

# 一种减少网络振动的智能路由选择算法设计

邵天竺 王晓亮 陈文龙 唐晓岚 徐 敏

(首都师范大学信息工程学院 北京 100048)

(nestea\_god@hotmail.com)

## Design of an Intelligent Routing Algorithm to Reduce Routing Flap

Shao Tianzhu, Wang Xiaoliang, Chen Wenlong, Tang Xiaolan, and Xu Min

(College of Information Engineering, Capital Normal University, Beijing 100048)

**Abstract** Recently, researchers have begun to focus on data-driven network protocol design methods to replace traditional protocol design methods that rely on human experts. While the resulting intelligent routing technology is rapidly developing, there are also problems to be solved urgently. This paper studies the large-scale routing flapping caused by the current intelligent routing algorithm in the routing update process and the resulting decrease in forwarding efficiency of network. A smart routing algorithm, named FSR(flap suppression routing), for route flapping suppression is proposed. While pursuing the uniform link load of the entire network and making full use of the forwarding resources of the entire network, FSR seeks an update plan that is most similar to the existing routing strategies. This reduces routing flapping in each routing update cycle, reduces route convergence time, and improves overall network forwarding efficiency. Experiments have shown that FSR algorithm can significantly improve the routing convergence speed, increase the network throughput by about 30% compared with the control algorithms, and significantly reduce the path length and the probability of congestion.

**Key words** routing algorithm; machine learning; deep neural network; traffic planning; routing flap

**摘要** 近来,研究人员开始关注基于数据驱动的智能网络协议设计方法,以此取代依赖人类专家的传统协议设计方式。智能化路由技术也随之得到快速发展,但仍存在亟待解决的问题。研究了当前智能路由算法在路由更新过程中带来的大范围路由抖动以及转发效率下降问题,提出了一种路由抖动抑制的智能路由选择算法 FSR(flap suppression routing),在追求全网链路负载均匀、转发资源高利用率的同时,寻求与现有路由策略最相似的更新方案,使得每个路由更新周期的路由抖动减小,缩短路由收敛时间,提升网络整体转发性能。实验表明:FSR 算法能显著提升路由收敛速度,与对照算法相比提升约 30% 的网络吞吐量,同时降低路径长度和拥塞概率。

**关键词** 路由算法;机器学习;深度神经网络;流量规划;网络振动

**中图法分类号** TP393

---

收稿日期:2020-12-25;修回日期:2021-04-08

基金项目:国家重点研发计划项目(2018YFB1800403);国家自然科学基金项目(61872252);北京市自然科学基金项目(4202012)

This work was supported by the National Key Research and Development Program of China (2018YFB1800403), the National Natural Science Foundation of China (61872252), and Beijing Natural Science Foundation (4202012).

通信作者:王晓亮(wangxiaoliang@cnu.edu.cn)

近年来互联网和移动通信产业的快速发展,网络系统不断向着规模化、异构化、动态化的方向演进<sup>[1]</sup>.同时,伴随着5G网络的成熟,万物互联将带来网络终端数量、网络流量以及应用形式上的新一轮爆发性增长,并对数据转发速率、超低延迟、高能效比和大规模连接提出更高的要求<sup>[2]</sup>.由此带来的服务能力与复杂性问题使得当前的网络系统面临无数新的挑战,现有尽力而为的路由转发算法难以满足这些应用所带来的多样化的网络服务质量需求.

随着网络环境的不断发展,当面对突发流量或大流量时,基于最短路径的传统路由协议可能导致严重的网络拥塞.互联网需要一种更加智能的路由策略,将网络状态与路由策略融合,提升网络服务质量.然而互联网庞大的网络规模和数据流量使得智能机制的设计充满挑战.

机器学习技术近来已取得高速发展,在计算机网络方面,有监督和无监督的人工神经网络(ANN)技术从路由策略到入侵检测的各种领域中得到广泛应用.尽管传统的浅层人工神经网络经常被用于主动网络管理的流量预测,然而其性能实际上是相对受限的<sup>[3]</sup>,因为单纯增加ANN的隐藏层数量难以改善网络操作决策(例如调度、路由等)的性能.然而,深度学习系统(例如Deep Belief Networks, Deep ANNs和Deep Boltzmann)的快速发展给网络领域的研究提供了新的突破点,它的算法性能显著提高<sup>[4]</sup>.

以深度学习技术为基础的智能路由算法设计,其核心思想是利用大数据驱动的网络特征获取机制,从部署范围的历史数据中寻找路由选择与时间、或路由选择与流量间的映射关系,从而指导后续路由优化.

相比传统数学模型支持下的路由策略设计,数据驱动的智能路由算法具有复杂度低和通用性强等优势.其中,复杂度低主要体现在其避免了针对网络复杂特性和多样需求的数学建模工作,以历史流量数据为基础,通过神经网络的迭代训练代替精确建模,大幅降低核心映射的获取复杂度;通用性强则体现在相同的机器学习模型,可以面向不同的网络环境和需求场景,采用不同的数据集来求解对应的差异化问题.

然而,随着智能化路由技术的发展,新的技术挑战也随之而来.通过实验我们发现,目前的智能路由算法在基于历史数据进行路由更新时,很容易因为局部流量的微小变动而导致大范围的、无关流量的调整,在每个调整周期造成大范围的链接断连、丢包

以及路径切换,对网络传输性能带来了负面影响.显然,这是智能路由算法迫切需要优化的问题.

本文中,我们提出了一种对路由抖动敏感的智能路由调度算法FSR(flap suppression routing).FSR的核心思想是在实现全网链路负载均匀分配、最大化利用全网转发资源的同时,寻求与当前路由选择变动最小的更新方案,使得每个更新周期带来的路由抖动尽可能小,路由快速收敛,提升网络整体转发效率.实验发现,FSR算法能够明显提升路由收敛速度,与对照算法相比提升约30%的网络吞吐量,并显著减少网络丢包和链路拥塞.

## 1 问题描述

路由选择是互联网网络层的核心机制,稳定而高效的路由选择可以保障一个网络系统良好的运行.传统路由协议的核心建立在最短路径的选择上,而并未考虑网络当前的负载分布和未来可能的流量分布情况,所以传统路由面对复杂网络状态特别是突发流量的情况下容易造成网络拥塞.如图1(a)左图所示,在节点A和节点B之间的链路由于突发流量造成拥塞时,基于最短路径的路由算法无法有效避让.更令人失望的是,由于传统路由选择算法不具备从历史流量分布中学习规律的能力,所以造成的拥塞情况在流量规律性很强的场景下会反复发生,严重降低网络使用体验.

因此,设计一种新型的智能化路由策略是网络发展的迫切要求.它应融合网络当前状态信息,从网络的历史行为中寻找规律,进行智能化路由选择,提升网络运行效率,实现如图1(a)右图所示的效果.目前,已有一些相关工作利用深度学习进行路由选择.其中,算法LRD<sup>[5]</sup>提出了独特的流量矩阵,将流量矩阵作为深度学习的特征输入,并且在最大化链路利用率的问题上给出了较好的解决方案.算法RDL<sup>[6]</sup>则首次将CNN神经网络应用在流量工程上,与LRD不同的是,RDL以流量需求矩阵做为特征输入,对于突发性流量具有更好的适应性.

然而,经过实验研究发现,目前智能路由算法仍然存在进一步改进的空间.由于目前的智能路由算法主要是学习历史流量的分布特征与路由选择直接的映射关系,并将其应用到当前流量环境的路由选择问题中,所以当网络流量分布发生变化时,很容易引起较大范围的路由抖动.如图1(b)左图所示,预期节点D和节点B间链路可用性降低时,原本仅需

