

## 在 Keystone SoC 上使用 Pktlib 加速 LTE 层 2 处理

Adam Yao

Multi-core DSP / FAE

### 摘要

层 2(L2) 在长期演进计划(LTE) 中由分组数据汇聚协议(PDCP), 无线链路控制协议(RLC) 和介质访问控制协议(MAC) 组成。Pktlib 是 TI 公司提供的基于其最新的 Keystone SoC 的软件开发包(PDK) 中的一个组件, 可以灵活地实现包的申请, 合并, 分割, 拷贝, 释放。通过对 L2 处理特性的研究, 提出了采用 Pktlib 在 Keystone SoC 上实现 L2 处理的方法。其实现简单, 效率高, 能广泛应用于 LTE 基站中。

### 文档修改记录

Version	Date	Author	Notes
1.0	Aug 2013	Adam Yao	First release
2.0	Sep 2013	Adam Yao	Rerised based on received comments

## 目录

1. LTE 层 2 处理的特点 .....	3
2. QMSS 和 Pktlib .....	4
3. 基于 Pktlib 的层 2 内存管理 .....	5
4. 基于 Pktlib 的层 2 零拷贝数据处理 .....	6
4.1 包的合并 .....	6
4.2 包的拷贝 .....	7
4.3 包的分割 .....	8
5. Pktlib 的性能 .....	8
6. 结束语 .....	9
参考文献 .....	9

## 图

图 1. LTE 层 2 下行结构 .....	3
图 2. 硬件队列和包格式图 .....	4
图 3. 基于 Pktlib 的层 2 下行内存管理 .....	6
图 4. 包的合并 .....	7
图 5. 包的拷贝 .....	7
图 6. 包的分割 .....	8

## 表

表 1. Pktlib 主要函数和功能 .....	4
表 2. Pktlib 主要函数的性能 .....	8

## 1. LTE 层 2 处理的特点

无线长期演进计划 (LTE) 是第三代合作伙伴计划(3GPP)在 2004 年启动的未来移动通信标准研究。3GPP 在 2008 年 12 月发布了第一个 LTE 标准 Release8, 到目前为止已经更新到 Release11, 其中 Release10 和 Release11 也被看成是 LTE-Advanced 的标准。

层 2 在 LTE 中位于层 1(物理层)和层 3 (RRC, RRM, S1-AP 等协议) 之间, 它的作用是屏蔽层 1 的细节, 向层 3 提供可用的服务接入点(SAP)。LTE 中的层 2 在下行方向上依次通过 PDCP, RLC 和 MAC 3 个子层 (如图 1)。每个子层处理之前的数据叫做服务数据单元(SDU), 处理之后的数据叫做协议数据单元(PDU), PDU 是对 SDU 的数据加以一些特殊的处理, 同时添加一个本子层特定的协议头所形成的。上行方向处理与下行类似, 只是 3 个子层通过的顺序相反。PDCP 层主要完成数据包头的压缩与解压缩 (ROHC), 数据内容的加密与解密, 以及向上层 PDU 的顺序递交。RLC 层根据 MAC 层的调度结果对 PDCP 的 PDU 进行拆分和重组, 同时实现 ARQ 的功能。MAC 层主要实现与调度和 HARQ 相关的功能, 包括逻辑信道向传输信道的映射, 对 RLC PDU 的复用与解复用, HARQ 纠错与重传等。

从实现的角度看, 一般将 MAC 层中的调度和优先级管理作为一个单独的模块, 称为调度器(Scheduler)。调度器要根据空口的状态和网络当前要发送的数据量以及优先级, 对空口资源进行实时的分配, 是一个计算密集型的模块。将 PDCP, RLC 和 MAC 层中的复用组包功能定义为另一个模块, 称为用户面, 这部分主要根据协议的规定完成加头, 去头, 合并, 拆分, 加密, 解密等处理, 是一个数据吞吐密集型的模块。因为 LTE 采用全 IP 的网络架构, 所有用户面处理的包都是 TCP/IP 的包, 本文所介绍的 Pktlib 就是针对这种包类型, 结合 TI 公司 Keystone SoC 中的硬件队列管理器(QMSS)所开发的一个软件包, 它能提高层 2 用户面的处理效率, 实现基于硬件队列的内存管理和真正的数据零拷贝, 同时接口简单, 便于用户移植。

