

基于控制信息集约的SDN流表更新策略

周创明, 于明秋✉

(空军工程大学防空反导学院, 西安, 710051)

摘要 在控制器带内模式下,为减小流表更新时间,简化流表下发过程,提出了基于控制信息集约的流表更新策略。从路径创建和路径切换2种场景讨论流表更新过程,引入源路由机制和跟踪包机制,集约化流表更新报文形式,统筹规划下发方式,旨在保证流表更新的一致性的同时,降低控制器端的收发信息量和更新时间。仿真实验表明:在不同链路时延和传输速率的条件下,更新策略相较于基于分类的更新策略(C-FUS)和基于路径及反馈的更新策略(PF-FUS),流表更新时间大大降低,更新过程不会造成较大的网络传输波动;更新策略下的控制器端收发信息量与PF-FUS相近,都远低于C-FUS。

关键词 软件定义网络;流表更新;源路由;跟踪包

DOI 10.3969/j.issn.1009-3516.2019.04.010

中图分类号 TP393 **文献标志码** A **文章编号** 1009-3516(2019)04-0066-06

A Control Message Intensive Strategy Based Flow Table Update for SDN

ZHOU Chuangming, YU Mingqiu✉

(Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China)

Abstract: In the controller in-band connection mode, a control message intensive based flow update strategy is proposed to reduce the flow table update time and simplify the process of pushing flows. The flow table update is discussed from two aspects, i. e. path creation and path switching. The source routing and tracking packet are used to intensify the form of flow table update message and plan the way to send it. This strategy is aimed at guaranteeing the consistency of the flow table update and simultaneously reducing the amount of information of controllers receiving and sending and updating time. The simulation results show that rather than classification based flow update strategy and path and feedback based policy update strategy, by this strategy the flow table update time is reduced severely and no great fluctuation is caused in data transmission under different link delay and transmission rate conditions. In addition, the amount of information that controllers receive and send by this strategy is similar to PF-FUS, and both the amounts of information are lower than that of C-FUS.

Key words: software definition network; flow table update; source route; trace packet

软件定义网络(Software Defined Networking, SDN)主张将控制平面与数据平面分离,在控制平面作统一决策,数据平面只负责数据转发,从而实现了网络的集中控制^[1-2]。但在SDN体制中,数据平

收稿日期: 2017-11-21

基金项目: 国家自然科学基金(61806219)

作者简介: 周创明(1967—),男,湖南益阳人,博士,副教授,主要从事数据库应用与智能信息处理研究。E-mail:179820572@qq.com

通信作者: 于明秋(1992—),男,河北沧州人,硕士生,主要从事网络与信息安全研究。E-mail:15511743667@163.com

引用格式: 周创明,于明秋. 基于控制信息集约的SDN流表更新策略[J]. 空军工程大学学报(自然科学版), 2019, 20(4): 66-71. ZHOU Chuangming, YU Mingqiu. A Control Message Intensive Strategy Based Flow Table Update for SDN[J]. Journal of Air Force Engineering University (Natural Science Edition), 2019, 20(4): 66-71.

面交换机依然是分布式部署。由于控制器与交换机之间的通信必然存在延迟,当转发策略更新时,需对流表下发的顺序与时间进行合理规划,否则将出现控制逻辑不一致的情况,造成网络转发的混乱^[3]。

针对流表更新一致性问题,文献[4]将新、旧2套规则的共同流表项提取出来,作为中间规则,更新过程中首先写入中间规则,待全网端到端最长时延后,将新规则补充完整,同时删除旧规则;文献[5]则把需要更新的交换机和流表分类,对不同类型的交换机和流表分时段处理,保证了流表更新的一致性。然而这2种方案均需要等待全网端到端最长时延,耗时较长,严重影响了网络性能。

文献[6]采用标签来区分新、旧规则,使新流与旧流分别执行不同的规则,实现了新、旧流的并行传输;文献[7]利用反馈机制对旧流位置进行判断,使新路径更新与旧路径上对旧流的处理同步进行,使新、旧流可以并行传输。但存在的问题是:文献[6]提出的更新方案需要在交换机端维护2套规则,占用了大量存储空间;而文献[7]提出的方案需要在控制器端进行大量的决策判断,模型的适用性不广。

