面向软件定义核心网的 OpenFlow 分组转发优先制排队模型研究

熊 兵[1,2],左明科[1,2],黎 维[1,2],王 进[1,2]
(1. 长沙理工大学计算机与通信工程学院,湖南长沙 410114;
2. 长沙理工大学综合交通运输大数据智能处理湖南省重点实验室,湖南长沙 410114)

摘 要: 软件定义网络(Software-Defined Networking,SDN)作为一种数据转发与控制逻辑相解耦、并开放底层编程接口的创新网络架构,为降低核心网的部署运营成本、提升应用业务性能提供了全新的解决思路. 然而,在 SDN 架构下,逻辑上集中的控制平面容易出现性能瓶颈,进而加大分组转发时延,因此有必要理解其分组转发性能特性. 为此,本文首先介绍了软件定义核心网的典型部署场景,分析了控制平面的 Packet-in 消息到达过程和数据平面的分组到达过程,进而应用 $M/M/n/m$ 和 $M/M/1/m$ 排队模型分别刻画控制器集群的 Packet-in 消息处理过程和 OpenFlow 交换机的分组处理过程. 在此基础上,建立 OpenFlow 分组转发优先制排队模型,进而推导出不同优先级的分组转发时延及其累积分布函数 CDF. 最后,借助控制器性能测量工具 OFsuite_Performance 进行实验评估,结果表明:与现有模型相比,本文所提的 $M/M/n/m$ 模型更能准确估计控制器集群的实际性能. 同时,采用数值分析的方法对比了多种情况下不同优先级的分组转发时延及 CDF 曲线,为软件定义核心网的实际应用部署提供有效参考.

关键词: 软件定义核心网; 分组转发性能; 优先制排队模型; SDN 控制器集群; OpenFlow 交换机
中图分类号: TP393.0 **文献标识码:** A **文章编号:** 0372-2112 (2019)10-2040-10
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2019.10.004

A Prioritized Queueing Model of OpenFlow Packet Forwarding in Software-Defined Core Networks

XIONG Bing[1,2], ZUO Ming-ke[1,2], LI Wei[1,3], WANG Jin[1,2]
(1. *School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha, Hunan* 410114, *China*;
2. *Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Changsha University of Science and Technology, Changsha, Hunan* 410114, *China*)

Abstract: As an innovative network architecture decoupling data forwarding and control logic, and opening underlying programming interfaces, software-defined networking (SDN) provides a novel solution to reduce deployment and operation costs and improve business application performance in core networks. However, logically centralized control plane under the SDN architecture is prone to performance bottlenecks, and increases packet forwarding delay. Thus it is necessary to understand the characteristics of its packet forwarding performance. To this end, we first introduce typical deployment scenarios of software-defined core networks, and analyze Packet-in message arrival process in its control plane and packet arrival process in its data plane. Then the $M/M/n/m$ and $M/M/1/m$ queueing models are respectively applied to depict Packet-in message processing process of its controller clusters and packet processing process of its OpenFlow switches. On this basis, we establish a prioritized queueing model of OpenFlow packet forwarding, and derive packet forwarding delays of different priorities and its cumulative distribution function. Finally, experimental evaluation in virtue of the controller performance measurement tool OFsuite_Performance shows that our proposed $M/M/n/m$ model can accurately estimate actual performance of controller clusters compared with existing models. Meanwhile, we contrast packet forwarding delays of different pri-

收稿日期:2018-07-26;修回日期:2019-03-01;责任编辑:孙瑶
基金项目:国家自然科学基金(No. 61502056);湖南省自然科学基金(No. 2015JJ3010);湖南省教育厅资助科研项目(No. 15B009);湖南省研究生科研创新项目(No. CX2018B567, No. CX2017B487)

orities in various cases and their CDF curves by numerical analysis, which provides effective references for practical deployments of software-defined core networks.

Key words: software-defined core networks; packet forwarding performance; prioritized queueing models; SDN controller clusters; OpenFlow switches

1 引言

软件定义网络(Software-Defined Networking, SDN)作为一种新型网络架构,将逻辑控制与数据转发相解耦,显著地提升了网络的灵活性、可管控性和可编程能力,被认为是未来网络领域最有前景的发展方向之一[1],成为近年来未来网络领域广受关注的研究热点. 核心网将各种类型的接入网互联起来,实现大范围网络的数据交换,导致网络流量汇聚程度高,往往难以管控和优化. 对此,软件定义核心网(Software-Defined Core Networks, SDCN)应运而生,为提高数据传输效率、改进应用性能、降低部署成本提供全新的解决思路.

在 SDCN 架构下,控制器负责建立全局网络视图,对整个网络的数据交换设备进行集中式控制与管理,容易产生性能瓶颈,进而影响分组转发性能. 例如:NOX 控制器每秒大约可以处理 30000 条流请求[2],而一个拥有 100 台交换机的网络在最坏情况下每秒可产生 100 万条流请求[3]. 虽然可以采用控制器集群增强处理能力[4],但由于分组转发过程可能需要等待控制器下发流规则,容易导致分组转发时延过长,往往难以满足网络应用的实时性需求[5,6]. 因此,有必要研究并理解 SDCN 网络分组转发性能与局限,为软件定义核心网的实际部署提供指导依据.

为评估 OpenFlow 控制器性能,目前已有研究人员开发了 Cbench[7]、OFsuite_Performance[8] 和 hcprobe[9] 等基准工具. 利用这些基准工具,可测量控制器的响应时间、最大吞吐量和时延等关键性能指标,以便于设计者在构建 SDN 网络时选择合适的控制器. 虽然实验测量与模拟的方法广泛运用于性能评估,但模拟实验往往需要购置昂贵的网络设备,花费大量的实验测量时间. 而解析建模可针对给定的网络架构,给出其性能参量的闭式解,快速计算出对应的近似估计值,具有独特的优势.

目前,已有许多学者运用解析建模方法对控制器性能展开研究. 一些研究人员利用网络演算评估 SDN 控制器在最坏情况下的性能局限性. Azodolmolky 等人[10,11]首次应用网络演算理论对 SDN 控制器和交换机的行为进行建模分析,包括时延和队长边界、缓存长度,可为网络设计者快速获取整个 SDN 网络部署的性能视图. Osgouei 和 Koohanestani 等人[12]针对 SDN 虚拟化场景,提出了一种基于网络演算的解析性能模型,计算了每个虚拟网络的服务曲线,求解了虚拟 SDN 控制器的时延上界. 进一步,采用网络演算框架,提供了一个基于纯 OpenFlow 标准实现的 SDN 解析模型,研究了网络规模、流量特性、流表大小等 SDN 参数之间的相互影响,并计算了 OpenFlow 交换机的时延上限[13]. Bozakov 等人[14]采用基于测量的方法估计相应的服务曲线,进而通过使用一个恰当参数化的每交换令牌桶过滤器扩展接口,允许操作员配置传输控制消息的延迟界限,从而使 SDN 应用程序的行为更加可预测. Lin 等人[15]提出了基于随机网络演算理论的解析模型,评估交换机到控制器的性能,以真实测量控制器与交换机之间的端到端网络性能. 黄军[16]等人针对当前 QoS 提供机制未充分考虑多媒体数据流的异构性和性能差异的问题,应用网络演算,结合优先排队和通用处理器共享机制,建立了一个混合调度模型,求解了最坏情况下的端到端时延以及排队积压长度,可满足 SDN 多媒体应用的不同 QoS 建模需求.