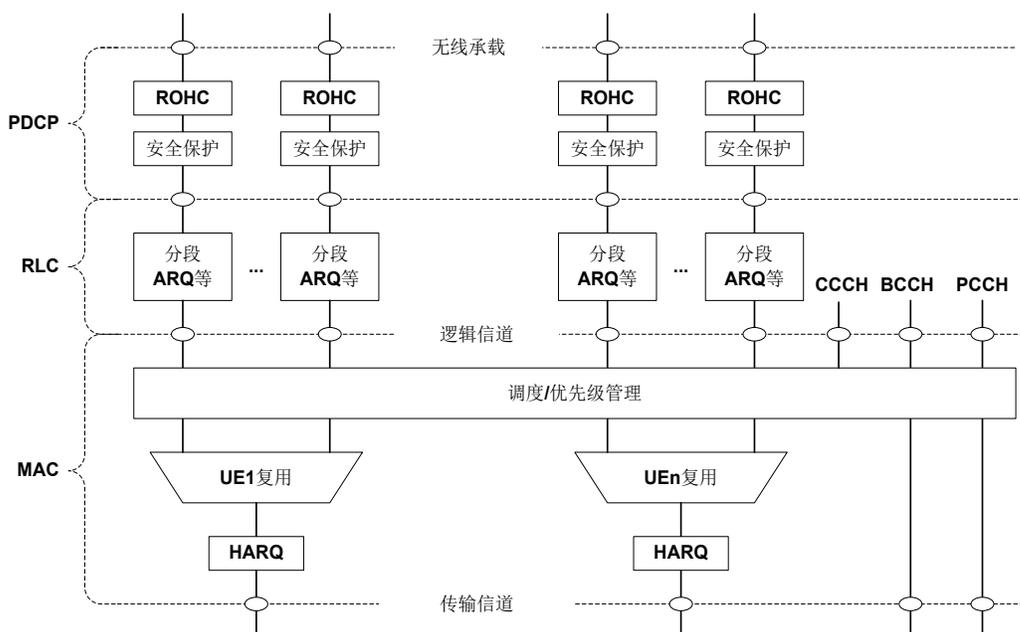


图 1. LTE 层 2 下行结构

## 2. QMSS 和 Pktlib

QMSS 是 Keystone 上集成的硬件队列管理器。每一个硬件队列都由一组特定的寄存器控制，用户通过读写寄存器完成包的入队和出队。队列中的包要满足 QMSS 特定的格式要求。一个包由包描述符(PD)和对应的缓存组成，包的所有属性信息都保存在 PD 中，包中的缓存可以分片，不同的分片之间地址可以不连续，每一个分片的属性保存在对应的缓存描述符(BD)中。满足这种格式的包可以压入硬件队列，同一个队列中包的前后链接关系由硬件维护。图 2 画出了硬件队列和其中的包的结构。在 Keystone 上，这种格式的包可以直接压入硬件加速器和外设（如以太网和 SRIO）的发送队列（这些发送队列也属于 QMSS 所管理的硬件队列），进行硬件加速处理或数据发送。