调整数据流 2 即可,即将数据流 2 的转发路径调整为 A—B 链路。但由于当前智能路由算法尚未考虑

路由振动,故无关的转发路径 1 也受到影响,重新进行了选路,这无疑会影响整个网络的传输效率。

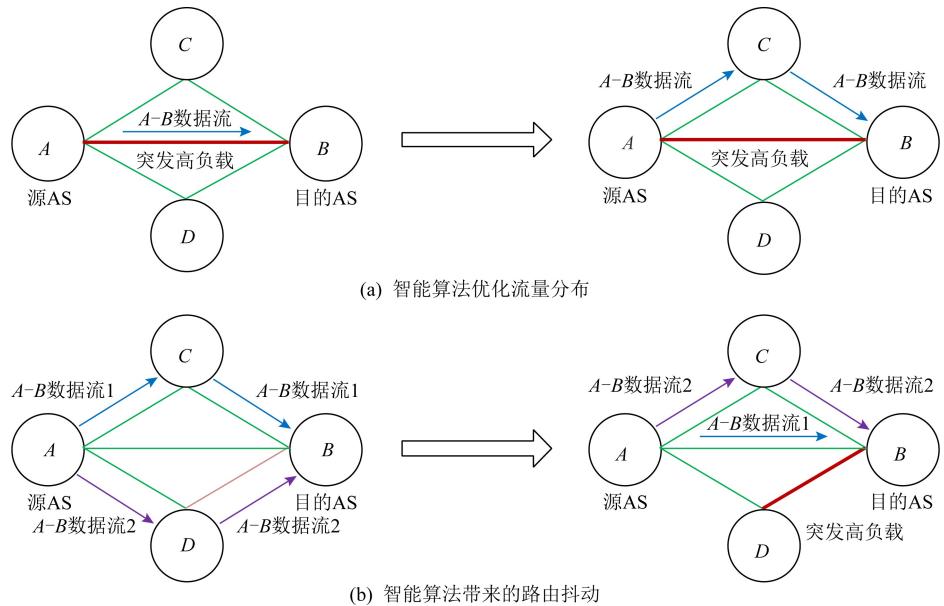


Fig. 1 Problem description diagram

图 1 问题描述示意图

我们在 NS3 仿真平台上设计了一个 18 条链路构成的拓扑,对该网络重放 30 min 的测试流量并每 5 min 重构网络路由,并观察 3 个重构周期,结果如表 1 所示:

Table 1 Impact of Oscillation Caused by Route Reconfiguration on Network Transmission

表 1 路由重构引起的振动对网络传输的影响

算法	实验内容	周期 1	周期 2	周期 3	平均值
RDL	受影响的单跳链路	9	10	10	9.7
	受影响的完整路径	4	4	6	4.7
LRD	传输带宽/Mbps	176	224	289	229.93
	受影响的单跳链路	10	15	11	12
	受影响的完整路径	5	6	4	5
	传输带宽/Mbps	135	276	293	234.43

RDL 算法平均每次路由更新都会导致约 9.7 条链路受到影响,比例达到 50% 以上,而受到影响的完整转发路径达到 4.7 条。LRD 算法相类似,在同样实验条件下,它会导致约 12 条链路受到影响,比例达到约 70%,受影响的转发路径达到 5 条。这无疑会影响整体网络的转发性能。在 300 Mbps 带宽的链路上,RDL 和 LRD 算法分别取得的平均带宽为

229.93 Mbps 和 234.43 Mbps,其中,在第 1 个路由更新周期调整完成后,2 种算法取得的平均带宽均受到严重影响,分别下降到 176 Mbps 和 135 Mbps,只能达到最大带宽的 58.67% 和 45%,传输效率受到显著影响。

于是,针对这一问题,本文提出了一种振动抑制的智能路由选择算法 FSR,在图 1(b)右图所示的情况下,仅调整受 D—B 链路影响的数据流 2,将因此带来的路由抖动收敛到最小状态,进而提升网络传输效率。由此,保证尽可能少的改动网络路由选择,同时提升网络资源的利用率。

## 2 FSR 算法设计

不同于现有研究工作,FSR 需要关注转发路径的变化,尽可能抑制端到端路由抖动,其主要设计思路有 2 个特征:

1) 在挖掘历史流量的特征信息时,不满足于传统机制对链路状态与时间关联规律的探寻,而是寻找路径-链路-时间 3 个维度的变化特征。本文设计了如图 2 所示的模型结构,通过链路、路径 2 个维度构建流量矩阵,再增加时间通道刻画二维流量矩阵的时间变化特性。以此做为输入数据,使得训练得到的神经网络可以更加全面地刻画流量特征,从而寻找

路由动荡最小的全局路由方案.

2) 本算法针对不同的网络规模, 其输入矩阵也存在较大差异.

考虑到随输入参数增加而可能出现的梯度消失或梯度爆炸问题, FSR 算法选择具备卷积层的 CNN 神经网络以更好地适应不同规模的网络输入.

因此, 本节将从算法的优化目标入手, 着重研究 FSR 算法的输入输出处理和 CNN 神经网络设计, 以此为基础完成抖动抑制算法的设计和实现.

FSR 算法中的 CNN 网络结构如图 2 所示. 整体结构由卷积层、池化层和全连接层构成, 每一层的参数说明为:

第 1 层(L1). 该层为输入层, 每个输入样本为 10@ $30 \times 18$  的输入矩阵, 其中 10 代表共有 10 个时间采集点, 30 为整个网络中有 30 条链路, 18 表示 18 个 OD 对.

第 2 层(C2). 该层为卷积层, 主要作用是对原始输入样本进行空间滤波, 因此该层与输入层之间的

连接是局部连接. 在该层使用 15 种滤波器, 每种滤波器去卷积输入矩阵就得到不同特征的映射, 即得到 15 个特征图. 卷积核的大小设置为  $2 \times 2$ , 每个特征图的大小为  $16 \times 16$ .

第 3 层(P3). 该层为池化层, 主要作用是减少模型特征提取方面的误差. 我们设置  $2 \times 2$  的池化窗口, 采用最大池化的方式对上一层数据进行池化操作.

第 4 层(F4). 该层为卷积层, 作用与第 2 层(C2)相同, 在该层使用 20 种滤波器, 每种滤波器去卷积输入矩阵就得到不同特征的映射, 即得到 20 个特征图. 卷积核的大小设置为  $2 \times 2$ , 每个特征图的大小为  $16 \times 16$ .

第 5 层(H5). 该层为池化层, 参数与第 3 层(P3)相同.

第 6 层(O6)与第 7 层(B7). 这 2 层为全连接层 (第 3 隐含层), 作用是配合前面的卷积层和后面的输出层, 组成分类部分, 因此该层前后都是全连接. 神经元个数分别为 60 个和 30 个.

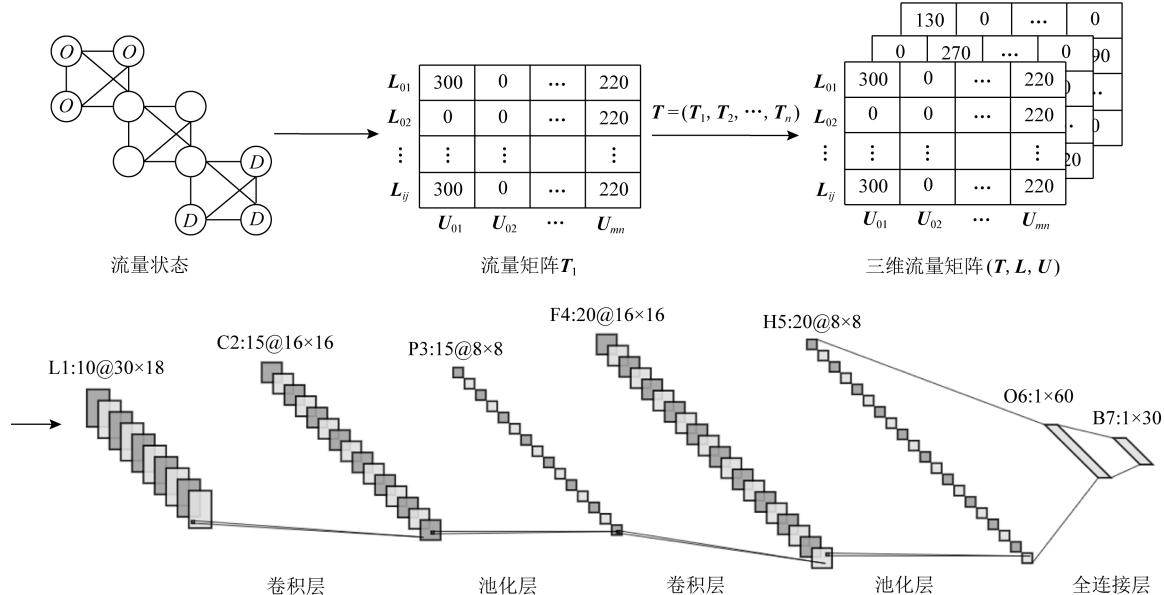


Fig. 2 Data processing and neural network design

图 2 数据处理与神经网络设计

## 2.1 设计目标

考虑到当前智能路由算法可能带来的路由抖动问题, 本文的优化目前将以全网流量的均匀分布作为约束条件, 以最小化相邻时间片的路由选择为优化目标, 设计整体智能路由算法.