不同于网络演算模型,排队模型更专注于 SDN 控制器处于平衡状态时的平均性能表现. Jarschel 等人[17]采用排队论的方法对 OpenFlow 架构建模为基于反馈的排队系统模型,交换机建模为 $M/M/1$ 模型,控制器建模为 $M/M/1$-S 模型,并测量了不同负载下的报文转发时延和丢包率. 但这种排队模型只考虑了单台 OpenFlow 交换机和控制器相连的情况,无法适用于多台 OpenFlow 交换机的情况. 左青云等人[18]采用多到达单服务排队模型对控制平面建模为 $M^m/M/1/B$ 排队模型,计算出控制器处于不同网络规模和负载下的排队时延. 然而,批量到达模型并不能准确刻画多个交换机发送的流请求过程. Mahmood 等人[19]针对 SDN 控制器负责数据平面多个节点的情形,运用 Jackson 网络对数据交换节点和控制器均建模为 $M/M/1$ 队列,进而提出了一个 OpenFlow 网络性能解析模型. AlGhadhban 等人[20]提出了 SDN 流安装过程的数学模型,考虑了主动/被动流安装模式对匹配概率的影响,推导评估了系统容量和阻塞概率,发现无论控制器服务时间使用哪种分布,系统响应时间 T 的分布特征保持稳定,形状接近大参数值的指数分布. Sood 等人[21]忽略 OpenFlow 交换机与控制器之间的交互,利用 $M/Geo/1$ 排队模型分析了 OpenFlow 交换机的流表大小、包到达速率、规则数量、规则位置等关键因素. 上述模型针对的均是一般性或假设性的 SDN 网络场景,与实际部署场景存在较大的差距.

熊兵等人[22]针对接入网场景,将 OpenFlow 交换机

的分组转发过程建模为 $M^X/M/1$ 排队模型,将 SDN 控制器的 packet-in 消息处理过程建模为 $M/G/1$ 排队模型,进而建立 OpenFlow 网络分组转发排队性能模型,并推导出分组转发时延的解析式. 李泰新等人[23]针对延时容忍网络的存储与转发机制,将软件定义卫星网络 SDSN 中的控制器建模为 $M/M/1$ 排队模型,运用 Jackson 网络将数据平面建模为 $M/M/1$ 排队模型,能够有效分析 SDSN 性能. 以上模型主要针对单个控制器的网络场景,没有考虑到 SDN 实际部署往往采用多控制器的情形. 对此,姚龙等人[24]将交换机到控制器的流建立请求建模为成批到达过程,形成排队模型 $M^k/M/n$,并推导出流的平均服务时间. 然而,批量到达模型并不能准确刻画多个交换机发送的流请求过程. 付永红等人[25]提出了一个集中式结构的多控制器休眠模型,将 Packet-in 消息数量和控制器工作状态建模为二维排队模型,进而建立控制器和消息时延的总开销函数,并应用遗传算法搜寻各个变量的最优值,以最小化系统开销. 本文针对核心网部署场景,在对 OpenFlow 交换机的分组处理过程和 SDN 控制器的 Packet-in 消息处理过程分别进行排队建模的基础上,提出一个 OpenFlow 分组转发排队性能模型,并给出主要网络性能参数,为软件定义核心网的实际部署提供参考依据.

2 软件定义核心网

在通信网络中,核心网控制着所有接入网以及各种业务系统的数据交换,是整个通信网络的枢纽,对网络服务质量起着决定性作用. 随着移动互联网的快速发展,新业务大量部署,新网元急剧增加,核心网的复杂度越来越高,使得网络资源无法统一编排调度,网络规模无法快速灵活调整. 通过引入 SDN 技术,可简化业务的部署,加快业务上线速度,实现网络资源的统一管理和灵活调度,可有效降低运营商的设备采购成本和网络运营成本. 软件定义核心网具有开放、灵活、可编程的特点,可为 ISP(Internet Service Provider)提供快捷、高效、智能的网络管理,已经成为近年来 ISP 发展的一大趋势.

软件定义核心网 SDCN 的典型部署场景如图 1 所示. 在用户接入层,网络用户可通过移动通信网、企业网、校园网、家庭接入网等各种网络接入到 ISP. ISP 将各种接入网通过 OpenFlow 交换机互联起来,构成核心交换层,实现所有网络流量的汇聚与交换. 核心交换层的 OpenFlow 交换机直接或间接连接到逻辑控制层的 SDN 控制器,以便统一调度和管理[26]. 在 SDCN 场景中,SDN 控制器需要集中管理核心交换层的所有 OpenFlow 交换机,通过链路发现、拓扑管理、路径计算建立全局网络视图,进而制定分组转发策略,并为上层应用提供资源管理、业务编排、数据可视化等服务.

在上述 SDCN 场景中,OpenFlow 交换机每收到一个网络分组,先提取其关键字段,然后查找 OpenFlow 流表. 若查找成功,则依据命中的流表项中的动作集转发处理分组;否则将该分组封装成流安装请求,即 Packet-in 消息,并提交给 SDN 控制器. 控制器根据全局网络视图生成流规则,并以 Packet-out 消息或 Flow-mod 消息的形式下发给相应的 OpenFlow 交换机. OpenFlow 交换机将该流规则添加到流表中,并据此转发处理该流的所有分组. 在该过程中,由于逻辑上集中的控制器容易出现性能瓶颈,导致流安装时延过大,进而影响分组转发性能.

