软件定义网络中基于流量管理的分布式防火墙策略

史久根 王 继* 张 径 徐 皓
(合肥工业大学计算机与信息学院 合肥 230009)

摘 要：在软件定义网络中将防火墙策略定义为访问控制型规则，并将其分布式地部署在网络中能够提高会话的服务质量。为了减少放置在网络中规则的数量，文中提出多路复用和合并的启发式规则放置算法(HARA)。算法考虑到了商品交换机TCAM存储空间和端点交换机相连链路的流量负载，通过建立以最小化规则放置数量为目标的混合整数线性规划模型，解决不同吞吐量的多路由单播会话的规则放置问题。实验结果表明，与nonRM-CP算法相比，在保证不同会话服务质量的前提下，该算法最多能节省56%的TCAM空间，平均能减少13.1%的带宽资源利用率。

关键词：软件定义网络；分布式防火墙策略；规则放置；TCAM；流量负载
中图分类号：TP393 **文献标识码**：A **文章编号**：1009-5896(2019)01-0091-08
DOI: 10.11999/JEIT180223

Distributed Firewall Policy Based on Traffic Engineering in Software Defined Network

SHI Jiugen WANG Ji ZHANG Jing XU Hao
(*School of Computer and Information*, *Hefei University of Technology*, *Hefei* 230009, *China*)

Abstract: Firewall policy is defined as access control rules in Software Definition Network (SDN), and distributing these ACL (Access Control List) rules across the networks, it can improve the quality of service. In order to reduce the number of rules placed in the network, the Heuristic Algorithm of Rules Allocation (HARA) of rule multiplexing and merging is proposed in this paper. Considering TCAM storage space of commodity switches and connected link traffic load of endpoint switches, a mixed integer linear programming model which minimize the number of rules placed in the network is established, and the algorithm solves the rules placement problem of multiple routing unicast sessions of different throughputs. Compared with the nonRM-CP algorithms, simulations show that HARA can save 18% TCAM at most and reduce the bandwidth utilization rate of 13.1% at average.

Key words: Software Defined Network (SDN); Distributed firewall policy; Rules allocation; Ternary Content Addressable Memory (TCAM); Traffic load

1 引言

软件定义网络(Software Defined Network, SDN)架构中，控制器用支持Open Flow协议的链路与特定的控制信息和各交换机进行通信连接[1]。此外，控制器通过发送相关命令，将规则安装在交换机的流表中来规定流的行为，但考虑到TCAM有限的空间，将规则全都沿着传输路径进行安装，这样会造成巨大的流表开销，再者，考虑到不同单播会话的流量需求，选择合适的路由能够减少对带宽资源的使用。

针对以上问题，部署防火墙策略能够有效地保证单播会话的QoS。由于防火墙策略可被定义为访问控制型规则(Access Control List rules)，即ACL规则，该类型规则决定数据包是否允许通过交换机，故主动式地部署ACL规则可使会话服务质量得到可靠保障[2]。为此，本文提出基于端点交换机流量管理的分布式防火墙策略，由于端点策略定义好了各组单播会话端点交换机对，即流从网络边缘的入口交换机进入，从网络边缘的出口交换机流出[3]。所以，本文依据单播会话吞吐量需求，部署分布式防火墙策略，实现相关链路负载更加均衡的目标。

在规则放置问题中，需要考虑多个网络资源限

收稿日期：2018-03-09；改回日期：2018-07-26；网络出版：2018-08-06
*通信作者： 王继 jiwang@mail.hfut.edu.cn
基金项目：国家重大科学仪器设备开发专项(2013YQ030595)
Foundation Item: The National Major Scientific Instruments Development Project (2013YQ030595)

制条件。在有限的TCAM空间以及链路带宽限制等条件下，文献[3]将网络抽象为 “Big Switch”，并将其划分为多条路径，通过估计每条路径上交换机总容量，将聚集的规则沿着传输路径进行放置；文献[4]提出PBD和CBD流表分割算法，将SDN流表分割成不同数量的子表来平衡各交换机TCAM；文献[5]提出非均匀流表的规则分割和放置模型，改善了规则缓存所造成的延时；文献[6]提出贪心的OFFICER启发式算法，对选择的默认路径利用偏转法得到多条转发路径，解决了在预定义的网络中规则放置及最大流问题；文献[7]提出多路由单播会话的规则复用计划，解决了以保证单播会话服务质量的前提下，规则分配和最小化规则放置数量的问题；文献[8]在链路带宽等网络资源总是有界的前提下，提出了帕累托优化模型，将链路利用率作为约束条件，得到帕累托最优解——规则数量；文献[9]提出能量感知的路由策略，解决了规则放置的线性规划模型；文献[10]提出自适应规则放置算法，该算法考虑到DROP、PERMIT规则的优先级所造成的规则循环依赖问题，构建了规则依赖关系图实现了规则合并。