具体来说, 令一个网络拓扑表示为无向图  $G = \langle V, E \rangle$ , 其中顶点集合  $V$  表示网络中的路由节点, 边集合  $E$  表示网络中的链路. 记网络中某一链路

$e \in E$  的负载为  $Load_e$ , 同时, 记时间片  $t$  内所有源—目的对间的转发路径集合  $Path^t = \{path_1^t, path_2^t, \dots, path_n^t\}$ , 其中  $path_n^t$  为组成路径  $n$  的链路集合. 则优化目标可以描述为

$$\begin{cases} \text{Max}(Load_e) < K, \forall e \in E, \\ \text{Min}\left(\sum_{path_i \in Path} (path_i^t - path_i^{t+1})\right). \end{cases}$$

为了求解这一优化问题, FSR 算法将采用深度

学习算法得到满足  $\text{Max}(Load_e) < K$  约束条件的可行解范围,之后在可行解范围内求解链路改变最小的最优解.

## 2.2 输入输出

FSR 算法中,满足约束条件的可行解范围由 CNN 深度学习神经网络给出.针对 CNN 神经网络,本节首先讨论算法的数据初始化处理.

区别于传统流量工程中对网络流量的矩阵描述,本文中所设计的流量矩阵,同时包含路径和链路的流量分布状态,在横向和纵向 2 个维度的特征关联性更强,这有助于 CNN 的卷积层更好地获取这 2 个维度间的关联特征.我们还在输入上设计了时间通道,使整个数据集更加具有时间连续上的特征.

为了描述网络历史流量特征,FSR 算法采用如图 2 所示的三维矩阵  $(\mathbf{T}, \mathbf{L}, \mathbf{U})$  作为算法输入.其中,  $\mathbf{T} = (\mathbf{T}_b, \mathbf{T}_{b-1}, \dots, \mathbf{T}_{b-h+1})$  为历史流量数据的时间片;  $\mathbf{L} = (\mathbf{L}_{01}, \mathbf{L}_{02}, \dots, \mathbf{L}_{ij})$  表示节点  $i$  到节点  $j$  间的链路上的流量;  $\mathbf{U} = (\mathbf{U}_{01}, \mathbf{U}_{02}, \dots, \mathbf{U}_{mn})$  表示节点  $m$  到节点  $n$  的完整路径上的流量.对矩阵中某一行求和,可以得知所有通过该链路的传输路径的总流量.而观察某一列的非空项,则可以得知该时间片内此源—目的(OD)对间的路径所包含的链路,横坐标代表了路径情况,而纵坐标表示链路情况.我们对于不同的路由拓扑输入的矩阵大小是不同的,所以对于一旦处理维度较大的输入矩阵时,ANN 的训练参数会急剧增加,容易出现梯度消失或爆炸的现象.而在前面加上了卷积层的 CNN 能更好地去适应不同的路由拓扑.

训练数据采集自 MAWI Working Group Traffic Archive 所提供的 WIDE 到上游 ISP 的传输链路上的跟踪流量.首先,将网络总体流量数据按照时间片分割为流量序列,这里时间片长度可以任意指定,时间片长度为 5 min.之后,将该流量序列输入到提前构建好的网络拓扑中(网络拓扑具体形式参加第 4 节实验部分),按照 OSPF 协议进行该流量序列的路由转发,并以 30 s 的间隔观察网络流量分布情况,对于任意链路的负载  $Load_e > K$  的情况,剔除该流量分布数据,将剩余的所有流量分布情况做为训练数据使用.将训练数据转化为网络流量特征矩阵的形式则可得到最终的训练集  $DataSet_{train}$ .

FSR 算法中,CNN 输出的结果是当前流量分布情况下的可行解范围,具体的形式是针对每一个 OD 对,给出其可行路径方案的概率值.FSR 算法是去解决一个多分类的问题,于是本方案不仅为最终

的结果进行了标签的标注,同时为每一个 OD 对的路径进行了序号的标注,于是可以准确地通过标签来反映出拓扑路径的选择.

首先,需要在算法运行之前,针对每一个 OD 对,给出其可选的路径方案集合  $Path$ ,为了控制神经网络算法的可行解空间,FSR 算法需要对  $Path$  集合进行预处理,设每个源—目的对间的可行路径数量上限为  $n$ ,于是,对于给定的网络拓扑  $G = \langle V, E \rangle$  中,FSR 算法需要的路径集合  $Path$  为

$$Path = \{Path_{(i,j)}, | \forall i, j \in V, Num(Path_{(i,j)}) \leq m, \\ Length(Path_{(i,j)}) \leq n\}.$$

FSR 算法中 CNN 在此基础上,针对每一个 OD 对  $(i, j)$ ,输出其  $Path$  集合中每一条路径的选择概率  $p$ ,即 CNN 的输出为

$$Output_{CNN} = \{(Path_{(i,j)}, p), | \forall i, j \in V\}.$$

之后,FSR 算法在 CNN 的输出集中,按照选择概率  $p$  降序排列,并计算每一个路径方案  $Path_{(i,j)}$  所对应的链路抖动幅度,最终将抖动幅度最小的方案做为 FSR 算法的输出,进行下一时间片的路由调整.

## 2.3 神经网络设计

CNN 神经网络中的神经元采用 ReLU 激活函数,具体形式为

$$f(x) = \max(0, x).$$

CNN 神经网络中,卷积层的正向传播函数为

$$a^l = \sigma(z^l) = \sigma(a^{l-1} * \mathbf{W}^l + \mathbf{b}^l),$$

其中,  $a^l$  为第  $l$  层神经元的输出,  $z^l$  为第  $l$  层神经元的输入,  $\mathbf{W}^l$  为从  $l-1$  层映射到  $l$  层的权值矩阵,  $\mathbf{b}^l$  为与上面参数对应的偏移值,  $\sigma$  为激活函数 ReLU.另外,FSR 中采用均方差来度量损失值,对应的损失函数可为

$$J(\mathbf{W}, \mathbf{b}, x, y) = \frac{1}{2} \|a^l - y\|_2^2,$$

其中,  $x$  为训练集输入,  $y$  为训练集中期望的输出结果.

按照 CNN 神经网络的基本思想我们给出全连接层也就是 DNN 神经网络正向传播的公式:

$$a^l = \sigma(z^l) = \sigma(\mathbf{W}^l a^{l-1} + \mathbf{b}^l).$$

根据随机梯度下降法可以给出全连接反向传播公式:

$$\delta^l = (\mathbf{W}^{l+1})^\top \delta^{l+1} \odot \sigma'(z^l).$$

**算法 1.** CNN 全连接层算法.

输入:最大迭代次数  $Max$ 、学习率  $\alpha$ 、输入的样本对  $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ ;

输出:各层的  $\mathbf{W}, \mathbf{b}$ .

for  $iter=1$  to  $Max$

  for  $i=1$  to  $m$

    set  $a^l = x_i$ ;

    for  $l=2$  to  $L$

      前向传播计算  $a^l = \sigma(z^l) = \sigma(\mathbf{W}^l a^{l-1} +$

$\mathbf{b}^l)$ ;

    end for

    for  $l=L$  to 2

      反向传播计算  $\delta^l = (\mathbf{W}^{l+1})^\top \delta^{l-1} \odot$

$\sigma'(z^l)$ ;

    end for

    for  $l=1$  to  $L$

      更新第  $l$  层的  $\mathbf{W}^{-l}, \mathbf{b}^l$ :

$$\mathbf{W}^l = \mathbf{W}^l - \alpha \sum_{i=1}^m \delta^{i,l} (\mathbf{a}^{i,l-1})^\top;$$

$$\mathbf{b}^l = \mathbf{b}^l - \alpha \sum_{i=1}^m \delta^{i,l};$$

    end for

  end for

end for

/\* 检查  $\mathbf{W}, \mathbf{b}$  的变化值, 如果小于停止迭代的

阈值, 就跳出循环输出各层的  $\mathbf{W}, \mathbf{b}$  结束 \*/  
针对池化层, 本文采用的是最大池化 (max

pooling) 的方法, 设置大小为  $2 \times 2$  的池化窗口来减少模型特征提取方面的误差, 这一误差主要来自 2 个方面:1) 邻域大小受限造成的估计值方差增大;2) 卷积层参数误差造成的估计均值偏移.

结合上文给出的卷积层正向传播函数可以得到池化层输出的反向传播函数:

$$\delta_k^{l-1} = \frac{\partial J(\mathbf{W}, \mathbf{b})}{\partial a_k^{l-1}} \frac{\partial a_k^{l-1}}{\partial z_k^{l-1}} = upsample(\delta_k^l) \odot \sigma'(z^{l-1}),$$

其中,  $upsample$  操作为最大池化的过程.