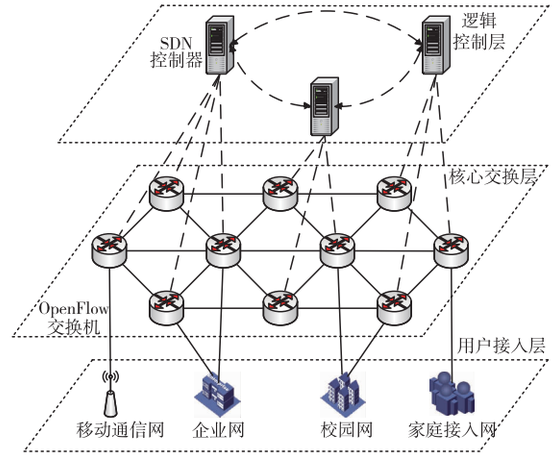


图1 软件定义核心网SDCN部署场景

3 SDN 控制器集群排队模型

在软件定义核心网中,网络用户数量大且分布广,其网络行为将产生大量的数据包流,进而通过接入网汇聚到核心交换层的 OpenFlow 交换机. 对于每条流的首个包,若 OpenFlow 流表中没有相应的流规则,则 OpenFlow 交换机将该包封装成 Packet-in 消息发送给 SDN 控制器集群. 在核心交换层,每个 OpenFlow 交换机由于汇聚大量网络包流,因而将会持续不断地产生 Packet-in 消息. SDN 控制器集群由于管控核心交换层的众多 OpenFlow 交换机,因而将汇聚大量 Packet-in 消息,形成队列等待处理. 图 2 展示了软件定义核心网中 Packet-in 消息的产生、汇聚和排队过程.

核心网测量研究结果表明:在骨干链路中,网络包流之间趋于相互独立,流到达过程往往服从泊松分布[27,28]. 这主要是因为在核心网中,网络包流的来源分布广,汇聚程度高. 根据 SDCN 分组处理过程可知,OpenFlow 交换机的流到达过程与 Packet-in 消息产生过程具有对应关系. 假设 OpenFlow 交换机中 SDN 控制平面预安装的流规则随机对应到流到达过程,根据泊松

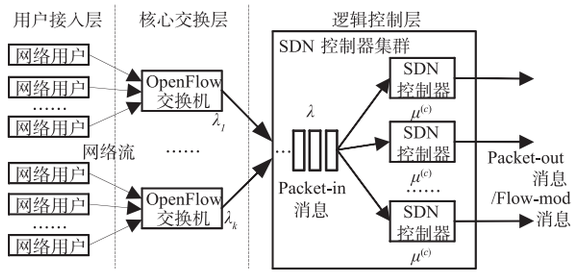


图2　Packet-in消息的产生、汇聚和排队过程

流性质可知,每台交换机发送的 Packet-in 消息流仍为泊松流. 假设所有交换机发送的 Packet-in 消息流相互独立,根据泊松流的可加性,可知控制器集群的 Packet-in 消息到达过程仍服从泊松分布.

当 Packet-in 消息到达控制器集群时,若存在空闲的控制器,则直接进行处理;否则,按到达顺序加入 Packet-in 消息队列等待被处理. 由于 Packet-in 消息的处理过程相互独立,且无记忆性,故其处理时间可视为近似服从负指数分布. 假设集群中所有控制器的处理能力相同,且处理 Packet-in 消息彼此独立. 由于 Packet-in 消息队列长度受限,可将控制器集群的 Packet-in 消息处理过程建模为 $M/M/n/m$ 排队模型.

在该模型中,假设集群中共有 n 个控制器,Packet-in 消息缓存容量为 m. 集群共管理 k 台 OpenFlow 交换机,其中第 i 台交换机的流到达过程服从参数为 $\lambda_i^{(f)}$ 的泊松分布. 对于任意流,假定对应的流规则没有预安装的概率为 q_i,则集群的 Packet-in 消息到达速率为 $\lambda^{(c)} = \sum_{i=1}^{k} q_i\lambda_i^{(f)}$. 又设控制器对各个 Packet-in 消息的处理过程相互独立,处理时间服从参数为 $\mu^{(c)}$ 的负指数分布. 记 $\rho = \lambda^{(c)}/n\mu^{(c)} < 0$,依据排队论可得,Packet-in 消息在集群中的平均逗留时间如式(1)所示.

$$E[T^{(c)}] = \frac{1}{\mu^{(c)}} + n^n\rho^{n+1}p_0 \cdot \frac{1-(m-n+1)\rho^{m-n}+(m-n)\rho^{m-n+1}}{n!(1-\rho)^2\mu^{(c)}[n-\sum_{k=0}^{n-1}(n-k)p_k]} \tag{1}$$

其中,$p_0 = \left(\sum_{k=0}^{n-1}\frac{(n\rho)^k}{k!} + \frac{(n\rho)^n(1-\rho^{m-n+1})}{n!(1-\rho)}\right)^{-1}$,

$$p_k = \begin{cases} \frac{(n\rho)^k}{k!}p_0, & 0 \leq k < n \\ \frac{n^n\rho^k}{n!}p_0, & n \leq k \leq m \end{cases}$$

4　OpenFlow 交换机排队模型

在核心网中,用户接入层的大量网络分组汇聚到核心交换层的 OpenFlow 交换机,形成队列等待处理. 目前,核心网流量测量结果表明:网络流量尽管在较粗糙的时间尺度上具有明显的自相似性,但由于分组来源广、复用程度高,破坏了短期内分组之间的时间相关性,因而在细尺度上可近似为泊松流[28].

当交换机收到一个分组时,首先将其放到入端口队列. 分组进行处理时,从中提取关键字段得到流关键字,进而查找流表以找到对应的流表项. 若查找成功,则根据匹配流表项中的动作集转发该分组;否则,将分组信息发送给 SDN 控制器集群以获取相应的流规则,进而转发该分组. 因此,OpenFlow 交换机对每个分组的交换处理过程相互独立,处理时间无记忆性,可用负指数分布来进行刻画. 由于分组具有优先级,分组缓存队列容量受限,故可将 OpenFlow 交换机的分组交换过程建模为非强占优先制 $M/M/1/m$ 排队模型.

该模型描述如下:(a) 网络分组共分为 N 个优先级(第 1 级最高),第 $i(1 \leq i \leq k)$ 台 OpenFlow 交换机的第 $j(1 \leq j \leq N)$ 级分组到达过程服从参数为 $\lambda_{ij}^{(s)}$ 的泊松分布;(b) 第 i 台交换机对每级分组的交换处理速率相同,处理时间服从参数为 $\mu_i^{(s)}$ 的负指数分布;(c) 每个交换机按优先级高低顺序依次交换处理分组,分组处理过程相互独立,分组缓存容量为 r. 记 $\rho_i^{(s)} = \lambda_i^{(s)}/\mu_i^{(s)}$,$\rho_{ij}^{(s)} = \lambda_{ij}^{(s)}/\mu_i^{(s)}$,根据排队论可得,第 1 级和第 $l(1 < l \leq N)$ 级分组的平均逗留时间分别如式(2)和式(3)所示.

$$E[T_{i1}^{(s)}] = \frac{1}{\mu_i^{(s)}(1-(1-p_r)\rho_{i1})} + \frac{1}{\mu_i^{(s)}} \tag{2}$$

$$E[T_{il}^{(s)}] = \frac{1}{\mu_i^{(s)}(1-(1-p_r)\sum_{j=1}^{l}\rho_{ij}^{(s)})} \cdot \frac{1}{(1-(1-p_r)\sum_{j=1}^{l-1}\rho_{ij}^{(s)})} + \frac{1}{\mu_i^{(s)}}, l > 1 \tag{3}$$

其中,$p_k = \frac{1-\rho_i^{(s)}}{1-(\rho_i^{(s)})^{r+1}}(\rho_i^{(s)})^k, \rho_i^{(s)} \neq 1$

5　OpenFlow 分组转发性能模型

根据上述建立的 OpenFlow 交换机排队模型和 SDN 控制器集群排队模型,可将 OpenFlow 网络中的分组转发过程建模为如图 3 所示的排队模型.