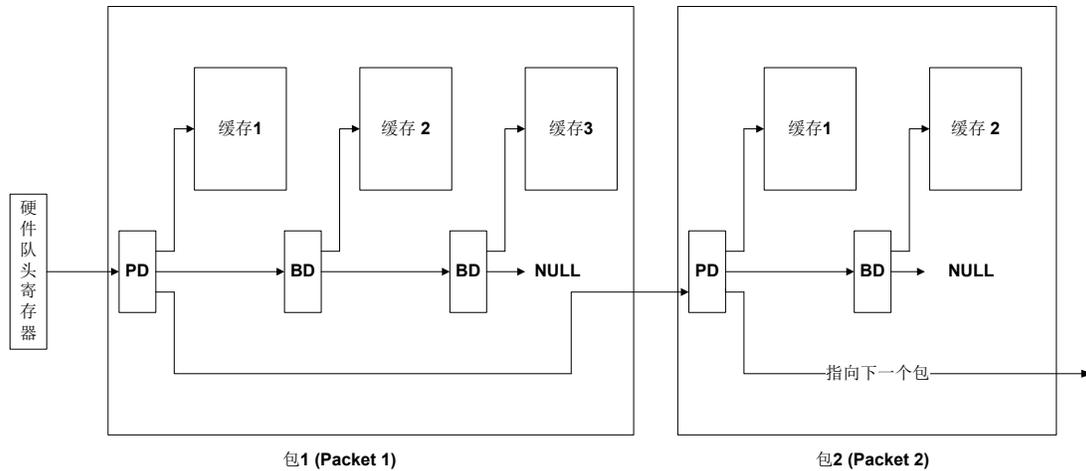


图 2. 硬件队列和包格式图

Pktlib 是 TI 基于 QMSS 封装的一个软件开发包，它采用 QMSS 的包作为基本的内存管理单元，将硬件队列虚拟为内存申请的堆，通过包的指针操作替代了内存的拷贝，拼接和分割。Pktlib 的主要函数接口和功能如下表。

表 1. Pktlib 主要函数和功能

Pktlib 函数	功能
Pktlib_createHeap	创建一个堆
Pktlib_findHeapByName	根据堆的名字，在本核定位并获得堆的句柄，用于定位由其他核创建的堆。
Pktlib_allocPacket	从堆上申请一个包
Pktlib_packetMerge	将两个包合并形成一个新的包
Pktlib_clonePacket	对一个包进行拷贝生成一个新的包
Pktlib_splitPacket	根据用户指定的分割字节位置，将一个包分割为两个包
Pktlib_freePacket	回收使用完毕的包

### 3. 基于 Pktlib 的层 2 内存管理

层 2 是一个基于包处理的模块，在处理中需要临时申请大量的内存，内存管理的效率是影响层 2 处理性能的关键因素。传统的层 2 处理一般直接采用操作系统的内存管理模块，完全通过软件来管理内存，这种处理方法在软件处理和加速器处理接口的地方必须做一次数据拷贝把加速器处理完的数据从加速器的缓存区中拷贝到软件的缓存中。这种拷贝在大量的包处理中极大地影响了效率。同时传统的内存管理方法因为无法通过软件直接保证多核互斥，在多核共享内存的管理上效率很低。

基于 Pktlib 的内存管理能解决传统内存管理的问题。以 LTE 下行处理为例，在 Keystone 上 PDCP 层的空口安全保护在安全加速器 (SA) 中完成，其后需要用软件完成 RLC 和 MAC 层的处理，然后将处理完的包交给物理层进行编码。层 2 和物理层的接口根据芯片功能划分有两种选择：物理层可以在其他芯片上完成（这时需要将层 2 处理完的数据打包通过以太网或 SRIO 发送出去），也可以在本芯片上完成（这时可以将层 2 处理完的数据直接发送给物理层的加速器 BCP）。层 2 软件和 SA, BCP, SRIO, EMAC 这些加速器或外设的接口都符合 QMSS 的包格式，并可以通过硬件队列来完成数据的传输。

图 3 画出了一个基于 Pktlib 的层 2 下行的内存管理架构。它的基本思想是在创建堆的时候将通过操作系统申请的固定长度的内存块直接挂接到 PD 上，然后压入专用的硬件队列（图 2 的 Mem FreeQ），这个硬件队列就相当于一个堆。应用在初始化完 Pktlib 的 Mem FreeQ 以后，通过调用 Pktlib\_allocPacket 函数从堆上申请（出队）内存包并将它压入 SA 的接收空闲队列 (Rx FreeQ)。当 SA 处理完一个 PDCP 包的空口加密工作后会把数据通过接收侧包 DMA (Rx Pkt DMA) 写入 Rx FreeQ 中包指向的内存，然后将包从 Rx FreeQ 出队，压入接收队列 (RxQ)。RLC 和 MAC 层软件接收到调度器的调度请求后，从 SA 的 RxQ 出队对应的包并进行 RLC 和 MAC 的组包工作。在这个处理过程中，因为涉及到包的分割，合并，添加包头的工作，RLC 和 MAC 模块会动态地申请新的内存，这时同样通过调用 Pktlib\_allocPacket 函数从 Mem FreeQ 中申请包。RLC 和 MAC 层处理完成后，软件可以将处理完的包直接压入和物理层接口的发送队列 (TxQ) 中，后面可以是通过加速器直接做物理层编码，也可以是通过接口发送数据。发送侧包 DMA (Tx Pkt DMA) 读入包之后，可以自动将包返回一个队列 (Tx ReturnQ)。Tx ReturnQ 的选择有两种情况需要分开考虑。对于可以立即释放的包（这个包中 PD/BD 所指向的缓存没有被其他的 PD/BD 指向），可以通过 QMSS 直接回收到 Mem FreeQ。还有一类包不能立即释放，这个包中内存同时还被另一个 PD 指向（这在包的拷贝，分割中经常出现）。对第二种类型的包，我们需要先让 QMSS 把包回收到一个临时的回收队列 (Tx ReturnQ) 中，然后调用 Pktlib\_freePacket 接口，Pktlib 会根据这个包的属性以及是否进行过拷贝，切割这些信息，决定是马上将包回收到 Mem FreeQ，还是等到所有指向同一个缓存的 PD 都使用完这个缓存后，再将包回收到 Mem FreeQ。从上面的描述我们可以看到，层 2 下行软件和处理流程前后的两个加速器/外设接口发生了数据交互，但是中间没有任何的数据拷贝，处理效率大大优于传统的方案。