综上所述，大部分研究都是在受限的网络资源下，从流和规则的关系来研究规则放置问题，但是它们未考虑到不同吞吐量需求对路由造成的影响，本文作为一种主动式的规则放置方法，取消了规则建立的延时以及减少控制链路中交互信息的数量。其中，本文利用入、出口交换机度数建立了候选路径集，然后，以最小化放置规则数量为目标，依据每组会话的吞吐量需求，选择从入口交换机不同的链路出发到不同链路的出口交换机的传输路径来放置ACL规则。

2 问题提出与系统模型

本节将讨论最小化规则数量和端点交换机相连链路的流量负载问题，并为求解这些问题构建系统模型。

2.1 问题描述

该模型需要解决的问题是，依据入、出口交换机度数建立候选路径集，根据会话集吞吐量需求，从候选路径集中选路并放置规则。

(1) 端点交换机相连链路的流量负载：为了均衡端点交换机相连链路的流量负载，应依据不同吞吐量需求从候选路径集中选择不同条数的传输路径，由于每条路径的瓶颈链路带宽决定了相应路径传输流量的大小，所以，当吞吐量远小于瓶颈链路的带宽时，则从候选路径集中选择任意一条路径进行传输，当吞吐量在链路带宽之间时，则选择多条满足带宽条件的路由。

(2) 最小化规则数量：经上述的选路过后，需放置ACL规则以保证会话的服务质量和流的可靠传输。本文提出规则的多路复用和合并的方式减少放置在交换机的规则数量，文献[7]首次提出规则多路复用(rule multiplexing)的概念，指在单播会话的多条传输路径中，放置在不同交换机的规则能被原会话的流匹配。文献[10]中提出规则合并(rule merging)的概念，指不同会话有相同匹配域和行为的规则能够进行合并。

2.2 系统模型

用图$G=(V,E)$表示SDN网络模型，V代表有限的Open Flow交换机集，E代表链路集，每条链路$(i,j)\in E$的路由花销用$w(i,j)$表示。控制器通过控制链路控制所有的交换机，每个交换机$i\in V$的TCAM存储容量为C_i，每条链路$(i,j)\in E$的带宽为$B(i,j)$。K表示单播会话集，每组会话(a,b)的吞吐量需求为d_k，P_k表示第k组会话候选路径集。将每组会话ACL规则集利用PBD流表分割算法[4]均匀地分割成R_k个规则子集，每个规则子集的规则数量为r_u，用r_u到$f(k)$映射函数表示规则子集R_k中应分配规则的数量，t^p表示路径p的传输速率，$t^p_{(i,j)}$表示路径p中链路(i,j)的传输速率。

定义 1 任意独立的网络拓扑图中，与节点关联的出弧的数目称为该节点的出度，即$\deg_i^+$，入弧的数目称之为入度，即$\deg_i^-$，两者的总和称为该节点的度数。在一个有向流网络$N=(V,E)$中，必须满足以下3个条件，才可以保证单播多路由的端点策略完整性：

(1) 每组会话的源边缘交换机a没有入弧，只有出弧；

(2) 每组会话目的边缘交换机b没有出弧，只有入弧；

(3) 源交换机经过多路转发到达目的交换机的吞吐量是守恒的，即 $\sum_{v\in V}t_{av}=\sum_{v\in V}t_{vb}$， 代表中间节点的交换机。

决策变量的定义：用式(1)表示是否从候选路径中选择该条传输路径，同理，用$l^p_{(i,j)}$表示传输路径p的链路(i,j)是否被选取，$l^p_{(i,j)}\in\{0,1\}$。

$$l^p=\begin{cases}1, & \text{传输路径}p\text{被选中},\forall k\in K,\ \forall p\in P_k\\ 0, & \text{其它}\end{cases} \tag{1}$$

传输路径是由一系列的交换机点i组成，其与规则子集u的关系用式(2)表示为

$$x_i^u=\begin{cases}1, & \text{规则子集}u\text{放置在交换机}i\text{上}\\ 0, & \text{其它}\end{cases} \tag{2}$$

在单播会话的候选路径集P_k中，从候选路径中选择路径p，并将相应规则子集u放置在传输路径的节点i上。其关系用式(3)表示为

$$x_i^{up}=\begin{cases}1, & \text{规则子集}u\text{放置在传输路径}p\text{的}\\ & \text{节点}i\text{上},\ \forall k\in K,\ p\in P_k\\ 0, & \text{其它}\end{cases} \tag{3}$$

