软件定义网络下两阶段大象流识别算法

刘向举,徐杨洋,方贤进,赵　犇
安徽理工大学 计算机科学与工程学院,安徽 淮南 232001

摘　要:目的 针对数据中心网络(Data Center Network,DCN)中数据流量多导致大象流与老鼠流识别精确度低的问题,提出一种基于软件定义网络(Software Defined Networking,SDN)下两阶段大象流识别算法。**方法** 将 SDN 与 DCN 结合,第一阶段,采用高斯分布动态阈值优化算法,通过对数据包阈值的设定,计算大象流误检率与漏检率,不断优化得到最优阈值,以此识别出可疑大象流;第二阶段,在依据流传输速率与流持续时间精确得到大象流的基础上,提出阈值约束、流量检测机制、Count 计数器等三方面改进对大象流识别阈值下限的约束,将网络中大象流的数据量与流持续时间进行周期内阈值计算,提高大象流的识别精确度。**结果** 实验结果表明:算法与已有相关算法相比,第一阶段可疑大象流平均字节数比网络流平均字节数多 11.3%;不同阈值下的算法准确度提高 1.7%,不同网络流量下的大象流平均检测时间降低至 6 ms 以内。**结论** 软件定义网络下两阶段大象流识别算法在第一阶段具有较强的大象流识别能力,同时算法的精确度有所提高,大象流的平均检测时间降低,提高了网络质量,能为进行网络流量调度策略的进一步研究提供相关性条件。

关键词:数据中心网络;软件定义网络;大象流;高斯分布;最优阈值
中图分类号:TP393　　**文献标识码:**A　　doi:10.16055/j.issn.1672-058X.2024.0003.012

Two-stage Elephant Flow Recognition Algorithm in Software Defined Network

LIU Xiangju, XU Yangyang, FANG Xianjin, ZHAO Ben
School of Computer Science and Engineering, Anhui University of Science and Technology, Anhui Huainan 232001, China

Abstract: Objective Aiming at the problem that the high data traffic in Data Center Network(DCN) leads to the low recognition accuracy of elephant flow and rat flow, a two-stage elephant flow recognition algorithm based on Software Defined Networking(SDN) was proposed. **Methods** Combining SDN and DCN, the dynamic threshold optimization algorithm of Gaussian distribution was adopted in the first stage to calculate the false detection rate and missing rate of elephant flows by setting the packet threshold, and the optimal threshold was continuously optimized to identify the suspicious elephant flows. In the second stage, based on the accurate elephant flow obtained from the suspicious elephant flow in the previous stage according to the transmission rate and duration of the flow, the threshold constraint, traffic detection mechanism and Count counter were proposed to improve the constraint of the lower limit of the elephant flow identification threshold, and the data volume and duration of the elephant flow in the network were calculated as the threshold value within the cycle, so as to improve the identification accuracy of elephant flows. **Results** The experimental results showed that compared with the existing algorithms, the average number of bytes of the suspicious elephant flows in the first stage was 11.3% more than that of the network flows. Under different thresholds, the accuracy of the algorithm

收稿日期:2022-12-11　**修回日期:**2023-03-02　**文章编号:**1672-058X(2024)03-0089-09
基金项目:国家自然科学基金项目(61572034);安徽省科技重大专项(18030901025).
作者简介:刘向举(1978—),男,硕士,副教授,从事物联网、智能控制和软件定义网络研究.
通讯作者:徐杨洋(1998—),男,硕士研究生,从事软件定义网络研究. Email:2294257862@qq.com.
引用格式:刘向举,徐杨洋,方贤进,等. 软件定义网络下两阶段大象流识别算法[J]. 重庆工商大学学报(自然科学版),2024,41(3):89—97.
LIU Xiangju, XU Yangyang, FANG Xianjin, et al. Two-stage elephant flow recognition algorithm in software defined network[J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2024, 41(3): 89—97.

was increased by 1.7%, and the average detection time under different network traffic was reduced to less than 6 ms. **Conclusion** The two-stage elephant flows recognition algorithm in the software-defined network has strong elephant flow recognition ability in the first stage. At the same time, the accuracy of the algorithm is improved, the average detection time of the elephant flows is reduced, and the network quality is improved, providing relevant conditions for further study of the network traffic scheduling strategy.

Keywords: data center network; software-defined network; elephant flows; Gaussian distribution; optimal threshold

1 引 言

DCN[1-3]作为大型设备网络,在对数据信息进行存储和处理中,具有信息实时共享性。随着互联网地不断发展,网络流量不断激增,流量传输效率以及流量的正确识别在网络中格外重要。对于DCN来说,不仅要对网络中信息存储容量以及转发速度有较高要求,而且需要对网络中的不同类型流量进行精确识别,从而实现DCN中流量的实时监测与高效流量调度,提高链路的有效利用率与数据传输的稳定性。目前,针对数据中心网络流量研究表明[4-8]:DCN中存在着两种不同类型的流量,其中一类是对时延敏感的老鼠流,另一类为对链路带宽敏感的大象流。老鼠流虽然在网络中流数量占比较大,但却携带少量的数据包;大象流虽然流数量占比较少,却携带大量的数据包。大象流需要更多的吞吐量才能满足流量的正常调度,相反,老鼠流应考虑满足时延需求。将SDN与DCN结合,利用OpenFLow技术将数据平面与控制平面进行分离,高效实时地进行网络流量的控制,提高对大象流与老鼠流的识别是实现流量调度[9]的基础,是提高DCN中流量传输效率的重要措施。

