



(12) 发明专利申请

(10) 申请公布号 CN 112134810 A

(43) 申请公布日 2020. 12. 25

(21) 申请号 202011003304.5

(22) 申请日 2020.09.22

(71) 申请人 中国人民解放军国防科技大学

地址 410073 湖南省长沙市开福区德雅路
109号

(72) 发明人 原玉磊 陈曙晖 时向泉 赵宝康
赵锋 陶静 虞万荣 韩彪
魏子令

(74) 专利代理机构 湖南企企卫知识产权代理有
限公司 43257

代理人 任合明

(51) Int. Cl.

H04L 12/851 (2013.01)

G06F 16/906 (2019.01)

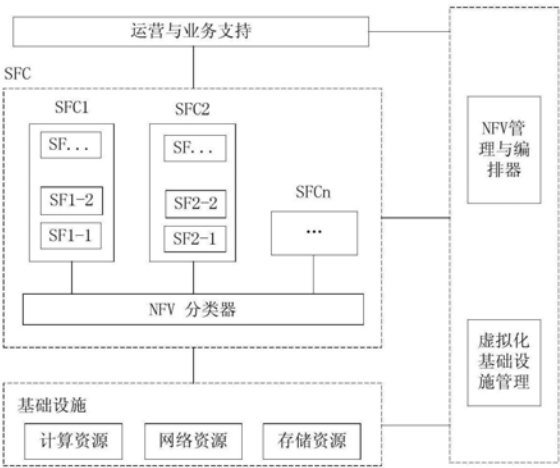
权利要求书4页 说明书10页 附图3页

(54) 发明名称

一种基于FPGA硬件加速的NFV网络流量分类
方法

(57) 摘要

本发明公开了一种基于FPGA硬件加速的NFV网络流量分类方法,目的是解决传统方法中分类器成为NFV技术性能瓶颈的问题。技术方案是构建由流表配置部件、和采用FPGA实现的流表管理部件、流表存储器、报文接收部件、流表查询部件、报文处理部件、报文发送部件组成的基于硬件加速的NFV分类器。NFV分类器一边进行初始化,监听NFV管理与编排器并从NFV管理与编排器接收服务功能链管理数据,完成报文流表的管理;一边监听网卡MAC层,接收网卡MAC层发送的输入报文,根据输入报文的流表数据结构完成报文分类处理,并发送给NFV服务功能链。本发明将NFV管理编排的服务功能链数据转为报文流表,将报文流表匹配及处理通过FPGA硬件加速来实现,大大提高了NFV分类器的分类速度。



1. 一种基于FPGA硬件加速的NFV网络流量分类方法,其特征在于包括以下步骤:

第一步,构建基于硬件加速的NFV分类器;NFV分类器由软件部分和硬件部分组成,其中软件部分包括流表配置部件,硬件部分包括流表管理部件,流表存储器,报文接收部件,流表查询部件,报文处理部件,报文发送部件;

流表配置部件为软件实现的功能模块,部署运行于NFV网络环境的操作系统中;流表管理部件、报文接收部件、流表查询部件、报文处理部件、报文发送部件为在网卡FPGA中实现的功能模块,部署运行于网卡FPGA中;流表存储器为网卡FPGA中实现的存储空间;

流表配置部件与流表管理部件、NFV管理与编排器相连,它从NFV管理与编排器中接收NFV服务功能链管理数据sfc_struct,并将服务功能链管理数据sfc_struct转换为流表管理数据,将流表管理数据发送给流表管理部件;

sfc_struct包括7个域:服务功能链管理命令即sfc_cmd,服务功能链编号即sfc_id,服务功能链管理数据的报文原始目的IP地址即sfc_dip1、报文原始端口号即sfc_dport1,服务功能链管理数据对应服务功能链报文的的目的IP地址即sfc_dip2、服务功能链报文的的目的端口号即sfc_dport2,服务功能链管理数据对应的网络协议号即sfc_proto;以下均以“->”表示sfc_struct的某个域;

流表管理数据flow_struct包括7个域:流表管理命令即flow_cmd,流表编号即flow_id,流表管理数据的报文原始目的IP地址即flow_dip1、报文原始端口号即flow_dport1,流表处理后报文的的目的IP地址即flow_dip2、流表处理后报文目的端口号即flow_dport2,流表管理数据对应网络协议号即flow_proto;

流表配置部件通过数组flowid_used[FLOWN]和链表数组sfc_flow[SFCN]来维护服务功能链编号和流表编号之间的关系;

数组flowid_used的第i个元素flowid_used[i]取值0或者1, $i=1,2,\dots,FLOWN$;flowid_used[i]取值为0时,表示流表编号i未被使用;flowid_used[i]取值为1时,表示流表编号i已被使用;其中FLOWN表示NFV分类器可支持的最大流表数量;

链表数组sfc_flow的第j个元素sfc_flow[j]为1个链表, $j=1,2,\dots,SFCN$,该链表存储属于编号为j的服务功能链的所有流表的编号,链表中一个元素为属于服务功能链j的一个流表的编号其中SFCN表示NFV分类器可支持的最大服务功能链数量;

服务功能链管理命令和流表管理命令有2种取值:0,1;服务功能链管理命令取值0表示增加服务功能链管理数据;取值1表示删除服务功能链管理数据;流表管理命令取值0表示增加流表管理数据;取值1表示删除流表管理数据;

流表管理部件与流表配置部件和流表存储器相连,它从流表配置部件接收流表管理数据,将流表管理数据转化成流表数据结构后存储到流表存储器;

流表存储器与流表管理部件和流表查询部件相连,是网卡内FPGA中的数据存储器,用于存储流表数据结构;流表数据结构包括3个域:流表编号即flow_id,流表处理报文目的IP地址即flow_dip2和流表处理后报文目的端口号即flow_dport2;

报文接收部件与以太网接口即网卡MAC层和流表查询部件相连,它从以太网接口接收外部输入报文,将输入报文发送给流表查询部件;

流表查询部件与报文接收部件、报文处理部件和流表存储器相连,它从报文接收部件接收输入报文,从输入报文中提取出五元组信息(源IP地址,目的IP地址,源端口,目的端

口,协议号),对五元组信息计算哈希值,并根据哈希值从流表存储器中查找对应的流表数据结构;查到后,将输入报文和流表数据结构一并发送给报文处理部件;

报文处理部件与流表查询部件和报文发送部件相连,它从流表查询部件接收输入报文和对应的流表数据结构,根据流表规则处理报文,并将处理后的报文发送给报文发送部件;

报文发送部件与报文处理部件相连,并通过网卡驱动和协议栈与NFV服务功能链相连,它从报文处理部件接收处理后的报文,将处理后报文发送给NFV服务功能链;