当不同组会话经过相同交换机的传输路径时，可合并的规则子集u^m与交换机i之间的关系可用式(4)表示，表明规则子集u在交换机i上是可合并的。

$$x_{ui}^m=\begin{cases}1, & \text{可合并的规则子集}u^m\text{放置在}\\ & \text{交换机}i\text{上}\\ 0, & \text{其它}\end{cases} \tag{4}$$

单播多路由的端点策略：各组会话所选路径的传输速率应满足相应的吞吐量需求，并用式(5)表示。若会话的吞吐量需求低于瓶颈链路带宽时，将式(5)取等号来表示单路由的传输方式，

$$\sum_{p\in P_k} t^p \geq d_k, \forall k\in K \tag{5}$$

用式(6)表示，每组会话的流经过中间交换机应满足流守恒：

$$\sum_{(a,v)} t_{(i,j)}^p - \sum_{(v,b)} t_{(j,i)}^p = 0, \forall a,b,v\in V, \forall k\in K, p\in P_k \tag{6}$$

用式(7)表示选择路径的传输速率要受限于该条路径的瓶颈链路带宽，

$$0\leq t^p \leq l_{(i,j)}^p t_{(i,j)}^p, \forall k\in K, \forall p\in P_k, (i,j)\in p \tag{7}$$

在同组单播会话的多条传输路径中，当不同的传输路径经过同一条链路时，该条链路上流量之和小于该条链路的带宽，用式(8)表示，

$$0\leq t_{(i,j)}^p \leq l^p B(i,j), \forall k\in K, \forall p\in P_k, (i,j)\in p \tag{8}$$

当不同组会话的传输路径经过同一条传输链路时，在该条链路的流量之和也要小于带宽限制，用式(9)表示，

$$\sum_{k\in K}\sum_{p\in P_k} t_{(i,j)}^p \leq B(i,j), \forall p\in P_k, (i,j)\in p \tag{9}$$

分布式防火墙策略：放置在不同传输路径相应节点的规则子集能被同组会话的不同流匹配。当右式x为1时，表明单播会话的规则会被放置在多条路径的相交节点上，用式(10)来表示规则的多路复用关系：

$$x_i^u \geq x_i^{up}, \forall k\in K, \forall p\in P_k, \forall u\in R_k \tag{10}$$

若选择了相应传输路径，就应将规则放置在交换机中。当右式x为1时，表明该路径已经被选取了，这时要在该条路径所对应的交换机中放置规则，用式(11)来表示：

$$\sum_{i\in p} x_i^{up} \geq \sum l^p, \forall k\in K, \forall p\in P_k, \forall u\in R_k \tag{11}$$

Open Flow交换机可以为不同的规则以添加VLAN ID等方式重新整合不同匹配域的规则，保证规则匹配域的功能性。由于交换机会放置不同会话的规则子集，当不同会话的规则放置在该交换机时，将同一节点i可合并的规则子集放在集合$\{x_{ki}^m, x_{kj}^m, \cdots\}$中。

考虑到两种特殊情况：在该交换机中所有的规则都能合并或所有的规则都不能合并，所以在该交换机的规则数量应从最大数量的规则子集到所有规则子集和之间取值。若所有的规则都能进行合并，那么就将规则数量最大的规则子集作为合并后规则数量能够取得的下界，若所有的规则都不能进行合并，那么放置在该交换机所有规则数量的总和作为合并后规则数量能够取得的上界。从这个范围中任意地选择相应数量的规则放置在该交换机，表示不同会话的规则进行了合并，此时$x_{ui}^m=1$，用式(12)表示。

$$\begin{aligned}&\max\{\ x_{ki}^m, \times r_m, x_{kj}^m\times r_m, \cdots\}\\ &\quad\leq x_{ui}^m \times r_m \leq \sum_{k\in K}\sum_{m\in R_k} x_i^m \times r_m\end{aligned} \tag{12}$$

式(13)表示放置在交换机的规则数量不能超过其TCAM，

$$\sum_{k\in K}\sum_{p\in P_k} x_i^u \times r_u \leq C_i, \forall u\in R_k, \forall i\in V \tag{13}$$

最后，该系统模型以最小化规则放置的数量R_{all}为目标，用式(14)表示。式(15)和式(16)计算所选路径的带宽资源利用率B_{use}和链路的路由花销L_{cost}。

$$\min\sum_{k\in K}\sum_{p\in \text{P}_\text{k}}\sum_{i\in N}(x_i^u\times r_u + x_{ui}^m \times r_m) \tag{14}$$

$$B_{\text{use}}=\sum_{k\in K}\sum_{p\in P_k}\sum_{(i,j)\in p}(t^p/B(i,j)) \tag{15}$$

$$L_{\text{cost}}=\sum_{k\in K}\sum_{p\in P_k}\sum_{(i,j)\in p}\left(l_{(i,j)}^p \times w(i,j)\right) \tag{16}$$