CNN 神经网络中的所有权重都用随机函数初始化, 即高斯函数和 Xavier 函数. 利用训练集不断优化权重设置, 直到获得 CNN 神经网络合理的权重矩阵.

训练阶段, FSR 算法把训练集输入上述 CNN 网络中, 不断地进行迭代计算来自动地调整模型参数. 同时, 在训练中采用了 10 重交叉验证 (10-fold cross-validation) 的方式, 这样可以有效地避免陷入局部优化问题.

在上述神经网络设计的基础上, FSR 算法采用如图 3 所示的结构, 利用历史流量数据进行 CNN 神经网络的训练, 完成训练后, FSR 算法通过获取实时数据进行流量特征的抓取并输出供优化的备选路由调整方案, 通过抖动抑制机制, 最终选取路由抖动最小的调整方案, 进行下一时间片的路由更新.

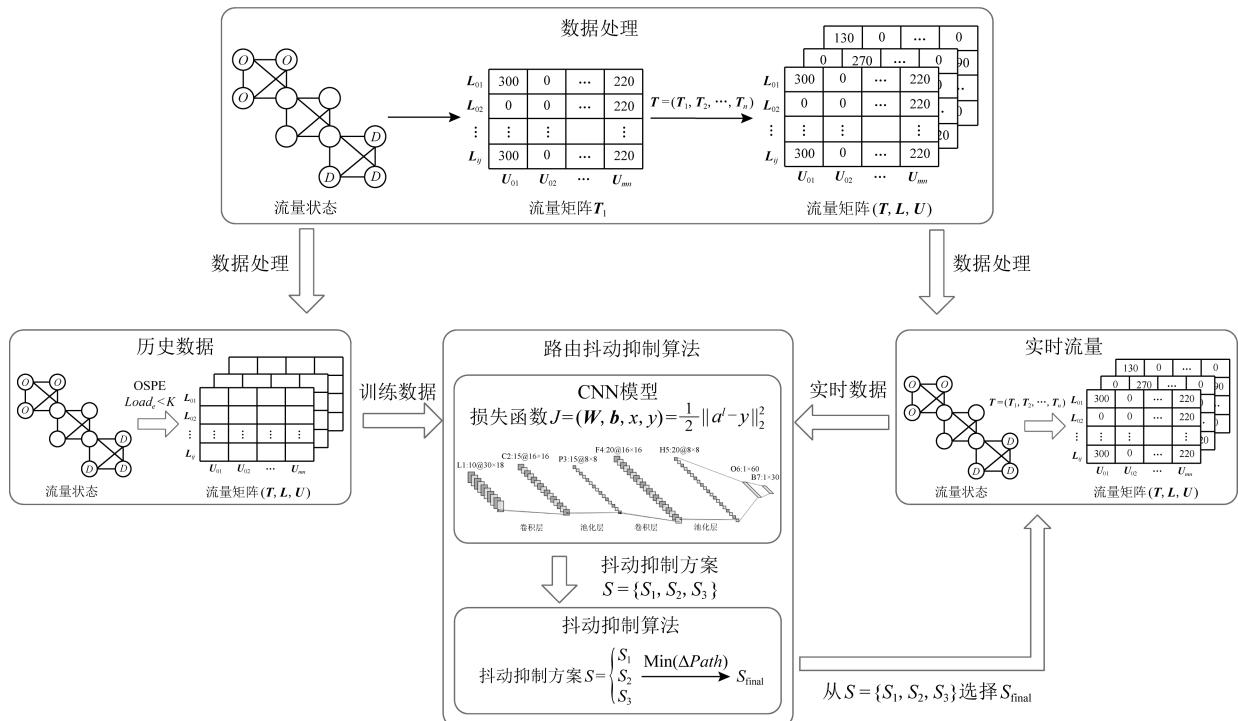


Fig. 3 Architecture of FSR algorithm

图 3 FSR 算法架构

## 2.4 抖动抑制

CNN 神经网络会给出下一时间片内可能的流量优化分布方案,但根据第 1 节讨论过的,这样的流量分布方案可能带来普遍的路由抖动、传输中断、数据重传等问题,进而导致网络整体传输性能下降.于是我们需要在待选方案中,选择尽可能减小网络振动的方案,可能的方法有:

1) 最小化路径变动.这是设计上最为简单的方法,我们将主观抹去所有传输路径本身的差异性,仅从路径变动的角度去衡量不同方案带来的网络振动.例如原有传输路径为  $A-B-C-D$ ,下一时间片给出的方案中,一项为  $A-E-C-D$ ,另一项为  $A-E-F-D$ ,那么我们将选择前者,因为它相对于现有传输路径,仅改变了一跳,带来的可能的网络抖动是最小的.这样的处理好处是方法简单、计算开销小、便于大规模部署,不足之处则是忽视了不同路径本身的重要程度.

2) 最小化流量变动.这是一个更复杂但更合理的方法.若使用这一方法,则我们不再单纯关注相邻时间片网络传输路径的变化跳数,而是要考虑被调整的路径所对应的总体流量大小.这是由于一条传输路径在相邻时间片内可能存在较大的路径调整,但是该路径上仅有 10 MB 流量,我们应该认为这对网络性能的影响非常有限.如果一条路径本身调整幅度不大,但该路径上承载了 10 GB 流量,那么我们有理由相信这对网络的影响远比前者更大.

3) 加权的流量变动最小化.在 1)2)方法的基础上,如果我们同时考虑流量的大小和优先级,那么我们可以得到一个更加合理但复杂度显著提升的抖动抑制方法.这需要我们更好地权衡算法有效性与算法开销间的平衡,甚至设计专门的机器学习机制来处理算法的复杂性.

于是,综合 1)~3) 路由抖动的评价思想,我们将路由抖动定义为

$$J(X) = \sum_{i=1}^N a_i X_i,$$

其中,  $X_i$  表示链路改变数量、改变链路中的负载、改变路径上的时延等可用的输入参数,  $a_i$  表示  $X_i$  所对应的权重系数.

本文我们将按照最小化路径变动中的思想,定义备选的路径方案相比上一时刻所最终决定的路径方案中每个 OD 对改变的链路数量分别为  $m_1, m_2, \dots, m_K$ , 其中  $K$  为 OD 对数量.于是,按照最小化路径变动的思想可得到:

$$J(X) = \sum_{i=1}^N a_i X_i = \sum_{i=1}^K m_i.$$

在本文中,FSR 算法采用第 1 种最小化路径变动机制来抑制路由抖动,如算法 2 所示.

### 算法 2. 抑制振荡算法.

输入:OD 对数量  $K$ 、每个 OD 对改变的链路数量  $m_1, m_2, \dots, m_K$ ;

输出:改变链路数量最少的路径  $J_{\min}$ .

set  $a_i = a_m = 1$ ;

for  $i = 1$  to  $k$

$$J = \sum_{i=1}^N a_i X_i = \sum_{i=1}^K m_i;$$

for  $s = 1$  to 3

$$J_{\min} = \text{Min}(J_1, J_2, J_3);$$

end for

end for

/\* 从备选的路径方案中选出抖动量最小值输出 \*/

具体来说,在运行阶段以 5 min 为一个时间间隔,在每个时间间隔结束时,把当前网络的流量情况整理为网络流量特征矩阵( $\mathbf{T}, \mathbf{L}, \mathbf{U}$ )输入 CNN 神经网络中.CNN 神经网络此时做为拥塞控制模型最终将提供可行路径方案的概率集合  $Output_{\text{CNN}}$ , 将此集合做为网络振动抑制算法的输入参数, 最终 FSR 算法可以给出当前时刻的最佳路由方案.

## 2.5 复杂度分析

我们对 FSR 算法的复杂度进行估算分析.从整体上来看,本文提出的 FSR 算法是由 CNN 神经网络以及抑制抖动算法所组成, 我们首先来讨论神经网络的复杂度.CNN 神经网络的复杂度可以直接按照表达式  $O(M^2 \times K^2 \times C_{\text{in}} \times C_{\text{out}})$  获得, 其中  $M$  为输出特征图(feature map)的尺寸,  $K$  为卷积核(kernel)的尺寸,  $C_{\text{in}}$  为输入通道数,  $C_{\text{out}}$  为输出通道数. 输出特征图尺寸又由输入尺寸  $X$ 、卷积核尺寸  $K$ 、填充  $Padding$ 、步长  $Stride$  这 4 个参数决定, 具体可表示为  $M = (X - K + 2 \times Padding) / Stride + 1$ . 对于不同拓扑的输入尺寸的增大, 时间复杂度也会随之升高. 另一方面, 本文提出的抑制振荡算法为最小值的排序算法, 抖动抑制算法时间复杂度为  $O(n)$ , 故整体算法的复杂度取决于神经网络的复杂度.