王宏等[10]通过Hits算法和Holds算法来减少误检率和过滤器的级数,同时避免使用多级过滤器,算法对识别精确度有一定的改善,但大象流阈值设置不准确;严军荣等[11]提出一种大象流两级识别方法,利用TCP队列发送进行可疑大象流的识别,并将超过可疑阈值的发送队列数据量进行存储,然后依据流持续时间将可疑大象流进行精确流判,大象流精确度有一定提升,但未考虑控制器开销;肖军弼等[12]为了在进行数据中心动态优先多路径调度时通过Pareto重尾模型计算大象流动态阈值,先对大象流采样阈值进行定义,计算采样数据包采样概率,在采样数据包阈值条件下,对总数据包的数量进行阈值判断,该算法在模型上符合大象流与老鼠流的分布模型,但仅通过Pareto重尾模型进行动态阈值计算的识别,精确度较低;白雪等[13]提出基于SDN的两级大象流负载均衡策略,通过两阶段对大象流进行识别,第一阶段通过大象流与老鼠流的分布情况进行阈值设置,第二阶段通过流持续时间剔除变质的大象流,从而增加大象流检测的精确度;汤倩[14]提出一种新的大象流检测方法—ADU(Auto Detect Upload),分为ADU-Server和ADU-Client两个部分,当产生可疑大象流时,ADU-Client会将生成的伪数据包通过交换机转发到控制器中,控制器通过伪源IP地址对其进行判断,但仅对不同大象流阈值下的平均耗时进行实验,未展开进行监测时间与精确度实验;金玲等[15]提出在DCN中的二级自适应大象流检测系统对大象流进行检测,在预处理阶段通过抽样进行可疑大象流识别,在第二阶段,基于上一阶段识别的可疑大象流,根据设置约束条件,通过考虑正误率与负错误率的最小值,获取精确大象流,从而提高大象流检测准确率;Tang等[16]提出一种有效的采样和分类(Efficient Sampling and Classification, ESCA)方法,通过预测大象流的到达间隔,筛选掉次样本从而降低流量检测时间与开销,使用基于数据流之间相关性的新监督分类算法有效地对样本进行分类,实现低带宽的特点,但两者都采用采样的方式进行大象流检测方法,处理样本开销大,检测频率高;刘奕等[17]提出一种基于动态流量学习(Dynamic Traffic Learning, DTL)自适应阈值的大象流检测,首先根据DCN中流量变化规律满足高斯分布,建立自适应阈值的高斯分布加权优化方法对网络中流量进行预测,其次通过控制器将动态阈值与预测结果进行分析,识别出大象流,但算法的检测时间较高;贾咏哲[18]首先根据大象流与老鼠流在DCN中的高斯分布状态进行大象流阈值约束,其次根据大象流最低阈值进行老鼠流初筛,明显降低检测时间,但准确率较低。

针对以上文献中出现的检测时间长、准确性差的问题,本文提出两阶段大象流识别算法。第一阶段,根据大象流与老鼠流在DCN中高斯分布的特点,设置动态阈值,不断进行阈值优化识别获得可疑大象流集合;第二阶段,将上一阶段的可疑大象流基于流持续时间与流传输速率双向考虑进行精确大象流判断,同时增加流检测机制,并考虑下限阈值设定与Count计数三方面的改进,增加大象流的实时性约束,减小流检测时间,提高大象流识别的精确度。

2 大象流检测模型

在网络拓扑与DCN中实时网络状态约束下,建立软件,定义网络下两阶段大象流识别算法。当网络数

据流下发时,控制器与交换机之间通过网络对数据流进行收集:第一阶段预分类出可疑大象流进行预处理;第二阶段精分类出大象流,构建出大象流检测模型。本文选择的网络拓扑结构为典型的胖树拓扑结构,将网络拓扑构造成有向图 $G=(N,L)$ 网络结构,N 表示网络中交换机节点的集合,$N=\{n_1,n_2,\cdots,n_k\}$。其中,$k=|N|$ 表示交换机个数;L 表示网络中链路的集合,$L=\{l_1,l_2,\cdots,l_m\}$,其中 $m=|L|$ 表示链路个数。本文算法主要包括三大模型:网络拓扑发现、流量监控与两阶段大象流识别。模型架构如图 1 所示。

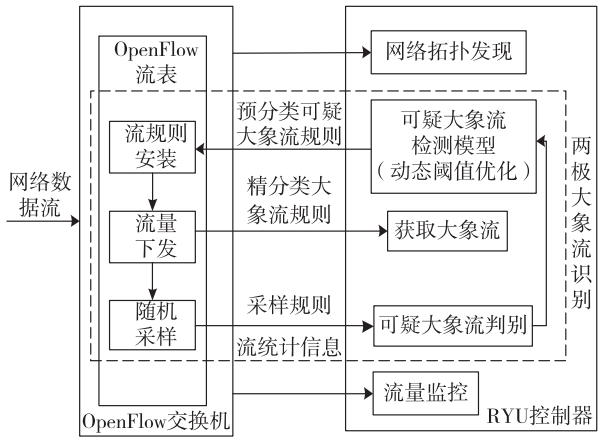


图 1　两阶段大象流识别模块架构

Fig. 1　Architecture of two-stage elephant flow recognition module

2.1　网络拓扑发现

RYU 控制器周期性向交换机发送链路层发现协议(Link Layer Discovery Protocol,LLDP)报文以获取 Packet_out 信息,当交换机接收到 RYU 控制器下发的信息时,交换机将 LLDP 报文封装到 Packet_in 中,再次发送到控制器从而获取网络拓扑信息。

2.2　流量监控

控制器向交换机发送请求,首先对流端口信息进行解析,其次通过 RYU 控制器下 ofproto_v1_3_parser 中的 OFPFlowStatsRequest()协议以及与 OFPPortStatsRequest()协议对流端口进行下发,控制器解析消息得到流量统计信息 tx_{pkts}、tx_{bytes}、tx_{error} 等,运用流量统计信息计算链路速率、丢包率以及链路可用带宽等。

定义 1　链路 i 的速率 $l_{ri}(\Delta t)$ 计算:

$$l_{ri}(\Delta t)=\frac{tx_{\text{bytes}(t+\Delta t)}-tx_{\text{bytes}(t)}}{\Delta t} \tag{1}$$

式(1)中,$tx_{\text{bytes}(t+\Delta t)}$ 与 $tx_{\text{bytes}(t)}$ 分别表示在 $t+\Delta t$ 与 t 时刻链路 i 传输的字节数。

定义 2　链路 i 的可用带宽:

$$l_{bi}=B_{\max}-sr_i(\Delta t) \tag{2}$$

式(2)中,$B_{\max}$ 表示链路 i 端口的最大带宽。

定义 3　路径的可用带宽:

$$B_{\text{path}}=\min\{l_{b1},l_{b2},l_{b3},\cdots,l_{b_i}\} \tag{3}$$

式(3)中,i 满足 $i\in[1,n]$;路径可用带宽为链路 i 的可用带宽的最小值。

2.3　大象流定义

大象流在网络流量实时监控中进行大量数据流传输,并在持续时间内进行数据传输。由数据中心的研究结果[19]可知,网络中有将近 20% 的大象流占据着 80% 的网络带宽,当数据流的传输速率超过链路容量的 10% 时,将该数据流定义为大象流。