3　算法分析

HARA算法由多路由选取算法MRSA(Multipath Routing Selection Algorithm)和分布式防火墙部署算法DFAA(Distributed Firewall Allocation Algorithm)组成。MRSA利用链路花销构建候选路径

集，DFAA从候选路径集中选择满足条件的传输路径，并将会话集的规则放置在交换机中。

3.1 算法描述

在文献[6]和文献[11]中已经证明了规则放置问题是一种NP-难问题，所以ACL规则放置问题也是NP-难的。当不考虑规则放置问题时，若将式(5)中d_k作为求解目标，那么该问题就成为了一种求最大流的多商品流问题，在文献[12]中也已证明该问题也是NP-难的。

下面首先详细描述MRSA算法的结构和流程，如表1所示。

表 1 多路由选取算法

算法1 多路由选取算法(MRSA)

输入：网络拓扑$G=(V,E)$，路由花销$w(i,j)$，单播会话集K

输出：$\widetilde{P}_k$

(1) $\widetilde{P}_k, A, B \leftarrow \varnothing$, $\Omega_{ab}^k \leftarrow \{a,b\}$, $\inf \leftarrow \infty$, $n,m \leftarrow 0$

(2) **for** each single session $k \in K$ in $G(V,E)$ **do**

(3) $A \leftarrow$ the nodes of connecting to a

(4) $B \leftarrow$ the nodes of connecting to b

(5) $\deg_a^+, \deg_b^- \leftarrow$ compute the degree of a and b in $G(V,E)$

(6) **while** $n \neq \deg_a^+$ in $G(V,E)$ **do**

(7) $\widetilde{P}_k \leftarrow$ Dijkstra (a,b)

(8) $n \leftarrow$ count the number of node a in $\widetilde{P}_k$

(9) the first link routing cost of previous candidate path in $G(V,E) \leftarrow$ inf

(10) **end while**

(11) **while** $n \neq \deg_b^-$ in $G(V,E)$ **do**

(12) $n \leftarrow$ count the number of different nodes connecting to b in $\widetilde{P}_k$

(13) $\widetilde{P}_k \leftarrow$ Dijkstra $(a,B) \cup \{b\}$

(14) **end while**

(15) $\deg_i^+ \leftarrow$ compute the degree of i from A in $G(V,E)$

(16) **while** $m \neq \deg_i^+$ in $G(V,E)$ **do**

(17) $m \leftarrow$ count the number of different nodes connecting i in $\widetilde{P}_k$

(18) the second link routing cost of previous candidate path in $G(V,E)$

(19) $\widetilde{P}_k \leftarrow$ Dijkstra $(A,b) \cup \{a\}$

(20) **end while**

(21) **end for**

(22) **return** $\widetilde{P}_k$

算法 1 MRSA是将入、出口边缘节点对，与入口节点相邻的节点和出口节点作为节点对，与出口节点相邻的节点和入口节点作为节点对，并用Dijkstra算法选取多条候选路径的过程。

(1) 初始化：初始化相关参数，将每组会话的边缘节点对放到集合Ω_{ab}^k中，其中a表示入口边缘节点，b表示出口边缘节点(行1)。

(2) 候选路径集的选取：将每组会话与节点a, b相邻的节点分别放到集合A, B中，计算节点a的出度和节点b的入度。用Dijkstra算法计算拓扑中入、出口节点对之间的最短路，并放在$\widetilde{P}_k$中。若集合$\widetilde{P}_k$中节点a的个数不等于其出度，就把当前路径第1条链路的路由松弛为∞，再计算一次最短路，以此类推，不断松弛拓扑中当前路径的相应路由花销，最后可得$\deg_a^+$条候选路径(行2—行10)。接下来，利用Dijkstra计算原拓扑中与出口节点相邻的节点和入口节点作为节点对的最短路，若集合$\widetilde{P}_k$中与节点b相连的不同节点的个数不等于其入度，那就计算节点a到集合B中各节点的最短路，将所有路径分别与b求并后，放在$\widetilde{P}_k$中(行11—行14)。同理，用Dijkstra计算原拓扑中与入口节点的相邻节点和出口节点作为节点对的最短路，松弛当前路径的第2条链路的路由花销为∞，与节点a求并后，放到$\widetilde{P}_k$，以此类推，最终可以得到$\deg_b^-$条候选路径(行15—行20)。最后，输出候选路径集$\widetilde{P}_k$(行22)。

