

基于时延的软件定义卫星网络控制域规划策略

郭子桢, 梁俊, 肖楠, 陈威龙, 陈金涛, 谢宝华

(空军工程大学信息与导航学院, 西安, 710077)

摘要 现有软件定义网络(SDN)多控制器划域研究大多基于拓扑相对固定的地面网络, 难以适应拓扑动态变化的卫星网络, 而现有卫星网络研究在处理时延问题时大多只考虑传播时延。针对这一问题, 在建立模型时引入控制器处理时延、排队时延对划域的影响, 提出了一种面向软件定义卫星网络(SDSN)的时延分析模型。借助该模型进行SDSN多控制域的划分。经过仿真验证, 与现有的地面SDN多控制域规划算法及卫星网络控制域划分方法相比, 该算法能够更好地适应卫星网络环境, 有效降低网络时延, 使控制器间负载均衡性能提升40%以上。同时迭代跳出机制使得算法运行速度快, 能够适应卫星网络拓扑的快速变化。

关键词 软件定义卫星网络; 多控制域规划; 负载均衡; 时延; 模拟退火算法

DOI 10.3969/j.issn.1009-3516.2020.06.006

中图分类号 TN92 文献标志码 A 文章编号 1009-3516(2020)06-0033-06

Multi-Domain Balanced Partition Based on Delay Optimization in Software Defined Satellite Networking

GUO Zizhen, LIANG Jun, XIAO Nan, CHEN Weilong, CHEN Jintao, XIE Baohua

(Information and Navigation College, Air Force Engineering University, Xi'an 710077, China)

Abstract Aimed at the problems that existing software defined network (SDN) multi-controller domain partitioning algorithms are mostly based on the ground networks with fixed topology, difficult to adapt to the satellite network with dynamic topology, and most of the existing satellite network researches are only in consideration of the propagation delay, a software defined satellite network (SDSN) delay analysis model is proposed in view of processing delay and queuing delay. On the basis of this model, satellite network multi-control domains are divided. The simulations prove that this algorithm can adapt to the satellite network environment better compared with the existing ground SDN multi-control domain planning algorithm and the satellite network control domain division method. Besides, the algorithm can effectively reduce the network delay, and increase the load balancing between the controllers by more than 40%. Moreover, the iteration stopping mechanism makes the algorithm run fast, and can adapt to the rapid changes of the satellite network topology.

Key words software defined satellite network; multi-domain partition; load balance; delay; simulated annealing algorithm (SAA)

收稿日期: 2020-06-08

基金项目: 国家自然科学基金(61871474); 国防科技项目基金(2201086)

作者简介: 郭子桢(1997—), 男, 河北邢台人, 硕士生, 主要从事空天信息网络研究。E-mail: g1034122134@163.com

通信作者: 梁俊(1962—), 男, 江苏江宁人, 教授, 博士生导师, 主要从事卫星通信研究。E-mail: liangjunhja@sina.com

引用格式: 郭子桢, 梁俊, 肖楠, 等. 基于时延的软件定义卫星网络控制域规划策略[J]. 空军工程大学学报(自然科学版), 2020, 21(6): 33-38. GUO Zizhen, LIANG Jun, XIAO Nan, et al. Multi-Domain Balanced Partition Based on Delay Optimization in Software Defined Satellite Networking[J]. Journal of Air Force Engineering University (Natural Science Edition), 2020, 21(6): 33-38.

将软件定义网络架构引入卫星网络,能够使数据平面的卫星仅需承担转发和硬件配置功能,就实现路由策略的灵活部署、网络配置简化及网络成本降低^[1]。随着卫星网络承载的业务越来越多,单个控制器显然无法满足网络的控制需求^[2],多控制器虽然能克服单控制器带来的单点失效和可扩展性差的缺点,却也带来了一个新问题:对于给定数量的控制器和交换机,如何合理规划交换机与控制器的配属关系。

现有 SDN 多域规划的研究已经较为成熟,能够为卫星网络相关研究提供一定的借鉴。WANG^[3]等提出基于 K-means 的控制器部署方法,通过对网络中节点进行聚类,在聚类中心处部署控制器并按照跳数最少原则划分控制域,这能够有效降低控制器通信时延。文献[4]提出的 K-Critical 算法通过预测机制和多分类集网络拓扑构建,可以有效实现系统的负载均衡。文献[5]提出了一种自适应多域划分算法(Adaptive Multi-Domain Approach, AM-DA),通过考虑时延和控制器负载,采用谱聚类方法划分控制域。文献[6]提出了一种基于双向匹配的控制域划分策略,同时从控制器和交换机 2 个角度考虑,以实现控制器间的负载均衡。然而这些算法大多适用于拓扑固定且网络尺度较小的地面网络,将其直接应用于卫星网络会导致网络性能不理想。