文献[4~7]都是针对控制器带外连接的网络进行的研究,此类连接模式的部署和维护都比较困难。相反,带内连接模式不需要为控制信息开辟专用通道,所有信息均在数据平面传输,不会额外增加管理负担^[8]。

本文从控制信息集约化的角度出发,研究控制器带内连接模式下的流表更新策略。

1 路径创建流表更新策略

1.1 控制器带内连接模式

Openflow^[9]协议中控制器与交换机之间通过安全通道连接,连接方式有2种:带外连接模式(Out-of-band)和带内连接模式(In-band)^[10]。带外连接模式在运行效率以及安全性上有一定的优势,但是需要有更多的网络资源作为支撑,而对于无线通信这类网络的负担较大。与此不同,在带内连接模式下,控制信息与流数据包采用相同的逻辑信道,控制信息经数据平面端口进入交换机,同样进行流表匹配等过程^[11]。带内连接模式可以充分利用网络资源,部署代价低,但流表更新过程复杂,这个过程是本文研究的重点。

1.2 基于源路由的流表下发机制

在源路由机制中,每个数据分组携带有从源节点到目的节点的完整路径信息,中间节点根据数据分组携带的路径信息进行转发。源路由机制分为严

格源路由和松散源路由^[12-14]。在SDN中,2个重要节点之间的数据传输将通过流表匹配实现。

基于源路由机制的流表更新原理为:流表更新指令中包含完整的路径信息,并严格按照路径信息传输,交换机节点根据流表更新指令携带的数据信息更新匹配项,根据转发过程中经过发送端口和下一跳节点正向或反向更新动作项。文献[15]便采用基于源路由的流表下发机制,解决控制器带外连接模式下的控制信道与数据信道的负载均衡问题。其特点是,大大减少了流表更新指令的下发。

控制器与交换机之间的通信路径时,以控制器所在节点作为源节点,根据严格路径信息传输流表更新指令,建立正反向传输路径。建立交换机节点之间的通信路径时,对指令下发到源节点的过程,只在路径信息中标注控制器节点到源节点,中间过程通过已有的路径传输;对节点间路径建立过程,则标注完整路径信息,从源节点开始按照严格源路由机制传输。待路径末端节点反馈更新完毕后,开始传输数据,因此路径创建不涉及流表更新一致性问题。

2 路径切换流表更新策略

路径切换发生在节点间通信路径失效或不满足业务传输条件时,将涉及到2条路径的更新,包括新路径流表添加和旧路径流表项删除,其主要考虑的问题是流表更新一致性^[16-17]。

设网络 G 中交换节点集合为 $S = \{s_1, s_2, \dots, s_N\}$,用户 h_1 与 h_2 通过网络通信。初始传输路径 $l_1 \in S$,切换后传输路径 $l_2 \in S$,则流表更新过程可以描述为:删除 l_1 节点中相关的流表项,同时在 l_2 节点中加入相关的流表项。考虑到 l_1 和 l_2 可能存在公共节点,表示为 $S' = \{s'_1, s'_2, \dots, s'_l\}$,其中节点按照数据传输方向排序。那么公共节点会存在以下情况:

1)若公共节点 s'_k 与 s'_{k+1} 在传输路径中为相邻节点,则 s'_i 无需进行更新;

2)若集合 $\{s'_1, s'_2, \dots, s'_k\}$ 前后节点在传输路径中都为相邻节点,且 s'_k 与 s'_{k+1} 不相邻,则称 s'_k 为边界交换机, s'_k 上游的节点都无需更新流表。边界交换机即为流表更新的起点。

在已有的研究中,更新开始首先删除边界交换机的相关流表项,这之前经过的相关数据包视为旧流,之后的视为新流。新流由于匹配不到流表项,会暂时上传到控制器。初始传输路径中除边界交换机外需要更新的节点为原路径交换机节点,切换后传输路径中除边界交换机外需要更新的节点称为新路

径交换机节点。考虑到更新一致性问题,原始路径交换机节点需要等待旧流传输完毕,才能进行流表更新,而新路径交换机节点没有这样的限制。

2.1 新路径交换机更新策略

新路径交换机节点的更新没有时序限制,应尽可能花费较少的时间。控制器带内连接模式下,同时向每个交换机发送指令无疑会增加控制器负担,同时占用过多的数据信道资源。为解决这一问题,本文在源路由机制的基础上对流表下发方式进行改进。

