Application Note

CPE

Dongle Host Driver Offload

CPE-AN2300-R

February 16, 2016

5300 California Avenue • Irvine, CA 92617 • Phone: 949-926-5000 • Fax: 949-926-5203

Revision History

Revision	*Date*	*Change Description*
CPE-AN2300-R	02/16/16	Initial release

Broadcom Corporation
5300 California Avenue
Irvine, CA 92617

Table of Contents

List of Figures

List of Tables

About This Document

Purpose and Audience

This document describes the Dongle Host Driver (DHD) offload feature. The scope of this document is restricted to a high level description of the feature, the configuration parameters, and the feature usage. The document is aimed at software engineers wanting to use the DHD offload feature.

Acronyms and Abbreviations

In most cases, acronyms and abbreviations are defined on first use. Table 1 shows the acronyms commonly used in this document.

Table 1: Acronyms and Abbreviations

AC	Access Category
BCMC	Broadcast/Multicast
BE	Best Effort Access Category
BK	Background Access Category
BSS	Basic Service Set
CFE	Common Firmware Environment
DHD	Dongle Host Driver
NIC	Network Interface Card
RDP	Runner Data Path
SSID	Service Set Identifier
TM	Traffic Management
VI	Video Access Category
VO	Voice Access Category
WFD	Wi-Fi Forwarding Driver

For a comprehensive list of acronyms and other terms used in Broadcom documents, go to: http://www.broadcom.com/press/glossary.php.

Document Conventions

The following conventions may be used in this document:

Convention	*Description*
Bold	User input and actions: for example, type **exit**, click **OK**, press **Alt+C**
Monospace	Code: #include <iostream> HTML: <td rowspan = 3> Command line commands and parameters: wl [-l] <command>
< >	Placeholders for *required* elements: enter your <username> or wl <command>
[ ]	Indicates *optional* command-line parameters: wl [-l] Indicates bit and byte ranges (inclusive): [0:3] or [7:0]

Technical Support

Broadcom provides customer access to a wide range of information, including technical documentation, schematic diagrams, product bill of materials, PCB layout information, and software updates through its customer support portal (https://support.broadcom.com). For a CSP account, contact your Sales or Engineering support representative.

In addition, Broadcom provides other product support through its Downloads and Support site (http://www.broadcom.com/support/).

Overview

The Dongle Host Driver (DHD) offload feature refers to the capability of a system to offload the traffic classification, modification and forwarding to and from WLAN to the hardware accelerator, thus reducing the host CPU utilization.

The term DHD Offload is used within the scope of this document as follows:

- Dongle Offload and DHD Offload may be used interchangeably.
- Offload specifically means “Host side processing offload”.
- The term does not apply to “NIC” mode WLAN drivers (including WLAN Dongles used in NIC mode).
- Dongle refers to a single WLAN card (43602, 4366 plus variants and such future devices).

DHD offload is currently supported on the following Runner Data Path (RDP) hardware accelerator based platforms:

- BCM963138
- BCM963148
- BCM96838X
- BCM96848X

The mechanism of flow learning/provisioning by flow-cache into the RDP accelerator is outside the scope of this document.

This document does not intend to highlight the exact code execution path but is limited to briefly provide the high level packet/processing flow.

Build Instructions

By default, the DHD Offload feature is enabled in 963138/96838 gateway profiles starting from release 4.16L.04.

DHD offload requires more memory than the non-offload case, this memory must be reserved at system boot time. Careful consideration is necessary when enabling this feature on low memory devices, see “Memory Reservation” on page 9.

To enable/disable the DHD offload feature before building:

1. Use the **make menuconfig** command at the Linux command prompt.
   ```
   $ make menuconfig
   ```
2. Navigate to menuconfig→Packet Acceleration
3. Select [ ] DHD Runner Acceleration Support
4. Save and exit

Memory Reservation

The amount of memory to be reserved for DHD Offload is specified in the CFE boot loader parameters, as shown in the example below for the BCM963138 reference board. These values are passed to Linux during system boot.

```
Base MAC Address                  : 02:10:18:89:82:01
PSI Size (1-128) KBytes           : 48
Enable Backup PSI [0|1]           : 0
System Log Size (0-256) KBytes    : 0
Auxillary File System Size Percent: 0
MC memory allocation (MB)         : 4
TM memory allocation (MB)         : 44
DHD 0 memory allocation (MB)      : 14
DHD 1 memory allocation (MB)      : 7
DHD 2 memory allocation (MB)      : 0
```

DHD 0/1/2 refers to the radio index that is assigned to the specific WLAN dongle during the PCIe scan (see “Radio Number Assignment” on page 23). Specified memory is reserved prior to Linux boot and will not be available for Linux usage. In the above example, the first WLAN dongle is allocated 14 MB and the second WLAN dongle is allocated 7 MB. See “Flow-Ring Allocation Controls” on page 23 for more details.

Using the default configuration (see “Flow Ring Profile” on page 23) and offload for all access categories, each WLAN station would require ~216 KB of memory. Therefore, based on the maximum number of clients/stations supported by the dongle, the required memory will be:

- Dongle with 32 station support: ~7 MB
- Dongle with 64 station support: ~14 MB

Note: In the above example, “TM Memory allocation” for 963138 platforms is also changed to 44 MB from the default 20 MB. This increase in TM memory is recommended in order to increase the total number of RDP packet buffers and to accommodate the large buffering requirements towards the WLAN dongle.

For 96838X platforms, the recommended TM memory requirement is 36 MB.

DHD Offload

Interaction with Dongle

All communication with Dongle is via descriptor rings irrespective of offload or not. Dongle firmware remains the same in both cases and dongle is configured to work in offload/non-offload modes. The following type of descriptor rings are used:

- Packet Receive Path related rings
  - RxBuf Post Ring (RX_POST): DHD/RDP puts the descriptor for the packet receive buffer in this ring. Dongle uses these buffers to receive the packets from WLAN Clients and passes them back to DHD/RDP (using RxPkt Complete Ring). Only one RX_POST ring per Dongle is created.
  - RxPkt Complete Ring (RX_CMPL): Dongle puts the descriptors for received packets in this ring and signals the DHD/RDP to process the packet(s). Only one RX_CMPL ring per Dongle is created.
- Packet Transmit Path related rings
  - TxPkt Post Flow-ring (TX_POST/Flow-Ring): DHD/RDP puts the descriptor for packets that need to be transmitted towards Dongle (i.e., WLAN Station) and signals the Dongle to process the packet(s). Up to four flow-rings are created per station for each access category (VO, VI, BE, BK). Each flow-ring can be a different size. The size is configurable through the boot time profile, see "Flow-Ring Allocation Controls" on page 23. There is also a flow-ring for Broadcast/Multicast traffic for every SSID/BSS.
  - TxPkt Complete Ring (TX_CMPL): Dongle puts the descriptors related to transmitted packets in this ring and signals the DHD/RDP to indicate completion of transmission, so that DHD/RDP can free up the packet buffers. Only one TX_CMPL ring per Dongle is created.