第二步,NFV分类器按2.1~2.5的流程进行初始化,监听NFV管理与编排器并从NFV管理与编排器接收服务功能链管理数据,完成报文流表的管理;同时NFV分类器按2.6~2.11的流程监听网卡MAC层,接收网卡MAC层发送的输入报文,根据输入报文的流表数据结构完成报文分类处理,并发送给NFV服务功能链;具体步骤如下:

2.1流表配置部件进行初始化,具体步骤如下:

2.1.1根据NFV网络需求确定FLOWN和SFCName的值;

2.1.2流表配置部件初始化数组flowid_used的每个元素为0,即令flowid_used[1],...,flowid_used[i],...,flowid_used[FLOWN]均=0;

2.1.3流表配置部件初始化链表数组sfc_flow的每个元素为空,即令sfc_flow[1],...,sfc_flow[j],...,sfc_flow[SFCN]均为空;

2.1.4流表配置部件初始化时间t=当前操作系统时间;

2.2流表配置部件监控NFV管理与编排器是否发送了服务功能链管理数据,如果发送了服务功能链管理数据,转步骤2.3;如果没有发送,转步骤2.2.1;

2.2.1获取当前操作系统时间t1,若 $(t1-t) < T$,转2.2继续监控;若 $(t1-t) \geq T$,转第三步;所述T为超时阈值;

2.3流表配置部件更新t为当前操作系统时间,接收服务功能链管理数据sfc_struct,并根据服务功能链管理数据sfc_struct更新数组flowid_used和sfc_flow,构建流表管理数据,具体步骤如下:

2.3.1从数组flowid_used中找出第一个为0的元素,如果所有元素均为1,则找出第一个元素;

2.3.3将i1插入链表sfc_flow[sfc_struct->sfc_id]中,其中sfc_struct->sfc_id为sfc_struct的sfc_id,sfc_flow[sfc_struct->sfc_id]为链表数组sfc_flow的第sfc_struct->sfc_id个元素;

2.3.4定义流表管理数据flow_struct,令:

flow_struct->flow_id=i1;

flow_struct->flow_cmd=sfc_struct->sfc_cmd;

flow_struct->flow_dip1=sfc_struct->sfc_dip1;

flow_struct->flow_dport1=sfc_struct->sfc_dport1;

flow_struct->flow_dip2=sfc_struct->sfc_dip2;

flow_struct->flow_dport2=sfc_struct->sfc_dport2;

flow_struct->flow_proto=sfc_struct->sfc_proto;

2.4流表配置部件将flow_struct发送给流表管理部件;

2.5流表管理部件从流表配置部件接收flow_struct,根据flow_struct构建流表数据

结构,并更新流表存储器,具体步骤如下:

2.5.1对flow_struct->flow_dip1、flow_struct->flow_dport1和flow_struct->flow_proto计算哈希值,令为flow_hash;

2.5.2如果flow_struct->flow_cmd为0,转步骤2.5.3,否则转步骤2.5.4;

2.5.3将flow_struct->flow_id、flow_struct->flow_dip2、flow_struct->flow_dport2依次存入流表存储器中第flow_hash个流表数据结构,分别作为第flow_hash个流表数据结构的flow_id、flow_dip2和flow_dport2,转步骤2.2;

2.5.4将流表存储器中第flow_hash个流表数据结构的数据全部设置为0,转步骤2.2;

2.6报文接收部件初始化时间 $t_2=0$,并开始计时;

2.6.1报文接收部件监控网卡MAC层是否有输入报文到达,如果有,转步骤2.7,如果没有,则转步骤2.6.2;

2.6.2获取当前计时时间 t_3 ,若 $(t_3-t_2)<T$,转2.6.1继续监控;若 $(t_3-t_2)\geq T$,转第三步;

2.7报文接收部件更新 t_2 为当前计时时间,从网卡MAC层接收输入报文,令为pkt,将pkt发送给流表查询部件;

2.8流表查询部件从报文接收部件接收pkt,根据pkt在流表存储器中查找匹配的流表数据结构,具体步骤如下:

2.8.1从pkt中提取报文目的IP地址pkt_dip,目的端口pkt_dport和协议号pkt_proto;

2.8.2对pkt_dip、pkt_dport和pkt_proto计算哈希值pkt_hash;

2.8.3在流表存储器的第pkt_hash位置处分别提取出流表数据结构的流表编号,记为pkt_fid、流表处理报文目的IP地址,记为pkt_fip、流表处理后报文目的端口号,记为pkt_fport;

2.8.4如果pkt_fid为0,将pkt转发至报文发送部件,转步骤2.11;否则,将pkt以及pkt_fip、pkt_fport转发至报文处理部件,转步骤2.9;

2.9报文处理部件从流表查询部件接收pkt、pkt_fip、pkt_fport,修改报文pkt目的IP地址、计算并更新IP校验和,得到报文pkt1,并将pkt1发送给报文发送部件;

2.10报文发送部件从报文处理部件接收pkt1,并将pkt1根据pkt1的目的IP地址转发至对应的NFV服务功能链,完成报文分类,转步骤2.6.1继续监听接收输入报文;

2.11报文发送部件从流表查询部件接收报文pkt,并将pkt根据pkt目的IP地址转发至对应的NFV服务功能链,完成报文分类,转步骤2.6.1继续监听接收输入报文。

第三步,此时表示NFV分类器已经有长达T的时间未从NFV管理与编排器接收到输出服务功能链管理数据,或报文接收部件已经有长达T的时间未从网卡MAC层接收到输入报文,说明网络出现了故障,结束。

2.如权利要求1所述的一种基于FPGA硬件加速的NFV网络流量分类方法,其特征在于所述服务功能链管理命令和流表管理命令取值相同。

3.如权利要求1所述的一种基于FPGA硬件加速的NFV网络流量分类方法,其特征在于所述FLOWN=1000,SFCN=100;所述T设置为30分钟。

4.如权利要求1所述的一种基于FPGA硬件加速的NFV网络流量分类方法,其特征在于2.3.1步所述从数组flowid_used中找出第一个为0的元素的方法为:

2.3.1.1定义循环变量i1,令i1=1;

2.3.1.2如果flowid_used[i1]=0,结束,否则转步骤2.3.1.3;

2.3.1.3令i1=i1+1,如果i1≤FLOWN,转步骤2.3.1.2,否则转步骤2.3.1.4;

2.3.1.4令i1=1,转步骤2.3.2。

5.如权利要求1所述的一种基于FPGA硬件加速的NFV网络流量分类方法,其特征在于2.5.1步和2.8.2所述计算哈希值的方法是CRC32算法。

6.如权利要求1所述的一种基于FPGA硬件加速的NFV网络流量分类方法,其特征在于2.9步所述报文处理部件计算并更新IP校验和,得到报文pkt1,并将pkt1发送给报文发送部件的方法是:

2.9.1报文处理部件将报文pkt的目的IP地址修改为pkt_fip,将报文pkt的目的端口号修改为pkt_fport,得到报文pkt1;

2.9.2报文处理部件计算并更新报文pkt1的IP校验和;

2.9.3报文处理部件将处理后的报文pkt1发送给报文发送部件。

一种基于FPGA硬件加速的NFV网络流量分类方法

技术领域

[0001] 本发明涉及网络功能虚拟化技术领域,尤其涉及一种基于FPGA硬件加速的NFV网络流量分类方法。

背景技术

[0002] 传统网络架构使用大量专用硬件设备,来提供防火墙、入侵防御、网络地址转换等网络功能。随着大数据、云计算等新兴行业的迅速发展,应用对网络的要求越来越高,部署新的应用往往需要新的专用设备和网络的支持,为网络运营成本和运维带来很大挑战。为应对这些问题,欧洲电信标准协会(European Telecommunication Standards Institute, ETSI)提出网络功能虚拟化(Network Function Virtualization, NFV)。NFV技术利用普通商用服务器为网络功能提供虚拟机、容器等运行环境,用虚拟机或容器中运行的软件来替换网络中硬件设备,提高网络功能的灵活性,降低投资成本、研发成本和运维成本。

[0003] 在NFV的应用场景中,NFV系统往往要提供多个网络功能服务。不同的网络功能通过相应的服务功能链来实现。服务功能链由若干个虚拟网络功能连接组成。

[0004] 根据IETF(The Internet Engineering Task Force:互联网工程任务组)的SFC工作组以及RFC7665标准,网络服务功能链通过多个功能节点来实现,功能节点包括:Classifier(分类器),SFF(Service Function Forwarder,服务功能转发器),SF(Service Function,服务功能节点),SFC Proxy(服务功能代理)。如图1所示,NFV分类器是整个NFV系统网络流量的总入口,并负责将不同的网络流量分发至对应的服务功能链。在NFV技术架构中,分类器往往成为整个系统的性能瓶颈。

[0005] 因此研究NFV分类器加速技术,提高NFV分类器的性能对于提升NFV技术架构的性能有重要意义。

[0006] 在提升NFV技术架构性能方面,毕军等(参见毕军,郑智隆,孙晨,于恒.一种NFV中网络功能并行处理基础架构:中国专利,申请号201810050345.6)提出一种并行处理的NFV网络架构。该架构通过多个网络功能处理模块的并行处理来提高系统处理性能。但是报文流在各个网络功能处理模块间的分类分发(即将不同的网络流分发至不同的SFC)仍是通过单一的软件分类器(即NFV分类器)来完成。该软件实现的NFV分类器将成为整个系统性能的瓶颈。

[0007] 虽然以下公开文献涉及提高报文转发效率的问题,但都没有提及如何提高NFV分类器的NFV网络流量分类的速度。如:赵立强等(参见赵立强,宋凤飞,梁凯,卢晓迪.一种基于NFV的网络切片选择方法:中国专利,申请号201810194397.0)基于不同用户对网络资源的需求以及运营商网络中不同切片的资源余量,针对用户接入提出一种网络切片选择方法。该方法实质上是运营商网络资源的调度方法,无法解决NFV网络中大量用户请求分类的问题。

[0008] 胡森标(参加胡森标,倪鹏浩,逯利军,钱培专.优化NFV数据转发性能的系统及其方法:中国专利,申请号201810153806.2)提出一种基于硬件加速器的在NFV网络环境中针

对QoS功能实现的硬件加速方法。该方法利用硬件加速器代替NFV网络的数据平面实现对不同用户QoS的管理,提高了NFV网络转发性能,但是并没有针对不同用户流量做NFV的分类处理。同时,胡森标(参加胡森标,丁强,逯利军,钱培专.基于NFV技术加速报文ACL匹配处理的方法及系统:中国专利,申请号201810126568.6)等针对NFV网络环境中,通用处理架构实现ACL网元时计算量大、处理效率低的问题,提出一种硬件加速的ACL匹配处理方法,通过将VNF网元的ACL功能卸载至硬件加速器实现,降低服务器的处理压力,同时提高转发性能。

[0009] 张新玲(参见张新玲.一种基于NFV的报文匹配系统和方法:中国专利,申请号201911159328.7,2020年2月25日)提出一种基于硬件加速器的NFV报文匹配方法,该方法通过将报文规则匹配功能卸载至硬件加速器中进行,从而实现了高性能NAT功能,减轻了NFV网元设备的CPU负载,提高了报文转发效率,缩小了报文转发时延。石江涛等(参见石江涛,陈彦灵,吴安.一种NFV加速资源与通用计算资源智能切换方法及系统:中国专利,专利号CN 106533987 B)提出一种NFV环境中加速资源和通用计算资源的智能切换方法。该方法根据NFV环境中加速资源和通用计算资源的利用率和空闲率以及服务网元的资源需求,自动在加速资源和通用计算资源中切换服务网元的实现,从而整体降低NFV环境的资源利用和消耗。

[0010] 从上述研究成果可以看出,使用硬件加速的方法来实现NFV网络环境中某个环节的功能,从而提升NFV技术架构的性能已经成为目前NFV技术研究的一个趋势。但是如何提高NFV分类器的NFV网络流量分类的速度却没有公开文献涉及。目前在NFV技术领域的研究中,关于NFV分类器主要采用通用服务器的软件来实现。在大流量的NFV应用场景中,软件实现的分类器无法满足高速流量分类的需求。

发明内容

[0011] 本发明要解决的技术问题是针对传统NFV技术架构中,分类器使用软件实现,分类效率难以提高,成为NFV技术性能瓶颈的问题。

[0012] 本发明基于FPGA提出一种硬件加速的NFV分类器,利用网卡中的FPGA实现NFV分类功能。该方法将NFV管理编排的服务功能链数据转为报文流表,并将报文流表匹配及处理通过FPGA硬件加速来实现,可大大提高NFV分类器的分流速度。

[0013] 本发明包括以下步骤:

[0014] 第一步,构建基于硬件加速的NFV分类器。NFV分类器由软件部分和硬件部分组成,其中软件部分包括流表配置部件,硬件部分包括流表管理部件,流表存储器,报文接收部件,流表查询部件,报文处理部件,报文发送部件,采用FPGA实现。

[0015] 流表配置部件为软件实现的功能模块,部署运行于NFV网络环境的操作系统中;流表管理部件、报文接收部件、流表查询部件、报文处理部件、报文发送部件为在网卡FPGA中实现的功能模块,部署运行于网卡FPGA中;流表存储器为网卡FPGA中实现的存储空间。