设 d_j 为新路径交换机集合中第 j 个交换机到所属控制器的最小跳数,其中 $j=1,2,\dots,n$, n 为新路径交换机的总数,按照数据传输的方向排序。文中以跳数衡量时延大小,设 τ_j 为第 j 个交换机在新路径上距离边界交换机的跳数,对于传输路径中的相邻节点,定义时延差:

$$tc_{ab} = |\tau_b - \tau_a| + d_a - d_b \quad (1)$$

式中: $a, b \in [1, n]$, 且若 $tc=0$, 则控制器与节点 b 的一条最小延时传输路径会经过节点 a 。此时可以以节点 a 为源节点,采用源路由机制建立节点 a 到节点 b 的传输路径,这样能够保证与独立向节点 b 下发流表更新指令有相同的时延。

为防止极端情况发生(如 $d_j, j=1,2,\dots,n$ 为恒值),应有效约简流表更新指令,拓展时延差的概念。定义阈值 Γ ,若路径 $a \rightarrow b$ 上的节点对于源节点 a 都满足 $tc \leq \Gamma$,便可以采用源路由机制更新路径 $a \rightarrow b$ 。这里要求阈值 Γ 为一个正值。通过这种方法可以将新路径交换机划分为一定数量的交换机集合,更新流表时,控制器只需向所有源交换机发送基于源路由机制的流表更新指令,约简了控制信息。集合划分算法可以描述为:

```

Begin
while (有交换机未归入集合中)
for  $i=1:n$ 
    比较获取  $d_j$  最小值,并依顺序将对应的交换机加入集合  $c = \{c_j\}_{1 \times m}$ 。
end
for  $j=1:m$ 
    以数据传输方向为基准,以交换机  $c_j$  为源节点,分别正向、反向计算  $tc$  值,将满足  $tc \leq \Gamma$  的交换机放入正向集合和反向集合中,记为集合  $B_{2j}$  和  $B_{2j-1}$ 。
end
for  $k=1:m-1$ 
    若  $B_{2k} \cap B_{2k+1} \neq \emptyset$ ;
    则判断,若交换机  $c_{k+1} \in B_{2k}$  或  $c_k \in B_{2k+1}$  &  $c_{k+1} \in B_{2k}$ ,则删除集合  $B_{2k+2}$ ,令  $B_{2k}$  为路径  $c_k \rightarrow c_{k+1}$  的交换机集合;若交换机  $c_k \in B_{2k+1}$ ,则删除集合  $B_{2k}$ ,令  $B_{2k+1}$  为路径  $c_{k+1} \rightarrow c_k$  的交换机集合。

```

否则设 $card()$ 为集合中元素个数。判断,若 $card(B_{2k}) \leq card(B_{2k+1})$, 则 $B_{2k+1} = B_{2k+1} - (B_{2k} \cap B_{2k+1})$; 若 $card(B_{2k}) > card(B_{2k+1})$, 则 $B_{2k} = B_{2k} - (B_{2k} \cap B_{2k+1})$ 。

end

排除已归入集合的交换机;

end

2.2 原路径交换机更新策略

原路径交换机的更新需要在旧流传输完毕后才能进行,否则会在流表更新一致性问题。原路径交换机的更新主要是删除相关不再使用的流表项,以节省交换机存储空间。本文在源路由机制的基础上,将更新过程约简为一个跟踪包的传输。

跟踪包的具体构造如图1所示。包头内容包含:①采用松散源路由机制,重要节点为边界交换机节点和原路径交换机节点的末节点;②需更新交换机的标识;③与旧流数据包相同的优先级。

设定相同的优先级后,只需控制跟踪包下发的时间,便可保证交换机在转发旧流数据包之后才会处理跟踪包。另外在传输过程中可能存在不需要更新的交换机,因此有必要对需更新交换机做出标识。