在图 3 中,第 i 个 OpenFlow 交换机的第 j 级分组到达速率为 $\lambda_{ij}^{(s)}$,总到达速率为 $\lambda_i^{(s)} = \sum_{j=1}^{N}\lambda_{ij}^{(s)}$,分组处理速率为 $\mu_i^{(s)}$. 假定到达第 i 个交换机的分组属于新流的概率为 q_i,则第 i 个交换机的 Packet-in 消息发送速率为 $q_i\lambda_i^{(s)}$,进而可知控制器集群的 Packet-in 消息到达速率为 $\lambda^{(c)} = \sum_{i=1}^{k} q_i\lambda_i^{(s)}$. 此外,假定集群中每个控制器的

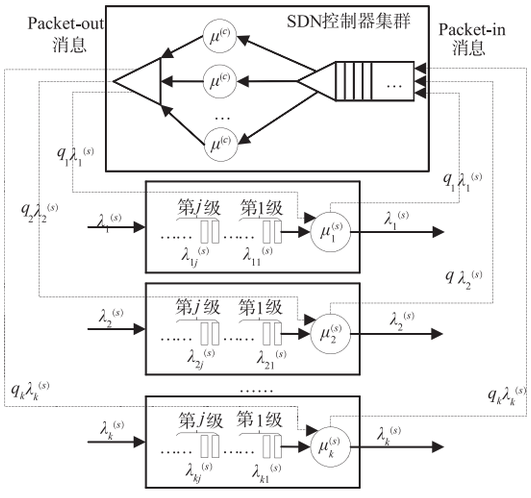


图3 OpenFlow分组转发排队性能模型

Packet-in 消息处理速率为 $\mu^{(c)}$.

在软件定义核心网中,传播时延和发送时延都极小,可忽略不计,因此本文主要考虑排队时延和处理时延,合称为逗留时间. 根据上述 OpenFlow 分组转发过程可知,分组通过第 i 个交换机的转发时间 W_i 可分成两种情况:交换机直接转发和控制器集群参与的间接转发. 前一种情况的分组转发时间即为分组在第 i 台交换机中的逗留时间 $T_i^{(s)}$;后一种情况的分组转发时间则包括分组在第 i 台交换机中的逗留时间 $T_i^{(s)}$ 和对应的 Packet-in 消息在控制器集群中的逗留时间 $T^{(c)}$. 因此,第 j 级分组通过第 i 个交换机的转发时间 W_{ij} 如式(4)所示.

$$W_{ij}=\begin{cases} T_{ij}^{(s)}, & \text{概率为 } 1-q_i \\ T_{ij}^{(s)}+T^{(c)}, & \text{概率为 } q_i \end{cases} \tag{4}$$

根据式(1)(3)和(4),可求得第 j 级分组通过第 i 个交换机的平均转发时间 $E[W_{ij}]$ 如式(5)所示. 进一步,可推导得分组转发时间 W_{ij} 的概率密度函数 PDF 和累计分布函数 CDF 如定理 1 所示.

$$\begin{aligned} E[W_{ij}] &= E[T_{ij}^{(s)}]+q_i(E[T^{(c)}]) \\ &= \frac{1}{\mu_i^{(s)}}+\frac{q_i}{\mu^{(c)}}+q_i n^n \rho^{n+1} p_0 \\ &\quad \cdot \frac{1-(m-n+1)\rho^{m-n}+(m-n)\rho^{m-n+1}}{n!(1-\rho)^2\mu^{(c)}\left[n-\sum_{k=0}^{n-1}(n-k)p_k\right]} \\ &\quad + \begin{cases} \dfrac{1}{\mu_i^{(s)}(1-(1-p_r)\rho_{ij})}, & j=1 \\ \dfrac{1}{\mu_i^{(s)}(1-(1-p_r)\sum_{k=1}^{j}\rho_{ik}^{(s)})} \\ \quad \cdot \dfrac{1}{1-(1-p_r)\sum_{k=1}^{j-1}\rho_{ik}^{(s)}}, & j>1 \end{cases} \end{aligned} \tag{5}$$

定理 1 对于 OpenFlow 网络,假定第 i 个交换机收到的分组属于新流的概率为 q_i,第 j 级分组在第 i 个交换机的逗留时间 $T_{ij}^{(s)}$ 和对应 Packet-in 消息在控制器集群中的逗留时间 $T^{(c)}$ 均服从负指数分布. 设 $\alpha_{ij}=1/E[T_{ij}^{(s)}]$,$\alpha_c=1/E[T^{(c)}]$,则第 j 级分组通过第 i 个 OpenFlow 交换机的转发时间 W_{ij} 的概率密度函数 PDF 和累积分布函数 CDF 分别如式(6)和式(7)所示.

$$W_{ij}(t)=\left(1-q_i\frac{\alpha_i}{\alpha_i-\alpha_c}\right)\alpha_i e^{-\alpha_i t}+q_i\frac{\alpha_i}{\alpha_i-\alpha_c}\alpha_c e^{-\alpha_i t} \tag{6}$$

$$\begin{aligned} \widetilde{W}_{ij}(t) &= \left(1-q_i\frac{a_i}{a_i-a_c}\right)(1-e^{-a_i t}) \\ &\quad +q_i\frac{a_i}{a_i-a_c}(1-e^{-a_i t}) \end{aligned} \tag{7}$$

证明 根据假设可得,第 j 级分组和 Packet-in 消息分别在第 i 个交换机和控制器集群的逗留时间的概率密度函数如式(8)所示. 由于第 i 个交换机收到的分组属于新流的概率为 q_i,结合 OpenFlow 网络分组转发过程,可得转发时间 W_{ij} 的拉普拉斯变换结果如式(9)所示. 进而进行拉普拉斯逆变换,可得到转发时间 W_{ij} 的概率密度函数 PDF 如式(10)所示. 最后对其积分,可得转发时间 W_{ij} 的累积分布函数 CDF 如式(7)所示.

$$f_{ij}(t)=a_i e^{-a_i t},f_c(t)=a_c e^{-a_i t} \tag{8}$$

$$W_{ij}(s)=\left(1-q_i\frac{a_i}{a_i-a_c}\right)\frac{a_i}{a_i+s}+q_i\frac{a_i}{a_i-a_c}\frac{a_c}{a_c+s} \tag{9}$$

$$W_{ij}(t)=\left(1-q_i\frac{a_i}{a_i-a_c}\right)a_i e^{-a_i t}+q_i\frac{a_i}{a_i-a_c}a_c e^{-a_i t} \tag{10}$$

证明完毕.