定义 4

$$\lambda=\frac{tx_{\text{bytes}(t+\Delta t)}-tx_{\text{bytes}(t)}}{l_{bi}\Delta t} \tag{4}$$

式(4)中,当 $\lambda>0.1$ 时,将其定义为大象流,以便后期进行大象流与老鼠流分离。

3　TIEF 算法设计

根据 DCN 中大象流与老鼠流的存在状态,设计并提出两阶段识别大象流(Two-stage Identification Elephant Flows,TIEF)算法。第一阶段根据数据流在 DCN 中流量呈高斯分布,通过对数据包阈值的设定,计算大象流误检率与漏检率,不断优化得到最优阈值;第二阶段提出基于流传输速率与流持续时间的大象流识别算法,通过两个周期 T_c 时间的传输速率 θ_e 计算可疑流的传输字节数,将可疑流流持续时间 d_i 与流的持续时间阈值 θ_d 比较,若 $d_i\geqslant\theta_d$,将可疑流判定为大象流,并在此基础上引入下限阈值、大象流实时性约束以及 Count 计数器表示不同时间的重要性,提高阈值的预测精度。

3.1　预分类可疑大象流检测

由定义 4 可得:当 $\lambda>0.1$ 时,将其定义为大象流;若在周期 Δt 中,检测到数据流量为 m,数据包个数为 N,数据流 i 接收到的数据包为 p_i,若数据流 i 产生的流量大小 m_i 大于所设定的 λ,即 $m_i\geqslant m\lambda$,则将数据流 i 视为可疑大象流。通过 RYU 控制器对网络进行实时监控,将产生的数据流进行大象流阈值 θ 划分。为了进一步精确大象流阈值 θ,在预先设定 θ 后进行动态阈值优化,如图 2 所示。在优化过程中,将老鼠流识别为大象流的错误识别设为误检率(False Positive Rate,其值用 F_{PR} 表示)和遗漏下来的大象流漏检率(False Negative Rate,其值用 F_{NR} 表示)。将大象流错误率 E 降为最低,计算出划分的大象流阈值 θ,误检率 F_{PR} 与漏检率 F_{NR} 如式(5)与式(6)所示:

$$F_{\text{PR}}=\frac{FP}{FP+TN}=P[p_i\geqslant\theta\,|\,m_i<m\lambda] \tag{5}$$

$$F_{\text{NR}}=\frac{FN}{FN+TP}=P[p_i<\theta\,|\,m_i\geqslant m\lambda] \tag{6}$$

式(5)、式(6)中,FP 为标记为老鼠流而被当作大象流

误检的流数目;*FN* 为标记为大象流而当成老鼠流漏检的流数目。大象流与老鼠流在数据中心网络中遵循一维高斯分布,大象流的数据量服从 $x_e \sim (\mu_e, \sigma_e^2)$,老鼠流的数据量服从 $x_m \sim (\mu_m, \sigma_m^2)$。当大象流错误率最小时,将两者高斯分布的交点设为最优阈值 θ,则数据量超过最优阈值的老鼠流叫做误检流,数据量小于最优阈值的大象流叫做漏检流。因此,大象流的概率密度如式(7)所示:

$$f(x_e)=\frac{1}{\sqrt{2\pi\sigma_e^2}}\exp\left[-\frac{(x_e-\mu_e)^2}{2\sigma_e^2}\right] \tag{7}$$

同理,老鼠流的概率密度函数如式(8)所示:

$$g(x_m)=\frac{1}{\sqrt{2\pi\sigma_m^2}}\exp\left[-\frac{(x_m-\mu_m)^2}{2\sigma_m^2}\right] \tag{8}$$

对于 DCN 中的老鼠流,若当前数据量超过最优阈值 θ,则大象流被误检,此时的误检率 F_{PR} 如式(9)所示:

$$F_{PR}(i)=\int_{\theta}^{+\infty} g(x_m)\,\mathrm{d}x \tag{9}$$

对于 DCN 中的大象流,若当前数据量小于最优阈值 θ,则大象流将被漏检,此时的漏检率 F_{NR} 如式(10)所示:

$$F_{NR}(j)=\int_{-\infty}^{\theta} f(x_e)\,\mathrm{d}x \tag{10}$$

在 DCN 中由于大象流与老鼠流混合存在,将两者进行区分的首要任务是通过大象流误检率与漏检率进行衡量,因此漏检率要将数据流视为老鼠流进行处理,误检率要将数据流视为大象流进行处理。设 DCN 中大象流占总数据流比例为 p_e,老鼠流占总数据流比例为 p_m,则对于任意一条网络流误检率与漏检率如式(11)与式(12)所示:

$$F_{PR}=p_m\cdot F_{PR}(i)=p_m\cdot\int_{\theta}^{+\infty} g(x_m)\,\mathrm{d}x \tag{11}$$

$$F_{NR}=p_e\cdot F_{NR}(j)=p_e\cdot\int_{-\infty}^{\theta} f(x_e)\,\mathrm{d}x \tag{12}$$

将漏检率与错误识别率进行综合评价得出任意流的识别错误率如式(13)所示:

$$E_r=F_{PR}+F_{NR}=p_m\cdot\int_{\theta}^{+\infty} g(x_m)\,\mathrm{d}x+p_e\cdot\int_{-\infty}^{\theta} f(x_e)\,\mathrm{d}x \tag{13}$$

若使得错误识别率最低,则令 E_r 对数据量 x 的偏导得 0,如式(14)所示:

$$\frac{\partial E_r}{\partial x}=0 \tag{14}$$

设大象流与老鼠标准差同为 σ,如式(15)所示:

$$\sigma_e=\sigma_m=\sigma \tag{15}$$

由式(13)、式(14)、式(15)可得最优阈值 θ 如式(16)所示:

$$\theta=\frac{(\mu_e^2-\mu_m^2)+2\sigma\ln\left(\frac{p_e}{p_m}\right)}{2(\mu_e-\mu_m)} \tag{16}$$