[0016] 流表配置部件与流表管理部件、NFV管理与编排器相连,它从NFV管理与编排器中接收NFV服务功能链管理数据(命名为sfc_struct),并将服务功能链管理数据sfc_struct转换为流表管理数据,将流表管理数据发送给流表管理部件。

[0017] sfc_struct包括7个域:服务功能链管理命令(即sfc_cmd),服务功能链编号(sfc_id),服务功能链管理数据的报文原始目的IP地址(即sfc_dip1)、报文原始端口号(即sfc_

dport1),服务功能链管理数据对应服务功能链报文的目的IP地址(即sfc_dip2)、服务功能链报文的目的端口号(即sfc_dport2),服务功能链管理数据对应的网络协议号(即sfc_proto)。以下均以“->”表示sfc_struct的某个域,如sfc_struct->sfc_id表示sfc_struct的sfc_id。

[0018] 每个流表管理数据(命名为flow_struct)包括7个域:流表管理命令(即flow_cmd),流表编号(flow_id),流表管理数据的报文原始目的IP地址(即flow_dip1)、报文原始端口号(flow_dport1),流表处理后报文的目的IP地址(即flow_dip2)、流表处理后报文目的端口号(flow_dport2),流表管理数据对应网络协议号(即flow_proto)。以下均以“->”表示flow_struct的某个域,如flow_struct->flow_id表示flow_struct的flow_id。

[0019] 流表配置部件通过数组flowid_used[FLOWN]和链表数组sfc_flow[SFCN]来维护服务功能链编号和流表编号之间的关系。

[0020] 数组flowid_used的第i个元素flowid_used[i] ($i=1,2,\dots,FLOWN$) 取值0或者1。flowid_used[i]取值为0时,表示流表编号i未被使用;flowid_used[i]取值为1时,表示流表编号i已被使用。其中FLOWN表示NFV分类器可支持的最大流表数量,根据NFV网络需求来确定。

[0021] 链表数组sfc_flow的第j个元素sfc_flow[j] ($j=1,2,\dots,SFCN$) 为1个链表,该链表存储属于编号为j的服务功能链的所有流表的编号,链表中一个元素为属于服务功能链j的一个流表的编号其中SFCN表示NFV分类器可支持的最大服务功能链数量,根据NFV网络需求来确定。

[0022] 服务功能链管理命令和流表管理命令有2种取值:0,1。服务功能链管理命令取值0表示增加服务功能链管理数据;取值1表示删除服务功能链管理数据。流表管理命令取值0表示增加流表管理数据;取值1表示删除流表管理数据。一般情况下,服务功能链管理命令和流表管理命令取值相同。

[0023] 流表管理部件与流表配置部件和流表存储器相连,它从流表配置部件接收流表管理数据,将流表管理数据转化成流表数据结构后存储到流表存储器。

[0024] 流表存储器与流表管理部件和流表查询部件相连,是网卡内FPGA中的数据存储器,用于存储流表数据结构。流表数据结构包括3个域:流表编号(即flow_id),流表处理报文目的IP地址(即flow_dip2)和流表处理后报文目的端口号(即flow_dport2)。

[0025] 报文接收部件与以太网接口(网卡MAC层)和流表查询部件相连,它从以太网接口接收外部输入报文,将输入报文发送给流表查询部件。

[0026] 流表查询部件与报文接收部件、报文处理部件和流表存储器相连,它从报文接收部件接收输入报文,从输入报文中提取出五元组信息(源IP地址,目的IP地址,源端口,目的端口,协议号),对五元组信息计算哈希值,并根据哈希值从流表存储器中查找对应的流表数据结构。查到后,将输入报文和流表数据结构一并发送给报文处理部件。

[0027] 报文处理部件与流表查询部件和报文发送部件相连,它从流表查询部件接收输入报文和对应的流表数据结构,根据流表规则处理报文,并将处理后的报文发送给报文发送部件。

[0028] 报文发送部件与报文处理部件相连,并通过网卡驱动和协议栈与NFV服务功能链相连,它从报文处理部件接收处理后的报文,将处理后报文发送给NFV服务功能链。

[0029] 第二步,NFV分类器按2.1~2.5的流程(2.1~2.5步由流表配置部件、流表管理部件和流表存储器配合工作)进行初始化,监听NFV管理与编排器并从NFV管理与编排器接收服务功能链管理数据,完成报文流表的管理;同时NFV分类器按2.6~2.11的流程(2.6~2.11步由报文接收部件、流表查询部件、报文处理部件以及报文发送部件配合完成)监听网卡MAC层,接收网卡MAC层发送的输入报文(以太网报文),根据输入报文的流表数据结构完成报文分类处理,并发送给NFV服务功能链。

[0030] 具体步骤如下:

[0031] 2.1流表配置部件进行初始化。具体步骤如下:

[0032] 2.1.1根据NFV网络需求确定FLOWN和SFCName的值,可选的,设置FLOWN=1000,SFCN=100。

[0033] 2.1.2流表配置部件初始化数组flowid_used的每个元素为0,即令flowid_used[1],...,flowid_used[i],...,flowid_used[FLOWN]均=0, $i=1,2,\dots,\text{FLOWN}$ 。

[0034] 2.1.3流表配置部件初始化链表数组sfc_flow的每个元素为空,即令sfc_flow[1],...,sfc_flow[j],...,sfc_flow[SFCN]均为空, $j=1,2,\dots,\text{SFCN}$ 。

[0035] 2.1.4流表配置部件初始化时间 t =当前操作系统时间;

[0036] 2.2流表配置部件监控NFV管理与编排器是否发送了服务功能链管理数据,如果发送了服务功能链管理数据,转步骤2.3;如果没有发送,转步骤2.2.1;

[0037] 2.2.1获取当前操作系统时间 t_1 ,若 $(t_1-t)<T$,转2.2继续监控;若 $(t_1-t)\geq T$,转第三步。所述 T 为超时阈值,一般设置为30分钟。

[0038] 2.3流表配置部件更新 t 为当前操作系统时间,接收服务功能链管理数据sfc_struct,并根据服务功能链管理数据sfc_struct更新数组flowid_used和sfc_flow,并构建流表管理数据,具体步骤如下:

[0039] 2.3.1从数组flowid_used中找出第一个为0的元素,如果所有元素均为1,则找出第一个元素,方法为:

[0040] 2.3.1.1定义循环变量 i_1 ,令 $i_1=1$ 。

[0041] 2.3.1.2如果flowid_used[i_1]=0,转步骤2.3.2,否则转步骤2.3.1.3。

[0042] 2.3.1.3令 $i_1=i_1+1$,如果 $i_1\leq\text{FLOWN}$,转步骤2.3.1.2,否则转步骤2.3.1.4。

[0043] 2.3.1.4令 $i_1=1$,转步骤2.3.2。

[0044] 2.3.2令flowid_used[i_1]=1。

[0045] 2.3.3将 i_1 插入链表sfc_flow[sfc_struct->sfc_id]中。其中sfc_struct->sfc_id为sfc_struct的sfc_id,sfc_flow[sfc_struct->sfc_id]为链表数组sfc_flow的第sfc_struct->sfc_id个元素。

[0046] 2.3.4定义流表管理数据flow_struct,令:

[0047] flow_struct->flow_id= i_1 ;

[0048] flow_struct->flow_cmd=sfc_struct->sfc_cmd;

[0049] flow_struct->flow_dip1=sfc_struct->sfc_dip1;

[0050] flow_struct->flow_dport1=sfc_struct->sfc_dport1;

[0051] flow_struct->flow_dip2=sfc_struct->sfc_dip2;

[0052] flow_struct->flow_dport2=sfc_struct->sfc_dport2;

- [0053] flow_struct->flow_proto=sfc_struct->sfc_proto;
- [0054] 2.4流表配置部件将flow_struct发送给流表管理部件。
- [0055] 2.5流表管理部件从流表配置部件接收flow_struct,根据flow_struct构建流表数据结构,并更新流表存储器。具体步骤如下:
- [0056] 2.5.1利用CRC32算法(见www.backplane.com/matt/crc64.html)对flow_struct->flow_dip1、flow_struct->flow_dport1和flow_struct->flow_proto计算哈希值,令为flow_hash。
- [0057] 2.5.2如果flow_struct->flow_cmd为0,转步骤2.5.3,否则转步骤2.5.4。
- [0058] 2.5.3将flow_struct->flow_id、flow_struct->flow_dip2、flow_struct->flow_dport2依次存入流表存储器中第flow_hash个流表数据结构,分别作为第flow_hash个流表数据结构的flow_id、flow_dip2和flow_dport2。转步骤2.2。
- [0059] 2.5.4将流表存储器中第flow_hash个流表数据结构的数据全部设置为0。转步骤2.2。
- [0060] 2.6报文接收部件初始化时间 $t_2=0$,并开始计时;
- [0061] 2.6.1报文接收部件监控网卡MAC层是否有输入报文到达,如果有,转步骤2.7,如果没有,则转步骤2.6.2。
- [0062] 2.6.2获取当前计时时间 t_3 ,若 $(t_3-t_2)<T$,转2.6.1继续监控;若 $(t_3-t_2)\geq T$,转第三步。
- [0063] 2.7报文接收部件更新 t_2 为当前计时时间,从网卡MAC层接收输入报文(命名为pkt),将pkt发送给流表查询部件。
- [0064] 2.8流表查询部件从报文接收部件接收pkt,根据pkt在流表存储器中查找匹配的流表数据结构,具体步骤如下:
- [0065] 2.8.1从pkt中提取报文目的IP地址pkt_dip,目的端口pkt_dport和协议号pkt_proto。
- [0066] 2.8.2利用CRC32算法对pkt_dip、pkt_dport和pkt_proto计算哈希值pkt_hash。
- [0067] 2.8.3在流表存储器的第pkt_hash位置处分别提取出流表数据结构的流表编号(记录为pkt_fid)、流表处理报文目的IP地址(记录为pkt_fip)、流表处理后报文目的端口号(记录为pkt_fport)。
- [0068] 2.8.4如果pkt_fid为0,将pkt转发至报文发送部件,转步骤2.11。否则,将pkt以及pkt_fip、pkt_fport转发至报文处理部件,转步骤2.9。
- [0069] 2.9报文处理部件从流表查询部件接收pkt、pkt_fip、pkt_fport,修改报文pkt目的IP地址、计算并更新IP校验和,得到报文pkt1,并将pkt1发送给报文发送部件。具体步骤如下:
- [0070] 2.9.1报文处理部件将报文pkt的目的IP地址修改为pkt_fip,将报文pkt的目的端口号修改为pkt_fport,得到报文pkt1。
- [0071] 2.9.2报文处理部件计算并更新报文pkt1的IP校验和(计算方法参见RFC1071(IETF发布的一个技术标准))。
- [0072] 2.9.3报文处理部件将处理后的报文pkt1发送给报文发送部件。
- [0073] 2.10报文发送部件从报文处理部件接收pkt1,并将pkt1根据pkt1的目的IP地址转

发至对应的NFV服务功能链,完成报文分类。转步骤2.6.1继续监听接收输入报文。

[0074] 2.11报文发送部件从流表查询部件接收报文pkt,并将pkt根据pkt目的IP地址转发至对应的NFV服务功能链,完成报文分类。转步骤2.6.1继续监听接收输入报文。

[0075] 第三步,此时表示NFV分类器已经有长达T的时间未从NFV管理与编排器接收到输出服务功能链管理数据,或报文接收部件已经有长达T的时间未从网卡MAC层接收到网络数据,说明网络出现了故障,结束。

[0076] 采用本发明可以达到如下技术效果:

[0077] 1) 本发明第一步构建了基于FPGA硬件加速的NFV分类器,该系统由软件部分和FPGA硬件部分组成。软件部分直接与NFV架构中的NFV管理与编排器相连,使得该分类器可以很好的嵌入在NFV技术架构中,根据NFV管理编排的要求完成网络流量分类工作。

[0078] 2) 本发明在FPGA中实现的报文接收部件、流表查询部件、报文处理部件和报文发送部件配合完成第三步网络流量分类,实现了NFV分类器的硬件加速,通过硬件加速提高了NFV分类器对报文流的分类速度。

附图说明

[0079] 图1是ETSI提出的NFV体系结构图。

[0080] 图2是本发明总体流程图。

[0081] 图3是本发明第一步构建的NFV分类器逻辑结构图。

具体实施方式

[0082] 图1是ETSI提出的NFV体系结构图,从图1可清楚地看出NFV分类器在NFV体系结构中的位置。NFV分类器是整个NFV系统网络流量的总入口,并负责将不同的网络流量分发至对应的服务功能链。在NFV体系结构中,分类器往往成为整个系统的性能瓶颈。NFV分类器从NFV管理与编排器接收服务功能链管理数据,将其转换为内置的网络流分类规则;从NFV基础设施接收网络流数据,根据内置的网络流分类规则完成网络流分类;将分类后的网络流数据分发至不同的服务功能链(SFC)。

[0083] 图2是本发明总体流程图。如图2所示,本发明包括以下步骤:

[0084] 第一步,构建基于硬件加速的NFV分类器。NFV分类器如图3所示,由软件部分和硬件部分组成,其中软件部分包括流表配置部件,硬件部分包括流表管理部件,流表存储器,报文接收部件,流表查询部件,报文处理部件,报文发送部件。

[0085] 流表配置部件为软件实现的功能模块,部署运行于NFV网络环境的操作系统中;流表管理部件、报文接收部件、流表查询部件、报文处理部件、报文发送部件为在网卡FPGA中实现的功能模块,部署运行于网卡FPGA中;流表存储器为网卡FPGA中实现的存储空间。

[0086] 流表配置部件与流表管理部件、NFV管理与编排器相连,它从NFV管理与编排器中接收NFV服务功能链管理数据(命名为sfc_struct),并将服务功能链管理数据sfc_struct转换为流表管理数据,将流表管理数据发送给流表管理部件。

[0087] sfc_struct包括7个域:服务功能链管理命令(即sfc_cmd),服务功能链编号(sfc_id),服务功能链管理数据的报文原始目的IP地址(即sfc_dip1)、报文原始端口号(即sfc_dport1),服务功能链管理数据对应服务功能链报文的目的IP地址(即sfc_dip2)、服务功能

链报文的目的端口号(即sfc_dport2),服务功能链管理数据对应的网络协议号(即sfc_proto)。以下均以“->”表示sfc_struct的某个域,如sfc_struct->sfc_id表示sfc_struct的sfc_id。

[0088] 每个流表管理数据(命名为flow_struct)包括7个域:流表管理命令(即flow_cmd),流表编号(flow_id),流表管理数据的报文原始目的IP地址(即flow_dip1)、报文原始端口号(flow_dport1),流表处理后报文的目的IP地址(即flow_dip2)、流表处理后报文目的端口号(flow_dport2),流表管理数据对应网络协议号(即flow_proto)。以下均以“->”表示flow_struct的某个域,如flow_struct->flow_id表示flow_struct的flow_id。

[0089] 流表配置部件通过数组flowid_used[FLOWN]和链表数组sfc_flow[SFCN]来维护服务功能链编号和流表编号之间的关系。

[0090] 数组flowid_used的第i个元素flowid_used[i] ($i=1,2,\dots,FLOWN$) 取值0或者1。flowid_used[i]取值为0时,表示流表编号i未被使用;flowid_used[i]取值为1时,表示流表编号i已被使用。其中FLOWN表示NFV分类器可支持的最大流表数量,根据NFV网络需求来确定。

[0091] 链表数组sfc_flow的第j个元素sfc_flow[j] ($j=1,2,\dots,SFCN$) 为1个链表,该链表存储属于编号为j的服务功能链的所有流表的编号,链表中一个元素为属于服务功能链j的一个流表的编号其中SFCN表示NFV分类器可支持的最大服务功能链数量,根据NFV网络需求来确定。

[0092] 服务功能链管理命令和流表管理命令有2种取值:0,1。服务功能链管理命令取值0表示增加服务功能链管理数据;取值1表示删除服务功能链管理数据。流表管理命令取值0表示增加流表管理数据;取值1表示删除流表管理数据。服务功能链管理命令一般和流表管理命令取值相同。

[0093] 流表管理部件与流表配置部件和流表存储器相连,它从流表配置部件接收流表管理数据,将流表管理数据转化成流表数据结构后存储到流表存储器。

[0094] 流表存储器与流表管理部件和流表查询部件相连,是网卡内FPGA中的数据存储单元,用于存储流表数据结构。流表数据结构包括3个域:流表编号(即flow_id),流表处理报文目的IP地址(即flow_dip2)和流表处理后报文目的端口号(即flow_dport2)。

[0095] 报文接收部件与以太网接口(网卡MAC层)和流表查询部件相连,它从以太网接口接收外部输入报文,将输入报文发送给流表查询部件。

[0096] 流表查询部件与报文接收部件、报文处理部件和流表存储器相连,它从报文接收部件接收输入报文,从输入报文中提取出五元组信息(源IP地址,目的IP地址,源端口,目的端口,协议号),对五元组信息计算哈希值,并根据哈希值从流表存储器中查找对应的流表数据结构。查到后,将输入报文和流表数据结构一并发送给报文处理部件。

[0097] 报文处理部件与流表查询部件和报文发送部件相连,它从流表查询部件接收输入报文和对应的流表数据结构,根据流表规则处理报文,并将处理后的报文发送给报文发送部件。

[0098] 报文发送部件与报文处理部件相连,并通过网卡驱动和协议栈与NFV服务功能链相连,它从报文处理部件接收处理后的报文,将处理后报文发送给NFV服务功能链。

[0099] 第二步,NFV分类器按2.1~2.5的流程(2.1~2.5步由流表配置部件、流表管理部

件和流表存储器配合工作)进行初始化,监听NFV管理与编排器并从NFV管理与编排器接收服务功能链管理数据,完成报文流表的管理;同时NFV分类器按2.6~2.11的流程(2.6~2.11步由报文接收部件、流表查询部件、报文处理部件以及报文发送部件配合完成)监听网卡MAC层,接收网卡MAC层发送的输入报文(以太网报文),根据输入报文的流表数据结构完成报文分类处理,并发送给NFV服务功能链。

[0100] 具体步骤如下:

[0101] 2.1流表配置部件进行初始化。具体步骤如下:

[0102] 2.1.1根据NFV网络需求确定FLOWN和SFCName的值,可选的,设置FLOWN=1000, SFCN=100。

[0103] 2.1.2流表配置部件初始化数组flowid_used的每个元素为0,即令flowid_used[1],...,flowid_used[i],...,flowid_used[FLOWN]均=0, $i=1,2,\dots, \text{FLOWN}$ 。

[0104] 2.1.3流表配置部件初始化链表数组sfc_flow的每个元素为空,即令sfc_flow[1],...,sfc_flow[j],...,sfc_flow[SFCN]均为空, $j=1,2,\dots, \text{SFCN}$ 。

[0105] 2.1.4流表配置部件初始化时间 t =当前操作系统时间;

[0106] 2.2流表配置部件监控NFV管理与编排器是否发送了服务功能链管理数据,如果发送了服务功能链管理数据,转步骤2.3;如果没有发送,转步骤2.2.1;

[0107] 2.2.1获取当前操作系统时间 t_1 ,若 $(t_1-t)<T$,转2.2继续监控;若 $(t_1-t)\geq T$,转第三步。所述 T 为超时阈值,一般设置为30分钟。

[0108] 2.3流表配置部件更新 t 为当前操作系统时间,接收服务功能链管理数据sfc_struct,并根据服务功能链管理数据sfc_struct更新数组flowid_used和sfc_flow,并构建流表管理数据,具体步骤如下:

[0109] 2.3.1从数组flowid_used中找出第一个为0的元素,如果所有元素均为1,则找出第一个元素,方法为:

[0110] 2.3.1.1定义循环变量 i_1 ,令 $i_1=1$ 。

[0111] 2.3.1.2如果flowid_used[i_1]=0,转步骤2.3.2,否则转步骤2.3.1.3。

[0112] 2.3.1.3令 $i_1=i_1+1$,如果 $i_1\leq \text{FLOWN}$,转步骤2.3.1.2,否则转步骤2.3.1.4。

[0113] 2.3.1.4令 $i_1=1$,转步骤2.3.2。

[0114] 2.3.2令flowid_used[i_1]=1。

[0115] 2.3.3将 i_1 插入链表sfc_flow[sfc_struct->sfc_id]中。其中sfc_struct->sfc_id为sfc_struct的sfc_id,sfc_flow[sfc_struct->sfc_id]为链表数组sfc_flow的第sfc_struct->sfc_id个元素。

[0116] 2.3.4定义流表管理数据flow_struct,令:

[0117] flow_struct->flow_id= i_1 ;

[0118] flow_struct->flow_cmd=sfc_struct->sfc_cmd;

[0119] flow_struct->flow_dip1=sfc_struct->sfc_dip1;

[0120] flow_struct->flow_dport1=sfc_struct->sfc_dport1;

[0121] flow_struct->flow_dip2=sfc_struct->sfc_dip2;

[0122] flow_struct->flow_dport2=sfc_struct->sfc_dport2;

[0123] flow_struct->flow_proto=sfc_struct->sfc_proto;

- [0124] 2.4流表配置部件将flow_struct发送给流表管理部件。
- [0125] 2.5流表管理部件从流表配置部件接收flow_struct,根据flow_struct构建流表数据结构,并更新流表存储器。具体步骤如下:
- [0126] 2.5.1利用CRC32算法(见www.backplane.com/matt/crc64.html)对flow_struct->flow_dip1、flow_struct->flow_dport1和flow_struct->flow_proto计算哈希值,令为flow_hash。
- [0127] 2.5.2如果flow_struct->flow_cmd为0,转步骤2.5.3,否则转步骤2.5.4。
- [0128] 2.5.3将flow_struct->flow_id、flow_struct->flow_dip2、flow_struct->flow_dport2依次存入流表存储器中第flow_hash个流表数据结构,分别作为第flow_hash个流表数据结构的flow_id、flow_dip2和flow_dport2。转步骤2.2。
- [0129] 2.5.4将流表存储器中第flow_hash个流表数据结构的数据全部设置为0。转步骤2.2。
- [0130] 2.6报文接收部件初始化时间 $t_2=0$,并开始计时;
- [0131] 2.6.1报文接收部件监控网卡MAC层是否有输入报文到达,如果有,转步骤2.7,如果没有,则转步骤2.6.2。
- [0132] 2.6.2获取当前计时时间 t_3 ,若 $(t_3-t_2)<T$,转2.6.1继续监控;若 $(t_3-t_2)\geq T$,转第三步。
- [0133] 2.7报文接收部件更新 t_2 为当前计时时间,从网卡MAC层接收输入报文(命名为pkt),将pkt发送给流表查询部件。
- [0134] 2.8流表查询部件从报文接收部件接收pkt,根据pkt在流表存储器中查找匹配的流表数据结构,具体步骤如下:
- [0135] 2.8.1从pkt中提取报文目的IP地址pkt_dip,目的端口pkt_dport和协议号pkt_proto。
- [0136] 2.8.2利用CRC32算法对pkt_dip、pkt_dport和pkt_proto计算哈希值pkt_hash。
- [0137] 2.8.3在流表存储器的第pkt_hash位置处分别提取出流表数据结构的流表编号(记录为pkt_fid)、流表处理报文目的IP地址(记录为pkt_fip)、流表处理后报文目的端口号(记录为pkt_fport)。
- [0138] 2.8.4如果pkt_fid为0,将pkt转发至报文发送部件,转步骤2.11。否则,将pkt以及pkt_fip、pkt_fport转发至报文处理部件,转步骤2.9。
- [0139] 2.9报文处理部件从流表查询部件接收pkt、pkt_fip、pkt_fport,修改报文pkt目的IP地址、计算并更新IP校验和,得到报文pkt1,并将pkt1发送给报文发送部件。具体步骤如下:
- [0140] 2.9.1报文处理部件将报文pkt的目的IP地址修改为pkt_fip,将报文pkt的目的端口号修改为pkt_fport,得到报文pkt1。
- [0141] 2.9.2报文处理部件计算并更新报文pkt1的IP校验和(计算方法参见RFC1071)。
- [0142] 2.9.3报文处理部件将处理后的报文pkt1发送给报文发送部件。
- [0143] 2.10报文发送部件从报文处理部件接收pkt1,并将pkt1根据pkt1的目的IP地址转发至对应的NFV服务功能链,完成报文分类。转步骤2.6.1继续监听接收输入报文。
- [0144] 2.11报文发送部件从流表查询部件接收报文pkt,并将pkt根据pkt目的IP地址转

发至对应的NFV服务功能链,完成报文分类。转步骤2.6.1继续监听接收输入报文。

[0145] 第三步,此时表示NFV分类器已经有长达T的时间未从NFV管理与编排器接收到输出服务功能链管理数据,或报文接收部件已经有长达T的时间未从网卡MAC层接收到网络数据,说明网络出现了故障,结束。

