

文章编号:1007-5321(2021)06-0089-07

DOI:10.13190/j.jbupt.2021-091

基于 VPP 的高性能 UPF 原型

刘 宇^{1,2,3}, 温向明^{1,2,3}, 王鲁晗^{1,2,3,4}, 路兆铭^{1,2,3}, 杜科良^{1,2,3}

(1. 北京邮电大学 信息与通信工程学院, 北京 100876; 2. 北京邮电大学 网络体系构建与融合北京市重点实验室, 北京 100876;
3. 北京邮电大学 先进信息网络北京实验室, 北京 100876; 4. 智邮开源通信研究院, 北京 100876)

摘要: 为满足未来应用多样化的需求,需要更加灵活的移动通信系统网络架构,在软件定义网络/网络功能虚拟化的支持下,第 5 代移动通信系统(5G)的网络功能可以是运行在通用服务器上的一组软件,但运行于通用服务器的用户面网元(UPF)性能低于传统专用转发硬件的性能. 为了提高 UPF 软件化后数据处理的性能,设计并实现了一个基于矢量包处理(VPP)的 UPF 原型系统. 该系统由包转发控制协议、VPP 以及相应的接口组成. 设计了 2 个实验以验证其功能和性能. 实验结果表明,所提出的 UPF 可与商用基站、商用终端以及开源 5G 核心网互联互通,同时,其吞吐量相比于基于 Linux 内核的方法有显著提升.

关 键 词: 第 5 代移动通信系统核心网; 矢量包处理; 用户面功能

中图分类号: TN919.5

文献标志码: A

High-Performance UPF Prototype Based on VPP

LIU Yu^{1,2,3}, WEN Xiang-ming^{1,2,3}, WANG Lu-han^{1,2,3,4}, LU Zhao-ming^{1,2,3}, DU Ke-liang^{1,2,3}

(1. School of Information and Communications Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China;
2. Beijing Key Laboratory of Network System Architecture and Convergence, Beijing University of Posts and Telecommunications, Beijing 100876, China;
3. Beijing Laboratory of Advanced Information Networks, Beijing University of Posts and Telecommunications, Beijing 100876, China;
4. Witcomm Open Source Communication Research Institute, Beijing 100876, China)

Abstract: The fifth generation of mobile communications (5G) technology is expected to be flexible enough to meet the diverse requirements of future applications. Enabled by software defined network/network function virtualization, 5G network functions could be software running on general servers. But the performance will decrease compared to traditional dedicated hardware paradigm, especially for user plane functions (UPFs). In order to improve the performance of data processing in softwareized UPF, we design and implement a UPF prototype following vector packets processing (VPP) technology, which consists of packet forwarding control protocol module, open-source VPP module, and corresponding interfaces. Two experiments are constructed for the validation of its functionalities and performance. The results show that the proposed UPF can interconnect with commercial base station, commercial terminals, and open-source 5G core network. Meanwhile, its data throughput makes obvious improvement over Linux kernel-based approaches.

Key words: the fifth generation of mobile communications system core network; vector packet processing; user plane function

收稿日期: 2021-05-12

基金项目: 国家重点研发计划项目(2019YFB1803301); 中央高校基本科研业务费专项资金项目(2020RC04)

作者简介: 刘 宇(1997—), 男, 博士生.

通信作者: 王鲁晗(1989—), 男, 讲师, E-mail: wluhan@bupt.edu.cn.