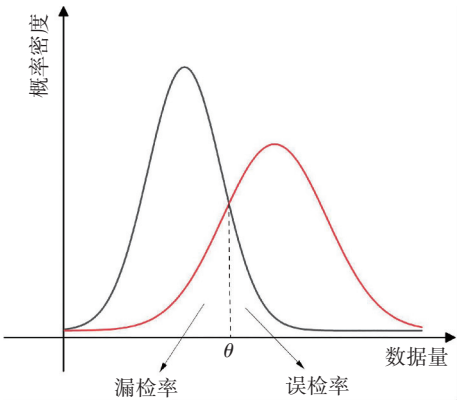


图 2 误检率与漏检率分布图

Fig. 2 Distribution of F_{PR} and F_{NR}

第一阶段预分类出的可疑大象流检测算法流程如图 3 所示。

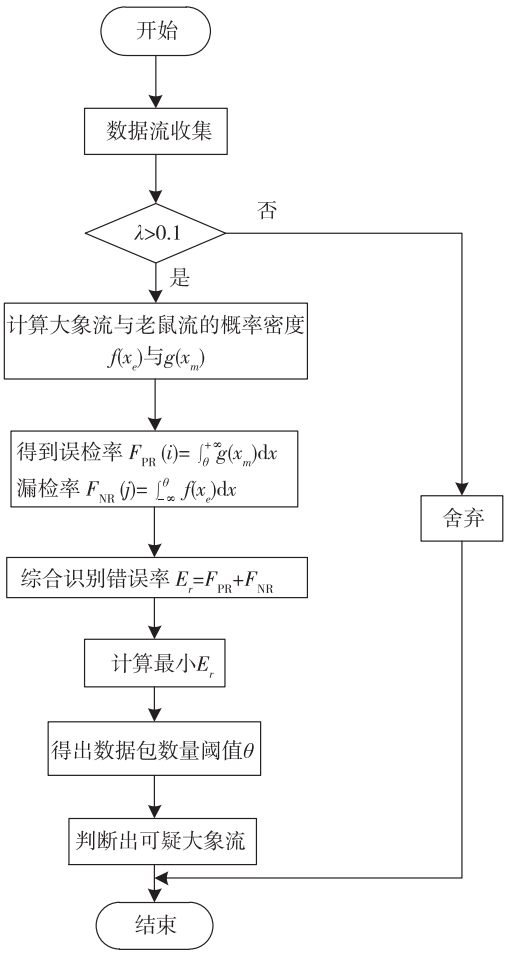


图 3 预分类出的可疑大象流检测算法流程图

Fig. 3 Flow chart of pre-classified suspicious elephant flow detection algorithm

3.2 精分类出大象流

第二阶段精分类出大象流中基于流传输速率与流

持续时间的大象流识别(Flow Transmission Rate and Flow Duration,FTRFD)算法。算法依据第一阶段预分类出的可疑大象流的流传输速率与流持续时间实现对大象流的精确判断。第一阶段产生的数据包通过 OpenFlow 交换机发送到 RYU 控制器,RYU 控制器根据上一阶段产生的大象流在两个周期 T_c 时间内的传输速率 θ_e 计算可疑流的传输字节数与传输速率,判断两个周期内的传输字节数,进行真实大象流判断,通过数据流的传输速率与流持续时间对可疑大象流进行精确筛选。FTRFD 算法步骤如下:

步骤 1　对单位时间内数据包进行计算,当有新流下发时,存储其流信息。

步骤 2　RYU 控制器根据上一阶段产生的可疑大象流在周期 T_c 时间内的传输速率 θ_e 计算可疑流的传输字节数 b,若两周期内的字节数相等,则认定为老鼠流;若传输字节数不相等,计算可疑流流持续时间 d_i,进行步骤 3。

步骤 3　将可疑流流持续时间 d_i 与流持续时间阈值 θ_d 比较,若 $d_i \geqslant \theta_d$,将可疑流判定为大象流。

步骤 4　输出大象流集合 S_e。

FTRFD 算法流程如表 1 所示。

表 1　FTRFD 算法流程

Table 1　Flow of FTRFD algorithm

算法 1:FTRFD 算法
输入:可疑大象流 f_i,可疑大象流集合 M_e,网络流 i 数据量大小 D_i,周期 T_c,流持续时间阈值 θ_d。
输出:大象流集合 S_e;
步骤 1　if $D_i>\theta$ then　# 网络流 i 可判定为可疑大象流 $f_i=i$; 将可疑大象流 f_i 存储到可疑大象流集合 M_e 中; end if

步骤 2　for f_i in M_e do
通过计时器计算流持续时间 d_i;
两个周期 T_c 时间的传输速率 θ_e 计算可疑流的传输字节数 b_1 与 b_2
if $b_1=b_2$　then
将其流视为老鼠流;
else　更新流持续时间 d_i;
end if
end for

步骤 3　if $d_i \geqslant \theta_d$ then 将其判定为大象流并存储到大象流集合 S_e 中;
步骤 4　得到大象流集合 S_e。

3.3　约束改进

在 FTRFD 算法的基础上,通过增加下限阈值、对大象流实时约束、加入 Count 计数器等方法对其进行改进,以解决对大象流检测识别时出现少量老鼠流干扰以及无法及时动态进行大象流阈值更新等问题。

(1) 阈值约束。大象流阈值设定与大象流数量以及 DCN 中的总流量具有相关性,当阈值设置过大,会使得大象流数占总流量的比例过小,若阈值设置过小,将会使得大象流出现误检。根据 DCN 流特征,由于数据中心网络中 80%的网络数据量小于 10 KB,因此将数据流量下限阈值设置为 10 KB,即 $Th_{\text{lower}}=10$ KB。当低于 10 KB 时,将其确定为老鼠流,从而提高大象流检测的精确度。

(2) 大象流实时性约束。在网络流进行数据下发时,根据上一周期内流量信息作为先验知识对当前周期流量变化进行分析,对网络中大象流的数据量与流持续时间引入流量检测机制 β:

$$\beta=D_i/d_i \tag{17}$$

式(17)中,网络流 i 的数据量大小为 D_i,流持续时间为 d_i。周期内大象流阈值识别算法如表 2 所示。

表 2　周期内大象流阈值识别算法

Table 2　Elephant flow threshold recognition algorithm in period

算法 2:周期内大象流阈值识别算法
输入:可疑大象流 f_i,大象流集合 S_e,网络流 i 的数据量大小 D_i,流持续时间 d_i,周期 T_c。
输出:周期内大象流阈值 γ;
步骤 1 令流量检测机制 $\beta=\infty$;