Descriptor Ring Allocation

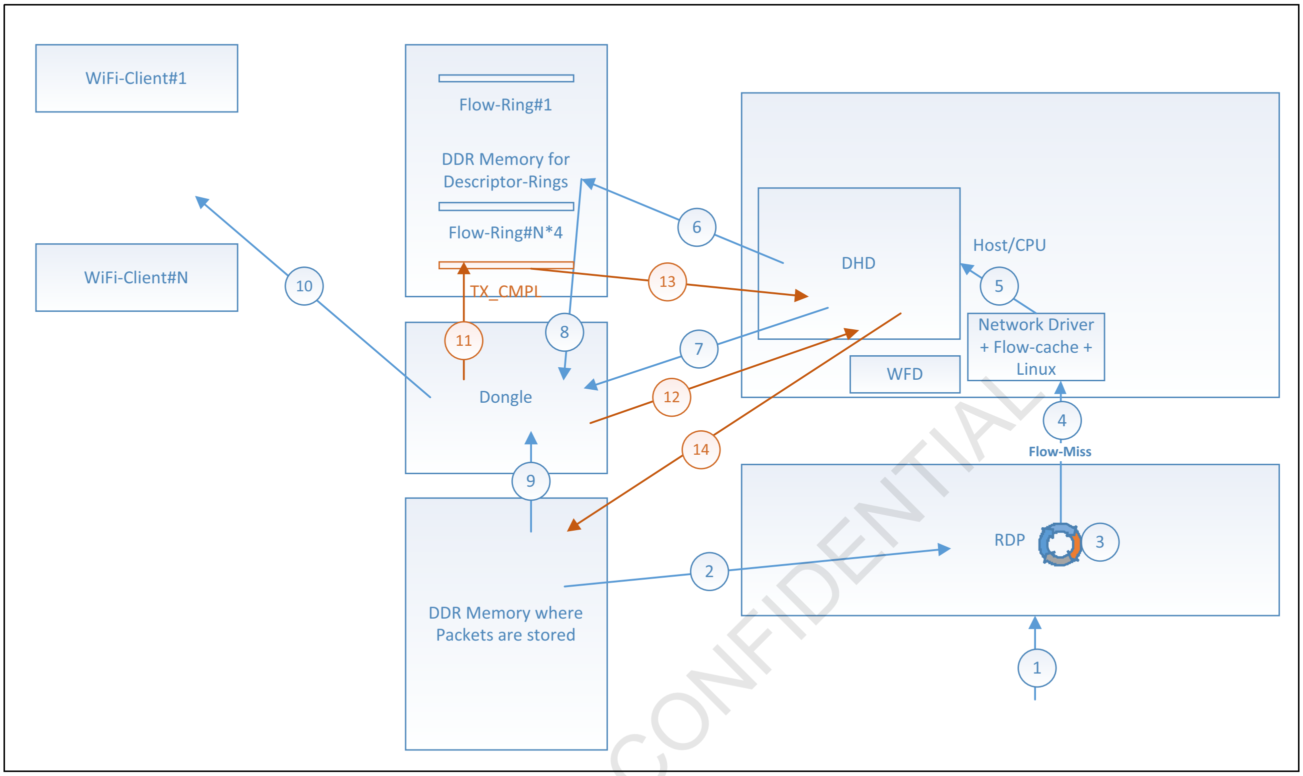
DHD allocates all the rings at initialization time based on the configured sizes and provides the required information to Dongle.

Non-Offload/DHD Path

When the DHD Offload feature is disabled, based on the number of stations supported by dongle, DHD allocates the Flow-Rings to be a size of 512 each. When required, DHD has mechanism to provide additional buffering for each station in addition to these 512 size flow-rings.

Packet Receive (DONGLE to DHD)

Figure 1: Packet Receive (DONGLE to DHD)



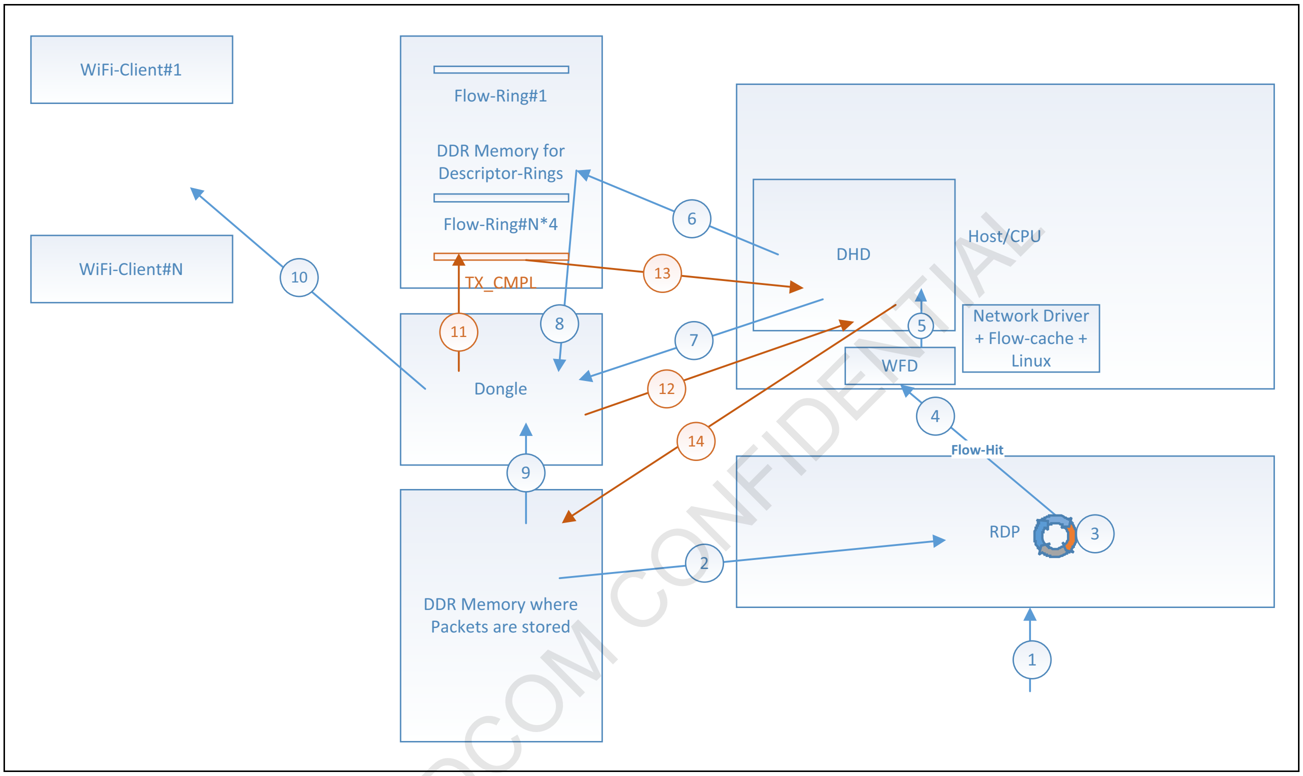
1. DHD allocates packet buffers to receive the packet from Dongle.
2. DHD puts the Receive packet buffer information in RX_POST Descriptor ring.
3. Dongle fetches the receive packet buffer information from the RX_POST descriptor ring.
4. Dongle receives the packet from WiFi-Client into the packet buffer.
5. Dongle processes the packet.
6. Dongle puts the information about received packet in RX_CMPL descriptor ring.
7. Dongle notifies the DHD about packet receive.
8. DHD fetches the descriptor from the RX_CMPL ring.
9. DHD fetches the packet from the packet buffers.
10. DHD forwards the packet to Network Driver (either through Linux or flow-cache).
11. Network Driver gives the packet to RDP for transmission.
12. RDP transmits the packet towards LAN/WAN.
13. RDP sends the packet buffer back to Network Driver to free.
14. Network Driver frees up the buffer back to Packet buffer memory.

Packet Transmission (DHD to DONGLE)

Packet transmission towards dongle can take two paths based on if the flow is via RDP or not, a Transmit Flow-miss Case or a Transmit Flow-hit Case.

Transmit Flow-miss Case

Figure 2: Packet Transmit Flow-miss Case



1. The packet is received by the RDP from LAN/WAN port.
2. The packet is received in the DDR memory buffer.
3. RDP processes the packets and makes the forwarding decision (flow-miss).
4. RDP sends the packet to Host CPU Network driver.
5. Network driver receives the packet and forwards it to DHD Driver (either through Linux Kernel, or Flow-cache).
6. DHD updates the TX_POST descriptor in the flow-ring with required information.
7. DHD notifies the Dongle about the packet.
8. Dongle reads the TX_POST descriptor from flow-ring and gets the required information.
9. Dongle grabs the packets from DDR.
10. Dongle transmits the packets towards WiFi-Client.
11. Dongle updates the descriptor with transmitted packet buffer information in TX_CMPL Ring.