近年来,随着高清晰视频流量的快速增长以及车联网新技术的发展,终端接入设备对移动通信网络的带宽、时延有了更高的要求^[1]。然而,当前移动通信网络面临众多挑战,如网元功能之间的相互耦合、网元功能与专用硬件耦合,网络灵活性不足,使得网络的升级换代需要消耗大量的资源。为了满足第 5 代移动通信系统(5G, the fifth generation of mobile communications system)定义的三大场景,即增强移动宽带场景、大规模物联网场景以及高可靠低时延连接场景对移动通信网络的要求,在 5G 网络架构中,采用软件定义网络(SDN, software defined network)解决控制面与数据面耦合的问题^[2],实现控制面与数据面独立演进,通过网络功能虚拟化(NFV, network function virtualization)技术解决网络功能与专用硬件耦合的问题^[3],在通用服务器上运行网络功能程序,以提高网络功能扩展与部署的灵活性^[4]。在 SDN/NFV 技术的驱动下^[5],5G 核心网架构设计为专用软件部署在通用硬件平台的模式,网络部署方式从原有的专用硬件部署方式改为软件灵活部署方式^[6-8]。

为了满足不断涌现的新服务和新应用的需求,降低网络运营商的运行维护成本,必须提高 5G 移动网络的灵活性、可扩展性和可部署性。因此,基于微服务的思想第 3 代合作计划提出了服务化的 5G 核心网架构,对传统的移动核心网架构进行了重构,各网元被划分为更加轻量级的网络功能(NF, network functions)模块,每个 NF 模块只负责单一的网络功能,如图 1 所示^[9]。因此,可围绕用户面功能网元(UPF, user plane functions)进行研究。UPF 网元具有用户数据的数据包检查、分组路由和转发、用户平面部分策略规则实施、合法拦截、流量使用报告、外部协议数据单元与数据网络互连的会话点、用户

平面的服务质量处理、将一个或多个“结束标记”发送和转发到源无线接入网节点等功能^[10]。

虽然 SDN/NFV 技术提高了网络功能的定制化与管理部署的灵活性,但是设计通用服务器的初衷是进行非实时性、通用计算处理,而专用转发设备是为大量简单的数据计算而设计,并且性能优越,所以专用软件加通用硬件的模式在转发效率上弱于专用硬件。对于 5G 核心网中,负责用户数据转发的网元是 UPF 网元,UPF 网元对终端用户数据进行检测、过滤、处理、转发,面对海量的用户数据包,UPF 网元的转发速率直接影响到终端用户(UE, user equipment)的用户体验和垂直行业的服务质量,所以在满足软件化部署的灵活性前提下仍要保证数据面转发的高效性。

1 相关工作

目前,UPF 网元主要是使用硬件加速技术辅助通用处理器实现对数据流的高速转发。王立文等^[11]总结并分析了 UPF 网元的转发流程,介绍了当前的硬件加速技术和加速方案,并指出由硬件承载的 UPF 业务越多,中央处理器(CPU, central processing unit)的负载就越小,同时硬件加速卡的定制化程度也越高,这使得 CPU 与硬件之间的应用程序接口(API, application programming interface)数量大幅度增加,并且使用硬件加速技术会增加功耗和成本。然而仅使用 Linux 内核系统转发数据包的性能较差,Linux 内核系统仅以每核心约 0.7 Mpacket/s 的速率转发 64 byte 的 IPv4 数据包。Linux 内核系统具有丰富的网络功能,有很强的通用性。但是在数据包处理过程中 Linux 内核系统频繁发生缓存未命中,消耗了大量的 CPU 周期^[12],使得 Linux 在高速数据包处理方面表现较差,无法达到 Linux 内核系统转发数据包的性能要求。

Cerovic 等^[13]介绍了多种软件处理框架,在与其他框架对比中,数据平面开发套件转发框架(DPDK, data plane development kit)的每个数据包在应用程序和网卡之间移动占用的 CPU 周期数更少,花费的处理移动数据包成本更低,在数据包批量移动时数据包批处理大小为 32 的 DPDK 成为最佳解决方案,因此采用集成了 DPDK 的 L_2 包转发组件的矢量包处理程序(VPP, vector packet processing)作为基础,设计了高性能 UPF 网元原型系统实例。VPP