步骤 2　for f_i in S_e do
if　$D_i/d_i<\beta$ then
更新期内流量检测机制 β 且且 $\beta=D_i/d_i$;
end if
end for

步骤 3　计算周期内大象流阈值 $\gamma=\beta T_c$。

(3) Count 计数器。在有限次的循环中不断获取最优阈值,通过加入 Count 计数器来减少流检测时间,当满足条件时,自动剔除可疑大象流。

3.4　TIEF 算法流程与算法复杂度分析

3.4.1　TIEF 算法流程

TIEF 算法在 DCN 中对大象流与老鼠流进行精确识别,主要包括两阶段:第一阶段对数据包阈值进行设定,通过计算大象流误检率与漏检率,不断优化得到最优数据包阈值;第二阶段提出 FTRFD 算法,并引入下限阈值、对大象流实时约束、加入 Count 计数器等方法对其进行改进,最终得到大象流集合 S_e。TIEF 算法流程如表 3 所示。

表 3 TIEF 算法流程

Table 3 Two-stage identification elephant flow algorithm

算法 3:TIEF 算法

输入:可疑大象流 f_i,大象流集合 S_e,网络流 i 的数据量大小 D_i,流持续时间 d_i,周期 T_c。

输出:周期内大象流阈值 γ;

步骤 1 预分类大象流检测 ;

根据公式(16)计算预分类最优阈值 θ;

if $\theta<Th_{\text{lower}}$ then #预分类阈值小于下限阈值

$\theta=Th_{\text{lower}}$; #更新阈值

end if;

if $D_i>\theta$ then # 网络流 i 可判定为可疑大象流

$f_i=i$;

将可疑大象流 f_i 存储到可疑大象流集合 M_e 中;

end if

步骤 2 精分类出大象流

for Count=0 do

Count=Count+1;

通过算法 2 根据流传输速率与流持续时间的大象流识别;

步骤 3 更新大象流集合 S_e。

3.4.2 TIEF 算法复杂度分析

TIEF 算法首先通过步骤 1 中 DCN 流量所呈现的高斯分布状态预分类出大象流检测,算法的时间主要存在于预先设定 θ 后进行动态阈值优化,其复杂度为 $O(N)$;步骤 2 的时间复杂度主要与计数器 Count 及周期 T_c 有关,在 N 次计数与更新期内,流量检测机制中使用的时间复杂度为 $O(T_cN^2)$,因此 TIEF 算法的时间复杂度为 $O(N^2)$。由于 TIEF 算法经过有限次迭代后收敛,具有实时性与有效性。

4 仿真实验与结论

4.1 实验环境

实验所使用的机器配置为 Intel © Core™ i7-0710U CPU@ 1.10 GHz,16.0 GB RAM,操作系统为 Ubuntu16.0.4-desktop-64 位 4.0GB。本实验所采用的仿真环境平台为 Mininet,交换机为 Open vSwich(OVS)虚拟交换机,控制器为 RYU 控制器,实验拓扑选择为胖树网络拓扑($k=4$),实验拓扑如图 4 所示。

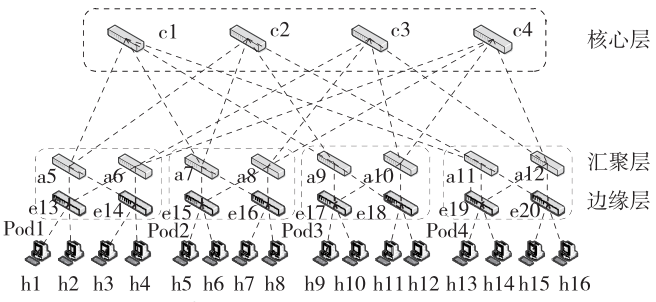


图 4 胖树($k=4$)数据中心网络拓扑结构

Fig. 4 The data center network topology of fat-tree($k=4$)

4.2 参数设置

实验中设置 RYU 控制器监督周期为 5 s,拓扑发现周期为 2 s,链路带宽设置为 10 Mbit/s。TIEF 算法在第一阶段预分类出的可疑大象流参数设置如表 4 所示。

表 4 预分类出的可疑大象流参数设置

Table 4 Parameter settings for pre-classified suspicious elephant flows

参 数	数 值
p_e	0.1
p_m	0.9
σ_e	0.01
σ_m	0.01
μ_e	5 MB 或 10 MB
μ_m	5 KB 或 10 KB

(1) 数据包平均字节比较。对预分类可疑大象流检测进行分析,对数据文件大小数量的改变进行多次实验,利用 RYU 控制器对可疑大象流集合的数据包平均字节数进行计算,并与所给予的网络流数据包平均字节数比较,包字节数定义为

$$\overline{P}_{\text{bytes}}=\frac{F_{tb}}{N_{tb}} \tag{18}$$

式(18)中,F_{tb} 为传输的总字节数,N_{tb} 为传输的总数据包数。

图 5 所示为第一阶段流数据包平均字节数比较。实验中,在第一阶段预分类出的可疑大象流检测中选择不同数据文件进行模拟,并进行 5 次实验,通过 RYU 控制器将可疑大象流的包平均字节数与网络流平均字节数进行比较可知:5 次实验中,大象流平均字节数均超过网络平均字节数,且在实验 1 中可疑大象流平均字节数比网络流平均字节数多 11.3%,可疑大象流平均字节数接近传输单元字节数。由此可知:在第一阶段具有较强的大象流识别能力。