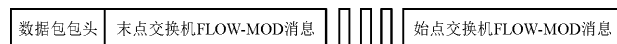


图1 跟踪包结构

跟踪包的主体内容是 $FLOW_MOD^{[18-19]}$ 消息集合,包含所有需要下发到交换机的 $FLOW_MOD$ 消息。其构造顺序是由原路径交换机节点的末节点到始节点的顺序排列,便于分割出各个交换机对应的 $FLOW_MOD$ 消息。跟踪包进入交换机的处理流程见图2。

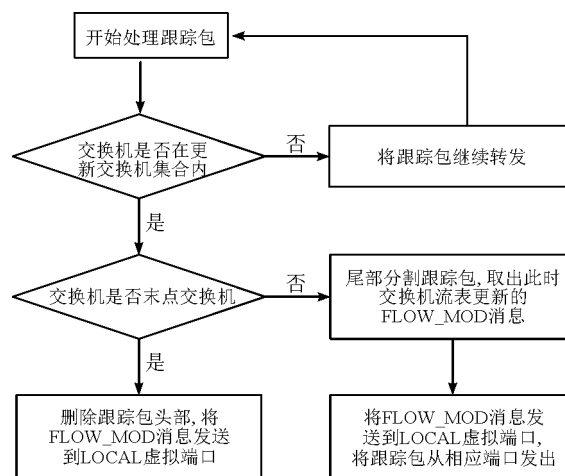


图2 跟踪包处理流程

跟踪包在传输的过程中会经过不执行更新的交换机,所以首先判断交换机是否需要更新,若不需更新,则继续转发。若到达的交换机需要更新,从尾部分割跟踪数据包,取出相应的 $FLOW_MOD$ 消息,

将跟踪包转发后,更新流表。若此时交换机是末点交换机则只需要删除数据包包头。由于指定了 2 段传输过程的源目节点,跟踪包通过流表匹配经中间节点转发。

2.3 更新过程

本文研究控制器带内连接模式下的流表更新一致性策略,更新流程分 4 个阶段:

1) 控制器采用松散源路由机制下发跟踪包,重要节点依次为边界交换机节点和原路径交换机节点的末节点。跟踪包的主要内容是删除相关流表项的 FLOW_MOD 消息。

2) 跟踪包到达边界交换机,边界交换机删除相关表项,新流将上传控制器,跟踪包跟随旧流传输。

3) 边界交换机旧表项删除后,更新新路径交换机流表。同时,跟踪包随旧流传输,更新旧路径交换机。

4) 新路径交换机更新完成后,控制器对边界交换机安装新流表,控制器将缓存数据包下发,沿新路

径传输。

从更新流程可以看出,控制器与交换机的交互得到有效约简。跟踪包跟随旧流传输,实时更新经过交换机的流表,降低了更新时间。

3 更新方案一致性分析

对流表更新流程做时序上的分析,如图 3 所示。 t_2 时刻之后新流上传控制器,旧流按照原规则传输。 t_3 时刻后,新路径交换机更新完毕,控制器将新流下发,新流按照新规则传输。

从逻辑上分析,当跟踪包到达边界交换机,边界交换机删除旧表项。新流到达,没有匹配的流表,均上传至控制器。跟踪包随旧流传输,在旧流经过后更新流表,旧流仍然按照原规则操作。当所有交换机更新完毕,控制器将缓存的数据包下发,按照新的规则传输。因此本方案可以确保数据处理得到一致性。

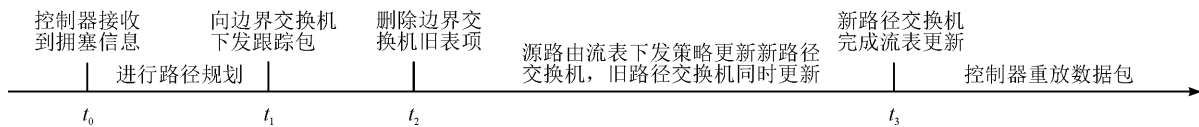


图 3 流表更新时序

4 实验分析

实验在 Mininet2.1.0 环境中进行,网络场景的搭建如图 4 所示,采用 Floodlight 开源控制器(Java),对路径切换时的流表更新问题进行仿真验证。数据传输路径由 $s_1 \rightarrow s_2 \rightarrow s_5 \rightarrow s_6 \rightarrow s_7$ 切换至 $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_4 \rightarrow s_7$,控制器部署于 s_3 的位置。实验初始,通过严格源路由流表更新机制创建控制器到各交换机路径的流表,以及路径 $s_1 \rightarrow s_2 \rightarrow s_5 \rightarrow s_6 \rightarrow s_7$ 的流表。为验证本文方案的有效性,将文献[5]基于分类的更新策略 C-FUS 和文献[7]基于反馈的更新策略 PF-FUS 作为对比。