12. Dongle notifies the DHD about the TX_CMPL ring update.

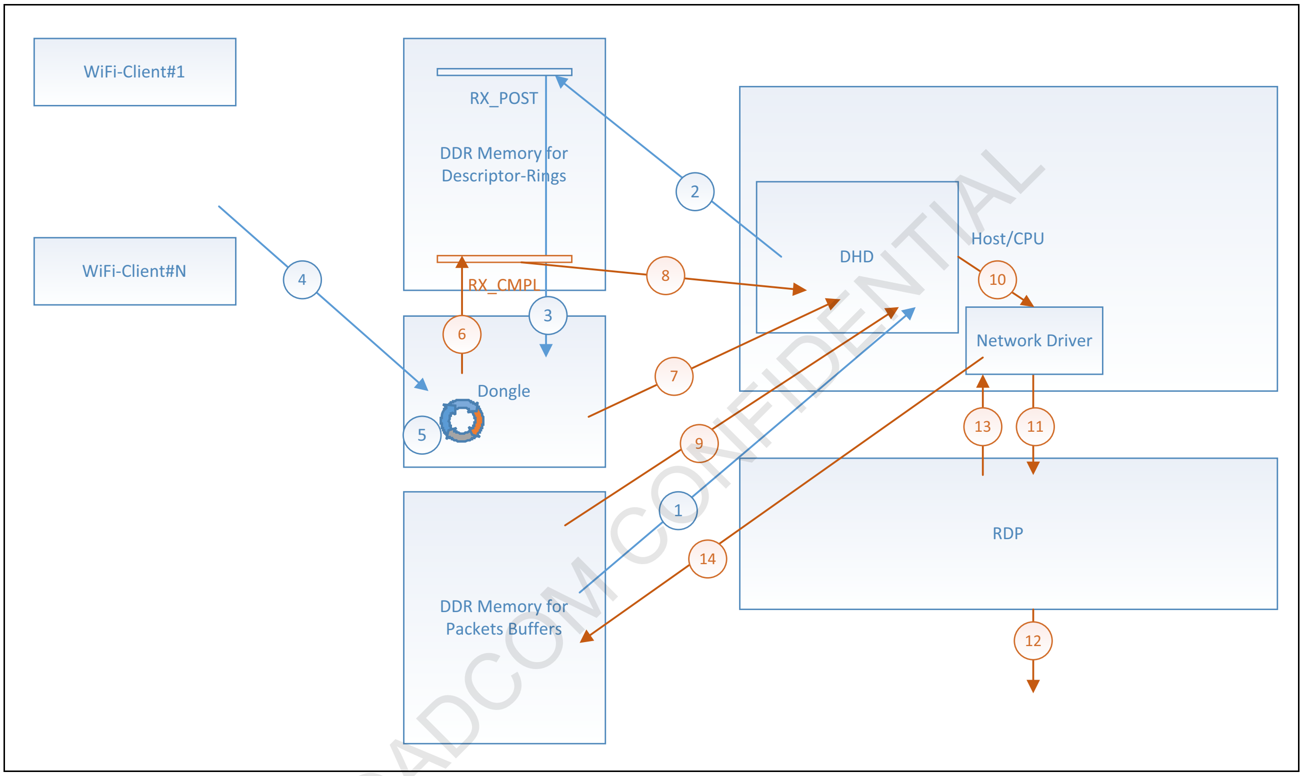
13. DHD takes the packet buffer information from the TX_CMPL ring.

14. DHD frees the packet buffer back to the packet pool (from wherever it came from).

Transmit Flow-hit Case

The difference between the Flow-miss and Flow-hit cases is highlighted below.

Figure 3: Packet Transmit Flow-hit Case



1. The packet is received by RDP from LAN/WAN port.

2. The packet is stored in the DDR memory.

3. RDP processes the packets and makes the forwarding decision (flow-hit).

4. RDP sends the packet to Host CPU WFD.

5. WFD receives the packet and forwards it to DHD Driver.

6. DHD updates the TX_POST descriptor in the flow-ring with required information.

7. DHD notifies the Dongle about the packet.

8. Dongle reads the TX_POST descriptor from flow-ring and gets the required information.

9. Dongle grabs the packets from DDR.

10. Dongle transmits the packets towards WiFi-Client.

11. Dongle updates the descriptor with transmitted packet buffer information in TX_CMPL Ring.
12. Dongle notifies the DHD about the TX_CMPL ring update.
13. DHD take the packet buffer information from the TX_CMPL ring.
14. DHD frees the packet buffer back to the packet pool (from wherever it came from).

DHD Offload Path

The interface with Dongle remains the same irrespective of offload or host/CPU DHD path. DHD offload provides a method such that descriptor rings can be managed by the RDP (instead of DHD). This capability enables the RDP to directly send/receive the packets to/from Dongle without DHD/Host involvement.

Allocations of descriptor rings remain the responsibility of DHD and the required information is passed to RDP to manage those rings.

Ring Allocations

When DHD Offload feature is enabled in the build profile, the receive path from the dongle is always handled by RDP.

For the remainder of this document and better understanding, following nomenclature will be used:

- Number of flow-rings managed by RDP = "N"
- Number of flow-rings managed by DHD = "M"
- Therefore the total number of flow-rings required to support all the stations on a Dongle is "N+M".

Each ring type management is detailed below:

- RX_CMPL: Managed by RDP.
- RX_POST: Managed by RDP.
- TX_POST/Flow-Rings: Allocation and Management of flow-rings (used to transmit data packets towards Dongle) is determined by the memory reservation done at system boot time (information that is supplied by the boot loader – see "Memory Reservation" on page 9".
  - If the memory is not enough to offload all flow-rings, DHD will allocate some flow-rings "N" in the reserved memory for RDP to manage and will allocate remaining "M" from Linux to be managed by DHD.
  - If there is enough memory reserved to offload all the flow-rings to RDP, then DHD will allocate all flow-rings from the reserved memory and hand over the management to RDP (i.e., M=0).
  - If there is no memory reserved for flow-rings, DHD will allocate all flow-rings from Linux memory pool (as it would do in offload disabled builds) and manage the flow-rings as well (i.e., N=0).
- TX_CMPL: This ring is managed by RDP if N > 0, otherwise managed by DHD.

Packet Receive (Dongle to RDP)

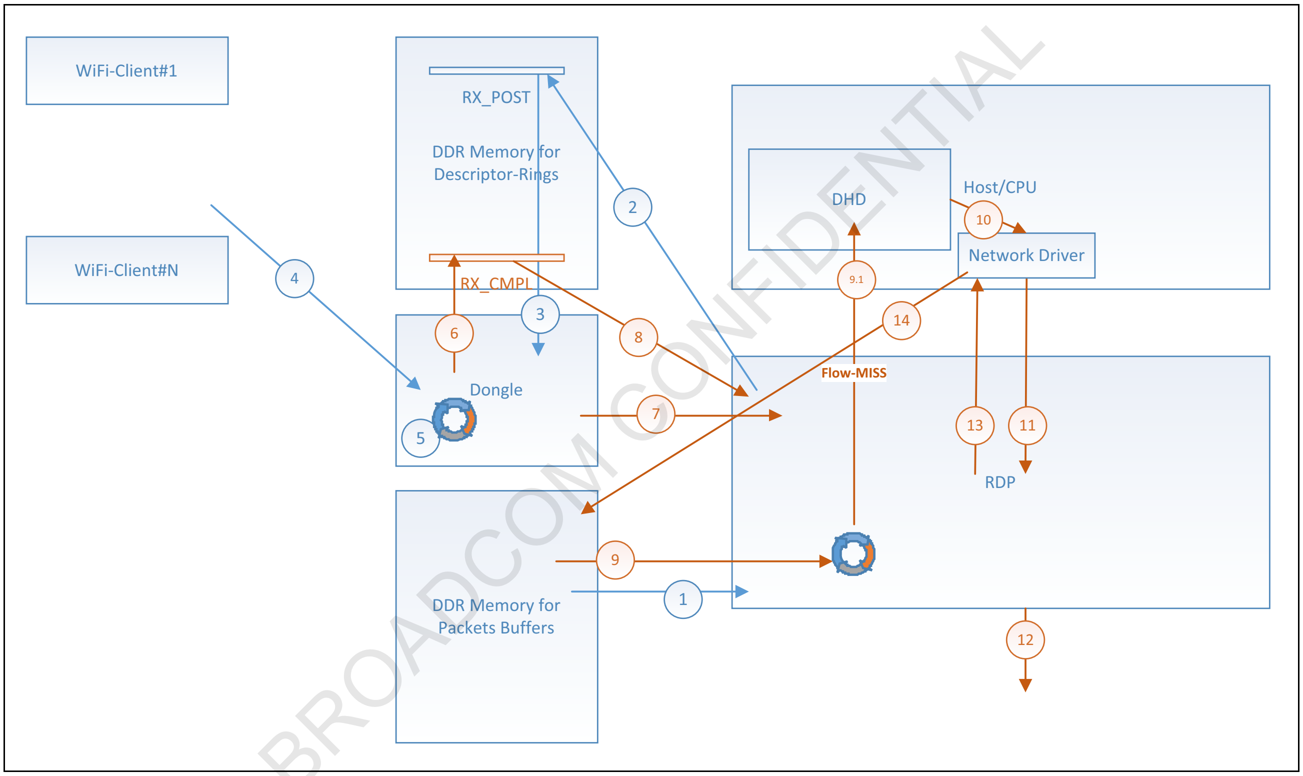
When DHD offload is enabled, all packets from Dongle are always received by the RDP (accelerated or not).

Receive Flow-Miss Case

The Flow-miss path (packet trapped to DHD for processing) will occur whenever RDP does not have the flow to process/forward the packet. This mechanism remain same as any other packet acceleration done by RDP for other interfaces (LAN/WAN).

The changes from the DHD receive path are highlighted in the text description to emphasis the fact that Dongle does not care if the packet is received by the RDP or DHD.