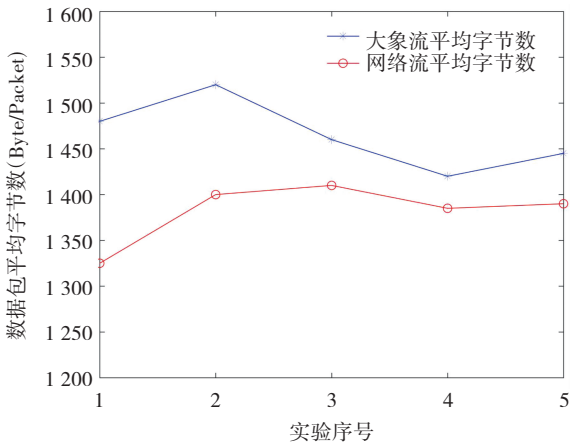


图 5 流数据包平均字节数

Fig. 5 The average number of bytes of flow packets

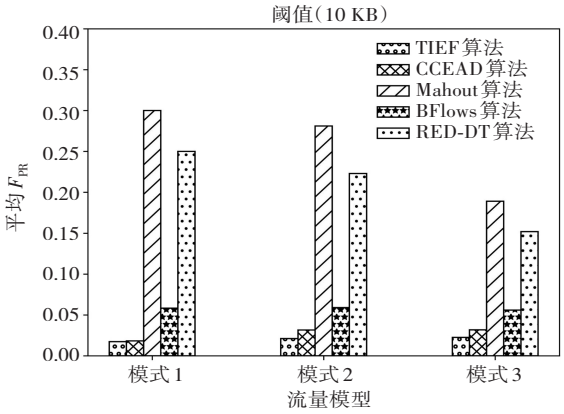
(2) 不同静态阈值下的算法精确度。使用 iperf 流量产生工具在胖树网络拓扑间产生流量,选择其中一条链路,本文选择主机 h1 生成网络流量,再将传输生成的网络流量传输到 h8,实验所进行的测试分为 3 组,选择对比算法为 Mahout 算法[20]、CEEAD 算法[18]、BFlows 算法[21]、RED-DT 算法[22],在 3 组测试模型中将其设置为自变量,计算因变量平均 F_{PR} 与平均 F_{NR}。

表 5　实验实测参数取值

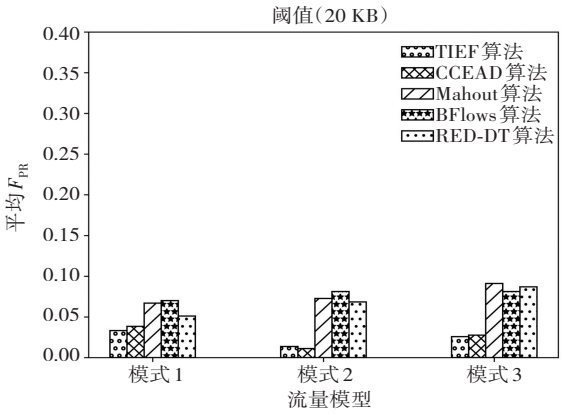
Table 5　Values of the parameters measured in the experiment

参　数	模式 1	模式 2	模式 3
p_e	0.1	0.1	0.1
p_m	0.9	0.9	0.9
σ_e	0.01	0.01	0.01
σ_m	0.01	0.01	0.01
μ_e	10 MB	10 MB	5 MB
μ_m	10 KB	5 KB	5 KB

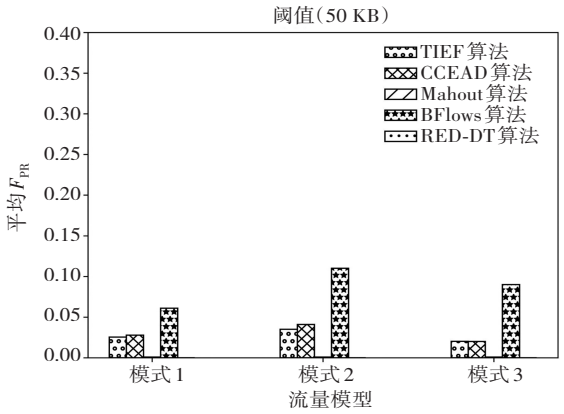
图 6 所示为在阈值为 10 KB、20 KB、50 KB 进行大象流识别测量算法时的精确度。本文在不同测试模式下进行仿真实验 20 次,取 20 次实验结果的平均值为最终实验结果,求得算法在 3 种测试模式下的平均 F_{PR} 与平均 F_{NR}。F_{PR} 与 F_{NR} 为 DCN 中大象流检测下互相影响的指标,当 F_{PR} 上升时,F_{NR} 会相对降低。将不同阈值下的 F_{PR} 与 F_{NR} 进行求和计算,当算法精确度高时,平均 F_{PR} 与平均 F_{NR} 的和相对较小。由图 6 可知:Mahout 算法作为静态大象流识别算法,其识别主要取决于静态阈值的设置,若静态阈值设置满足当前要求,则算法的精确度较高,当网络测试模式改变时,该算法的精确度会大幅度降低,当静态阈值为 50 KB 时,Mahout 算法的平均 F_{PR} 几乎为 0,但平均 F_{NR} 较高。BFlows 算法通过在初始化时进行速率阈值设置的方式来代替静态阈值的方法,根据网络传输速率对大象流与老鼠流进行区分,算法在较短周期内会有较大的波动,因此算法精确度不是最优。RED-DT 算法由于只在第二阶段采用流持续时长来设置持续时长阈值,未考虑大象流的阈值约束与实时性约束,算法在阈值为 50 KB 时,有较低的平均 F_{PR},但平均 F_{NR} 较高。CEEAD 算法整体效果较好,但与本文算法相比,本文在精分类出的大象流中,对流持续时间与传输速率进行综合考虑,且在实时性约束与计数周期内完成,阈值为 10 KB 时,TIEF 算法的精确度较 CEEAD 算法平均提高 1.7%;阈值为 20 KB 时,TIEF 算法的精确度较 CEEAD 算法平均提高 1.28%;阈值为 50 KB 时,TIEF 算法的精确度较 CEEAD 算法平均提高 1.21%。因此本文的算法精确度最高。