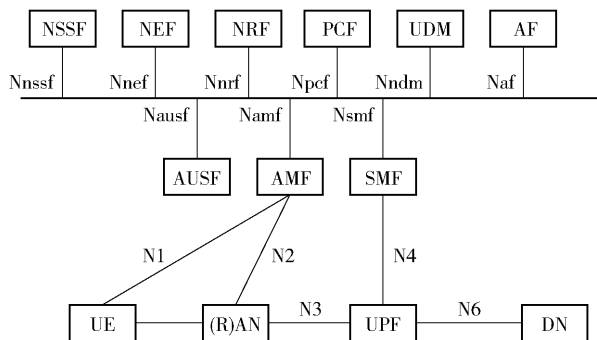


图 1 核心网架构

是矢量数据包处理技术的开源版本,包含一种高性能的数据包处理堆栈,提供了一个可扩展的框架,拥有高性能、模块化、灵活性等优点和丰富的功能集,并且允许任何人无需更改核心代码而插入新的图形节点. VPP 通过将多个数据包看作一个数据包矢量,一次性地对多个数据做包批量处理,减少了平均单个数据包的处理时间,此外,VPP 将控制层和数据层分离,将数据包处理、内存管理、处理器调度等任务转移到 Linux 的用户空间态去完成,而 Linux 的内核仅处理部分控制指令,从而可减少系统中断、上下文切换、系统调用、系统调度等开销. 此外通过采用无锁技术解决资源竞争问题,使用多核编程技术代替多线程技术,并设置 CPU 的亲 and 性,将线程和 CPU 核进行一比一绑定,减少了彼此之间调度切换,并且使用大页内存代替普通的内存,减少缓存未命中情况的发生.

针对 5G 系统中网元软件化后如何保证处理数据包的高性能问题,设计了 UPF 网元控制面程序,将会话管理功能(SMF,session management function)网元下发的规则传递到 VPP 中,并且按照规则对数据包进行处理,实现在通用处理器上对数据包的高速率转发.

2 系统架构设计

基于 VPP 的高性能 UPF 原型架构如图 2 所示. 其中包括报文转发控制协议(PFCP, packet forwarding control protocol)协议栈处理模块、基于 VPP 的数据包转发模块、PFCP 处理模块和 VPP 交互模块.

1) PFCP 协议栈处理模块. 处理 SMF 与 UPF 网元之间的 PFCP 协议消息,包括封装、解封装以及消息处理过程.

2) 基于 VPP 的数据包转发模块. 应用 SMF 下发的规则进行数据包的检测、过滤、处理和转发.

3) PFCP 与 VPP 交互模块. 使用 Golang 程序提供了与 VPP 通信的 API,使 PFCP 协议模块生成的处理规则通过 GoVPP 提供的上层接口发送到 VPP.

图 3 所示为 UPF 的内部模块结构,其中 UDP 服务端模块负责监听是否收到 SMF 发送的 PFCP 协议消息. UDP 客户端负责向 SMF 发送 PFCP 协议消息. 分类器负责对收到的 PFCP 协议消息进行分类,通过消息类型字段判断消息类型,送到对应的处理模块.