Keystone 中的硬件队列可以自动完成多核访问的互斥，由此带来的好处是多核可以同时使用一个 Pktlib 创建好的堆，同时不需要加入效率较低的互斥锁保护。对于多核同时进行层 2 处理的方案，这一点也能在一定程度上节省软件的开销。

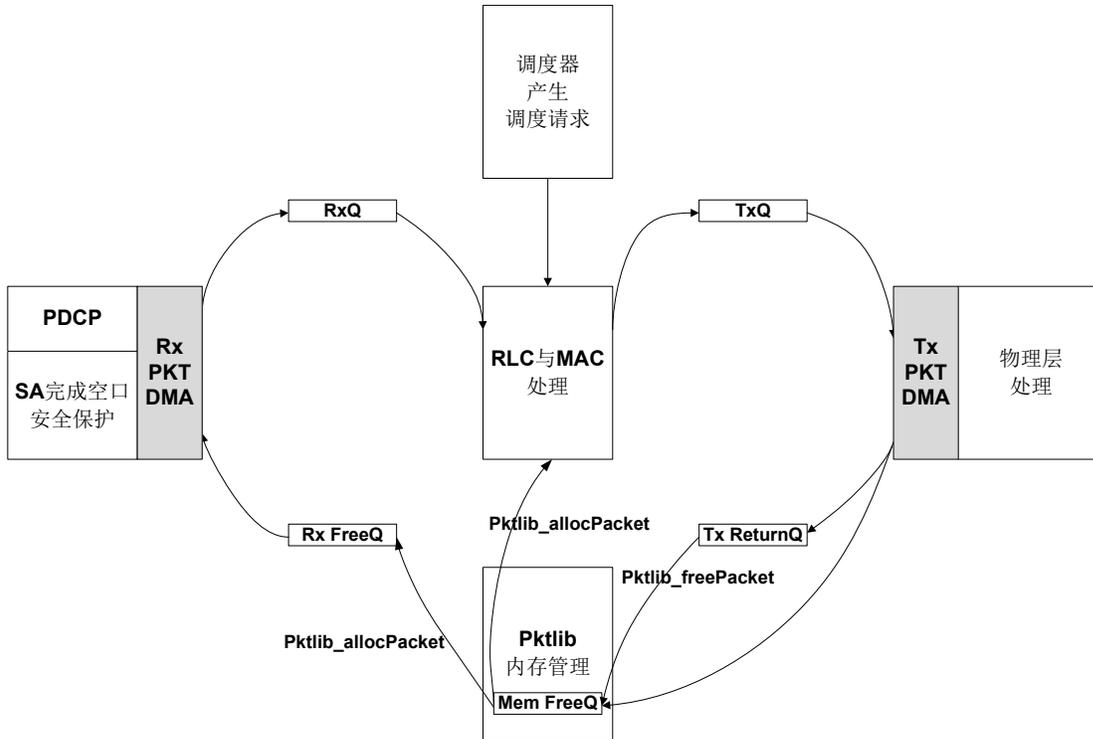


图 3. 基于 Pktlib 的层 2 下行内存管理

## 4. 基于 Pktlib 的层 2 零拷贝数据处理

层 2 处理中经过 PDCP 层处理的包保存在硬件队列中，RLC 和 MAC 模块需要根据调度器提供的调度结果对 PDCP 处理的包进行重新封装，在这个过程中需要添加 RLC 协议头，MAC 协议头，还需要对 PDCP 包进行切割，重组(因为调度结果是根据当前空口状态产生的，跟 PDCP 的包长不可能匹配)。传统的做法是采用内存拷贝，软件开销比较大。采用 Pktlib 可以通过相应的函数实现零拷贝，提高层 2 处理效率。