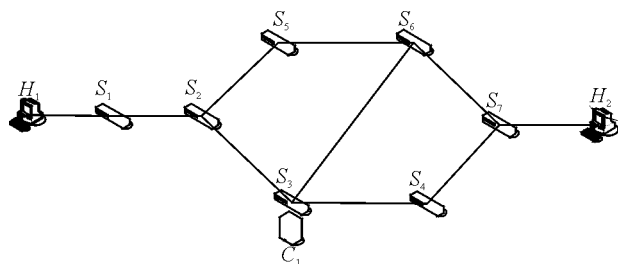


图 4 实验网络拓扑图

图 5 展示的是在链路传输时延为 0.2 ms 时,随着传输速率的改变,3 种方案流表更新时间的变化。其中,C-FUS 更新时间最长,主要原因是 C-FUS 在

新路径交换机流表更新完成后,需要等待端到端最长时延才能更新旧路径交换机,耗时较长。而 PF-FUS 和本文方案对新、旧路径交换机的更新是同步进行的。相较之下,PF-FUS 流表更新时耗更长,这是因为 PF-FUS 是一种基于控制器带外连接的更新策略,控制器需向每个交换机发送 FLOW_MOD 消息,FLOW_MOD 消息与反馈信息的传输都需要消耗时间,这一问题在文后的图中可以明显看出。

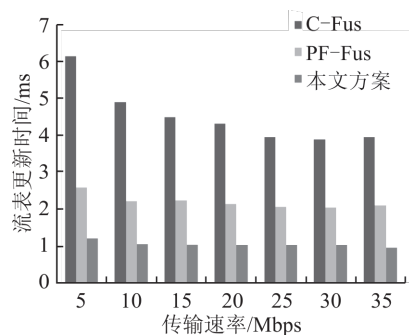


图 5 链路时延为 0.2 ms 时流表更新时间比较

图 6 显示的是传输速率为 5 Mbps 时,流表更新时间与链路时延的关系。随着链路时延的增大,C-FUS 和 PF-FUS 中控制器收发信息较多,信息传输时耗影响了流表更新时间。而本文方案的流表更新时间基本不变,主要是由于本文方案将控制信息约简,控制器不需要多次收发信息。

从控制信息集约化方面进行研究,统计控制器收发信息的次数,如图7所示。整个过程中,与流表更新直接相关的控制器信息收发过程有2类,其一是边界交换机新流数据包的上传,其二是流表更新所需要的控制器与交换机之间的交互。结合3种方案更新过程中数据包上传数的比较,如图8所示,C-FUS由于流表更新时间较长,随着数据包在边界交换机 s_2 积累,上传的数据包数目明显多于其他2种方案。相比之下,PF-FUS实现了新、旧流的并行传输,边界交换机 s_2 的数据包积累不多;而本文方案虽然要等待交换机更新完毕才下发数据包,但由于更新时间较短,对 s_2 的数据包积累影响不大,所以PF-FUS和本文方案的数据包上传数目较小。

从另一个方面讲,C-FUS和PF-FUS都是基于控制器带外连接的更新方案,控制器与交换机交互频繁,因此PF-FUS的控制器收发信息量明显高于本文方案,见图8。结合2种实验,能够看出本文提出的流表更新方案实现了控制信息的集约化,在减轻控制器负载的同时,也缓解了链路的压力。