## 3 实验分析

我们设计了 4 组实验分别对 FSR 算法与对照组算法 RDL、算法 LRD 进行了综合的性能对比和

实验结果分析.实验内容包括拥塞率、吞吐量、传输路径的平均跳数和算法运行时间.

### 3.1 实验设计

在本实验中,我们期望 FSR 算法训练得到的神经网络,能够对网络流量的变化产生恰当的反应,并指导网络中的路由选择以较低的变动成本做出有效响应,使得全网流量合理分摊,所有链路均得到高效利用.

为了验证算法有效性,我们在 NS3 仿真平台上设计了 4 组实验,使用的仿真设备硬件配置为: CPU 为 I7-8700K, GPU 为 NVIDIA TESLA T4, 内存为 64 GB, 分别从网络拥塞率、网络最大吞吐量、网络的平均路径长度和运行时间这 4 个方面对算法性能进行了综合的对比与分析评价.另外,由于路由

选择与网络拓扑具有相关性,为了验证算法在不同网络拓扑下的性能,构造了 4 种拓扑结构同时进行实验,如图 4 所示.

关于本实验中 4 种不同的拓扑设计,其目的是为了验证在不同节点和可选路由规模的情况下,FSR 算法对于网络流量优化及路由抖动抑制的综合效果.FSR 算法和当前智能路由算法能够优化网络流量分布的前提,是网络中任意 2 个节点间存在足够数量的可选路由路径.所以对于结构过于简单、节点间连接不够丰富的拓扑,路由调整空间受限,实验无法分析出智能路由算法与传统路由算法性能上的差别.因此,本实验中,我们按照节点数量逐渐增加,可选路由数量逐渐增加的思路,设计了如图 5 所示的 4 种拓扑结构.

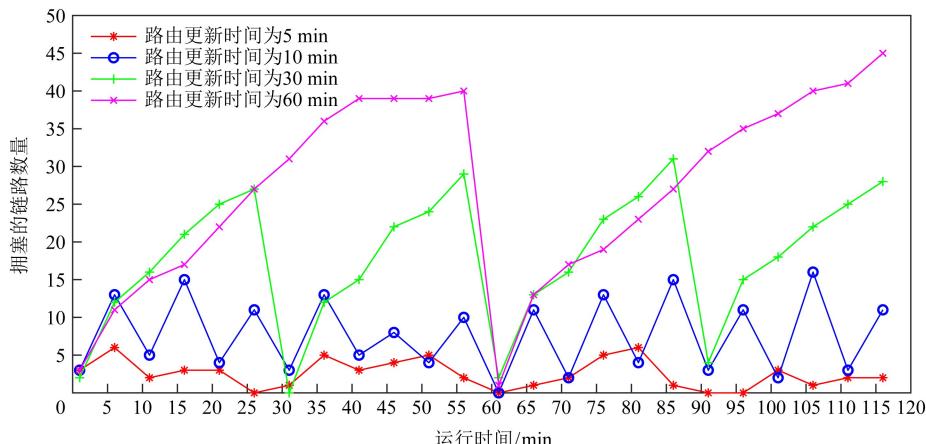


Fig. 4 Number of congested links in different time slices

图 4 不同时间片下拥塞链路数量

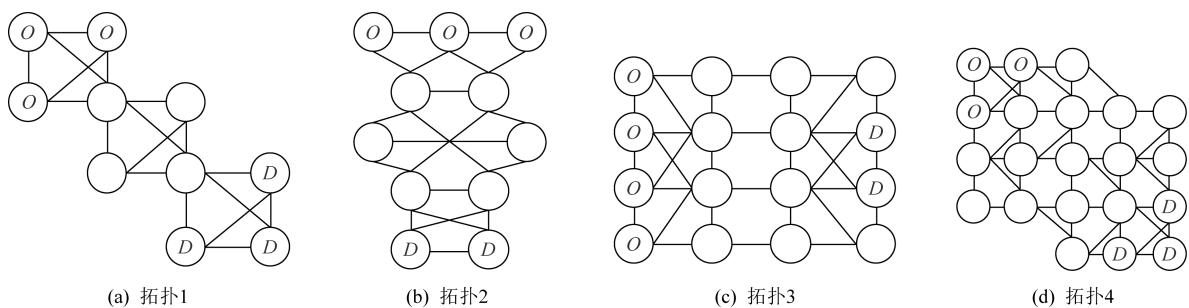


Fig. 5 Experimental topology

图 5 实验拓扑

本节实验中使用到的网络流量来自于 MAWI Working Group Traffic Archive 所提供的 WIDE 到上游 ISP 的传输链路上的跟踪流量,时间跨度为 2009-03-30—2009-04-06.同时,为了横向对比并验证 FSR 算法的性能,我们选取了 RDL, LRD 两种算

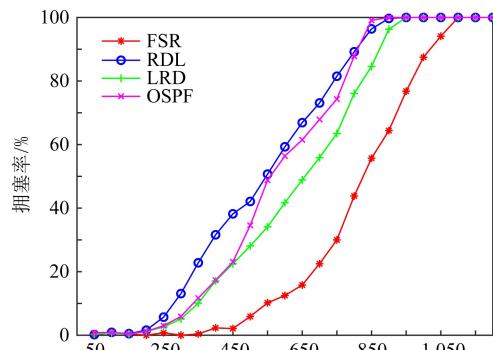
法.其中,LRD 算法在流量工程中将流量矩阵作为深度学习的特征输入,在最大化链路利用率的问题上给出了很好的解决方案.RDL 算法则将 CNN 神经网络应用在流量工程的问题上,并以流量需求矩阵做为特征输入,能适应突发性流量.

### 3.2 时间片选择

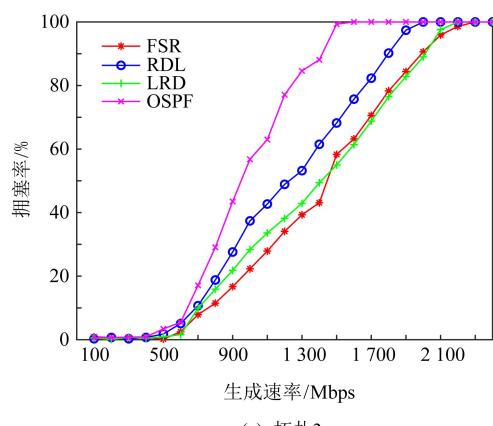
本文我们将路由选择按照时间片进行周期更新,时间片在本实验中具体设定为 5 min。关于时间片的取值,其实质是网络资源充分利用与路由稳定性间的平衡问题。我们分别以 5 min, 10 min, 30 min, 60 min 作为路由重构的时间片选择,使用拓扑 4 的网络环境,以 2 000 Mbps 生成速率在 120 min 内统计拥塞链路的数量,实验结果如图 4 所示。可以看出,时间片间隔越长,路由稳定性越好,对于网络连接的影响越小,但是另一方面,时间片间隔越长,由于路由灵活性降低,网络中出现链路利用率不均的情况越来越严重,造成网络中链路拥塞的数量越来越大。所以得出时间片的选取将是针对不同网络环境进行资源利用率和路由稳定性间的权衡取舍,在本实验中,5 min 的时间间隔是我们在实验中挑选的一个实践数值,该数值在我们设置的网络环境中能够较好地平衡网络流量分配和传输稳定性,并证明本文提出方案的有效性。

### 3.3 实验结果分析

本节将详细讨论每一组实验的具体设计和实验



(a) 拓扑1



(c) 拓扑3

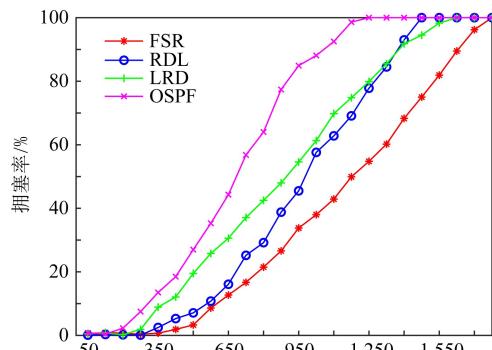
结果,并对实验结果进行详细分析和讨论。实验结果显示 FSR 算法在得到的网络性能上全方面优于对照算法组,证明 FSR 算法有效抑制了路由重构时带来的网络性能下降,与对照算法相比提升约 30% 的网络吞吐量,同时显著降低路径长度和拥塞概率。

