目录

[General description 2](#_Toc40984819)

[缩写词/术语 3](#_Toc40984820)

[WiFi硬件加速相关技术问题 4](#_Toc40984821)

[PCIe 界面描述符 4](#_Toc40984822)

[帧结构 4](#_Toc40984823)

[Queue 4](#_Toc40984824)

[Forwarding Table 5](#_Toc40984825)

[Re-order buffer 5](#_Toc40984826)

[系统资源 5](#_Toc40984827)

[组播 5](#_Toc40984828)

[驱动代码分析 5](#_Toc40984829)

|  |  |  |
| --- | --- | --- |
| 作者 | 蓝麒 | |
| 版本信息/修改记录 |  | |
| 版本 | 描述 |
| 2020-05-22 | 1.0 | 初始版本 |
|  |  |  |

## General description

随着WiFi吞吐量的增长，给CPU处理报文的收发带来很大的负担，体现在CPU loading 高达100%，无法获得最大的吞吐量。业内针对此问题引入offload方案，将处理报文的部分甚至全部工作。方案大体上分为 F/W offload 和 H/Woffload。下面列举已知的方案。

* Case1: F/W offload

常见的USB/PCIe wifi card 里面运行了firmware, firmware 执行报文收发路径的一些处理。

* Case2: H/W offload + F/W offload

Quantenna 在查找转发表、转发和队列管理上使用硬件实现，而使用独立的MuC执行报文封装聚合和发送调度工作，又使用另一颗MuC执行解封装、解聚合和re-order 的处理。

* Case3: Full H/W offload

目前高通IPQ807X 平台在 RX 方向实现了完全的硬件处理。

本文档对Galachip 实现硬件加速设计进行顶层描述，尽可能涵盖实现目标所需要考虑的各个组件功能。各个组件的细节可能由另外的相关文档专门描述。

## 缩写词/术语

|  |  |
| --- | --- |
| DA | Destination *MAC* Address |
| SA | Source *MAC* Address |
| dip, sip | Destination IP address, source IP address |
| Port | 在本文需要根据前后上下文，可能指物理的或虚拟的端口，也可能是TCP/UDP port. |
| VAP | virtual AP. 软件上的虚拟AP，和 multi-BSS 相关  有个OS struct net\_device{} 与之对应。 |
| Node | 代表一个station. 对端station成为 peer node. 本地成为 local node. 每个VAP 可以看做一个station, 即每个VAP 一个local node。 |
| TID | Traffic identification. 支持QoS 的station(node) 具有16个TID, 用以区分steam 的优先级。某些实现中会额外添加第17个TID，用于发送管理报文。 |
| Radio device | 一个Radio设备工作于2.4GHz band 或者 5GHz band。 可支持多个 VAP。有个OS struct net\_device{} 与之对应。 |
| 高优先级帧 | EAPOL, DHCP, ARP, QoSNULL |
| WMAC | Wireless MAC, 属于硬件范畴 |
| 异步/同步 | 本文中同步是指 针对一个报文发送的过程的顺序执行。异步是指针对一个报文发送的过程发生中断。中断往往是由于node 状态、由QoS 引入的queue 或者AMSDU/AMPDU 聚合导致暂缓发送。 |
| DPI | Data path interface  硬件提供给软件的接口，用于灵活设置一些数据报文的datapath. 例如某些bit 匹配目标模式，将报文送至CPU处理 或者丢弃。 |
| packet/frame | 本文没有区分 packet, frame，统一理解为 报文。本文也混用报文，封包这种用语。 |
| NAWDS | Non-association WDS. 是比较早期流行的一种无线桥模式。目前比较多用无线桥是 Repeater 模式。 |
| 组播 | Multicast. 也称多播。本文中组播包含了广播，如有排除广播的情况需特别指出。 |
| Datapath | 数据报文从设备入口 到设备出口需要经过许多处理节点。所有这些节点连接构成的流程称为datapath. |
| SW datapath | 针对单个数据报文的处理需要CPU参与的datapath |
| HW datapath | 针对单个数据报文的处理不需要CPU参与的datapath |

## WiFi硬件加速相关技术问题

### PCIe 界面描述符

系统完成PCIe 枚举，PCIe 驱动建立通信通道。

WiFi Chip 通过PCIe 上报给SoC 的数据有数据报文和控制数据。这个可以通过分析驱动获取跟多信息。

SoC 下传给 WiFi Chip 的数据有数据报文和控制数据。数据报文和控制数据可能使用同一的描述符。这个可以通过分析驱动获取更多信息。

\*\* 所提的描述符可能也没有那么复杂，应该就是一个DMA操作描述符。

### 帧结构

列举所有系统需要处理的帧类型，帧结构，帧长度范围，包含：

1. LLC 交付给WiFi MAC 的帧类型。

Ethernet II, Ethernet SNAP

1. AMSDU, MPDU
2. AMPDU (次要，一般来说是WiFi chip 处理)

### Queue

理论上需要执行的Queue:

1. AMSDU TX Queue
2. Legacy power save TX queue (区分data, mgmt, mcast)
3. UAPSD power save TX queue
4. Fragmentation TX Queue
   1. 这个太复杂了，考虑实际需求和意义不大，可以不做这个
5. Per-node Per-tid TX Queue
6. PCIe TX ring buffer
7. PCIe RX ring buffer
8. Per-node Per-tid RX queue
9. De-Fragmentation RX Queue
   1. 自身可能不会分片，但是不能假定对方不会分片
10. Re-order buffer

实际操作需要执行的Queue?

### Forwarding Table

发往无线的报文需要通过DA 查找peer node. 收到的报文也需要通过SA 查找peer node. 软件往往实现hash查找表。这个转发表与 bridge 的转发表很类似，每一个node可以理解为一个port.

硬件需要实现这个转发表，提供接口给其他硬件模块和软件访问。转发表的维护由软件实现。

### Re-order buffer

### 系统资源

需要分析实现需要的 memory 资源。并具体划分各个部分的功能和占用的memory空间大小。

### 组播

二层组播 与 三层组播（IGMP）

### 驱动代码分析

硬件加速的实现是基于软件。分析驱动以便对驱动进行深度的客制化。

一些特殊帧需要由驱动处理。软件能够维护硬件中的流。