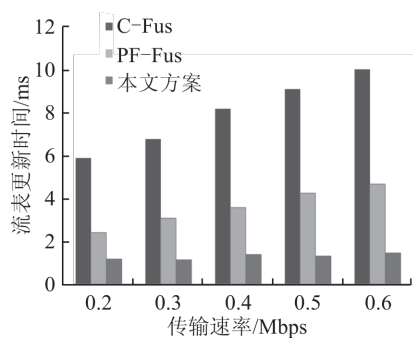


图6 传输速率为5 M时流表更新时间比较

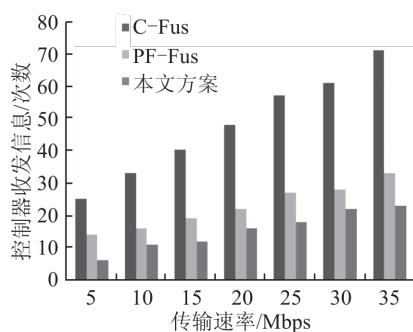


图7 链路时延为0.2 ms控制器收发信息量的比较

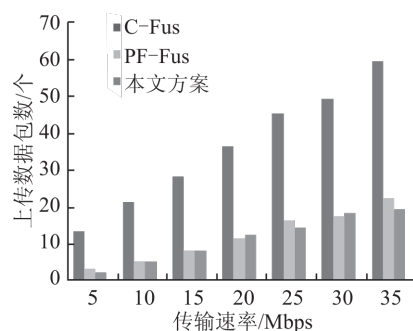


图8 链路时延为0.2 ms时上传数据包数比较

5 结语

本文提出了一种基于控制信息集约的流表更新方案,采用源路由机制和跟踪包机制,解决了控制器带内连接模式下的流表更新问题,使控制器在整个更新过程中只需要收发少量信息。该方案在保证流表更新一致性的同时,缓解了控制器端的压力,降低了更新时耗。本文仅从流表更新过程进行了研究,对网络拥塞等问题没有深入分析,未来将把网络流量控制的研究集中到控制器端,进一步完善流表更新策略,提高研究实用性。

参考文献(References):

- [1] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. Openflow: Enabling Innovation in Campus Networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74.
- [2] BIANCO A, BIRKE R, GIRAUDO L, et al. Openflow Switching: Data Plane Performance[C]//2010 IEEE International Conference on Communication (ICC). Cape Town, South Africa: IEEE, 2010: 1-5.
- [3] 刘江, 胡晓露, 黄韬, 等. 基于多向搜索的SDN流表更新一致性方案[J]. 北京邮电大学学报, 2016, 39(3): 54-59.
- [4] LIU J, HU X L, HUANG T, et al. Multi-Directional Searching Based Consistent Flow Updating Scheme in Software Defined Networking[J]. Journal of Beijing University of Posts and Telecommunications, 2016, 39(3): 54-59. (in Chinese)
- [5] MCGEER R. A Safe, Efficient Update Protocol for Openflow Networks [C]//Proceedings of the First Workshop on Hot Topics in Software Defined Networks. New York: ACM, 2012: 61-66.
- [6] 周焯, 杨旭, 李勇, 等. 基于分类的软件定义网络流表更新一致性方案[J]. 电子与信息学报, 2013, 35(7): 1746-1752.
- [7] ZHOU Y, YANG X, LI Y, et al. Classification Based Consistent Flow Update Scheme in Software Defined Network[J]. Journal of Electronics & Information Technology, 2013, 35(7): 1746-1752. (in Chinese)
- [8] REITBLATT M, FOSTER N, REXFORD J, et al. Abstractions for Network Update [J]. ACM SIGCOMM Computer Communication Review, 2012, 42(4): 323-334.
- [9] 李玲芮, 李俊, 常向青, 等. 基于路径及反馈的软件定义网络策略更新方案[J]. 计算机应用研究, 2016, 33(5): 1461-1467.
- [10] LI L R, LI J, CHANG X Q, et al. Path and Feedback