Figure 4: Packet Receive Flow-Miss Case



1. RDP allocates a packet buffer to receive the packet from Dongle.
2. RDP puts the Receive packet buffer information in RX_POST Descriptor ring.
3. Dongle fetches the receive packet buffer information from the RX_POST descriptor ring.
4. Dongle receives the packet from WiFi-Client into the packet buffer.
5. Dongle processes the packet.
6. Dongle puts the information about received packet in RX_CMPL descriptor ring.
7. Dongle notifies the RDP about the packet.
8. RDP fetches the descriptor from the RX_CMPL ring.

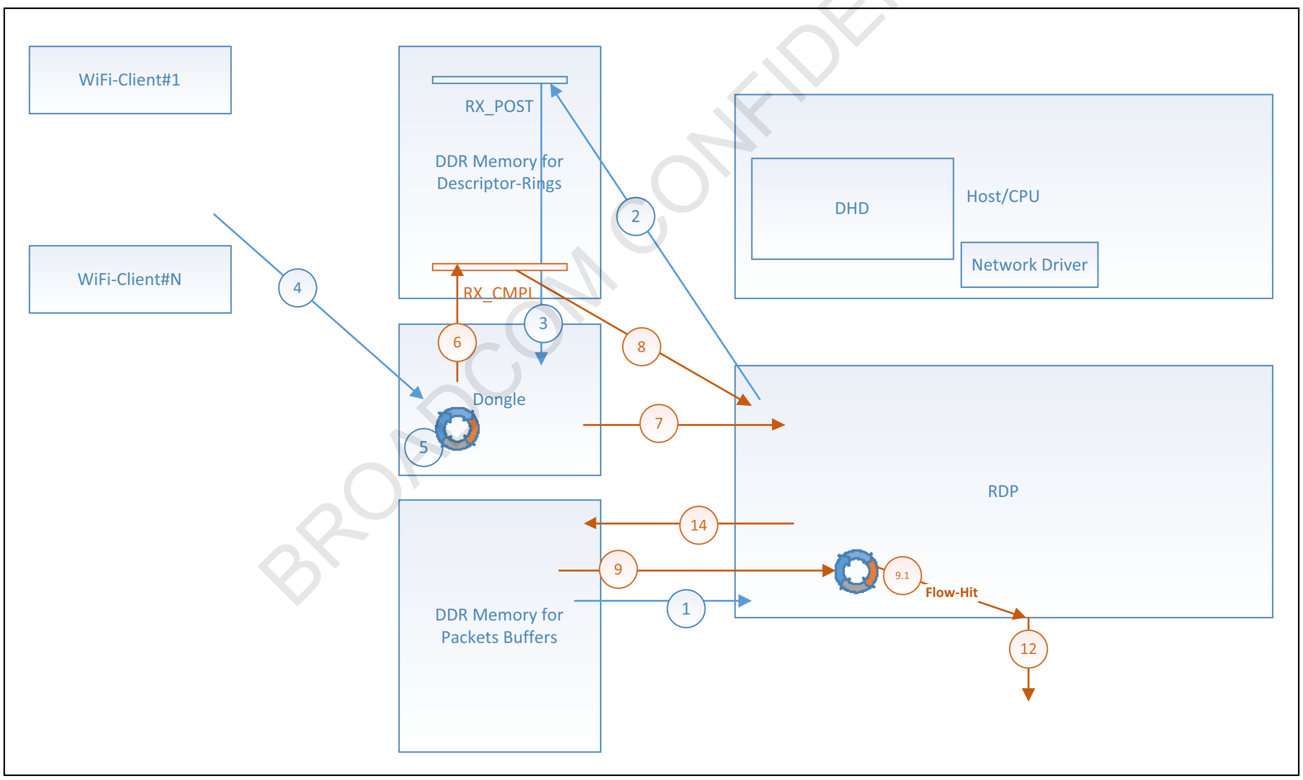
9. RDP fetches the packet from the packet buffers.
   RDP has no flow for this packet → Flow-miss → the packet is trapped to DHD.
10. DHD forwards the packet to Network Driver (either through Linux or flow-cache).
11. Network Driver gives the packet to RDP for transmission.
12. RDP transmits the packet towards LAN/WAN.
13. RDP sends the packet buffer back to Network Driver to free.
14. Network Driver frees up the buffer back to Packet buffer memory.

Receive Flow-hit Case

The Flow-hit path will occur whenever RDP has the flow to process/forward the packet. This mechanism remains the same as any other packet acceleration done by RDP for other interfaces (LAN/WAN).

Flows are pushed to RDP by flow-cache the same way as done for any other interface. The changes from the "Flow-miss" receive path are highlighted in the text description below.

Figure 5: Packet Receive Flow-hit Case



1. RDP allocates a packet buffer to receive the packet from Dongle.
2. RDP puts the Receive packet buffer information in RX_POST Descriptor ring.
3. Dongle fetches the receive packet buffer information from the RX_POST descriptor ring.
4. Dongle receives the packet from WiFi-Client into the packet buffer.

5. Dongle processes the packet.
6. Dongle puts the information about received packet in RX_CMPL descriptor ring.
7. Dongle notifies the RDP about packet.
8. RDP fetches the descriptor from the RX_CMPL ring.
9. RDP fetches the packet from the packet buffers.
   9.1. RDP has flow for this packet → Flow-hit → the packet is forwarded to the egress interface.
10. RDP transmits the packet towards LAN/WAN.
11. RDP frees up the buffer back to Packet buffer memory.

Packet Transmission (RDP/DHD to Dongle)

The packet transmission path in DHD Offload enabled builds can take three different paths depending upon Flow-ring allocation and flow-hit/miss in RDP (see “Ring Allocations” on page 14.)

- Flow-ring managed by RDP (one among the “N” flow-rings): All traffic towards this flow-ring must go through RDP.
- Flow-ring managed by DHD (one among the “M” flow-rings): All traffic towards this flow-ring must go through DHD.

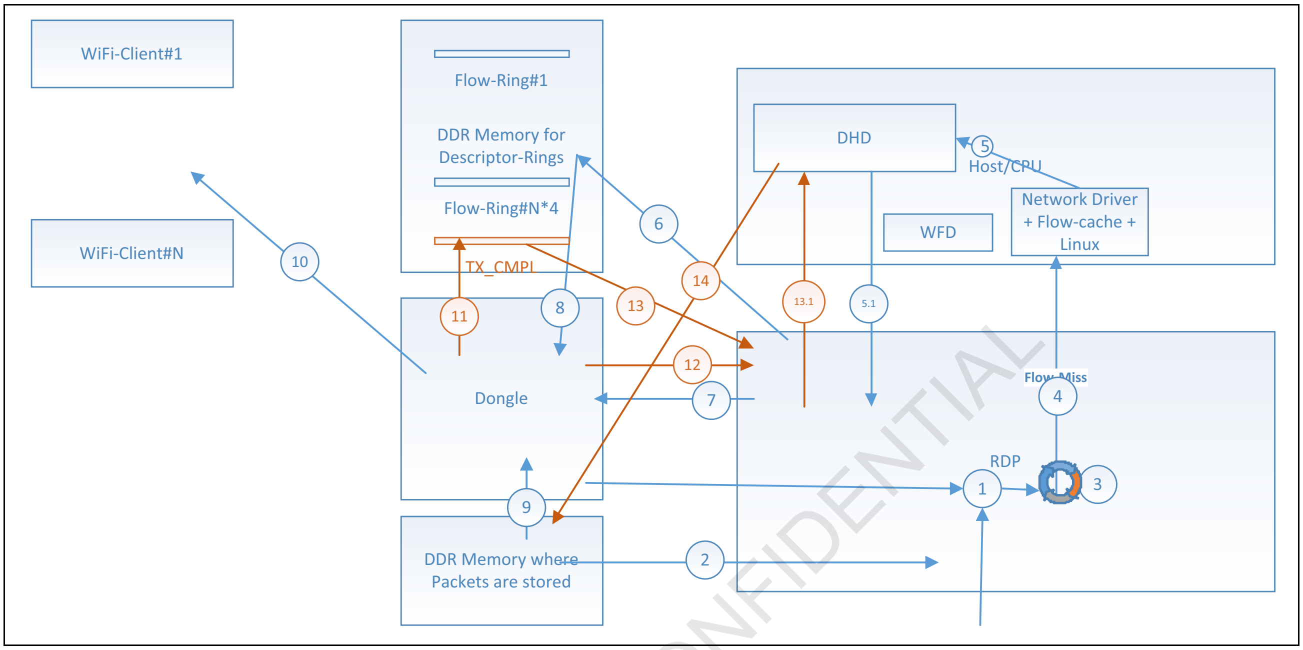
RDP Managed Flow-rings

This case illustrates the data path flow for two scenarios, flow-miss and flow-hit, when the Flow-ring is managed by the RDP.