6 实验

6.1 控制器集群排队模型评估

为有效评估本文所提的控制器集群排队模型,实验采用 SDNCTC 开发的 OFsuite_Performance 工具测量 SDCN 控制器集群的实际性能参数. 该工具通过多块网卡连接到控制器集群,然后模拟多台 OpenFlow 交换机不断向控制器集群发送 Packet-in 消息,当收到控制器集群下发的对应 Packet-out 消息时,记录消息时间间隔作为 Packet-in 消息的响应时延. 实验通过模拟不同数量的交换机不断加大 Packet-in 消息发送速率,直至控制器集群的消息缓存耗尽. 在此过程中,最大的 Packet-out 消息下发速率,即为集群的 Packet-in 消息处理速率 $\mu^{(c)}$. 然后,在 Packet-in 消息发送速率达到处理速率 $\mu^{(c)}$ 之前,统计每个发送速率下 Packet-in 消息的平均响应时延,以检验排队模型的准确性.

本实验中,控制器采用 OpenDaylight Beryllium 版本,运行在 Linux 服务器上. 在 Linux 系统中,控制器的缓存大小一般默认为 87380B,而每个 Packet-in 消息的大小约为 148B,所以消息缓存容量 m 设为 512[29]. 利用 OFsuite_Performance 工具模拟核心网拓扑结构,设置每台交换机的 Packet-in 消息发送速率为 1000 个/s. 对于 1 台控制器,模拟的交换机数量从 1 依次增长到 14;对于 3 台控制器组成的集群,则交换机数量从 1 依次增长到 23. 每组实验进行 5 次迭代测试,计算 Packet-in 消息的平均响应时延. 表 1 给出了上述控制器集群的 Packet-in 消息处理速率测试结果.

表 1　控制器集群的 Packet-in 消息处理速率测试

	交换机数量/台	性能参数		
		Packet_in 速率/个/s	Packet_out 速率/个/s	平均响应时延/ms
(*a*) 1 台控制器	10	10000	10000	0. 913
	11	11000	11000	1. 331
	12	12000	12000	3. 152
	13	13000	11853	60. 35
	14	14000	11632	186. 04
(*b*) 3 台控制器	19	19000	19000	0. 735
	20	20000	20000	0. 873
	21	21000	21000	1. 052
	22	22000	21352	23. 70
	23	23000	21024	89. 21

从表 1(*a*)可以看出,当 Packet-in 消息发送速率不超过 12000 个/s 时,Packet-out 消息下发速率与 Packet-in 消息速率保持一致;当 Packet-in 消息发送速率超过 12000 个/s 时,Packet-out 消息下发速率将有所下降,同时测量得到的平均响应时延急剧上升. 因此,单控制器的 Packet-in 消息处理速率可近似为 12000 个/s. 同理,由表 1(*b*)可知,3 台控制器的 Packet-in 消息处理速率大约为 21500 个/s.

对于上述两种控制器集群,交换机数量 k 依次增加,使得 Packet-in 消息发送速率 $\lambda^{(c)}$ 逐步增加到处理速率 $\mu^{(c)}$. 将这些参数和消息缓存容量 m 代入式(1)和文献[22]中的平均逗留时间表达式,可得到 $M/M/n/m$ 和 $M^k/M/n$ 排队模型的 Packet-in 消息估计时延如图 4 所示. 同时,图 4 给出了 OFsuite_Performance 工具测量得到的 Packet-in 消息平均响应时延.

从图 4 可以看出,$M/M/n/m$ 模型的估计时延比 $M^k/M/n$ 模型更接近于工具测量时延. 时延随着交换机数量的增加而变大,这是因为 Packet-in 消息发送速率越高,控制器集群内的排队消息数量越多,排队等候时延越大. 工具测量时延整体上比 $M/M/n/m$ 模型的估计值大,这是因为工具测量时延除了包含 Packet-in 消息在控制器集群中的逗留时间外,还有发送时延和传播时延等. 此外,$M^k/M/n$ 模型的估计时延远大于工具测量时延,这是因为该模型将每批消息个数等同于交换机数量,而 $M^k/M/n$ 模型的估计时延与每批消息个数具有较强的正相关性[30].

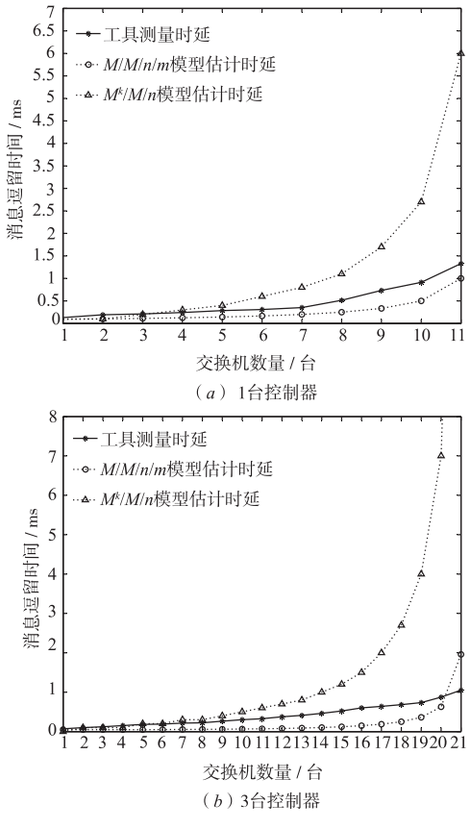


图4　不同排队模型估计时延的准确性对比

6.2　分组转发时延

由上可知,当集群由 3 台控制器组成时,Packet-in 消息处理速率为 $\mu^{(c)}=21500$ 个/s. 假定 OpenFlow 交换机的分组交换速率为 $\mu_{i(s)}=30000$ 个/s[22],分组到达速率 $\lambda_i^{(s)}=25000$ 个/s,分组属于新流的概率为 $q_i=0.04$[19],则单个交换机发送 Packet-in 消息的速率为 $\lambda_i^{(s)}*q_i=1000$ 个/s. 因此,可假定控制器集群管理 21 台交换机,则其 Packet-in 消息到达速率为 $\lambda^{(c)}=21000$ 个/s. 假设交换机的分组缓存容量为 $r=512$,到达的分组分为四个优先级,优先级从高到低的占比分别为 0. 125,0. 125,0. 25,0. 5. 将以上参数值分别代入式(1)(3)和(7),可得不同优先级分组的转发时延 CDF 对比如图 5 所示.