#### 3.3.1 拥塞率

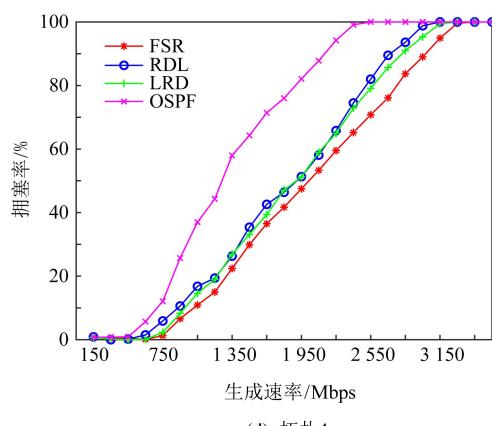
本实验考察各算法在网络流量不断增加的情况下,能否利用网络现有带宽资源,保障网络通信效率,尽可能减少网络拥塞。

在图 5 所述的 4 个网络拓扑中,分别测试 FSR, RDL 以及 LRD 算法在网络流量不断增加的情况下网络拥塞出现的概率。其中网络流量从 100 Mbps 一直增长到最大 3 600 Mbps, 直至所有链路均完全拥塞,即拥塞率达到 100% 为止。在这一过程中,每增加 100 Mbps 流量分析一次网络拥塞率。在本实验中,我们将网络中某一条链路的利用率大于某个阈值定义为该链路出现了拥塞,统计所有链路中处于拥塞状态的链路占比,作为网络当前的拥塞率。我们得到了如组图 6 所示的实验结果。

FSR 在每一种拓扑中,均表现出了最好的网络



(b) 拓扑2



(d) 拓扑4

Fig. 6 Congestion rate experiment

图 6 拥塞率实验

拥塞抑制能力,说明 FSR 在不同网络状况下,均具备较为良好的网络资源利用能力。具体来看,在拓扑 1 中,FSR 与对照组算法的差异最为明显,FSR 的优势最大。

以全网 10% 拥塞率为例,对照组算法中,RDL 算法在拓扑 1 中,当源端发出流量达到 280 Mbps 左右时,即出现 10% 的全网拥塞。LRD 算法则在相同情况下,当源端流量达到 350 Mbps 时,出现 10% 拥塞。OSPF 算法与 LRD 算法相当,相对比来看,FSR 算法可以坚持到源端流量达到 550 Mbps 时,才出现全网 10% 的拥塞。相比 RDL 和 LRD,FSR 算法分别提升 96% 与 57%。再以全网链路全部拥塞出现时的源端流量来看,RDL 算法与 LRD 算法非常接近,均在源端吞吐达到 950 Mbps 时,全网出现完全拥塞,

OSPF 算法则在约 860 Mbps 时出现完全拥塞。FSR 算法则可以坚持到源端吞吐达到 1 100 Mbps 时才出现全网拥塞。与对照组算法相比,在相同的网络环境下,最大可承受的源端吞吐量提升了 16%~28%。由此可见,FSR 算法可形成更加合理的路由选择,并更加充分利用全网网络资源,在相同的网络条件下,使得网络可以承担更大的流量压力。

### 3.3.2 网络吞吐

本实验中,网络源端的流量生成速率从 50 Mbps 开始持续增长,一直提升至网络整体吞吐量不再发生变化,得到的结果如图 7 所示。

从图 7 可以看出,随着网络拓扑逐渐复杂,网络节点逐渐增多,4 种算法的整体吞吐量都呈现增长趋势。

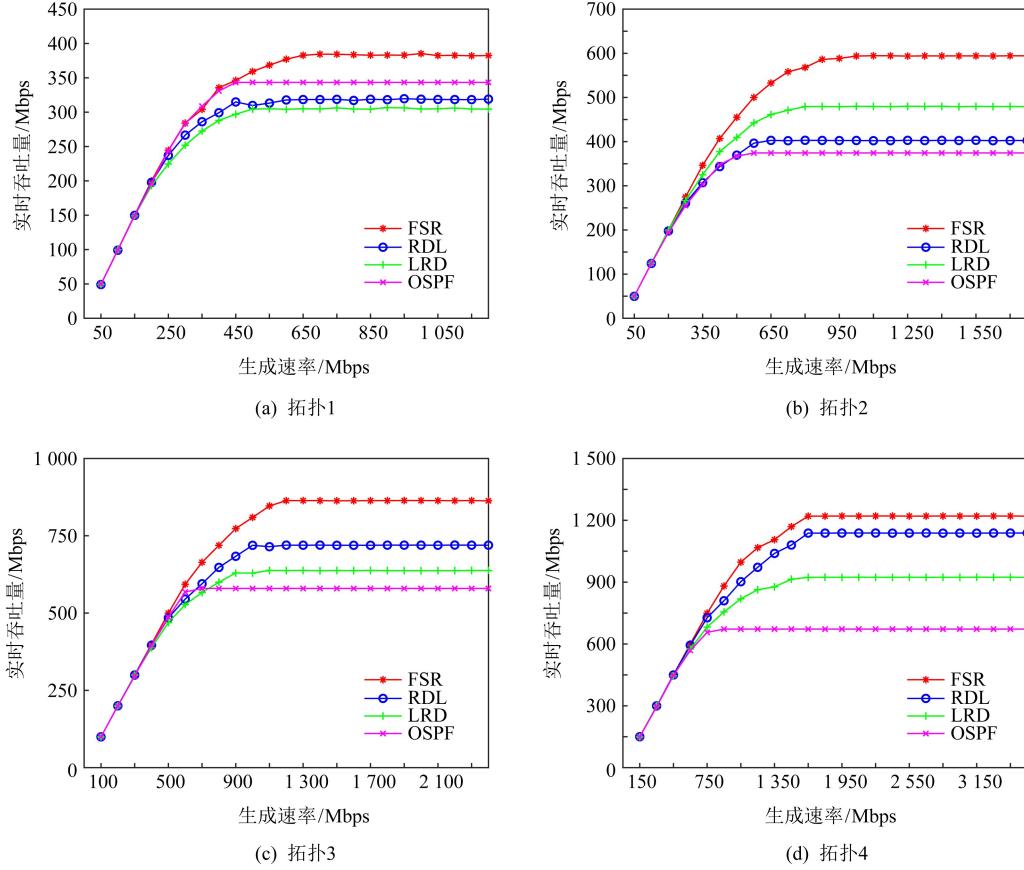


Fig. 7 Network throughput experiment

图 7 网络吞吐实验

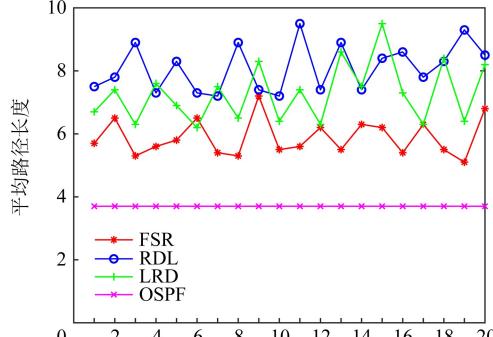
另一方面,从整体上来看,FSR 算法在相同网络拓扑结构下,相比对照组算法,能够达到更大的网络吞吐量。其中,拓扑 1 网络环境下,4 种算法差距最小,RDL 和 LRD 算法几乎相同,OSPF 略好于前两者,LRD 算法最终达到约 305 Mbps,RDL 算法最终

达到约 320 Mbps,略微优于 LRD 算法,OSPF 算法达到约 345 Mbps。FSR 算法最终可以达到 385 Mbps,相比对照组算法,分别提升约 20%,26% 和 12%。拓扑 3 中,FSR 算法取得了最大的吞吐量优势。对照组算法中,OSPF 算法达到约 580 Mbps,LRD 算法最终

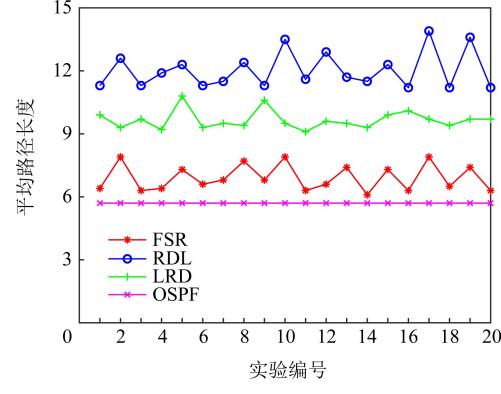
取得约 640 Mbps 的最大吞吐量, RDL 算法则可达到 720 Mbps. 与此相对比的是, FSR 算法可以达到最大 865 Mbps 的最大吞吐量, 相比对照组算法, 分别提升约 49%, 20% 和 35%.