Flow-Miss Case

The differences between the offload and the non-offload flow-miss case are highlighted in Figure 6.

Figure 6: RDP Managed Flow-miss Case

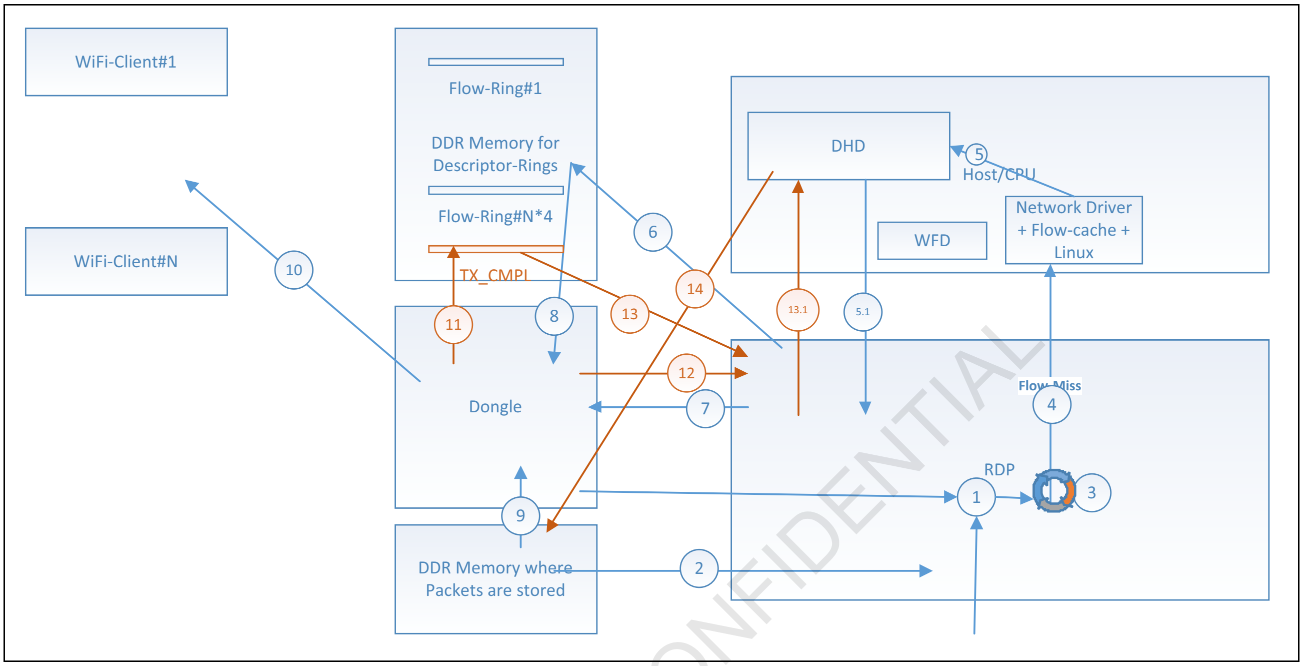


1. Packet is received by RDP from LAN/WAN port or WLAN-Dongle.
2. Packet is received in the DDR memory buffer.
3. RDP processes the packets and makes the forwarding decision (flow-miss).
4. RDP sends the packet to the Host CPU Network driver.
5. Network driver receives the packet and forwards it to the DHD Driver (either through Linux Kernel, or Flow-cache).
   5.1 DHD determines that the flow-ring is managed by RDP, and forwards the packet to the RDP for transmission.
6. RDP updates the TX_POST descriptor in the flow-ring with required information.
7. RDP notifies the Dongle about the packet.
8. Dongle reads the TX_POST descriptor from flow-ring and gets the required information.
9. Dongle grabs the packets from DDR.
10. Dongle transmits the packets towards WiFi-Client.
11. Dongle updates the descriptor with transmitted packet buffer information in TX_CMPL Ring.
12. Dongle notifies the RDP about the TX_CMPL ring update.
13. RDP takes the packet buffer information from the TX_CMPL ring.
    13.1 RDP sends the TX_CMPL information to DHD.
14. DHD frees the packet buffer back to the packet pool (from wherever it came from).

Flow-Hit Case

The differences between the offload and the non-offload flow-hit case are highlighted in Figure 7.

Figure 7: RDP Managed Flow-Hit Case



1. A packet is received by RDP from LAN/WAN port or WLAN-Dongle.
2. The packet is stored in the DDR memory.
3. RDP processes the packet and makes the forwarding decision (flow-hit).
4. RDP updates the TX_POST descriptor in the flow-ring with required information.
5. RDP notifies the Dongle about the packet.
6. Dongle reads the TX_POST descriptor from flow-ring and gets the required information.
7. Dongle grabs the packets from DDR.
8. Dongle transmits the packets towards WiFi-Client.
9. Dongle updates the descriptor with transmitted packet buffer information in TX_CMPL Ring.
10. Dongle notifies RDP about the TX_CMPL ring update.
11. RDP takes the packet buffer information from the TX_CMPL ring.
12. RDP frees the packet buffer back to the packet pool (from wherever it came from).

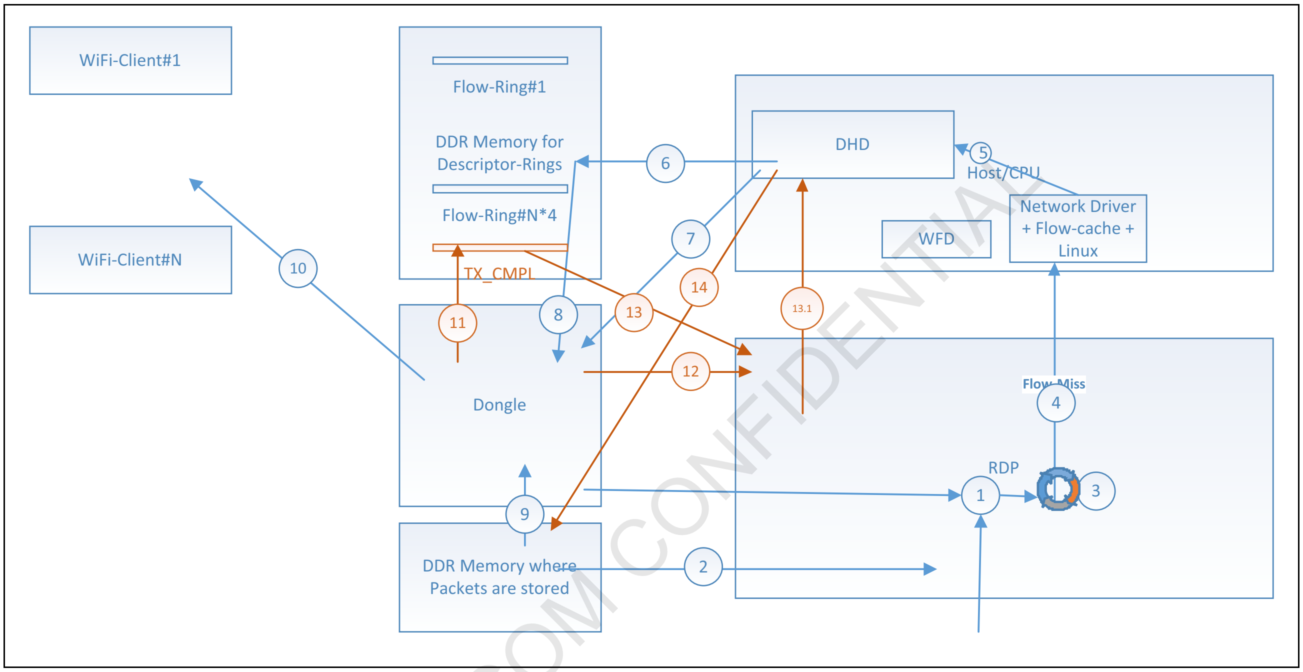
DHD Managed Flow-Rings

This case illustrates the data path flow for two scenarios, flow-miss and flow-hit, when the Flow-ring, to where the packet must to be transmitted, is managed by the DHD.

Flow-miss Case (N > 0)

The differences between the offload and the non-offload flow-miss scenario are highlighted in Figure 8.