从图5可以看出,优先级越高,CDF值增长速度越

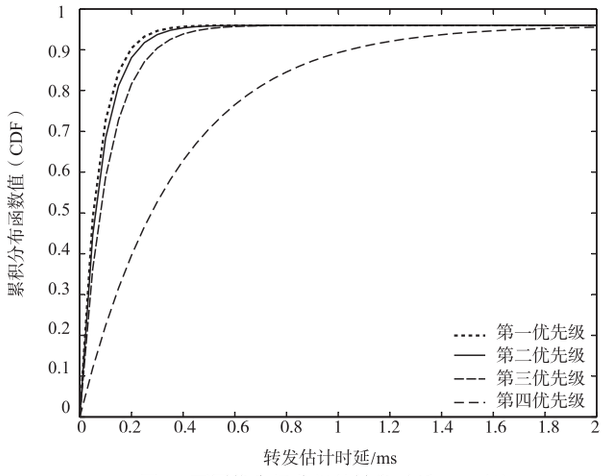
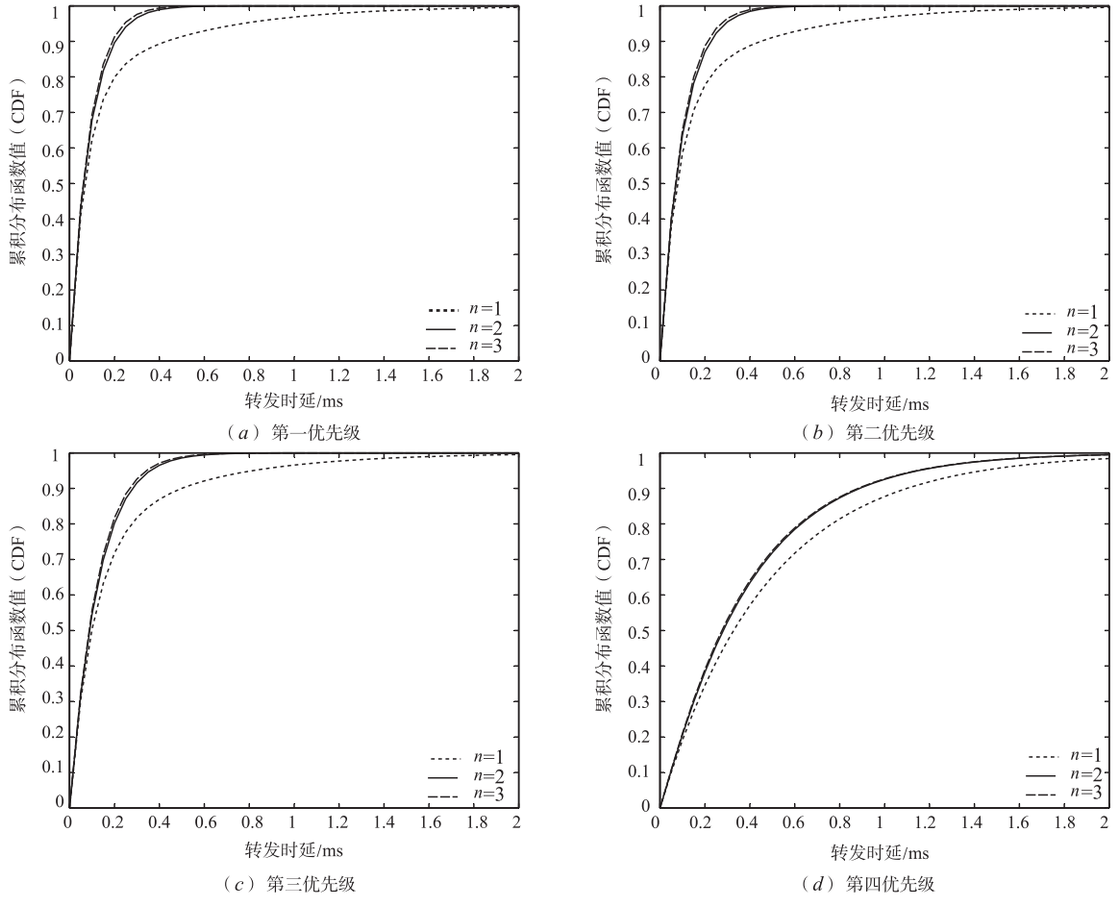


图5 不同优先级分组的转发时延CDF对比

快，而高优先级的 CDF 值相对接近，且明显高于最低优先级. 具体来说，从高到低的四个优先级，90% 分组分别可以在 0. 2ms、0. 25ms、0. 3ms、1. 1ms 实现转发，95% 分组则分别可以在 0. 35ms、0. 4ms、0. 5ms、1. 8ms 实现转发. 这意味着分组优先级越高，其平均转发时延越小，且高优先级分组的转发时延相差不大. 因此，对于实时性要求高的网络应用，将其分组设置一个合适的高优先级即可.

假定控制器集群分别由 1、2、3 台控制器组成，其 Packet-in 消息处理速率 $\mu^{(c)}$ 分别为 12000 个/s、17500 个/s 和 21500 个/s，分组属于新流的概率 $q_i = 0.2$[19]，其它参数同上. 将以上参数值分别代入式(1)(3)和(7)，可得四个优先级的分组转发时延 CDF 值如图 6 所示.