### 4.1 包的合并

层 2 处理中有很多包的合并操作，常见的场景有：

- (1) 在处理 PDCP，RLC，MAC 层协议的时候添加对应的协议头。
- (2) RLC 收到调度结果后需要将两个或多个 PDCP 的 PDU 打包为一个 RLC 的 PDU
- (3) MAC 复用需要将多个 RLC 的 PDU 打包为一个 MAC 的 PDU

如图 4 所示，通过调用 Pktlib\_packetMerge 函数可以将输入的两个 QMSS 格式的包合并为一个新的包。这一操作实际上是将包 1 的 Next 指针置为包 2 的 PD 地值，在合并的过程中没有任何的内存拷贝。

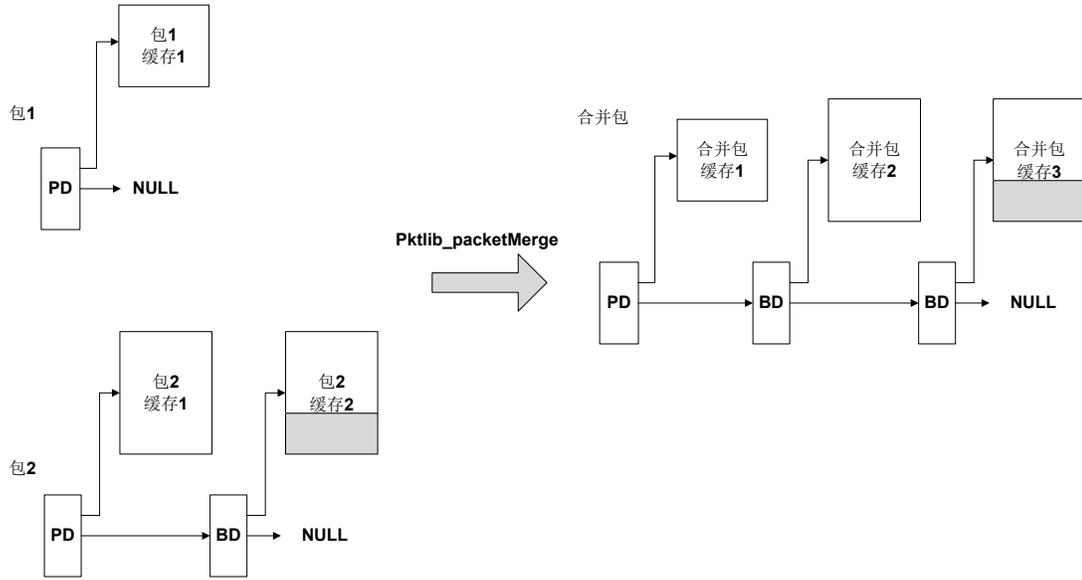


图 4. 包的合并

### 4.2 包的拷贝

层 2 处理中包的拷贝主要发生的 RLC 包进入重传缓冲区，RLC 协议在 AM 模式下对处理完成的包在提交给 MAC 进行处理的同时，还必须拷贝一份放入重传缓冲区。如果接收端发出了重传请求，下一次 RLC 会从重传缓冲区取出数据重发。只有在收到接收端的确认后，RLC 才能释放重传缓存区中的包。

如图 5 所示，Pktlib 在做包的拷贝时不是通过内存拷贝来实现的，而是通过将传入的一组空的 PD 按照原始包的结构重新链接一次。拷贝后，包 1 和包 2 的 PD/BD 不同，但是 PD/BD 所指向的内存是同一块。这种包的拷贝（其实包的切割也会出现相同的问题）要求在包的释放时只有在确认同一块内存所关联的所有包都被使用完了之后，才能进行内存的释放。在 Pktlib\_freePacket 函数中会查找当前包下所有 PD/BD 所指向内存所关联的所有 PD/BD 的当前情况，如果还有 PD/BD 在使用这块内存就不立即释放，只有在所有 PD/BD 都不使用这块内存了，才会将它释放。