算法 2 为了均衡端口交换机相连链路的流量负载，DPAA根据会话集吞吐量及网络受限条件选择不同数量的路径和放置规则的交换机节点，其主要流程如表2所示。

(1) 初始化：初始化参数(行1)，将$\widetilde{P}_k$作为DPAA的输入(行2)。

(2) 计算过程：对于每组会话的候选路径集而言，首先统计出该组会话所有路径的最小瓶颈带宽(行5)，然后依据会话的吞吐量选择相应的传输路径，首先对每个规则子集利用公式$f(u)=r_k, \forall u \in R_k$映射相应的规则数量，同时限制每个节点的TCAM不能被超出(行6)，若会话的吞吐量远低于所有路径的最小带宽，则从候选路径集中任意选择一条满足条件的路径，并将规则放置在传输路径的任意不同的节点上(行7，行8)，若吞吐量大于所有路径的最小带宽，则从候选路径集中选择多条满足条件的路径，并将规则子集放置在不同路径的相交节点上(行9，行10)。通过遍历所有放置规则的节点，当不同的会话在该节点放置规则时，利用式(12)对规则进行合并(行15—行19)。最后，输出规则放置的数量、网络带宽利用率及路由花销(行20)。

3.2 复杂度分析

用n表示节点个数，在算法1中n个节点中度最大的为α, Dijkstra算法的时间复杂度为$O(n^2)$，行6—行10、行11—行14及行16—行20的最大循环次数为$O(\alpha n^2)$；行2—行21的循环次数为k。故算法1的

表 2　分布式防火墙部署算法

算法2：分布式防火墙部署算法(DPAA)

输入：单播会话集K，会话集的吞吐量d_k，候选路径集$\widetilde{P}_k$，规则子集R_k

输出：规则数量R_{all}，带宽利用率B_{use}，路由花销L_{cost}

(1) $R_{\text{all}}, B_{\text{use}}, L_{\text{cost}} \leftarrow 0$

(2) $P_k \leftarrow$ sort $\widetilde{P}_k$ from algorithm 1 into a two-dimension array

(3) **for** each single session $k \in K$ in $G(V,E)$ **do**

(4)　**for** $p \in P_k$ **do**

(5)　　$B_{\min} \leftarrow$ sort the minimum bandwidth of all paths

(6)　　**if** $\sum_{i \in V} r_u \le C_i$ **and** $R_k \subseteq \forall_{i \in V}\{i\}$ do

(7)　　**if** $d_k < B_{\min}$ **do**

(8)　　$l^p \leftarrow 1$, $x_i^{up} \leftarrow 1$

(9)　　**else**

(10)　　$l^p \leftarrow 1$, $l^p_{(i,j)} \leftarrow 1$, $x_i^u \leftarrow 1$, $x_i^{up} \leftarrow 1$

(11)　　**end if**

(12)　　**end if**

(13)　**end for**

(14) **end for**

(15) **for** all nodes **do**

(16)　**if** each node satisfy (12) **do**

(17)　　$x_{ui}^m \leftarrow 1$

(18)　**end if**

(19) **end for**

(20) **return** R_{all}, B_{use}, L_{cost}

整体复杂度为$O(3kn^2)$。

在算法2中，R_k记为β, $|P_k|$记为γ，行4—行13的时间复杂度为$O(n^2|\beta|^2|\gamma|)$；行3—行14的时间复杂度为$O(kn^2|\beta|^2|\gamma|)$；行15—行19的理想化最大时间复杂度为$O(n)$；故算法2的时间复杂度为$O(k|\beta|^2 \cdot|\gamma|n^2+n)$。

算法2的时间复杂度计算过程为

$$O\left(n^2 \times \left(\sum_{k=1}^{K}|R_k|^2\right) \times \sum_{k=1}^{K}|L_k|\right) = O(n^2|\beta|^2|\gamma|)$$

$$O(k \times n^2|\beta|^2|\gamma|) = O(kn^2|\beta|^2|\gamma|)$$

$$O(kn^2|\beta|^2|\gamma|) + O(n) = O(k|\beta|^2|\gamma|n^2+n)$$

4　实验结果及分析

4.1　实验环境参数

(1)硬件环境为Inter Core i5-6300HQ CPU 2.30 GHz, RAM 8 GB的Windos 10家庭版操作系统；利用Gurobi2.7.1[13]线性规划优化器求解混合整数线性规划，且提供相关接口在Python上运行，相关算法均在Pycharm 2017.2.3上编写。

(2)实验所用仿真拓扑来自SNDlib，采用真实的网络拓扑nobel-eu拓扑[14]，该拓扑有28个节点，41条链路。实验相关参数设定如下：$K=7$, $w(i,j) \in [1,10]$，$d_k \in (80,180)$，$B(i,j) \in (90,200)$，$C_i \in (1000,2000)$，$|R_k|=40$，$r_u \in (20,50)$。