（a）第一优先级 （b）第二优先级

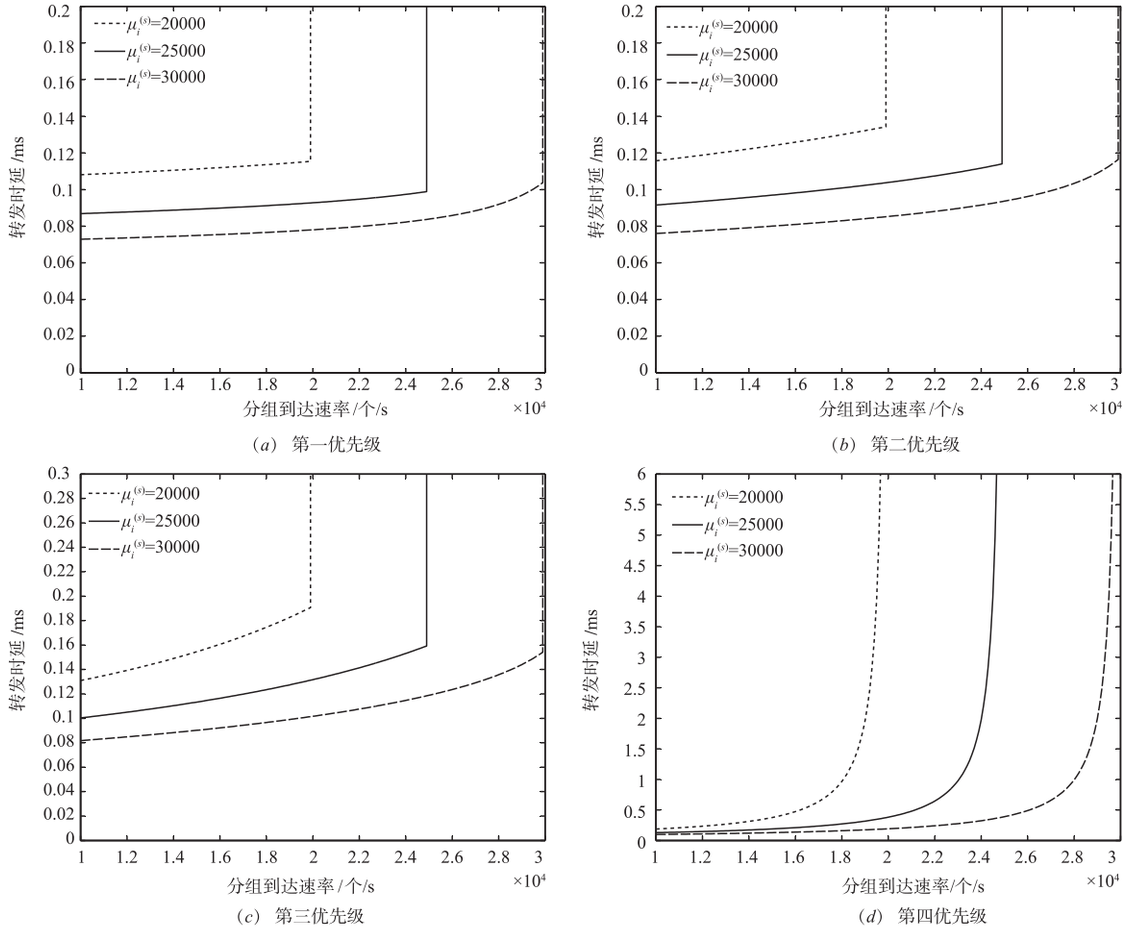
（c）第三优先级 （d）第四优先级

图6 不同控制器数量下的分组转发时延CDF对比

从图 6 可以看出，无论哪个优先级，控制器个数越多，即集群的 Packet-in 消息处理速率越大，CDF 值越快增长到接近 1 的水平. 这是因为交换机转发属于新流的分组需要等待控制器集群下发流规则，而集群处理速率越高，等待时间越短. 同时，无论哪个优先级，控制器数量 n 分别取 2 和 3 时的 CDF 曲线比较接近. 这是因为此时控制器集群的 Packet-in 消息处理速率远超过其到达速率. 此外，无论哪种控制器集群，前三个优先级的 CDF 曲线形状基本相同，而只有第四优先级明显不同. 这是因为前三个优先级的分组优先被转发，排队等待时间较短，而第四优先级分组需等待前三个优先级分组转发完毕，排队等待时间较长.

假定集群由 3 台控制器组成,Packet-in 消息处理速率为 $\mu^{(c)}=21500$ 个/s. OpenFlow 交换机的分组交换速率 $\mu_i^{(s)}$ 分别取 20000 个/s、25000 个/s 和 30000 个/s[22],分组需封装成 Packet-in 消息发送给控制器集群的概率 $q_i=0.04$,其它参数同上. 将以上参数值分别代入式(1)(3)和(5),可得不同分组处理速率下四个优先级的分组转发时延如图 7 所示.

从图 7 可以看出,对于给定的分组交换速率,在分组到达速率较小的情况下,优先级越高的分组转发时延越小;随着分组到达速率的增加,转发时延增长越平缓. 这是因为分组优先级越高,需要等待处理的同级或更高级分组越少,排队等待时间越短,因而转发时延越小,且增速越慢. 当分组到达速率接近分组交换速率时,最低优先级的分组转发时延快速上升,而较高优先级的分组转发时延则会发生跃变. 这是因为较高优先级的分组始终进行排队处理,而只有最低优先级的分组趋于无限排队等待.



(a) 第一优先级　(b) 第二优先级

(c) 第三优先级　(d) 第四优先级

图7　不同分组处理速率下的分组转发时延

7　结束语

本文针对软件定义核心网场景,应用 $M/M/n/m$ 和 $M/M/1/m$ 排队模型分别刻画控制器集群的 Packet-in 消息处理过程和 OpenFlow 交换机的分组处理过程,进而建立 OpenFlow 分组转发性能模型,推导出不同优先级的分组转发时延及其累积分布函数 CDF. 实验测量了 OpenDaylight 控制器集群在多种网络情形下的性能参数,进而计算排队模型的估计时延,结果表明 $M/M/n/m$ 模型的估计时延更接近于实际测量时延. 进一步,采用数值分析的方法,对比了各种情形下不同优先级的分组转发时延及 CDF 曲线. 实验发现:(a)分组优先级越高,CDF 值增长速度越快,平均转发时延越小,且高优先级分组的转发时延相差不大;(b)无论哪个优先级,控制器个数越多,CDF 值越快增长到接近 1 的水平,且控制器数量较多时的 CDF 曲线比较接近;(c)对于给定的分组交换速率,随着分组到达速率的增加,优先级越高的分组转发时延增长越平缓,且非最低优先级的分组转发时延会发生跃变.

参考文献

[1] 左青云,陈鸣,赵广松,等. 基于 OpenFlow 的 SDN 技术研究[J]. 软件学报,2013,24(5):1078 – 1097.
ZUO Q Y,CHEN M,ZHAO G S,et al. Research on OpenFlow-based SDN technologies[J]. Journal of Software,2013,24(5):1078 – 1097.

[2] TAVAKOLI A,CASADO M,KOPONEN T,et al. Applying NOX to the datacenter[A]. ACM Workshop on Hot Topics in Networks (HotNets)[C]. New York,USA:ACM,2009. 1 – 6.

[3] KANDULA S,SENGUPTA S,GREENBERG A,et al. The nature of data center traffic:measurements & analysis[A]. ACM SIGCOMM Conference on Internet Measurement (IMC)[C]. Chicago,Illinois,USA:ACM,2009. 202 – 208.

[4] 付永红,毕军,张克尧,等. 软件定义网络可扩展性研究综述[J]. 通信学报,2017,38(7):141 – 154.
FU Y-H,BI J,ZHANG K-Y,et al. Scalability of software defined network[J]. Journal on Communications,2017,38(7):141 – 154. (in Chinese)

[5] 张栋,郭俊杰,吴春明. 层次型多中心的 SDN 控制器部署[J]. 电子学报,2017,45(3):680 – 686.
ZHANG D,GUO J-J,WU C-M. Controller placement based on hierarchical multi-center SDN[J]. Acta Electronica Sinica,2017,45(3):680 – 686. (in Chinese)

[6] 胡涛,张建辉,邬江,等. SDN 中基于分布式决策的控制器负载均衡机制[J]. 电子学报,2018,46(10):2316 – 2324.
HU T,ZHANG J-H,WU J,et al. Controller load balancing mechanism based on distributed policy in SDN[J]. Acta Electronica Sinica,2018,46(10):2316 – 2324. (in Chinese)

[7] JARSCHEL M,LEHRIEDER F,MAGYARI Z,et al. A flexible OpenFlow-controller benchmark[A]. The 1st European Workshop on Software Defined Networking (EWSDN)[C]. Darmstadt,Germany,2012. 48 – 53.