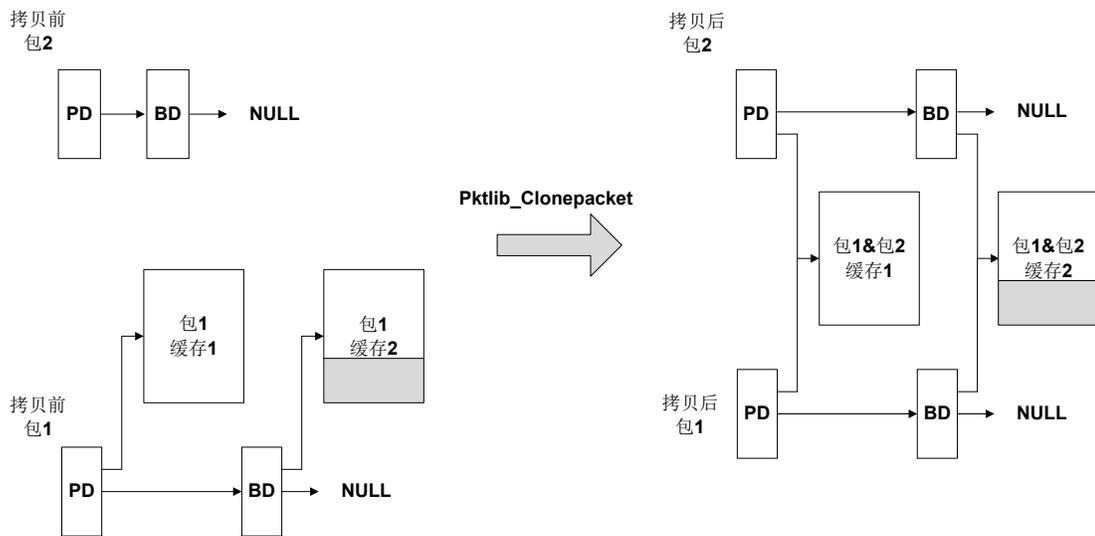


图 5. 包的拷贝

### 4.3 包的分割

层 2 处理中包的分割操作，主要有下面两个场景：

- (1) RLC 收到调度结果后需要将一个 PDCP 的 PDU 分割为两个或多个 RLC 的 PDU
- (2) MAC 复用需要一个 RLC 的 PDU 分割为两个或多个 MAC 的 PDU

如图 6 所示，Pktlib 在做包的分割时是通过传入一个空的 PD，然后用这个 PD 指向切割的字节位置构造出新的包。从图中可以看到切割后包 1 缓存 2 和包 2 缓存 1 在同一个内存块上，只有当这两个包都不在使用这块内存之后，这个内存块才能得到释放。

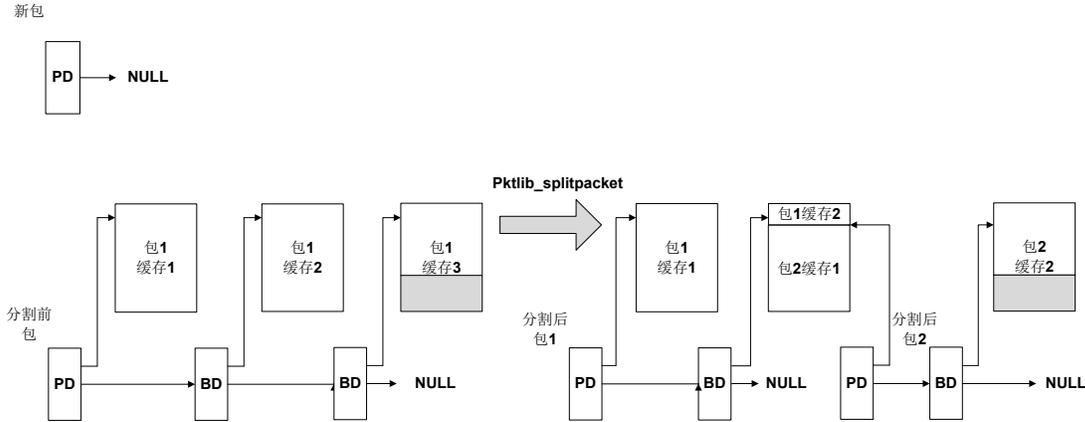


图 6. 包的分割

## 5. Pktlib 的性能

使用 Pktlib 的 API 能够完全替换传统的基于 OS 的内存管理，并且能实现零拷贝的包合并，复制和分割，在内存处理的效率上大大优于传统方案。下表列出了在 TI 公司 TMS320C6614 芯片上测试得到的函数性能。