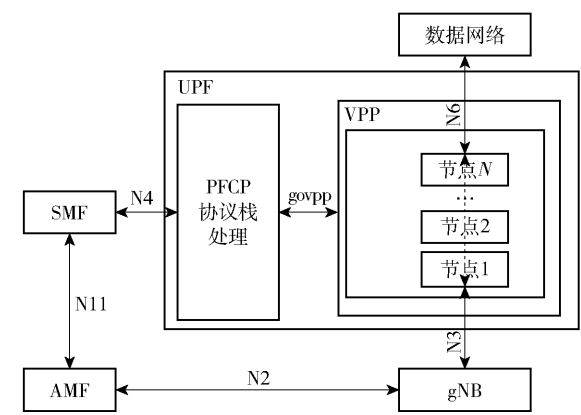


图 2 整体架构图

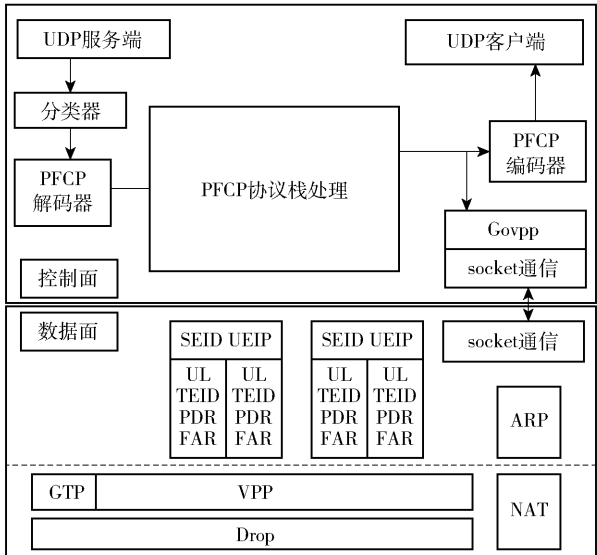


图 3 UPF 模块结构

PFCP 协议栈处理模块包含 Heartbeat 模块和 Association 模块,Heartbeat 模块负责心跳连接,时刻监听对端是否处于活跃状态. Association 模块负责连接管理,UPF 开机时会向 SMF 发送连接建立请求,并且会收到 SMF 返回的响应,连接相关消息都由该模块负责. PFCP 协议栈处理模块负责 PFCP 协议消息的处理,包括会话建立流程、话修改流程以及会话删除流程. Govpp 模块负责 Go 程序与 VPP 的通信,通过 Govpp 提供的 API 将消息发送到 VPP 中. VPP 中 GTP 模块负责封装和解封装 GTP 协议包头. VPP 负责应用 SMF 网元下发的规则对数据包进行检测、过滤、处理和转发. 地址解析协议(ARP, address resolution protocol)模块负责响应 ARP 消息请求. 网络地址转换(NAT, network address translation)模块负责用户数据包的源 NAT 转换和目的 NAT 转换.

3 系统原型实现

在数据包处理程序中,每一个数据包到达处理程序后都会触发一次中断,当网络中大量数据包到来时,会产生频繁的硬件中断请求。这些硬件中断可以打断之前较低优先级的软中断或者系统调用的执行过程,如果这种打断频繁,将会产生较高的性能开销,因此对于大数据量场景中采用轮询的方式,可以大幅提高数据包的处理转发速率。相对于传统标量报文处理方式,报文是按照到达的先后顺序来处理,第1个报文处理完成后处理第2个数据包,然后处理第3个数据包,直到处理完最后一个数据包。传统的方式还需要处理中断,并遍历调用,然后从中断返回,函数会频繁嵌套调用。造成CPU指令缓存(I-cache, instruct cache)抖动和I-cache未命中,除了提供更大的高速缓冲存储器空间之外,没有更好的解决方法。

矢量报文处理利用了数据报文处理的时间局部性特点,将一批报文分为一组,看作一个矢量包,如果第1次预取I-cache命中,则这一批报文都命中;否则这一批报文都未命中。而未命中时,则通过矢量包中的第1个报文为I-cache进行预热;然后可以针对矢量报文中后续数据包重复执行缓存中所得到的指令。从而在整个报文分组中分摊一个报文的高速缓存未命中时间,可使单个报文的处理开销显著降低。

将VPP中的功能插件称作节点,每个节点一次性处理多个数据包即数据包矢量。数据包在传递过程中,传递的并不是数据包,而是对数据包的控制。由此可见,矢量报文处理避免了标量报文处理的主要性能缺陷,解决了I-cache抖动问题,对I-cache进行预热可缓解读时延的问题,使得cache命中在数据包的处理过程更加稳定并且具有更优的性能。