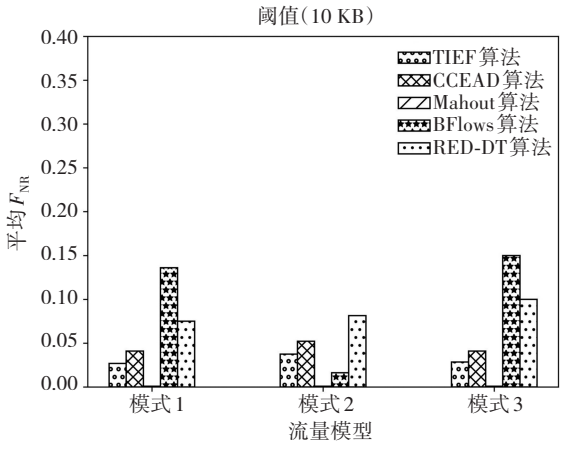
(a) 阈值为 10 KB 的平均 F_{PR}



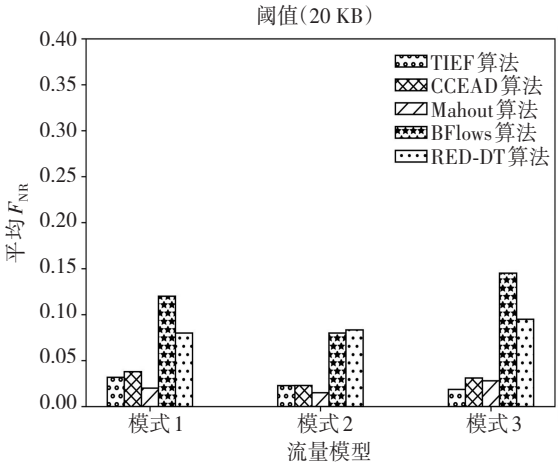
(b) 阈值为 20 KB 的平均 F_{PR}



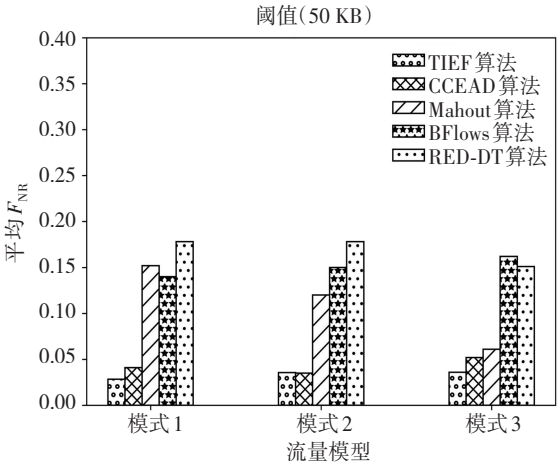
(c) 阈值为 50 KB 的平均 F_{PR}



(d) 阈值为 10 KB 的平均 F_{NR}



(e) 阈值为 20 KB 的平均 F_{NR}



(f) 阈值为 50 KB 的平均 F_{NR}

图 6 测试下的平均 F_{PR} 与平均 F_{NR}

Fig. 6 Average F_{PR} and average F_{NR} in the test

(3) 平均检测时间比较。图 7 所示为主机 h1 与 h8 在通信过程中大象流平均检测时间比较。实验选择 0.1 MB、0.5 MB、1 MB、5 MB、10 MB 等 5 种不同网络流量进行测量,针对 5 种不同网络进行仿真实验 20 次,取 20 次实验结果检测时间的平均值为最终实验结果。在本文中,大象流检测时间主要包括第一阶段的可疑大象流识别时间与第二阶段的流传输持续时间。由图 7 可以看出:当网络流量在 0.1 MB~1 MB 时,TIEF 算法可以快速转发且精确度高,平均检测时间在 2.6 ms 与 5 ms 之间;由于算法数据包转发正常且可以快速收集,网络流量在 0.1 MB~0.5 MB 时,CEEAD 算法、Mahout 算法、BFlows 算法、RED-DT 算法的平均检测时间相差不大,保持在 13 ms 以内;当网络流量超过 0.5 MB 时,随着网络流量的增加,网络链路负载的增加,BFlows 算法、RED-DT 算法的平均检测时间在不断上升,TIEF 算法均能稳定在 6 ms 以内。由于大象流在网络中存在时长在几秒以内,因此 TIEF 算法具有有效地识别率。

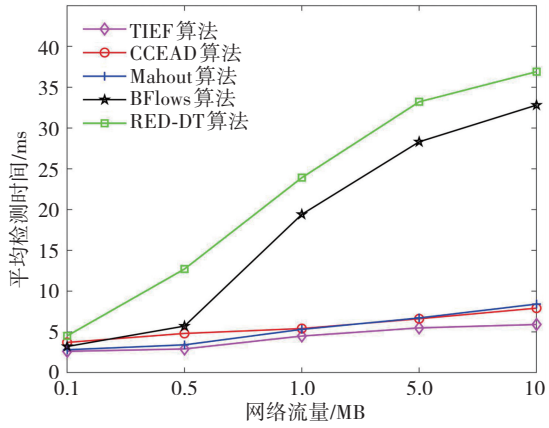


图 7 平均检测时间

Fig. 7 Average detection time

4.3 结 论

在 Ubuntu 环境下通过 Mininet 进行仿真,选用胖树网络拓扑,使用 TIEF 算法进行两阶段大象流识别。第一阶段通过高斯分布的动态阈值优化算法预处理出可疑大象流;第二阶段使用 FTRFD 算法,进一步添加阈值约束、大象流实时性约束以及 Count 计数器以增加精确度;最后对第一阶段数据包的平均字节、测试下的平均 F_{PR} 和平均 F_{NR}、平均检测时间三方面进行实验对比并分析。在 5 次实验中,大象流平均字节数均超过网络平均字节数;本文提出的 TIEF 算法在 3 种不同流量模式下,精确度最高;当网络流量不断增加时,平均检测时间不断上升,但均稳定在 6 ms 以内。TIEF 算法的识别能力更加准确,有利于为后续大象流的拥塞路由调度提供有力的基础。

5 结束语

研究了 DCN 中大象流与老鼠流共存时高效进行两阶段大象流识别的策略。在 SDN 网络架构下,通过 TIEF 算法,能够高效精确地将大象流从网络流中识别出,提高网络质量,有利于进行网络流量调度策略的进一步研究。通过实验表明:该算法具有更好的识别精确度,较低的流检测时间。通过仿真实验对大象流进行识别,选择有规则的网络拓扑,如何在真实环境中进行高效合理的大象流与老鼠流网络流量调度有待进一步地研究。

参考文献(References):

[1] 段晨, 彭伟, 王宝生. 数据中心网络路由研究进展[J]. 计算机工程与科学, 2022, 44(4): 631—644.
DUAN Chen, PENG Wei, WANG Bao-sheng. Research progress of data center network routing [J]. Computer Engineering and Science, 2022, 44(4): 631—644.