[8] Introduction to OFsuiteTesting Tools[EB/OL]. http://www.sdnctc.com/test_identify/test_identify/id/47. 2018-07-26.

[9] SHALIMOV A,ZUIKOV D,ZIMARINA D,et al. Advanced study of SDN/OpenFlow controllers[A]. The 9th Central & Eastern European Software Engineering Conference in Russia[C]. Moscow,Russian,2013. 1 – 6.

[10] AZODOLMOLKY S,WIEDER P,YAHYAPOUR R. Performance evaluation of a scalable software-defined networking deployment[A]. The 2nd European Workshop on Software Defined Networks (EWSDN)[C]. Berlin,Germany,2013. 68 – 74.

[11] AZODOLMOLKY S,NEJABATI R,PAZOUKI M,et al. An analytical model for software defined networking:A network calculus-based approach[A]. IEEE Global Communications Conference (GlobeCom)[C]. Atlanta,USA:IEEE,2013. 1397 – 1402.

[12] OSGOUEI A G,KOOHANESTANI A K,SAIDI H,et al. Analytical performance model of virtualized SDNs using network calculus[A]. The 23rd Iranian Conference on Electrical Engineering(ICEE)[C]. Tehran,Iran,2015. 770 – 774.

[13] KOOHANESTANI A K,OSGOUEI A G,SAIDI H,et al. An analytical model for delay bound of OpenFlow based SDN using network calculus[J]. Journal of Network & Computer Applications,2017,96:31 – 38.

[14] BOZAKOV Z,RIZK A. Taming SDN controllers in heterogeneous hardware environments[A]. The 2nd European Workshop on Software Defined Networks (EWSDN)[C]. Berlin,Germany,2013. 50 – 55.

[15] LIN C T,WU C M,HUANG M,et al. Performance evaluation modeling and performance evaluation of an OpenFlow architecture[A]. IEEE International on Teletraffic Congress(ITC)[C]. San Francisco,USA:IEEE,2011. 1 – 7.

[16] HUANG J,XU L,DUAN Q,et al. Modeling and performance analysis for multimedia data flows scheduling in software defined networks[J]. Journal of Network & Computer Applications,2017,83:89 – 100.

[17] JARSCHEL M,OECHSNER S,SCHLOSSER D,et al. Modeling and performance evaluation of an OpenFlow architecture[A]. IEEE International on Teletraffic Congress (ITC)[C]. San Francisco,USA:IEEE,2011. 1 – 7.

[18] 左青云,陈鸣,蒋培成. 基于排队模型的 OpenFlow 控制平面时延评估[J]. 华中科技大学学报(自然科学版),2013,8(1):44 – 49.
ZUO Q-Y,CHEN M,JIANG P-C. Evaluation delay of OpenFlow control plane based on queuing model[J]. Journal of Huazhong University of Science and Technology(Natural Science Edition),2013,8(1):44 – 49. (in Chinese)

[19] MAHMOOD K,CHILWAN A,ΦSTERBΦ O,et al. Modelling of OpenFlow-based software-defined networks:the multiple node case[J]. IET Networks,2015,4(5):278 – 284.

[20] ALGHADHBAN A,SHIHADA B. Delay analysis of new-flow setup time in software defined networks[A]. IEEE/IFIP Network Operations and Management Symposium (NOMS)[C]. Taipei,Taiwan,China:IEEE,2018. 1 – 7.

[21] SOOD K,YU S,XIANG Y. Performance analysis of software-defined network switch using M/Geo/1 model[J].

IEEE Communications Letters, 2016, 20 (12): 2522 -2525.

[22] XIONG B, YANG K, ZHAO J, et al. Performance evaluation of OpenFlow-based software-defined networks based on queueing model[J]. Computer Networks, 2016, 102 (1): 172 -185.

[23] LI T, ZHOU H, LUO H, et al. Modeling software defined satellite networks using queueing theory[A]. IEEE International Conference on Communications (ICC)[C]. Paris, France: IEEE, 2017. 1 -6.

[24] YAO L, HONG P, ZHOU W. Evaluating the controller capacity in software defined networking[A]. The 23rd International Conference on Computer Communication and Networks (ICCCN)[C]. Shanghai, China, 2014. 1 -6.

[25] FU Y, BI J, WU J, et al. A dormant multi-controller model for software defined networking[J]. China Communications, 2014, 11(3): 45 -55.

[26] YONG' AN S. Heterogeneous networking architecture based on SDN[J]. Chinese Journal of Electronics, 2017, 26(1): 166 -171.

[27] KAHE G, JAHANGIR A H. On the Gaussian characteristics of aggregated short-lived flows on high-bandwidth links[A]. The 27th International Conference on Advanced Information Networking and Applications Workshops [C]. Barcelona, Spain: IEEE, 2013. 860 -865.

[28] ARFEEN M A, PAWLIKOWSKI K, WILLIG A, et al. Internet traffic modelling: from superposition to scaling[J]. IET Networks, 2014, 3(1): 30 -40.

[29] 姜腊林,彭霞,熊兵. 基于混合制排队模型的 SDN 控制器性能评估研究[J]. 计算机工程与科学,2017,39(1): 86 -91.
JIANG L, PENG X, XIONG B. Performance evaluation of SDN controllers based on hybrid queuing model[J]. Computer Engineering & Science, 2017, 39(1): 86 -91. (in Chinese)

[30] 丛国超. 批量到达的多服务台排队模型及其仿真研究[D]. 镇江:江苏大学,2006. 29 -36.
CONG G-C. Multi-Server Queues with Batch Arrival and Its Simulation[D]. Zhenjiang: Jiangsu University, 2006. 29 -36. (in Chinese)

作者简介

熊　兵(通信作者)　男,1981 年 8 月出生,湖南益阳人. 毕业于华中科技大学,博士,现为长沙理工大学计算机与通信工程学院副教授,硕士生导师,主要从事未来网络、网络安全等领域研究.
E-mail: xiongbing@csust.edu.cn

左明科　男,1991 年 2 月出生,湖南永州人. 长沙理工大学计算机与通信工程学院硕士研究生,主要从事未来网络领域研究.
E-mail: 1363287285@qq.com

黎　维　男,1990 年 7 月出生,湖南长沙人. 毕业于长沙理工大学,硕士,主要从事未来网络领域研究.
E-mail: 136710815@qq.com

王　进　男,1979 年 11 月出生,江苏扬州人. 毕业于韩国庆熙大学,博士,现为长沙理工大学计算机与通信工程学院教授,主要从事物联网、无线传感网等领域研究.
E-mail: jinwang@csust.edu.cn