4.2　典型会话流分析

本节将从两种典型会话流来分析规则子集重合程度对规则合并的影响。该节所用的拓扑是polska网络拓扑[15]，它由12个交换机和18条链路组成，假设在该网络拓扑中，3组单播会话A, B, C的吞吐量需求分别为100, 120 和80 Mb/s，每组会话需要分配20条规则到交换机中，每条链路的带宽(单位：Mb/s)范围为(90, 200)，每个交换机的TCAM为100，从[1, 10]中任意取一组数作为链路的花销。

会话流A的端点策略为：从边缘入口节点0流入网络，从边缘出口节点11流出网络，同理，会话流B为：从节点2流入，从节点7流出，会话流C为：从节点3流入，从节点4流出。如图1，图2中所示，实线的代表会话流A，虚线的代表会话流B，点划线的代表会话流C。会话流A的候选路径有：$p1$：0—4—5—8—9—11，$p2$：0—1—2—6—8—10—11，$p3$：0—4—7—8—10—11，$p4$：0—1—5—8—9—11，会话流B的候选路径有：$q1$: 2—6—8—7, $q2$: 2—1—0—4—7, $q3$：2—1—5—4—7，会话流C的候选路径有：$l1$: 3—2—6—8—5—4, $l2$: 3—6—8—5—4, $l3$: 3—2—6—8—7—4。

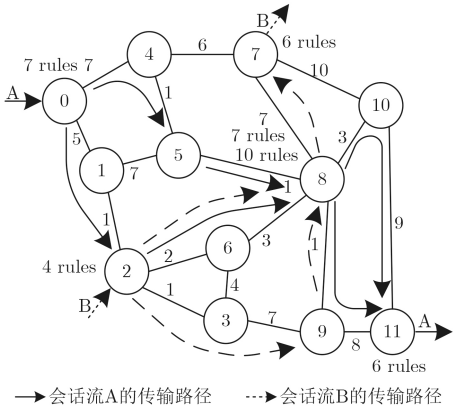

图 1 单播会话流场景1

在图1的两组会话都是多路由场景中，会话流A选择的路径分别为：传输速率为44 Mb/s的$p1$和传输速率为45 Mb/s的$p2$，会话流B选择的路径分别为：59 Mb/s的$q1$和71 Mb/s的$q2$，会话流A在节点0, 8, 11放置了复用规则，会话流B在节点2, 7, 8放置了复用规则，两组会话的规则在节点8进行了合并，通过表3对比发现，可合并的规则数量最多为5，若规则全都可以进行合并，那么合并后节点

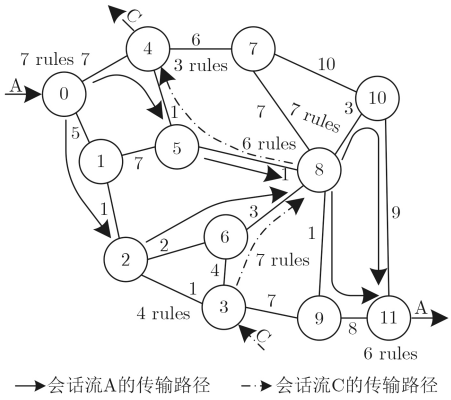

图 2 单播会话流场景2

表 3 不同场景下规则放置的节点及数量

典型会话场景	选择的节点：规则数量	规则总数量
场景1	会话流A: 0: 7, 8: 7, 11: 6 会话流B: 2: 4, 7: 6, 8: 10	[33, 40]
场景2	会话流A: 0: 7, 8: 7, 11: 6 会话流C: 3: 4, 4: 3, 6: 7, 8: 6	[34, 40]

8的规则数量为10，所以总的规则数量能够取得的下界为35，若规则都不能合并，那么放置在网络中规则的总数量为40。在图2的两组会话不都是多路由场景中，由于会话C的吞吐量需求低于最小链路带宽，仅选择了一条路径进行传输，所以会话流C选择了传输速率为80 Mb/s的路径$l2$，并将规则放置在了传输路径上的4个节点，其中在节点8，两组会话进行了规则合并，通过对比两组会话在同一节点放置规则的数量，可合并的规则数量为6，同理可知，网络中总的规则数量的范围为[34, 40]。

4.3 仿真分析

实验中每条链路的固定路由花销随机地从[1, 10]中取得，其中，正反向链路的路由花销是一样的。会话集的数量变化范围为[1,7]，相应吞吐量的需求从[80,180]中取得，每条链路带宽随机地从[90,200]取得，每个交换机的TCAM容量随机地从[1000,2000]取得。本文主要比较的算法是文献[6]中有候选路径的规则复用放置算法(Candidate Paths-Rule Multiplexing, RM-CP)以及有候选路径的无规则复用放置算法(non Candidate Paths-Rule Multiplexing, nonRM-CP)。RM-CP算法是将每组会话的规则放置在多条传输路径的相交节点上，nonRM-CP算法是将每组会话的规则沿着传输路径的节点任意地放置。