### 3.3.3 平均路径长度

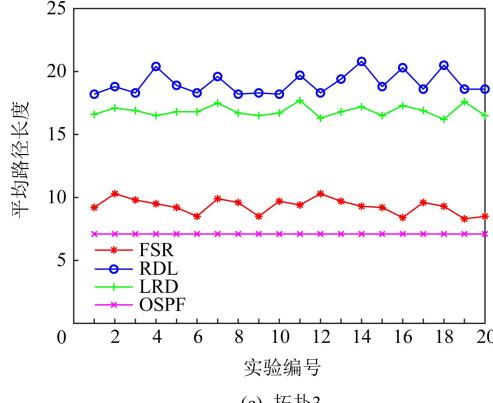
为了在网络资源受限或者流量分布不均的情况下, 充分利用网络资源提高全网吞吐能力, 路由路径的调整则必然会存在“舍近求远”的情况, 从而达到



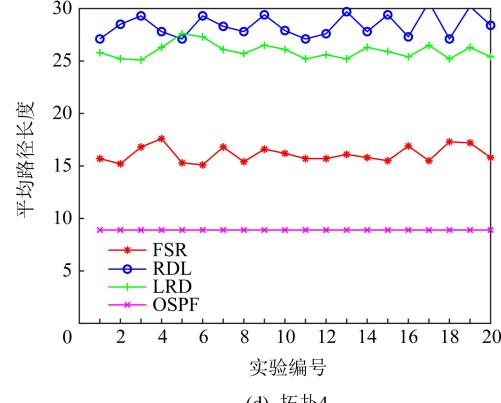
(a) 拓扑1



(b) 拓扑2



(c) 拓扑3



(d) 拓扑4

Fig. 8 Average path length experiment

图 8 平均路径长度实验

从实验结果可以看出, 随着网络结构更加复杂、网络规模不断增大, 除 OSPF 算法路径不变外, 其他各算法的平均路径长度都呈现增长趋势, 但除去路径不变的 OSPF 算法, FSR 在 4 种拓扑情况下平均路径均为最短. 其中, 拓扑 1 环境下, 3 种智能路由算法的平均路径长度差异最小, RDL 算法在经过 20 次更新周期后, 平均转发路径长度为 8.09, LRD 算法为 7.29, 而 FSR 算法的平均路径长度为 5.89. 在拓扑 3 和拓扑 4 中, FSR 的平均路径长度均大幅度低于对照算法组中的智能路由算法. 以拓扑 4 为例, RDL 为 28.34, LRD 为 25.94, 同情况下 FSR 则为

16.11, 平均路径长度分别降低了 43% 和 38%, 但相比 OSPF 算法依然提升了约 79%.

综上所述, 通过对 4 种拓扑下 3 个算法的测量可以发现, FSR 给出的路由选择, 其平均路径长度相比对照算法组, 有明显的降低, 可以以更高的效率完成数据的转发操作.

### 3.3.4 算法运行时间

除了网络性能之外, 一个路由选择算法还需要考虑算法自身的效率问题, 效率低下的算法可能导致路由计算开销较大, 从而影响算法可用性, 为了考察 FSR 算法的算法效率, 我们设计了一组实验, 在

网络负载的均衡.

然而, 一个恰当的平均路径长度也是一个优秀的路由选择算法需要考量的因素.

在 4 种网络拓扑中, 分别运行 OSPF, FSR, LRD 和 RDL 这 4 种算法, 同时, 利用实验设计阶段选取的网络流量, 并分别记录每个算法 20 次的网络路由调整结果, 计算全网源—目的对间的平均路径长度, 得到如图 8 所示的实验结果.

同样进行 20 次路由选择的条件下,记录 3 种智能路由选择算法的运行时间,并在 4 种不同的拓扑上分别进行该实验。

通过实验可以发现,在 4 种不同的拓扑下,算法运行时间略有差异,在 3 种算法的横向对比中,FSR 算法的总体运行时间最长。这主要是由于 FSR 算法与对比组算法相比,为了减少网络中的路由抖动而多了一个处理环节。这一设计一方面确实增强了网络整体运行效率,而另一方面也带来了一定的额外时间成本。

具体来说,FSR 算法在 20 次路由调整过程中,最短用时 3.35 s,最长用时 4.38 s,平均用时 3.73 s。而与之对比的是,对照组算法中,RDL 算法最短用时 2.28 s,最长用时 3.3 s,平均用时 2.69 s,LRD 算法最短用时 2.83 s,最长用时 3.68 s,平均用时为 3.17 s。FSR 算法运行时间确实长于对照组的 2 种算法,平均运行时间方面,FSR 相比 RDL 要多消耗 1.04 s,而相比 LRD 算法要多消耗 0.56 s。

通过上述实验可以看出,FSR 算法由于额外的抖动抑制操作,在运行时间上比对照组算法多出了额外的时间开销。但考虑到 FSR 算法在实际吞吐率测试中带来约 30% 的带宽提升,不难发现即便存在算法运行速度上的劣势,但 FSR 算法由于抑制了网络振动,仍旧可以实现比现有机制更好的网络传输效率。

## 4 相关工作

流量工程和路由策略是互联网技术的基础并受到了学术界和工业界的共同研究。这些研究广泛涉及了互联网的各种应用形式,包括传统网络<sup>[7]</sup>、数据中心网络<sup>[8]</sup>和核心骨干网<sup>[9]</sup>。相关研究的核心目标均是为了满足在复杂多变的应用环境中尽可能地优化满足灵活的服务质量要求,而这个问题的求解则建立在数学模型的建立和求解基础上。然而,正如引言所述,网络规模的庞大和流量需求的多变为这一类问题的数学建模工作提出了巨大的挑战,通常的做法是针对具体应用场景进行必要的简化假设,并以此使得数学模型可以有效求解。然而现实网络环境往往是多种应用需求、流量特性和网络结构的复杂叠加,简化假设在实际场景中往往难以满足。针对上述问题,目前还难以找到一个有效的通用模型,并在优化效果上达到实际使用的需求,因此,传统基于数学模型的网络管理机制越来越难以满足快速发展的网络应用的需求。

另一方面,随着计算能力的提高和数据的爆炸式增长,人工智能得到了极大的发展,强大的计算能力可以模仿更深的神经网络(DNN),而大数据可以提供足够的训练样本。其中人工神经网络和深度学习技术取得了最大的发展,其通过构建更多层次的深度神经网络,实现学习和识别抽象模式,并已广泛应用于图像分类、物体识别、通信以及其他各个相关领域<sup>[10-11]</sup>。

机器学习技术已被广泛应用于解决各类网络管理问题,包括拥塞控制<sup>[12-13]</sup>、资源分配<sup>[14]</sup>和视频流的比特率选择<sup>[15]</sup>等。其中,对于路由选择问题,早期 Q 路由<sup>[16]</sup>首次将 Q 学习<sup>[17]</sup>应用于网络路由上下文。与传统基于最短路径的路由策略相比,Q 路由通过历史规律的学习,可以更加有效地避免网络拥塞并降低传输时延。在 Q 路由中,每个路由器分别学习从数据包头到输出端口的映射。这涉及路由器以每个数据包分辨率不断交换有关其针对不同目的地的延迟信息。我们认为,在每个数据包级别上以分散方式进行操作,在可伸缩性和通信开销方面提出了重大挑战。

针对这些问题,近年来也取得了一定的研究成果<sup>[18-19]</sup>。其中,LRD 算法通过强化学习的方式来进行智能的路由选择,将经典的最小化最大链路利用率作为研究的目标,希望通过历史网络流量来驱动路由的选择。RDL 采用了深度卷积神经网络(CNN)来控制网络流量,利用当前请求发送的流量数据来合理地规划路由的路径选择,通过利用在线的方式不断地迭代 CNN 神经网络,以此来得到更具适应力、平均时延和丢包率更低的模型。然而,正如本文引言讨论的,当前智能路由算法对路由更新时的振动抑制还存在较大优化空间。

## 5 总 结

随着网络流量的快速增长以及网络应用模式的不断更新,路由机制及网络设备的转发能力面临着更为严峻的考验,传统的、相对固化的路由算法已经很难满足当前不断增长和变化的网络需求。

动态灵活的路由调整机制,以及基于历史数据学习网络流量和路由选择规律的能力,是智能路由算法核心优势所在。本文聚焦于当前智能路由算法在每个路由更新周期所带来的大范围的路由抖动以及由此引发的网络转发效率降低的问题,提出了一种

路由抖动敏感的智能路由选择算法,核心思想是在追求全网链路负载均匀分配、充分利用全网转发资源的同时,寻求与当前路由选择变动最小的更新方案,使得每个更新周期带来的路由抖动尽快收敛,提升网络整体转发效率.FSR 算法利用 CNN 生成当前网络拓扑环境的路由方案可行解,并从可行解中寻找路由抖动最小的最优解.实验表明:FSR 算法与对照算法相比提升约 30% 的网络吞吐量,同时显著降低网络丢包和拥塞概率.