VPP包含丰富的功能节点,不需要用户重新开发常用的节点。VPP节点图的拓扑方式使得VPP易于拓展,可以快速在节点图中添加新功能节点并且不影响到其他的节点。每个节点具有一个独立的线程并可实现不同的功能,可以由软件实现也可以由硬件实现,这使得后续的工作中能选择性地特定功能卸载到硬件中实现,例如将大量的简单运算放到硬件中实现,而在通用处理器中进行复杂的运算,并且根据场景的需求有选择性地将在VPP中的不同

节点功能放到硬件中实现,结合已有的高效硬件加速方案,提高数据包的处理转发速率。除了应用VPP自身拥有的原生节点外,还设计了新的节点以适配5G核心网数据包的特点。例如,在gNB与UPF网元之间传输的通用分组无线业务(GPRS, general packet radio service)隧道协议(GTP, GPRS tunneling protocol)数据包包含GTP协议的扩展包头,需要UPF网元对扩展包头进行处理,研究开发了对应于GTP扩展包头处理的相应功能。

完成数据面的设计后,实现了UPF的控制面和UPF控制面与VPP的通信机制。在UPF的配置文件中添加SMF地址,UPF启动后向SMF发送连接建立请求,SMF收到请求后发送连接建立响应,此刻UPF已与SMF建立连接。当用户完成在核心网中注册鉴权后,用户触发会话建立流程,AMF将用户与基站的信息透明传输到SMF,SMF收到会话建立请求后,向UPF发送会话建立请求,其中包含基站IP地址和分配的用户IP地址以及用户的上行数据包检测规则(PDR, packet detection rule)和上行转发行为规则(FAR, forwarding action rule)。UPF收到会话建立请求之后,缓存SMF发送的检测与转发规则,并将上述规则通过UNIX方式发送到VPP, VPP缓存SMF下发的规则,为后续上行数据包做准备。UPF为该基站的IP建立数据承载隧道,并分配隧道标识,将隧道标识通过SMF和AMF发送到基站;基站缓存隧道标识,并且为用户建立数据面承载。基站将为用户分配下行隧道标识。通过会话修改流程,将下行隧道标识经过AMF发送到SMF,触发SMF向UPF发送会话修改请求。会话修改请求消息中包含下行的检测规则和下行转发规则,UPF缓存下行规则,并将规则同步至VPP, VPP缓存规则参数,用于后续对数据包的检测、过滤、处理和转发。

当用户完成会话流程后,UPF已经缓存会话ID以及会话中包含的上行PDR、FAR规则与下行PDR、FAR规则。对于上行数据,当数据包到达UPF N3接口后,进行上行规则检测,检测GTP包头中隧道标识是否与已建立的隧道标识相同,如若相同则去除协议包头。对用户的数据包进行上行PDR规则检测,检测源IP地址是否为SMF已分配的IP地址,若与SMF分配的IP地址相同,则应用上行FAR规则进行转发,最后将数据包从N6接口发送到外部数据网络。

对于下行数据,当数据包到达 N6 接口之后,获取数据包的目的 IP,根据目的 IP 搜寻对应该 IP 的用户. 用为此用户缓存的规则,对 IP 数据包进行下行 PDR 规则检测,检测目的 IP 地址是否为本地用户,通过检测之后用下行 FAR 的转发规则添加 GTP 包头信息;最后通过 N3 接口将下行数据包转发到目的基站.

4 测试方法与结果

基于 OpenXG 的测试环境对 UPF 原型系统进行功能和性能测试,其中功能测试包括连接建立功能、心跳连接功能、会话建立功能、隧道建立功能、会话修改功能. 性能测试使用长度为 1 400 byte 的数据包进行上行吞吐率和下行吞吐率测试.