Figure 8: DHD Managed Flow-Miss Case (N > 0)



1. A packet is received by RDP from LAN/WAN port or WLAN-Dongle.
2. The Packet is received in the DDR memory buffer.
3. RDP processes the packets and makes the forwarding decision (flow-miss).
4. RDP sends the packet to the Host CPU Network driver.
5. Network driver receives the packet and forwards it to DHD Driver (either through Linux Kernel, or Flow-cache).
6. DHD updates the TX_POST descriptor in the flow-ring with required information.
7. DHD notifies the Dongle about the packet.
8. Dongle reads the TX_POST descriptor from flow-ring and gets the required information.
9. Dongle grabs the packets from DDR.
10. Dongle transmits the packets towards WiFi-Client.
11. Dongle updates the descriptor with transmitted packet buffer information in TX_CMPL Ring.
12. Dongle notifies the RDP about the TX_CMPL ring update.

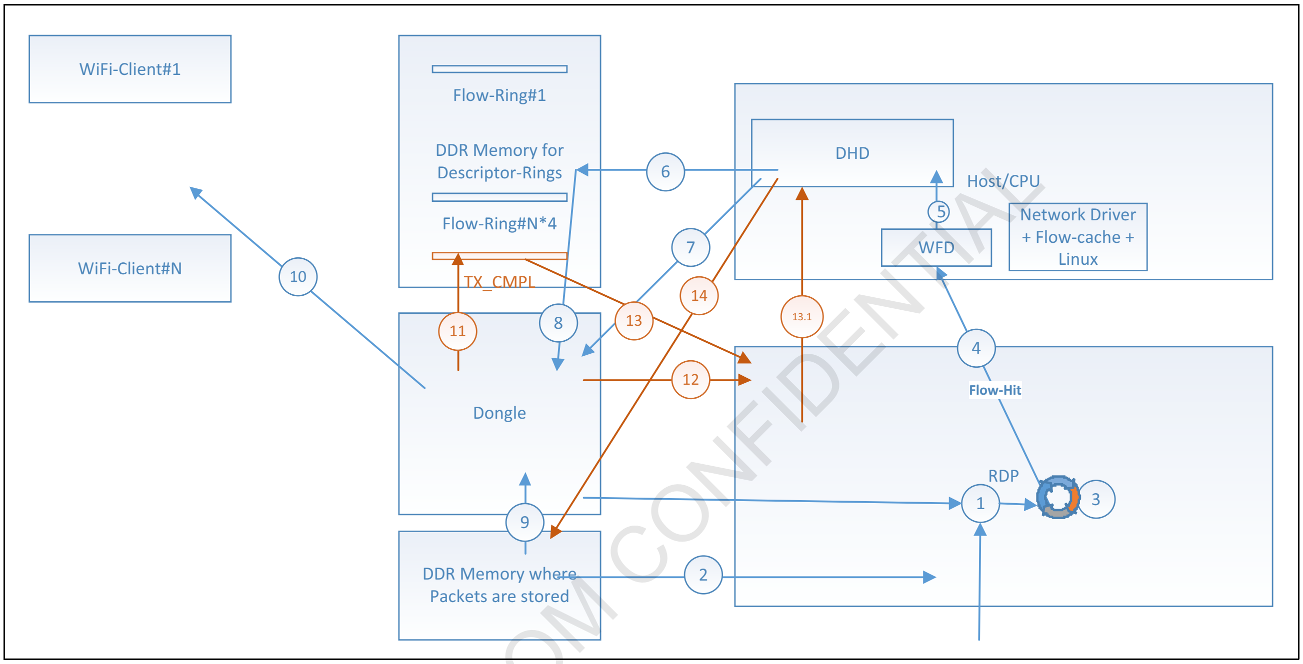
13. RDP takes the packet buffer information from the TX_CMPL ring.
13.1 RDP sends the TX_CMPL information to DHD.

14. DHD frees the packet buffer back to the packet pool (from wherever it came from).

Flow-Hit Case (N > 0)

The differences between the offload and the non-offload flow-hit scenario are highlighted in Figure 9.

Figure 9: DHD Managed Flow-Hit Case (N > 0)



1. Packet is received by RDP from LAN/WAN port or WLAN-Dongle.
2. Packet is stored in the DDR memory.
3. RDP processes the packets and makes the forwarding decision (flow-hit).
4. RDP sends the packet to Host CPU WFD.
5. WFD receives the packet and forwards it to DHD Driver.
6. DHD updates the TX_POST descriptor in the flow-ring with required information.
7. DHD notifies the Dongle about the packet.
8. Dongle reads the TX_POST descriptor from flow-ring and gets the required information.
9. Dongle grabs the packets from DDR.
10. Dongle transmits the packets towards WiFi-Client.
11. Dongle updates the descriptor with transmitted packet buffer information in TX_CMPL Ring.
12. Dongle notifies RDP about the TX_CMPL ring update.

13. DHD takes the packet buffer information from the TX_CMPL ring.
13.1 RDP sends the TX_CMPL information to DHD.

14. DHD frees the packet buffer back to the packet pool (from wherever it came from).

No RDP Managed Flow-rings

When RDP is not managing any flow-rings (N = 0; meaning that the DHD memory in boot loader was set to 0), TX_CMPL ring is also managed by DHD.

In such a case, the data path flow is the same as "DHD Managed Ring" except that TX_CMPL signal is handled by DHD directly (instead of RDP).

IPTV Multicast Handling

Multicast traffic acceleration/offload takes the same data path as unicast i.e., depending on the destination flow-ring, either it takes the WFD (flow-ring managed by DHD) or the direct RDP path (flow-ring managed by RDP.)

When multiple clients join the same stream and the flow-rings are managed by DHD, RDP only forwards one packet per SSID to DHD and the replication to all the clients is done by DHD.

Multicast flows are configured (not learned by default) in flow-cache/RDP, therefore, multicast traffic does not go through Linux iptable/ebtable rules/classification. By default all WLAN multicast traffic will be mapped to the Video Access Category Flow-ring (AC_VI).

IPTV Multicast traffic priority can be changed by setting the system level global multicast precedence. This is set using WebGUI and is displayed in the "cat proc/net/igmp_snooping" output. When global multicast precedence is set, it applies to all IPTV multicast traffic (including LAN clients).

Notes: Multicast acceleration through RDP is only supported for WAN to LAN/WLAN traffic. LAN/WLAN to LAN/WLAN multicast acceleration is done by flow-cache in the Host/CPU only.

If the WMF (Wireless Multicast Forwarding) feature is NOT enabled for a given SSID/BSS, all multicast traffic towards clients on that SSID/BSS will go through the WFD acceleration path.

Inter/Intra BSS and Inter Radio Traffic Handling

Inter/Intra BSS and Inter Radio traffic will always be accelerated using one of the following acceleration paths.

- Non-offload builds
  - Flow-cache acceleration applies.
- Offload builds
  - RDP accelerates all the received traffic from Dongle as below:
    - RDP forwards the traffic through the WFD path if TX flow-ring is DHD managed or if the destination is NIC WLAN.
    - RDP directly forwards the traffic (without host involvement) if TX flow-ring is managed by RDP.
  - For all other scenarios Inter/Intra BSS and Inter Radio traffic are learned and accelerated by flow-cache.

Flow-Ring Allocation Controls

The following controls are provided for configuring the N and M flow rings:

- Reserved memory (through boot loader)
- Flow-Ring Profile
  – Flow ring size (max. items)
  – Flow ring allocation priority
  – Flow ring selection policy

All these controls are set per dongle radio.

Note: It is recommended to select proper flow-ring controls based on product requirements. However, carefully consider the impact and implications before changing the default values.

Radio Number Assignment

During radio number assignment, dongle cards are scanned first. The radio number assignment starts from 0 and is incremented for each additional dongle card.

Note: NIC cards are scanned after dongle cards.

The following is the scan sequence:

- PCIe port #0
  – Port#0 … Port#N, if connected through a PCIe bridge
- PCIe port #1 (if present)
  – Port#0 … Port#N, if connected through a PCIe bridge

The WLAN interface number will be same as the DHD dongle radio number.

Reserved Memory

See “Memory Reservation” on page 9.

Flow Ring Profile

The Flow ring profile controls the maximum items and allocation weight and priority for each WLAN access category (AC) to determine if the created flow-ring should be managed by the RDP or DHD.

Max items control determines how much memory is needed for flow rings. Allocation weight is used to allocate RDP accelerated flow rings to an access category.

There is one flow ring profile per radio. The profile information is stored in the Persistent Scratch Pad (PSP) memory and is accessed using the DHD command line application (dhdctl).