- Based Policy Update Scheme for SDN [J]. *Application Research of Computers*, 2016, 33(5):1461-1467. (in Chinese)
- [8] 曹继军,肖立权,王克非,等. 超级计算互连网络带内管理的实现与评测[J]. *计算机学报*, 2016, 39(9): 1717-1732.
CAO J J, XIAO L Q, WANG K F, et al. The Implementation and Evaluation of In-Band Network Management in Supercomputing System[J]. *Chinese Journal of Computers*, 2016, 39(9): 1717-1732. (in Chinese)
- [9] NADEAU D T, GRAY K. 软件定义网络:SDN与Openflow解析[M]. 毕军,单业,张绍宇,等,译. 北京:人民邮电出版社,2014:9-19.
NADEAU D T, GRAY K. SDN: Software Defined Networks[M]. BI J, SHAN Y, ZHANG S Y, et al, Translated. Beijing: Posts & Telecom Press, 2014: 9-19. (in Chinese)
- [10] GORANSSON P, BLACK C. 软件定义网络—原理、技术与实践[M]. 王海,张娟,于卫波,等,译. 北京:电子工业出版社,2016:76-77.
GORANSSON P, BLACK C. Software Defined Networks: A Comprehensive Approach[M]. WANG H, ZHANG J, YU W B, et al, Translated. Beijing: Publishing House of Electronics Industry, 2016: 76-77. (in Chinese)
- [11] 刘韬,王文东,龚向阳,等. 一种软件定义网络中带内模式下控制网络的配置升级机制[J]. *北京邮电大学学报*, 2016, 39(1): 24-28.
LIU T, WANG W D, GONG X Y, et al. Updated Mechanism for In-Band Controlling Traffic of Software Defined Networking [J]. *Journal of Beijing University of Posts and Telecommunications*, 2016, 39(1): 24-28. (in Chinese)
- [12] 舒炎泰,高德云,王雷. 无线 Ad Hoc 网络中的多径源路由[J]. *电子学报*, 2002, 30(2): 279-282.
SHU Y T, GAO D Y, WANG L. Multipath Source Routing in Wireless Ad Hoc Network[J]. *Acta Electronica Sinica*, 2002, 30(2): 279-282. (in Chinese)
- [13] 谭华. Ad Hoc 网络中动态源路由的 QoS 改进[J]. *重庆科技学院学报(自然科学版)*, 2007, 9(3): 59-62.
TAN H. Modified Algorithm for Dynamic Source Routing Protocol with QoS Support in Ad Hoc Network[J]. *Journal. Chongqing University of Science Technology (Natural Science Edition)*, 2007, 9(3): 59-62. (in Chinese)
- [14] 张连芳,舒炎泰,袁明,等. Ad Hoc 网络多径源路由协议的实现和性能评价[J]. *天津大学学报*, 2004, 37(2): 95-99.
ZHANG L F, SHU Y T, YUAN M, et al. Implementation and Performance Evaluation of Multi-path Source Routing Protocol in Wireless Ad Hoc Network [J]. *Journal of Tianjin University*, 2004, 37(2): 95-99. (in Chinese)
- [15] 王小毛,黄传河,范茜莹,等. SDN 中基于负载均衡的流表下发方法[J]. *华中科技大学学报(自然科学版)*, 2016, 44(11): 75-81.
WANG X M, HUANG C H, FAN X Y, et al. Flow Table Update Method Based on Load Balancing for SDN[J]. *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, 2016, 44(11): 75-81. (in Chinese)
- [16] 高原,陈庶樵,扈红超. 一种改进的软件定义网络低开销一致性更新算法[J]. *计算机应用研究*, 2016, 33(7): 2131-2134.
GAO Y, CHEN S Q, HU H C. Improved Smaller Rulespace Overhead and Consistent Flow Update in Software Defined Network[J]. *Application Research of Computers*, 2016, 33(7): 2131-2134. (in Chinese)
- [17] 刘惠琳,林亚飞. 基于控制集群的 SDN 流表更新机制[J]. *长春工业大学学报(自然科学版)*, 2014, 35(4): 429-433.
LIU H L, LIN Y F. SDN Flow Table Update Mechanism Based on Controller Cluster [J]. *Journal of Changchun University of Technology (Natural Science Edition)*, 2014, 35(4): 429-433. (in Chinese)
- [18] 晃通,宫永直树,岩田淳. 图解 OpenFlow[M]. 李战军,薛文玲,译. 北京:人民邮电出版社,2016:9-52.
AKIMICHI, MIYANAGA Naoki, IWATA Jun. Diagrammatize OpenFlow [M]. LI Z J, XUE W L, Translated. Beijing: Posts & Telecom Press, 2016: 9-52. (in Chinese)
- [19] 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. (in Chinese)

(编辑:徐楠楠)