表 2. Pktlib 主要函数的性能

Pktlib 函数	处理的包长 (字节)	Cycle 数
Pktlib_allocPacket	800	267
Pktlib_freePacket	800	316
Pktlib_packetMerge	第一包 200, 第二包 500	58
Pktlib_clonePacket	拷贝前 800, 拷贝后 800	546
Pktlib_splitPacket	分割前 1500, 分割后第一包 800, 第二包 700	143

## 6. 结束语

TI 在 Keystone 上提供的 Pktlib 能够实现硬件和软件内存管理的统一，实现基于包的数据合并，拷贝，分割，以及软件与硬件加速器之间完全的零拷贝数据传递。这些特点正好适应了 LTE 层 2 中大批量包数据处理的要求。结合 Pktlib 进行 LTE 基站层 2 软件的开发，能极大地提高包的处理效率，在硬件规格不变的情况下提供更高的系统吞吐率。本方案为基于 Keystone SoC 的 LTE 基站开发提供了一个有益的参考。

## 参考文献

1. 3GPP. TS 36.321 V9.1.0 [S]. 3GPP, <http://www.3gpp.org>, 2009-12.
2. 3GPP. TS 36.322 V9.1.0 [S]. 3GPP, <http://www.3gpp.org>, 2010-03.
3. Erik Dahlman, Stefan Parkvall, Johan Skold. 4G LTE/LTE-Advanced for Mobile Broadband [M]. Elsevier, 2011
4. Syslib User Guide Rev.1.E [Z], Texas Instruments, 2012-02
5. Keystone Architecture Multicore Navigator User Guide (SPRUGR9E) [Z], Texas Instruments, 2012-05

## 重要声明

德州仪器(TI) 及其下属子公司有权根据 JESD46 最新标准, 对所提供的产品和服务进行更正、修改、增强、改进或其它更改, 并有权根据 JESD48 最新标准中止提供任何产品和服务。客户在下订单前应获取最新的相关信息, 并验证这些信息是否完整且是最新的。所有产品的销售都遵循在订单确认时所提供的TI 销售条款与条件。

TI 保证其所销售的组件的性能符合产品销售时 TI 半导体产品销售条件与条款的适用规范。仅在 TI 保证的范围内, 且 TI 认为有必要时才会使用测试或其它质量控制技术。除非适用法律做出了硬性规定, 否则没有必要对每种组件的所有参数进行测试。

TI 对应用帮助或客户产品设计不承担任何义务。客户应对其使用 TI 组件的产品和应用自行负责。为尽量减小与客户产品和应用相关的风险, 客户应提供充分的设计与操作安全措施。

TI 不对任何 TI 专利权、版权、屏蔽作品权或其它与使用了 TI 组件或服务的组合设备、机器或流程相关的 TI 知识产权中授予的直接或隐含权作出任何保证或解释。TI 所发布的与第三方产品或服务有关的信息, 不能构成从 TI 获得使用这些产品或服务的许可、授权、或认可。使用此类信息可能需要获得第三方的专利权或其它知识产权方面的许可, 或是 TI 的专利权或其它知识产权方面的许可。

对于 TI 的产品手册或数据表中 TI 信息的重要部分, 仅在没有对内容进行任何篡改且带有相关授权、条件、限制和声明的情况下才允许进行复制。TI 对此类篡改过的文件不承担任何责任或义务。复制第三方的信息可能需要服从额外的限制条件。

在转售 TI 组件或服务时, 如果对该组件或服务参数的陈述与 TI 标明的参数相比存在差异或虚假成分, 则会失去相关 TI 组件或服务的所有明示或暗示授权, 且这是不正当的、欺诈性商业行为。TI 对任何此类虚假陈述均不承担任何责任或义务。