A profile is specified using a profile identifier. DHD supports five built-in fixed profiles and three (one per radio) user defined profiles. Each profile has the following format:

```
<id>  <weight>:<max_items>  <weight>:<max_items>  <weight>:<max_items> …
```

where:

<id> = Refers to the Profile identifier

<weight>:<max_items> = the weight and max items combination, which is specified per access category (AC_BK, AC_BE, AC_VI, AC_VO), and for Broadcast/Multicast (BCMC) flow-ring, per interface.

Based on the allocation weight, DHD starts allocating RDP managed flow rings to a particular access category. Any time during the allocation process, if there is no more reserved memory available, the remaining flow rings are DHD managed.

Allocation weight specifies the number of flow-rings that are allocated during each AC (Access Category) scan.

Table 2: Flow Ring Profile Allocation Weight

Weight	*Description*
-1	Highest weight. All flow-rings are allocated at once.
0	Reserved for future use.
1	One flow-ring is allocated for each AC per scan. The scan is repeated until all required flow-rings are allocated.
2..N	“N” flow-ring is allocated for each AC per scan. The scan is repeated until all required flow-rings are allocated.

Table 3: Flow Ring Profiles

Field	*Values*	*Value Description*	*Default Values*				
<id>	0–6	0: Radio#1 user defined profile	0 1:1024	-1:2048	-1:1024	-1:512	1:512
		1: Radio#2 user defined profile	1 1:1024	-1:2048	-1:1024	-1:512	1:512
		2: Radio#3 user defined profile	2 1:1024	-1:2048	-1:1024	-1:512	1:512
		3: built-in profile (AC_BE, AC_VI priority)	3 1:1024	-1:2048	-1:1024	1:512	1:512
		4: built-in profile (each AC allocate first)	4 -1:1024	-1:2048	-1:1024	-1:512	1:512
		5: built-in profile (all AC same priority)	5 1:1024	1:2048	1:1024	1:512	1:512
		6: built-in profile (ascending AC priority)	6 1:1024	2:2048	4:1024	8:512	1:512
		7: built-in profile (all flow-rings same size, and priority)	7 1:2048	1:2048	1:2048	1:2048	1:2048
<weight>	-1, 1, 2, ….N	-1: Allocate all flow rings first 1-N: Allocate ‘N’ flow rings per AC per scan.	Based on profile id				
<max_items>	128…..N	N: size of ring for that AC	AC_BK:1024	AC_BE: 2048	AC_VI:1024	AC_VO:512	BCMC:512

Flow-Ring Profile User Interface

"dhdctl" provides IOCTL's to list and sets the flow ring profile for each radio. If no profile is stored in PSP, then DHD uses the default built-in profile id 3.

To view the list of available profiles and the current active profile use the command shown below. The active profile of the radio is the id marked with an asterisk "*".

```
# dhd -i <IF#> flowring_profile
[id] [ ac_bk ] [ ac_be ] [ ac_vi ] [ ac_vo ] [ ac_bcmc]
  0   01:1024   -1:2048   -1:1024   -1:0512   01:0512
  1   01:1024   -1:2048   -1:1024   -1:0512   01:0512
  2   01:1024   -1:2048   -1:1024   -1:0512   01:0512
  3   01:1024   -1:2048   -1:1024   -1:0512   01:0512
 *4   -1:1024   -1:2048   -1:1024   -1:0512   01:0512
  5   01:1024   01:2048   01:1024   01:0512   01:0512
  6   01:1024   02:2048   04:1024   08:0512   01:0512
  7   01:2048   01:2048   01:2048   01:2048   01:2048

Where
<IF#> wl interface of radio (wl0, wl1, wl2)
```

Example: To change the user defined profile settings:

```
# dhd -i <IF#> flowring_profile <p> 1:1024 1:2048 1:1024 1:512 1:512

Where
<IF#> wl interface of radio (wl0, wl1, wl2)
<p>   user profile id (0-2)
```

Example: A profile can be set active using a profile id or user profile id with <weight><items> settings. The code shown below sets the profile using only the profile id, with no profile parameters.

```
# dhd -i <IF#> flowring_profile <p>

Where
<IF#>  wl interface of radio (wl0, wl1, wl2)
<p>    profile id (0-7)
```

Note: A system reboot is required for these changes to become active.

Flow-Ring Policy Control

Flow ring policy is used to select the RDP offload flow ring or DHD managed flow ring at run time. This control is provided mainly to test the N+M flow ring feature and for tuning purpose. The following table shows the available flow ring selection policies. At a time one of the policies will be active.

Table 4: Flow-Ring Policy Control

Policy Name	Policy ID	Values	Value Description	Comments
Global	0	0–1	0: Prefer DHD flow ring first 1: Prefer RDP offloaded flow ring first	This is the default profile id with a default value of 1.
Intfidx	1	0..N N < 16	Prefer RDP offloaded flow rings for all stations connected to interface index <= n	43602 dongle N<4 4366 dongle N<16
Clients	2	0..N N < 128	Prefer RDP offloaded flow rings for all stations until N clients	43602 dongle N<32 4366 dongle N<128
aclist	3	<ac>:<0/1>	Prefer DHD/RDP for each AC specified in the list	<ac> values 0: ac_bk 1: ac_be 2: ac_vi 3: ac_vo 4: ac_bcmc
Maclist	4	<mac>	Prefer RDP flow rings for all the stations from the MAC address list	MAC address format xx:xx:xx:xx:xx:xx Up to four MAC addresses can be specified.
d11ac	5	0/1	1: Prefer RDP flow rings for all 11AC stations, other stations DHD flow rings 0: Prefer DHD flow rings for all 11AC stations, other stations RDP flow rings	Not currently available. For future implementation.

If no policy is stored in PSP, DHD uses a default user policy id#0.

“dhdctl” provides IOCTL’s to list and set the flow ring policy of each radio.

To get the list of policies available and current active policy, use the command below. The active policy of the radio is the id shown with an asterisk “*”.

```
# dhd -i <IF#> flowring_policy
[ Name  ] [id] [Policy]
  global    0
 intfidx    1
 clients    2
  aclist    3
 maclist   *4   00:90:4c:0f:50:93
   d11ac    5

where
<IF#> wl interface of radio (wl0, wl1, wl2)
```

Example: Set the global policy for all RDP flow rings.

```
# dhd -i <IF#> flowring_policy 0 1

Where
<IF#> wl interface of radio (wl0, wl1, wl2)
```

Example: Set the MAC address list policy with three stations.

```
# dhd -i <IF#> flowring_policy 4 00:90:4c:0f:50:91 00:90:4c:0f:50:92 00:90:4c:0f:50:93

Where
<IF#> wl interface of radio (wl0, wl1, wl2)
```

Example: Set the AC list policy BE, VI to select RDP offload and the others to select DHD.

```
# dhd -i <IF#> flowring_policy 3 0:0 1:1 2:1 3:0

Where
<IF#> wl interface of radio (wl0, wl1, wl2)
```

Note: A new setting is effective for all clients that connect after the set. It is recommended to disconnect all clients before setting, or reboot the system after setting.

Accelerated Data Path Overview

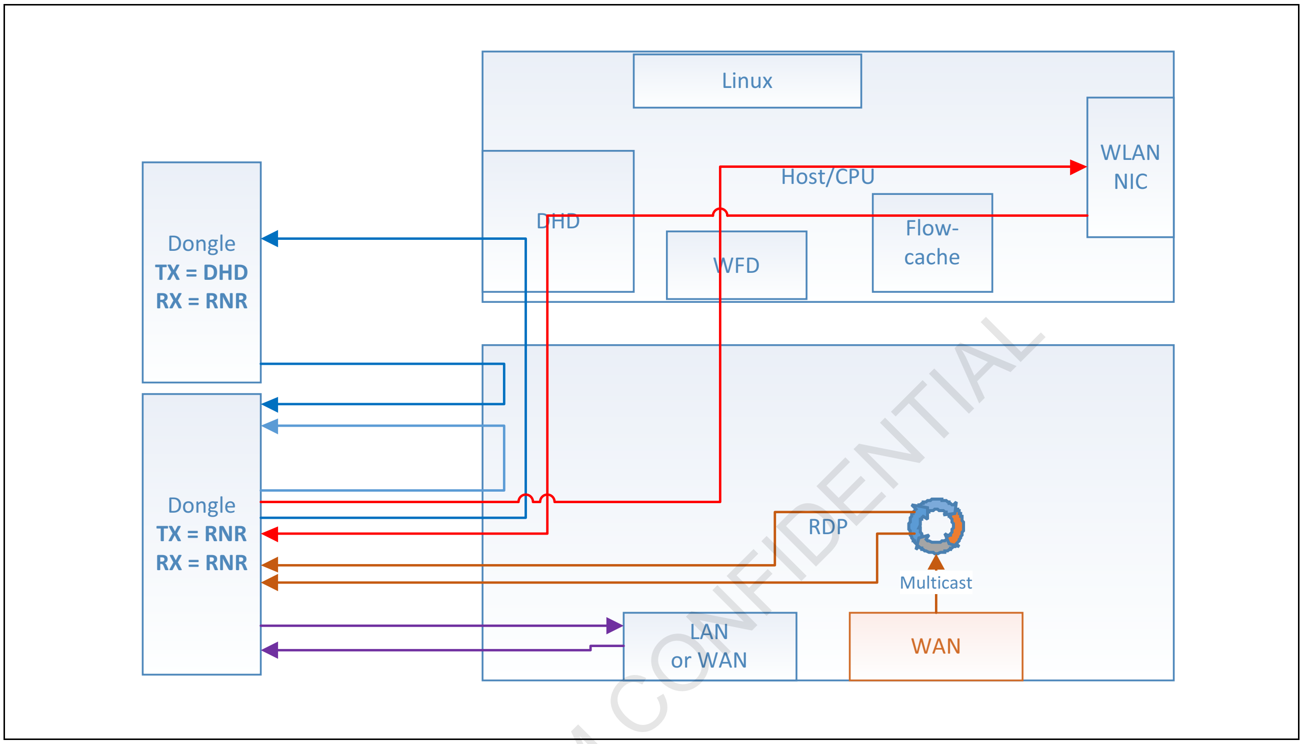
All the possible accelerated data path flows for Dongle when offload is enabled in the build are listed below. In the figures shown in this section, arrows are traces from RX to TX. The arrows show what path the packet will take for that kind of traffic scenario.