测试用到的网元包括接入和移动管理功能 (AMF, access and mobility management function) 网元、SMF、统一数据管理 (UDM, unified data management) 功能网元、基站、用户. AMF 使用 OpenXG 的开源版本,在虚拟机中部署运行,并在 AMF 的数据库中添加用户识别卡 (SIM, subscriber identity module) 信息,负责用户的接入、鉴权流程. SMF 使用 OpenXG 开源平台的开源 SMF,在虚拟机中部署运行,负责用户的会话建立、会话修改流程. UDM 网元使用 OpenXG 的开源版本,在虚拟机中部署运行,负责用户的会话策略存储. 在 PC1 中运行 AMF、SMF、UDM 的虚拟机. 基站使用 Amarisoft 公司的商用基站,该基站拥有 5G 标准接口并可与商用终端建立空口连接. 用户使用 Huawei CPE 5G,该设备支持以独立组网的方式接入核心网,并且安装了自制的 SIM (subscriber identity module) 卡, SIM 卡信息与数据库中的 SIM 卡信息对应.

本测试系统不包括网络存储功能网元. 在 AMF 网元配置文件中配置了 SMF 网元与 UDM 的接口地址;在 SMF 网元中配置了 UDM 网元的地址;UPF 网元中配置了 SMF 网元的接口地址. 此外,测试系统不包括鉴权服务功能网元. 在 AMF 网元中配置了数据库,用来存储 SIM 卡信息,用户的鉴权过程由 AMF 网元完成.

测试拓扑如图 4 所示. 基站通过网线与 UPF N3 接口相连,UPF 的 N3 接口与 N6 接口均可使用 10 Gbit/s 网卡. N6 接口连接外部数据网络,UPF 的 N4 接口、SMF 的 N4 和 N11 接口、AMF 的 N2 和 N11

接口以及 gNB 的 N2 接口均连接到交换机中. 在连接性测试中,UPF 开启后会向已配置 SMF IP 发送连接建立请求,SMF 收到后发送连接建立响应. UPF 收到 SMF 的响应后在本地存储 SMF 信息. 使用抓包工具 wireshark 捕获 UPF 网元网卡数据包,从 wireshark 的捕获结果中获得连接建立请求与响应;连接建立之后,SMF 与 UPF 网元在收到心跳请求后会发送心跳响应;终端用户触发会话建立流程,SMF 向 UPF 发送会话建立请求,包含上行 PDR 规则和上行 FAR 规则,UPF 收到数据包后返回会话建立响应,UPF 原型系统为用户缓存报文转发规则;基站触发会话修改流程,SMF 向 UPF 发送会话修改流程,包含下行的 PDR 检测规则和下行 FAR 转发规则,并且包含基站分配的隧道标识和基站 IP 地址,UPF 收到会话修改请求后生成会话修改响应消息,然后发送到 SMF.

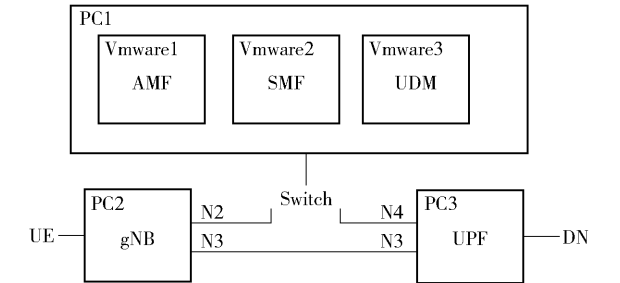


图 4 测试拓扑

使用基于 DPDK 的 Trex 工具进行性能测试. 将在 2 台服务器上运行 Trex 和 UPF, Trex 服务器的 2 个网卡分别连接到 UPF 主机的 N3 接口网卡和 N6 接口网卡. 在测试中,均使用了 10 Gbit/s 的网卡.