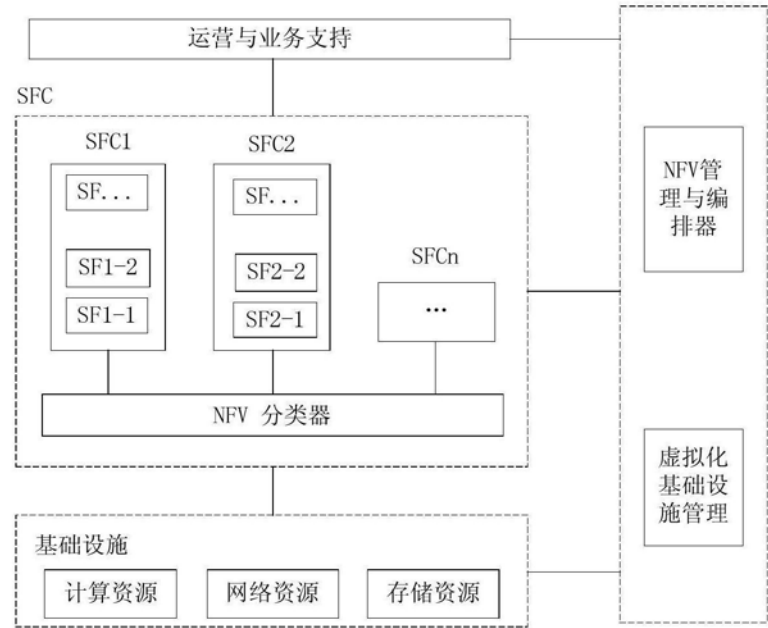


图1

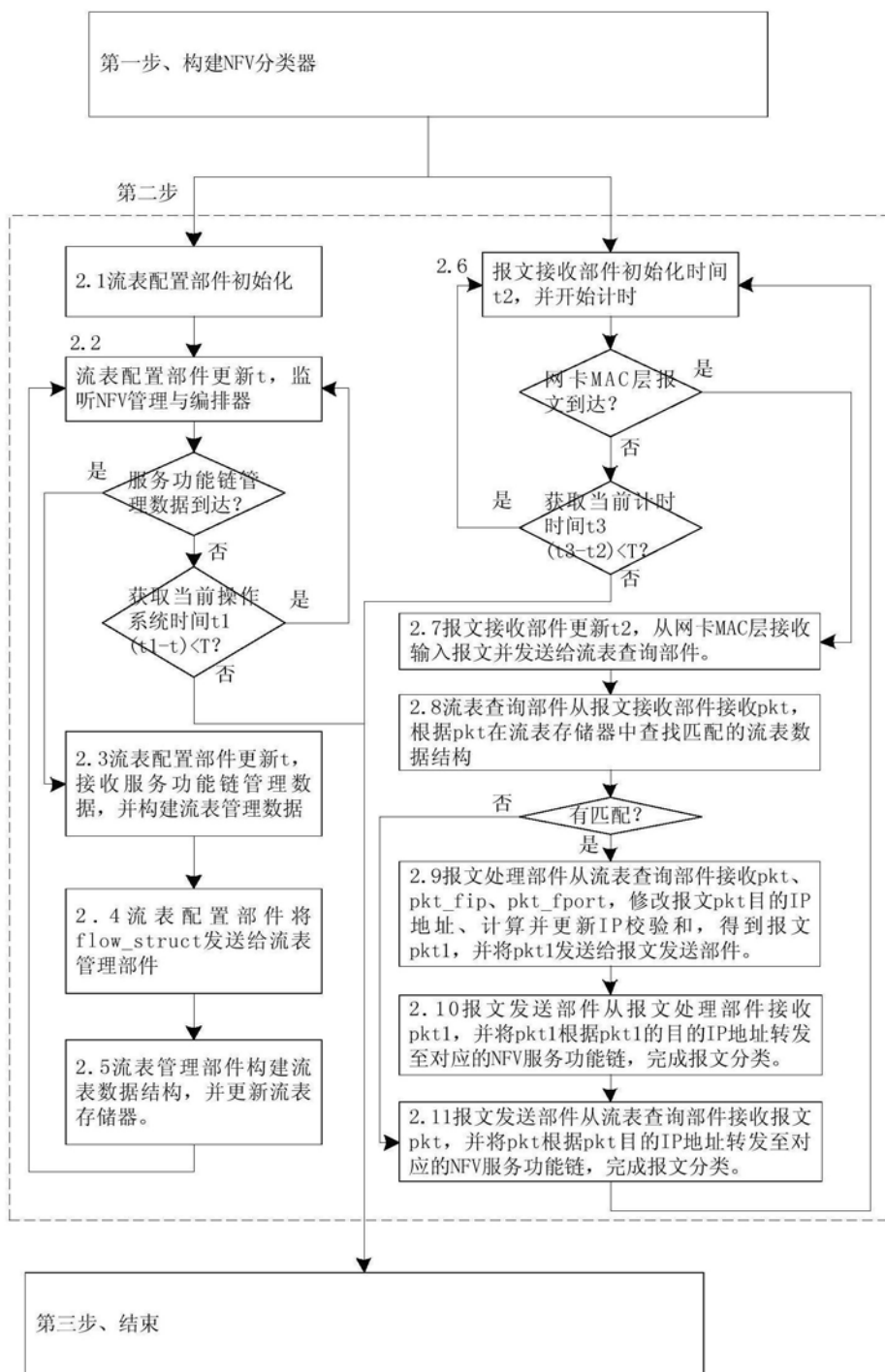


图2

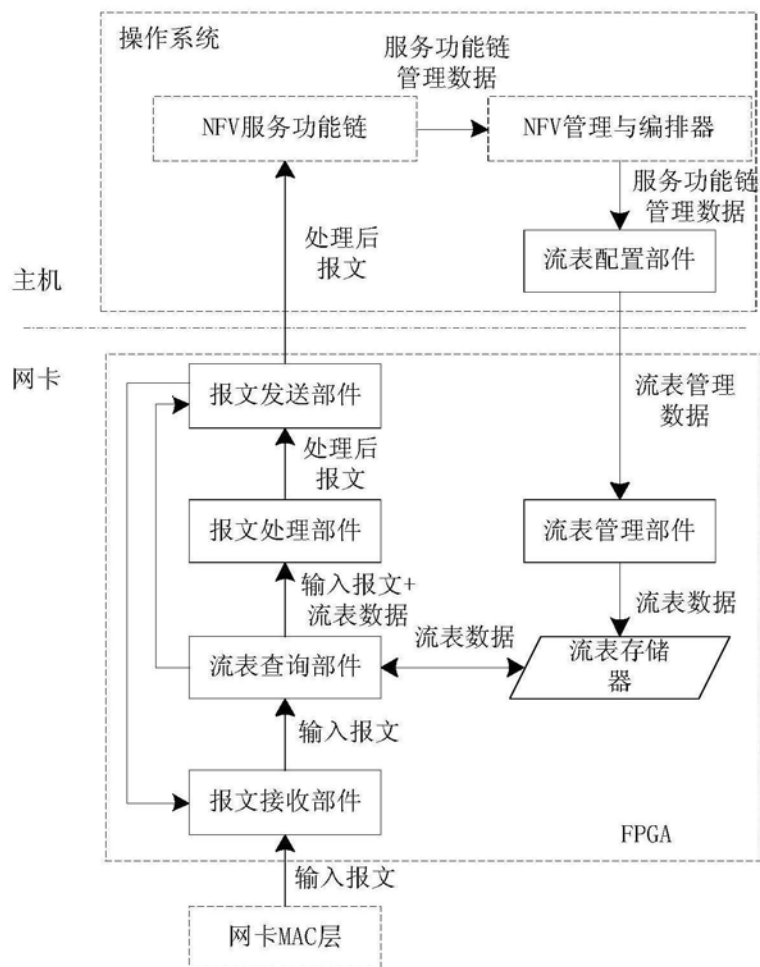


图3