[2] ZAHER M, ALAWADI A H, MOLNAR S. Sieve: A flow scheduling framework in SDN based data center networks[J]. Computer Communications, 2021, 171: 99—111.

[3] ZAW H T, MAW A. Traffic management with elephant flow detection in software defined networks(SDN)[J]. International Journal of Electrical and Computer Engineering, 2019, 9(4): 3203—3211.

[4] 曾嘉麒,刘外喜,卢锦杰. 软件定义数据中心基于残差网络的大象流预测机制[J]. 小型微型计算机系统, 2021, 42(9): 1938—1943.
ZENG Jia-qi, LIU Wai-xi, LU Jin-jie. Elephant flow prediction mechanism based on residual network in software defined data center[J]. Small and Micro Computer Systems, 2021, 42(9): 1938—1943.

[5] 刘向举,徐杨洋,方贤进,等. 软件定义网络下大象流拥塞路由调度算法[J]. 湖北民族大学学报(自然科学版), 2022, 40(2): 181—189.
LIU Xiang-ju, XU Yang-yang, FANG Xian-jin, et al. Elephant flows congestion routing scheduling algorithm in software defined networks [J]. Journal of Hubei University for Nationalities (Natural Science Edition), 2022, 40(2): 181—189.

[6] 杜鑫乐,徐恪,李彤,等. 数据中心网络的流量控制: 研究现状与趋势[J]. 计算机学报, 2021, 44(7): 1287—1309.
DU Xin-le, XU Que, LI Tong, et al. Flow control in data center networks: Research status and trends [J]. Chinese Journal of Computers, 2021, 44(7): 1287—1309.

[7] XU LEI. ABQ: Active buffer queueing in datacenters [J]. IEEE Network, 2020, 34(2): 232—237.

[8] 程克非,高江明,段洁,等. 面向 SDN 的数据中心网络更新研究综述[J]. 电讯技术, 2017, 57(10): 1224—1232.
CHENG Ke-fei, Gao Jiang-ming, Duan Jie, et al. Review of data center network update for SDN[J]. Telecommunication Technology, 2017, 57(10): 1224—1232.

[9] 高健文,黄友锐,徐善永,等. 基于膜计算的蚁群算法在配电网 WSNs 中路由研究[J]. 重庆工商大学学报(自然科学版), 2021, 38(3): 50—57.
GAO Jian-wen, HUANG You-rui, XU Shan-yong, et al. Ant colony algorithm based on membrane computing for WSNs routing in distribution network [J]. Journal of Chongqing Technology and Business University(Natural Science Edition) 2021, 38(3): 50—57.

[10] 王宏,龚正虎. Hits 和 Holds: 识别大象流的两种算法[J]. 软件学报, 2010, 21(6): 1391—1403.
WANG Hong, GONG Zheng-hu. Hits and Holds: Two algorithms for recognizing elephant flows[J]. Chinese Journal of Software, 2010, 21(6): 1391—1403.

[11] 严军荣,叶景畅,潘鹏. 一种大象流两级识别方法[J]. 电信科学, 2017, 33(3): 36—43.
YAN Jun-rong, YE Jing-chang, PAN Peng. A two-stage recognition method for elephant flows[J]. Telecommunication Science, 2017, 33(3): 36—43.

[12] 肖军弼,程鹏,谭立状,等. 基于 SDN 的数据中心动态优先级多路径调度算法[J]. 计算机与现代化, 2020(7): 21—26.
XIAO Jun-bi, CHENG Peng, Tan Li-zhuang, et al. Dynamic priority multi-path scheduling algorithm for Data Center based on SDN[J]. Computer & Modernization, 2020(7): 21—26.

[13] 白雪. 基于 SDN 的数据中心网络大象流负载均衡的研究与实现[D]. 兰州: 兰州交通大学, 2020.
BAI Xue. Research and implementation of elephant flow load balancing in data center network based on SDN[D]. Lanzhou: Lanzhou Jiaotong University, 2020.

[14] 汤倩. 基于 SDN 的数据中心网络大象流检测与调度策略研究[D]. 长沙: 湖南师范大学, 2019.
TANG Qian. Research on elephant flow detection and scheduling strategy in data center network based on SDN[D]. Changsha: Hunan Normal University, 2019.

[15] 金玲,束永安. 数据中心网络中基于 SDN 的大象流负载均衡的研究[J]. 计算机应用研究, 2019, 36(1): 203—205.
JIN Ling, SHU Yong-an. Research on load balancing of elephant flow based on SDN in data center network [J]. Application Research of Computers, 2019, 36(1): 203—205.

[16] TANG F, ZHANG H, YANG L T, et al. Elephant flow detection and load-balanced routing with efficient sampling and classification [J]. IEEE Transactions on Cloud Computing, 2019, 9(3): 1022—1036.

[17] 刘奕,李建华,陈玉. 基于自适应阈值的大象流检测方法[J]. 计算机工程与应用, 2022, 58(3): 159—164.
LIU Yi, LI Jian-hua, CHEN Yu. Elephant flow detection method based on adaptive threshold[J]. Computer Engineering and Applications, 2022, 58(3): 159—164.

[18] 贾咏哲. 基于 Ryu 的 SDN 拓扑发现机制及路由算法研究[D]. 南京: 南京理工大学, 2020.
JIA Yong-zhe. Research on topology discovery mechanism and routing algorithm of SDN based on Ryu[D]. Nanjing: Nanjing University of Science and Technology, 2020.

[19] 张坤. 高速网络中大象流精确识别方法研究[D]. 长沙: 长沙理工大学, 2021.
ZHANG Kun. Research on accurate identification method of elephant flow in high-speed network[D]. Changsha: Changsha University of Science and Technology, 2021.

[20] CURTIS A R, KIM W, YALAGANDULA P. Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection[C]//2011 Proceedings IEEE INFOCOM. IEEE, 2011: 1629—1637.

[21] 黄马驰. 基于 SDN 的数据中心网络流量调度策略研究[D]. 重庆: 重庆邮电大学, 2017.
HUANG Ma-chi. Research on network traffic scheduling strategy of data center based on SDN [D]. Chongqing: Chongqing University of Posts and Telecommunications, 2017.

[22] AKYILDIZ I F, LEE A, WANG P, et al. Research challenges for traffic engineering in software defined networks [J]. IEEE Network, 2016, 30(3): 52—58.

责任编辑:李翠薇