图3是3种算法在不同会话数量下，放置规则数的对比。从图中可以看出，3种算法放置的规则数随着会话数的增加，是逐渐递增的。在会话数较少时，HARA放置的规则数量与RM-CP算法相差不大，这是因为不同会话的传输路径相交的节点较少，可合并的规则较少。随着会话数的增多，相交的节点能够让更多的规则进行合并，尤其是当会话数目为7时，HARA比RM-CP算法最多能够节省18%的流表空间，比nonRM-CP算法最多能够节省56%的流表空间。从图4可以得知，HARA与non-RM-CP算法在不同单播会话数下，选择的传输路径数相差不大，但是放置的规则数量却远远小于nonRM-CP。

图5是3种算法在不同会话数下，已用带宽利用率的对比。已用带宽利用率指的是分配到每条传输路径的流量占相应路径的带宽资源比例。从图中可以看出，HARA的带宽利用率大部分是最低的，相应的剩余带宽资源也越多。这是因为HARA把每组会话的流量分配到不同的传输路径上，使得端口交

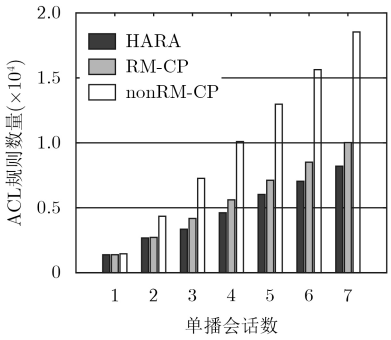

图 3 ACL规则数量对比

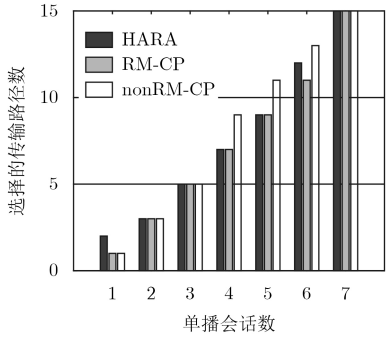

图 4 选择的路径数对比

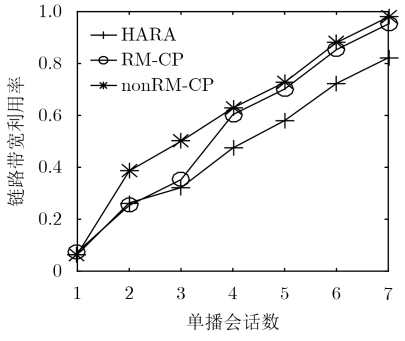

图 5 带宽利用率对比

换机相连链路的负载更加均衡。与RM-CP相比平均能节省10.3%的带宽资源，与nonRM-CP相比平均能节省13.1%的带宽资源。

图6是3种算法在不同会话数下，链路的路由花销对比。从图中可以看出，HARA花销值接近于RM-CP，但是低于nonRM-CP。当会话数目为4时，HARA低于nonRM-CP路由花销最多为114，当会话数为5时，HARA 最多高于RM-CP的花销为56。原因是RM-CP会选择与最短路中的链路有较多重复的路径，使得路由花销是最低的，nonRM-CP选取的路径数较多，使得相应的花销值总是较高的，而HARA在选择最短路的同时，还需要考虑端点链路的负载均衡，即选择了从源交换机不同链路出发，并从目的交换机不同链路进入的路径，所以两者的路由花销值总体相差不大，但总小于nonRM-CP。

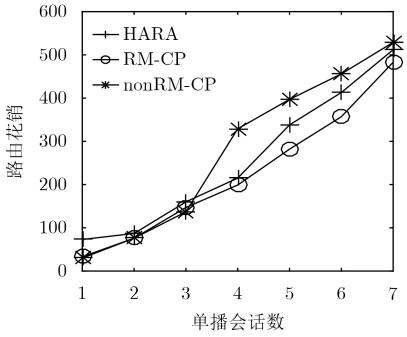

图 6 路由花销对比

5　结束语

本文提出了一种在端点交换机相连链路负载均衡的条件下，可应用在BCube, VL2等不同数据中心网络中[15]的规则放置算法。所提算法通过考虑到不同会话的吞吐量大小，选择了最佳的路由数量及节点位置，保证了会话集的QoS，节省了高昂的TCAM开销。然而在实际的网络当中流量请求是离散的[16]，所以，后续的工作，会从流的离散性特点以及规则分割所造成的规则依赖性特点[17]进一步研究网络流与ACL规则放置问题。

参 考 文 献

[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. doi: 10.1145/1355734.1355746.

[2] NGUYEN X N, SAUCEZ D, BARAKAT C, *et al*. Rules placement problem in openflow networks: A survey[J]. *IEEE Communications Survey & Tutorials*, 2016, 18(2): 1273–1286. doi: 10.1109/COMST.2015.2506984.