采用上述测试方法对 UPF 的功能和性能进行了测试,测试结果如图 5 所示. UPF 支持与 SMF 的连接建立、心跳连接建立、会话建立和会话修改流

```
PFCEP Association Setup Request
PFCEP Association Setup Request
PFCEP Association Setup Response
PFCEP Heartbeat Request
PFCEP Heartbeat Response
PFCEP Heartbeat Request
PFCEP Heartbeat Response
PFCEP Session Establishment Request
PFCEP Session Establishment Response
PFCEP Heartbeat Request
PFCEP Heartbeat Response
PFCEP Heartbeat Request
PFCEP Heartbeat Response
PFCEP Session Modification Request
PFCEP Heartbeat Request
PFCEP Heartbeat Response
PFCEP Heartbeat Request
PFCEP Session Modification Response
```

图 5 功能测试结果

程. 对 UPF 的性能进行了对比测试. 使用基于 VPP 的 UPF 性能测试结果如表 1 所示. 在上行测试中数据包长度为 1.4 KB 的速率为 7.73 Gbit/s,包转发率为 0.679 9 Mpacket/s,丢包率为 1.76%;在下行测试中数据包长度为 1.4 KB 的速率为 8.08 Gbit/s,包转发率为 0.680 5 Mpacket/s,丢包率为 0.01%. 使用基于内核转发的 UPF 性能测试结果如表 2 所示. 在上行测试中数据包长度为 1.4 KB 的速率为 0.041 Gbit/s,包转发率为 0.003 6 Mpacket/s,丢包率为 2.37%. 在下行测试中数据包长度为 1.4 KB 的速率为 0.045 Gbit/s,包转发率为 0.003 8 Mpacket/s,丢包率为 0.58%. 图 6 所示为实际测试场景.

表 1 基于 VPP 的 UPF 性能

传输方向	丢包率/%	包转发速率/ (Mpacket·s ⁻¹)	比特率/ (Gbit·s ⁻¹)
下行传输	0.01	0.680 5	8.08
上行传输	1.76	0.679 9	7.73

表 2 基于 Linux 内核的 UPF 性能

传输方向	丢包率/%	包转发速率/ (Mpacket·s ⁻¹)	比特率/ (Gbit·s ⁻¹)
下行传输	0.58	0.003 8	0.045
上行传输	2.37	0.003 6	0.041

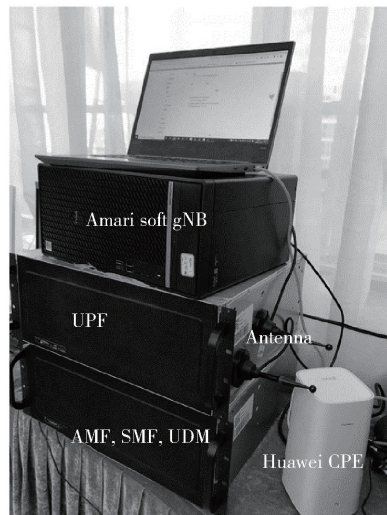


图 6 实际测试场景

从表 1 和表 2 可见,基于 VPP 的 UPF 相比于用 Linux 方式可达到更高的数据包转发速率和吞吐量. 由于 VPP 处于用户空间态,而通过 Linux 内核方式转发会在用户空间与内核空间进行频繁的数据拷

贝,导致数据包丢包率高于基于 VPP 方式的数据包丢包率,而数据包的转发速率和比特率低于基于 VPP 方式的数据包转发速率和比特率.

5 结束语

研究了基于 VPP 的高性能 UPF 原型系统的实现. 提出了基于矢量包处理方式的高性能 UPF 原型系统架构,并且在该架构的基础上实现了高吞吐量 UPF 原型系统,可以为 5G 核心网中 UPF 的部署提供解决方案,支持 5G 与垂直行业的结合. 在未来的工作中,将进一步提高该原型系统的性能,进一步完善其功能.

参考文献:

[1] Zhang Haijun, Dong Yanjie, Cheng Julian, et al. Fronthauling for 5G LTE-U ultra dense cloud small cell networks[J]. IEEE Wireless Communications, 2016, 23 (6): 48-53.

[2] Li Yong, Chen Min. Software-defined network function virtualization: a survey[J]. IEEE Access,2015,3:2542-2553.

