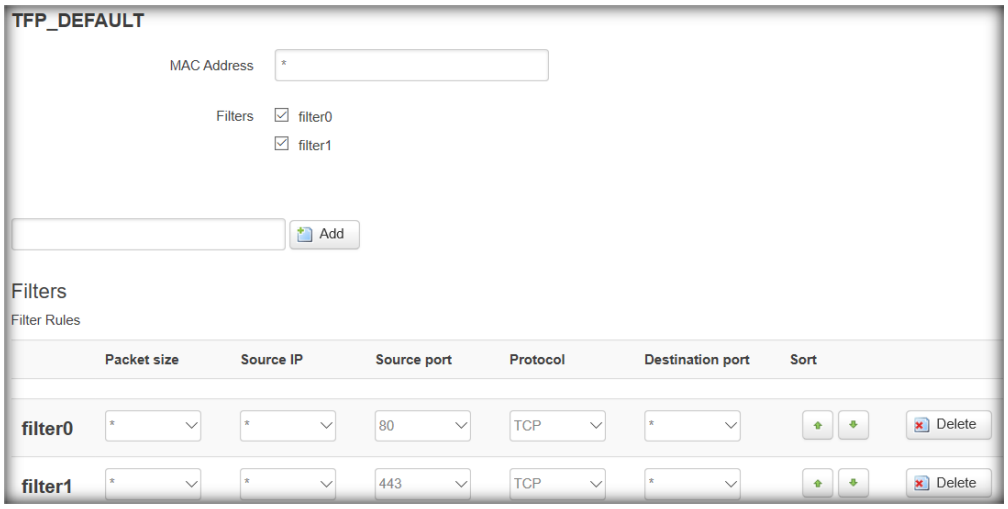


To prepare the data channel for use, move to the DATACHANNEL0 section under "Data Channels." Select the desired SSID for the data channel from the drop-down list. The default for the bridge is automatically selected and can be assumed to be correct. For an advanced setup, the bridge can be customized.

Lastly, tick the box next to the desired WLAN interface for this data channel.



Filters can be used to mark traffic for delivery over the data channel. The default filters are for http and https protocols and are good choices for testing. Filters can be added or changed for an advanced setup.



When finished, save the device's DCW configuration and begin the DCW Client configuration and connection.

10 TROUBLESHOOTING

In general, Wireshark is recommended for troubleshooting network issues because it provides visibility into the traffic passing through the network. For information on the protocol format of the signaling messages, it is advised to review the Dual Channel Wi-Fi protocol specification document.

10.1 Useful Wireshark Filters

- Show only Dual Channel Wi-Fi signaling messages on a Wi-Fi adapter not in monitor mode:
  - `eth.type == 0xb4e3`
- Show only Dual Channel Wi-Fi signaling messages on a Wi-Fi adapter in monitor mode:
  - `llc.type == 0xb4e3`

10.2 Validate the Health of All Software Components

To confirm the overall health of the Dual Channel Wi-Fi components on the device, the following basic steps should be executed; healthy components will yield no errors.

10.2.1 Ensure the DCWAPD Process Is Functional

To determine if the Dual Channel Wi-Fi AP daemon process is running, use the ps command and look for the DCWAPD process.

```
$ ps | grep dcwapd
```

10.2.2 Examine the UCI Data Model

The UCI data model controls the operation of the DCWAPD process and network reconfiguration. Examining the DCWAPD UCI configuration can be helpful in identifying abnormal operations.

```
$ uci show dcwapd.
```

10.2.3 Ensure the MAC Address Remapper Is Functional

Query the state of the MAC Address Remapper by using the "mrmctl" utility.

```
$ mrmctl show
```

If it is not desired to kit the "mrmctl" utility into the image, the state of the MAC Address Remapper can be queried by calling "cat" on the control device.

```
$ cat /proc/macremapctl
```

10.3 Debugging the MAC Address Remapper

Because the MAC Address Remapper is the component responsible for the actual traffic flow of Dual Channel Wi-Fi, it is imperative to have the means to troubleshoot it. The "mrmctl" utility is a command-line tool that allows the developer to debug the state of the MAC Address Remapper. Everything exposed through the software API is also exposed through the "mrmctl" utility. For more information on this process, please refer to the MAC Address Remapper documentation.

10.4 Station Does Not Bond in Dual Channel Wi-Fi Mode

- Confirm that the station is correctly joined to the primary channel SSID. If more than one AP is using the primary channel SSID, then it is highly advised to change the primary channel SSID to something unique to only this AP.
- Ensure the Dual Channel Wi-Fi client software is running and that there are no errors.
- Ensure the station’s Wi-Fi adapters support the radio frequencies and Wi-Fi standards being used. For example, using a USB Wi-Fi dongle that supports only 2.4 GHz for a data channel on 5 GHz will not work.
- Run Wireshark on the station’s primary channel interface to validate the presence of the DCW signaling frames.

- Check the data model on the AP for presence of the station's primary channel MAC address.
- Try swapping the station's Wi-Fi adapters. For example, if the internal adapter is used as the primary channel, try using the USB dongle as the primary channel adapter.

10.5 Station Bonds in Dual Channel Wi-Fi Mode but Traffic Does Not Flow

- Ensure the primary channel SSID and data channel SSID are different.
- Completely disable Dual Channel Wi-Fi on the AP, and test both radios individually.
- Check the station's RSSI values. If too low, try reducing the channel width of the data channel.

11 KNOWN ISSUES AND LIMITATIONS

11.1 SSID for Primary Channel and Data Channel Must Be Different

Because of the way the Dual Channel Wi-Fi signaling protocol works, it is imperative that the primary channel and data channel SSIDs are different. If these values are the same, the stations will unlikely be able to successfully complete channel bonding.

11.2 Filtering DHCP Traffic Through the Data Channel

Most dynamic host control protocol (DHCP) client implementations bind to a specific interface when operating. Filtering DHCP traffic (UDP ports 67 and 68) through the data channel can cause the client to not receive DHCP packets. If this happens, Dual Channel Wi-Fi stations may not be able to obtain or renew existing DHCP leases, therefore cutting off IP communication with the network. For this reason, it is not advisable to filter DHCP traffic onto the data channel.

* * *