[3] KANG Nanxi, LIU Zhenming, REXFORD Jennifer, *et al*. Optimizing the “one big switch” abstraction in software-defined networks[C]. Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies, Santa Barbara, California, USA, 2013: 13–24. doi: 10.1145/2535372.2535373.

[4] KANIZO Y, HAY D, and KESLASSY I. Palette: distributing tables in software-defined networks[C]. 2013 Proceedings IEEE INFOCOM, Turin, Italy, 2013: 545–549. doi: 10.1109/INFCOM.2013.6566832.

[5] HUANG Jenfeng, CHANG Gueyyun, WANG Chunfeng, *et al*. Heterogeneous flow table distribution in software-defined networks[J]. *IEEE Transactions on Emerging Topics in Computing*, 2016, 4(2): 252–261. doi: 10.1109/TETC.2015.2457333.

[6] NGUYEN X, SAUCEZ D, BARAKAT C, *et al*. OFFICER: A general optimization framework for openflow rule allocation and endpoint policy enforcement[C]. 2015 IEEE Conference on Computer Communications (INFCOM), Kowloon, China, 2015: 478–486. doi: 10.1109/INFOCOM.2015.7218414.

[7] HUANG Huawei, GUO Song, LI Peng, *et al*. Joint optimization of rule placement and traffic engineering for QoS provisioning in Software Defined Network[J]. *IEEE Transactions on Computers*, 2015, 64(12): 3488–3499. doi: 10.1109/TC.2015.2401031.

[8] ASHRAF U. Rule minimization for traffic evolution in software-defined networks[J]. *IEEE Communications Letters*, 2017, 21(4): 793–796. doi: 10.1109/LCOMM.2016.2636212.

[9] GIROIRE F, MOULIERAC J, and PHAN T K. Optimizing rule placement in software-defined networks for energy-aware routing[C]. 2014 IEEE Global Communications Conference, Austin, USA, 2014: 2523–2529. doi: 10.1109/GLOCOM.2014.7037187.

[10] ZHANG Shuyuan, IVANCIC F, LUMEZANU C, *et al*. An adaptable rule placement for software-defined networks[C]. 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Atlanta, USA, 2014: 88–99. doi: 10.1109/DSN.2014.24.

[11] NGUYEN X, SAUCEZ D, BARAKAT C, *et al*. Optimizing rules placement in openflow networks: Trading routing for better efficiency[C]. Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, 2014: 127–132. doi: 10.1145/2620728.2620753.

[12] GARG N and KONEMANN J. Faster and simpler algorithms for multicommodity flow and other fractional packing problems[C]. Proceedings 39th Annual Symposium on Foundations of Computer Science, Palo Alto, USA, 2007: 630–652. doi: 10.1137/S0097539704446232.

[13] OPTIMIZATION G. Gurobi Optimizer Reference Manual[M]. 2013.

[14] Sndlib: Library of test instance for survivable fixed telecommunication network design[OL]. http://sndlib.zib.de/home.action, 2006.

[15] 史久根, 许辉亮, 陆立鹏. 软件定义网络中数据中心虚拟机迁移序列问题的研究[J]. 电子与信息学报, 2017, 39(5): 1193–1199. doi: 10.11999/JEIT160792.
SHI Jiugen, XU Huiliang, and LU Lipeng. Research on the migration queue of data center's virtual machine in software defined networks[J]. *Journal of Electronics & Information Technology*, 2017, 39(5): 1193–1199. doi: 10.11999/JEIT160792.

[16] 伊鹏, 刘洪, 胡宇翔. 一种可扩展的软件定义数据中心网络流调度策略[J]. 电子与信息学报, 2017, 39(4): 825–831. doi: 10.11999/JEIT160623.
YI Peng, LIU Hong, and HU Yuxiang. A scalable traffic scheduling policy for software defined data center network[J]. *Journal of Electronics & Information Technology*, 2017, 39(4): 825–831. doi: 10.11999/JEIT160623.

[17] KATTA N, ALIPOURFARD O, REXFORD J, *et al*. CacheFlow: Dependency-aware rule-caching for software-defined networks[C]. Proceedings of the Symposium on SDN Research, Santa Clara, USA 2016: 1–12. doi: 10.1145/2890955.2890969.

史久根：男，1963年生，副教授，研究方向为嵌入式系统、计算机网络和无线传感器网络.

王 继：男，1992年生，硕士生，研究方向为软件定义网络、网络功能虚拟化.

张 径：男，1993年生，硕士生，研究方向为软件定义网络、网络功能虚拟化和嵌入式系统.

徐 皓：男，1994年生，硕士生，研究方向为软件定义网络和嵌入式系统.