客户认可并同意, 尽管任何应用相关信息或支持仍可能由 TI 提供, 但他们将独力负责满足与其产品及其应用中使用的 TI 产品相关的所有法律、法规和安全相关要求。客户声明并同意, 他们具备制定与实施安全措施所需的全部专业技术和知识, 可预见故障的危险后果、监测故障及其后果、降低有可能造成人身伤害的故障的发生机率并采取适当的补救措施。客户将全额赔偿因在此类安全关键应用中使用任何 TI 组件而对 TI 及其代理造成的任何损失。

在某些场合中, 为了推进安全相关应用有可能对 TI 组件进行特别的促销。TI 的目标是利用此类组件帮助客户设计和创立其特有的可满足适用的功能安全性标准和要求的终端产品解决方案。尽管如此, 此类组件仍然服从这些条款。

TI 组件未获得用于 FDA Class III (或类似的生命攸关医疗设备) 的授权许可, 除非各方授权官员已经达成了专门管控此类使用的特别协议。

只有那些 TI 特别注明属于军用等级或“增强型塑料”的 TI 组件才是设计或专门用于军事/航空应用或环境的。购买者认可并同意, 对并非指定面向军事或航空航天用途的 TI 组件进行军事或航空航天方面的应用, 其风险由客户单独承担, 并且由客户独力负责满足与此类使用相关的所有法律和法规要求。

TI 已明确指定符合 ISO/TS16949 要求的产品, 这些产品主要用于汽车。在任何情况下, 因使用非指定产品而无法达到 ISO/TS16949 要求, TI 不承担任何责任。

	产品		应用
数字音频	<a href="http://www.ti.com.cn/audio">www.ti.com.cn/audio</a>	通信与电信	<a href="http://www.ti.com.cn/telecom">www.ti.com.cn/telecom</a>
放大器和线性器件	<a href="http://www.ti.com.cn/amplifiers">www.ti.com.cn/amplifiers</a>	计算机及周边	<a href="http://www.ti.com.cn/computer">www.ti.com.cn/computer</a>
数据转换器	<a href="http://www.ti.com.cn/dataconverters">www.ti.com.cn/dataconverters</a>	消费电子	<a href="http://www.ti.com.cn/consumer-apps">www.ti.com.cn/consumer-apps</a>
DLP® 产品	<a href="http://www.dlp.com">www.dlp.com</a>	能源	<a href="http://www.ti.com.cn/energy">www.ti.com.cn/energy</a>
DSP - 数字信号处理器	<a href="http://www.ti.com.cn/dsp">www.ti.com.cn/dsp</a>	工业应用	<a href="http://www.ti.com.cn/industrial">www.ti.com.cn/industrial</a>
时钟和计时器	<a href="http://www.ti.com.cn/clockandtimers">www.ti.com.cn/clockandtimers</a>	医疗电子	<a href="http://www.ti.com.cn/medical">www.ti.com.cn/medical</a>
接口	<a href="http://www.ti.com.cn/interface">www.ti.com.cn/interface</a>	安防应用	<a href="http://www.ti.com.cn/security">www.ti.com.cn/security</a>
逻辑	<a href="http://www.ti.com.cn/logic">www.ti.com.cn/logic</a>	汽车电子	<a href="http://www.ti.com.cn/automotive">www.ti.com.cn/automotive</a>
电源管理	<a href="http://www.ti.com.cn/power">www.ti.com.cn/power</a>	视频和影像	<a href="http://www.ti.com.cn/video">www.ti.com.cn/video</a>
微控制器 (MCU)	<a href="http://www.ti.com.cn/microcontrollers">www.ti.com.cn/microcontrollers</a>		
RFID 系统	<a href="http://www.ti.com.cn/rfidsys">www.ti.com.cn/rfidsys</a>		
OMAP应用处理器	<a href="http://www.ti.com.cn/omap">www.ti.com.cn/omap</a>		
无线连通性	<a href="http://www.ti.com.cn/wirelessconnectivity">www.ti.com.cn/wirelessconnectivity</a>	德州仪器在线技术支持社区	<a href="http://www.deyisupport.com">www.deyisupport.com</a>

邮寄地址: 上海市浦东新区世纪大道 1568 号, 中建大厦 32 楼 邮政编码: 200122  
Copyright © 2013 德州仪器 半导体技术 (上海) 有限公司