[3] Han Bo, Gopalakrishnan V, Ji Lusheng, et al. Network function virtualization: challenges and opportunities for innovations[J]. IEEE Communications Magazine, 2015, 53(2): 90-97.

[4] Riggio R, Bradai A, Harutyunyan D, et al. Scheduling wireless virtual networks functions[J]. IEEE Transactions on Network and Service Management, 2016, 13 (2): 240-252.

[5] 赵明宇, 严学强. SDN 和 NFV 在 5G 移动通信网络架构中的应用研究[J]. 移动通信, 2015, 39(14): 64-68.

Zhao Mingyu, Yan Xueqiang. Research on application of SDN and NFV in 5G mobile communication network architecture[J]. Mobile Communications, 2015, 39(14): 64-68.

[6] Akyildiz I F, Lin S C, Wang Pu. Wireless software-defined networks (W-SDNs) and network function virtualization (NFV) for 5G cellular systems: an overview and qualitative evaluation[J]. Computer Networks, 2015, 93: 66-79.

[7] Akyildiz I F, Wang Pu, Lin S C. SoftAir: a software defined networking architecture for 5G wireless systems[J]. Computer Networks, 2015, 85: 1-18.

[8] Wang Hucheng, Chen Shanzhi, Xu Hui, et al. SoftNet:

- a software defined decentralized mobile network architecture toward 5G[J]. IEEE Network, 2015, 29(2): 16-22.
- [9] 3GPP. 3GPP TS 23. 501—2020, System architecture for the 5G system (release 16) [S]. Sophia Antipolis: 3rd Generation Partnership Project, 2020: 29-404.
- [10] 3GPP. 3GPP TS 29. 244—2020, Interface between the control plane and the user plane nodes (release 16) [S]. Sophia Antipolis: 3rd Generation Partnership Project, 2020: 20-294.
- [11] 王立文, 王友祥, 唐雄燕, 等. 5G 核心网 UPF 硬件加速技术[J]. 移动通信, 2020, 44(1): 19-23, 32.
- Wang Liwen, Wang Youxiang, Tang Xiongyan, et al. Research on UPF hardware acceleration technology in 5G core networks[J]. Mobile Communications, 2020, 44(1): 19-23, 32.
- [12] Hong Cheol-Ho, Lee Kyungwoon, Hwang Jaehyun, et al. Kafe: can OS kernels forward packets fast enough for software routers? [J]. IEEE/ACM Transactions on Networking, 2018, 26(6): 2734-2747.
- [13] Cerović D, Del Piccolo V, Amamou A, et al. Fast packet processing: a survey[J]. IEEE Communications Surveys & Tutorials, 2018, 20(4): 3645-3676.
-

(上接第 88 页)

- [5] Mirjalili S, Lewis A. The whale optimization algorithm [J]. Advances in Engineering Software, 2016, 95: 51-67.
- [6] Elhosseini M A, Haikal A Y, Badawy M, et al. Biped robot stability based on an A-C parametric whale optimization algorithm [J]. Journal of Computational Science, 2019, 31: 17-32.
- [7] Arora S, Singh S. Butterfly optimization algorithm: a novel approach for global optimization[J]. Soft Computing, 2019, 23(3): 715-734.
- [8] Heidari A A, Mirjalili S, Faris H, et al. Harris hawks optimization: algorithm and applications[J]. Future Generation Computer Systems, 2019, 97: 849-872.
- [9] Liu Peng, El Basha M D, Li Yangjunyi, et al. Deep evolutionary networks with expedited genetic algorithms for medical image denoising [J]. Medical Image Analysis, 2019, 54: 306-315.
- [10] Serizawa T, Fujita H. Optimization of convolutional neural network using the linearly decreasing weight particle swarm optimization [EB/OL]. (2020-01-16) [2021-01-21]. <https://arxiv.org/abs/2001.05670>.