- Downstream Multicast with replication
- LAN/WAN to/from Dongle
- Dongle to/from Dongle
- Dongle to/from NIC

Traffic Flows To/From Fully Offloaded Dongle

Figure 10 shows the traffic flow when both RX and TX paths from/to Dongle are going through the RDP.

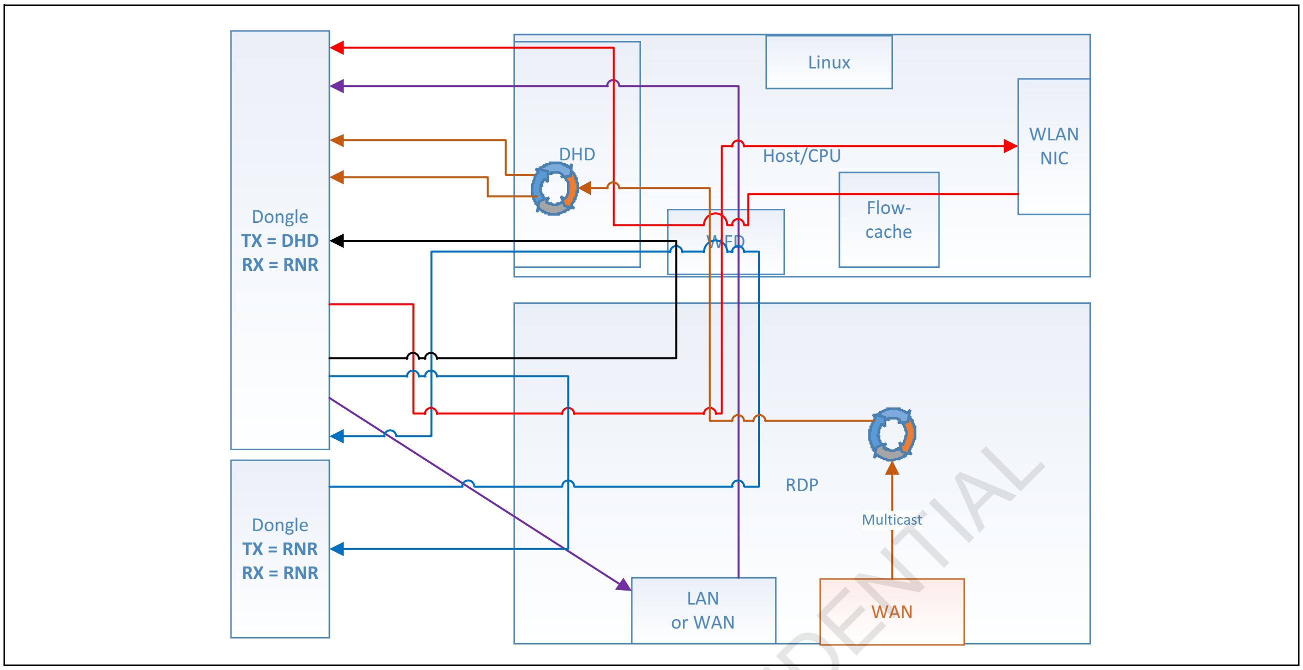
Figure 10: Traffic Flows To/From Fully Offloaded Dongle



Traffic Flows To/From RX-Only Offloaded Dongle

Figure 11 on page 29 shows the traffic flow when the RX path goes through RDP and TX goes through DHD. This scenario occurs if the user does not allocate any memory for that Dongle during system boot up (through boot loader).

Figure 11: Traffic Flows To/From RX-Only Offloaded Dongle



DHD Offload Debugging

While debugging any issues related to the DHD Offload, the user should look for the following information:

1. Check the amount of memory allocated through the boot loader (see “Memory Reservation” on page 9“ for details).
2. Make sure that the correct memory is allocated during system boot (as configured in the boot loader). Look for these outputs during system boot to find out the actual memory allocated.

```
creating a MT_MEMORY_NONCACHED device at physical address of 0x0d200000 to virtual address at
0xcd200000 with size of 0x2c00000 byte for RDPA tm
creating a MT_MEMORY_NONCACHED device at physical address of 0x0c000000 to virtual address at
0xcc000000 with size of 0xe00000 byte for DHD dhd0
creating a MT_MEMORY_NONCACHED device at physical address of 0x0b800000 to virtual address at
0xcb800000 with size of 0x700000 byte for DHD dhd1
```

3. Get the output of the following commands:
   - “dhdctl –i wl0 dump”
   - “dhdctl –i wl1 dump”

```
DHD Runner:
  Status     : radio#  0 tx_Offl 1  rx_Offl 1
  Profile    : prfl_id 3 id_valu  01:1024   -1:2048   -1:1024   -1:0512   01:0512
  Policy     : plcy_id 0 id_valu 1 (HW)
  tx_flowring: [ac_bk] sw 0 hw 64 [ac_be] sw 0 hw 64 [ac_vi] sw 0 hw 64 [ac_vo] sw 0 hw 64 [bc_mc]
sw 0 hw 8
  h2r_notif  : tx_post 74 rx_cmpl 5 tx_cmpl 22
  r2h_req    : rx_cmpl 11 tx_cmpl 0 wk_dngl 85
```

```
CtrlPost: RD 37 WR 37
CtrlCpl:  RD 85 WR 85
RxPost:   RBP 0 RD 576 WR 10
RxCpl:    RD 11 WR 11
TxCpl:    RD 73 WR 74
active_tx_count 0  pktidmap_avail -1  rxbufpost 0
dhd cumm_ctr 0
Num: HW Flow If Prio :Dest_MacAddress: Qlen CLen L2CLen Pkts  Overflows  RD  WR  Acked tossed noack
  0.  1   2  0    0 33:33:00:00:00:16    0    0  0        0        0      69   69    NA     NA    NA
  1.  1  202  0   0 ac:9e:17:45:a1:ac    0    0  0        0        0       5    5    NA     NA    NA
```

The code above the configuration information as follows:

- The Flow-Ring Profile used.
- The policy used.
- RX_POST, RX_CMPL, TX_CMPL and TX_POST ring current Read/RD and Write/WR pointers.
- Provides information about the Flow-Rings:
  – Flow-Ring-ID “Flow” (e.g., 202)
  – Is the flow-ring managed by hardware (i.e., RDP) or software (i.e., DHD) – shown in the “HW” column.
  – Priority (or Access Category) of the flow-ring – shown in the “Prio” column.
  – MAC Address of WLAN Client using that flow-ring – shown in the “Dest MacAddress: column.

4. If the issue is related to unicast acceleration, dump the RDP unicast flows.

5. If the issue is related to multicast acceleration, dump the running multicast flows along with WLAN multicast flow information using command “bs /b/e wlan_mcast”

Connecting everything®

Broadcom Corporation
5300 California Avenue
Irvine, CA 92617

CPE-AN2300-R February 16, 2016

Phone: 949-926-5000
Fax: 949-926-5203
E-mail: info@broadcom.com
Web: www.broadcom.com