未来,我们将会结合具体的网络应用场景,进一步研究更有针对性的其他路由抖动收敛机制,同时重点研究算法时间开销的优化方法,提升 FSR 算法的执行效率.

**作者贡献声明:** 邵天竺,负责实验方案设计与仿真、数据分析,参与方案设计、论文编写;王晓亮(通信作者),负责方案设计、论文编写,参与实验方案设计;陈文龙,参与方案设计;唐晓岚,参与实验方案设计;徐敏,参与方案设计、数据分析.

## 参 考 文 献

- [1] Chen Shanzhi, Xu Hui, Liu Dake, et al. A vision of IoT: Applications, challenges, and opportunities with China perspective [J]. Internet of Things Journal IEEE, 2014, 1(4): 349–359
- [2] Boccardi F, W. Heath R, Lozano A, et al. Five disruptive technology directions for 5G [J]. IEEE Communications Magazine, 2014, 52(2): 74–80
- [3] Andrews J G, Buzzi S, Choi W, et al. What will 5G be? [J]. IEEE Journal on Selected Areas in Communications, 2014, 32(6): 1065–1082
- [4] Goh H, Thome N, Cord M, et al. Top-down regularization of deep belief networks [C] //Proc of Advances in Neural Information Processing Systems 26. Cambridge, MA: MIT Press, 2013: 1878–1886
- [5] Valadarsky A, Schapira M, Shahaf D, et al. Learning to route [C] //Proc of the 16th ACM Workshop on Hot Topics in Networks. New York: ACM, 2017: 185–191
- [6] Tang Fengxiao, Mao Bomin, Fadlullah Z, et al. On removing routing protocol from future wireless networks: A real-time deep learning approach for intelligent traffic control [J]. IEEE Wireless Communications, 2018, PP(99): 1–7
- [7] Chiesa M, Rétvári G, Schapira M. Lying your way to better traffic engineering [C] //Proc of the 12th Int Conf on Emerging Networking Experiments and Technologies. New York: ACM, 2016: 391–398
- [8] Al-Fares M, Radhakrishnan S, Raghavan B, et al. Hedera: Dynamic flow scheduling for data center networks [C] //Proc of the 7th USENIX Symp on Networked Systems Design and Implementation. New York: ACM, 2010, 10(8): 89–92
- [9] Kandula S, Katabi D, Davie B, et al. Walking the tightrope: Responsive yet stable traffic engineering [J]. ACM SIGCOMM Computer Communication Review, 2005, 35(4): 253–264
- [10] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks [J]. Communications of the ACM, 2017, 60(6): 84–90
- [11] O’shea T, Hoydis J. An introduction to deep learning for the physical layer [J]. IEEE Transactions on Cognitive Communications and Networking, 2017, 3(4): 563–575
- [12] Dong Mo, Li Qingxi, Zarchy D, et al. Re-architecting congestion control for consistent high performance [C] //Proc of the 12th Symp on Networked Systems Design and Implementation. New York: ACM, 2015: 395–408
- [13] Winstein K, Balakrishnan H. TCP ex Machina: Computer-generated congestion control [J]. Computer Communication Review, 2013, 43(4): 123–134
- [14] Mao Hongzi, Alizadeh M, Menache I, et al. Resource management with deep reinforcement learning [C] //Proc of the 15th ACM Workshop on Hot Topics in Networks. New York: ACM, 2016: 50–56
- [15] Mao Hongzi, Netravali R, Alizadeh M, et al. Neural adaptive video streaming with pensieve [C] //Proc of the Conf of the ACM Special Interest Group on Data Communication. New York: ACM, 2017: 197–210
- [16] Majer M, Bobda C, Ahmadinia A, et al. Packet routing in dynamically changing networks on chip [C] //Proc of the 19th IEEE Int Parallel and Distributed Processing Symp. Piscataway, NJ: IEEE, 2005. DOI:10.1109/IPDPS.2005.323
- [17] Watkins C J C H, Dayan P. Q-learning [J]. Machine Learning, 1992, 8: 279–292
- [18] Mao Bomin, Fadlullah Z, Tang Fengxiao, et al. Routing or Computing? The paradigm shift towards intelligent computer network packet transmission based on deep learning [J]. IEEE Transactions on Computers, 2017, 66(11): 1946–1960
- [19] Fadlullah Z, Tang Fengxiao, Mao Bomin, et al. State-of-the-art deep learning: Evolving machine intelligence toward tomorrow’s intelligent network traffic control systems [J]. IEEE Communications Surveys & Tutorials, 2017, 19(4): 2432–2455



**Shao Tianzhu**, born in 1995. Master candidate. His main research interests include two-dimensional routing and forwarding and artificial intelligence routing.

**邵天竺**,1995 年生,硕士研究生,主要研究方向为二维路由转发和人工智能路由。



**Wang Xiaoliang**, born in 1986. Received his PhD degree from the Department of Computer Science & Technology of Tsinghua University in 2017. Lecturer in the College of Information Engineering at Capital Normal University. His main research interests include network architecture and network security.

王晓亮,1986 年生.2017 年从清华大学计算机科学与技术系获得博士学位.目前在首都师范大学信息工程学院担任讲师.主要研究方向为网络体系结构和网络安全.

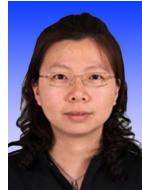


**Chen Wenlong**, born in 1976. PhD, professor. His main research interests include Internet architecture, high performance router and network protocol.

陈文龙,1976 年生.博士,教授.主要研究方向为互联网体系结构、高性能路由系统和网络协议.



**Tang Xiaolan**, born in 1987. PhD, associate professor. Member of CCF and IEEE. Her main research interests include Internet of vehicles, Internet architecture, urban computing. 唐晓岚,1987 年生.博士,副教授.CCF 和 IEEE 会员.主要研究方向为车联网、互联网体系结构、城市计算.



**Xu Min**, born in 1978. PhD, associate professor. Member of IEEE and CCF. Her main research interests include biometrics recognition, computer vision, and intelligent educational technology.

徐敏,1978 年生.博士,副教授.IEEE 和 CCF 会员.主要研究方向为生物特征识别、计算机视觉和智能教育技术.

## 《计算机研究与发展》征订启事

《计算机研究与发展》(Journal of Computer Research and Development)是中国科学院计算技术研究所和中国计算机学会联合主办、科学出版社出版的学术性刊物,中国计算机学会会刊.主要刊登计算机科学技术领域高水平的学术论文、最新科研成果和重大应用成果.读者对象为从事计算机研究与开发的研究人员、工程技术人员、各大专院校计算机相关专业的师生以及高新企业研发人员等.

《计算机研究与发展》于 1958 年创刊,是我国第一个计算机刊物,现已成为我国计算机领域权威性的学术期刊之一.并历次被评为我国计算机类核心期刊,多次被评为“中国百种杰出学术期刊”.此外,还被《中国学术期刊文摘》、《中国科学引文索引》、“中国科学引文数据库”、“中国科技论文统计源数据库”、美国工程索引(Ei)检索系统、日本《科学技术文献速报》、俄罗斯《文摘杂志》、英国《科学文摘》(SA)等国内外重要检索机构收录.

国内邮发代号:2-654;国外发行代号:M603

国内统一连续出版物号:CN11-1777/TP

国际标准连续出版物号:ISSN1000-1239

### 联系方式:

100190 北京中关村科学院南路 6 号《计算机研究与发展》编辑部

电话: +86(10)62620696(兼传真); +86(10)62600350

Email:crad@ict.ac.cn

<http://crad.ict.ac.cn>