目前,已经有许多研究者开始关注 SDSN 网络控制器部署问题:杨力^[7]等提出一种基于双门限的控制器动态部署策略,通过定义控制器的负载状态以期实现控制器负载的动态调整,然而仿真实验却并未直接与卫星网络拓扑相结合。WU^[8]等提出一种低开销的网络扩展方法,以控制器与交换机的最短距离为优化指标。XU^[9]等借助 STK 工具对 SDSN 网络的时延进行了分析,将低轨卫星(Low Earth Orbit, LEO)按照轨道分组并从每组选出最靠近赤道且运动方向相反的 2 颗卫星部署控制器,共部署 12 个控制器,但这种部署方案要求星座中所有节点均安装控制器模块且动态调整其开闭状态,网络状态变化时的重规划开销过大。HU^[10]等设计了一种 SoftLEO 架构,将 LEO 卫星按照轨道分为 6 组,选择 6 个纬度相近的节点作为每组卫星的控制器部署位置以减小控制器间的通信开销,然而网络的可靠性和负载均衡性能不够理想。WU^[11]等设计了一种 APSO 算法全面的分析了控制器的状态同步开销、信息收集开销、流建立开销、网络重规划开销等并以此确定控制器部署位置。文献[12]通过线性规划优化流建立时延完成控制器部署及划域。文献[13]在部署控制器时将网络端到端时延和负载

均衡指数作为优化目标。这些研究大多仅考虑传播时延,忽略了排队时延与处理时延,同时卫星网络中节点转发、处理能力有限,不同的负载将带来排队时延与处理时延的巨大差异进而影响划域结果,简单地忽略这些影响或者认为所有控制器都有相同的排队时延与处理时延的做法显然是不合理的,这将导致网络性能的下降。

本文针对现有算法时延描述不够全面的问题,综合考虑排队时延、处理时延、传播时延构建了软件定义卫星网络时延分析模型,并设计了基于模拟退火的多控制域均衡划分算法(Multi-Domain Balanced Partition Algorithm Based on Simulated Annealing, MDBPSA),通过构建控制器匹配列表的方法寻找近似最优的划域结果,提升网络时延性能、实现网络负载均衡。同时由于卫星网络拓扑快速动态变化,时间片持续时间短,要求算法有较快的运算速度,本文特别设计了迭代跳出机制以加快模拟退火算法的求解速度,仿真结果表明改进后的算法求解时间远小于时间片持续时间且性能良好,能够满足卫星网络需求。

1 软件定义卫星网络架构

考虑到卫星地面站全球布站困难且成本高昂、同步轨道卫星(Geosynchronous Earth Orbit, GEO)处理转发能力无法适应 LEO 卫星数量和业务的快速增长、GEO 卫星对地通信时延较大等问题^[14-17],本文采用在地面网络控制/监测中心(Network Control Center/Network Management Center, NCC/NMC)部署主控制器,在 GEO 卫星上部署区域控制器,并从基于 OpenFlow 的 LEO 卫星中挑选从属控制器的多层次部署方案。该架构利用地面强大的计算能力和充足的存储空间控制整个网络,GEO 卫星与地面站保持稳定连接,并选取从控制器作为 GEO 的补充。

整个 SDSN 网络根据 GEO 卫星数量及其覆盖区域划分为若干控制域,每个控制域包含一个域控制器、多个从控制器以及许多 LEO 卫星。每个控制域又被划分为若干个控制子域,每个控制子域内由从控制器管理其所属的 LEO。运行过程中,GEO 卫星接收流建立请求,并通过分发控制信息改变其控制域内的 LEO 状态。NCC/NMC 访问 LEO 卫星,并通过 GEO 卫星管理整个网络,GEO 卫星与 NCC/NMC 间的连接实现了全球流建立,能够对每一个控制动作做出响应。从控制器通过星间链路(Inter Satellite Links, ISLs)收集状态信息或将控

制信息分发到所属子域内的 LEO 卫星。这种网络架构能极大地减少 GEO 卫星广播引起的干扰、控制器-交换机之间的信令延迟。下一步基于该架构对卫星网络中控制子域划分问题展开。

2 模型构建

SDSN 控制子域划分问题可以运用图论相关知识进行描述。SDSN 网络可以表示为 $G = (V, E)$, 其中 V 表示节点集合, E 表示节点间链路集合。假设网络中所有 LEO 卫星均可承担交换机功能, 其中的一部分部署了从控制器, LEO 卫星数量为 N , 从控制器数量为 M 。从控制器集合用 $C = \{C_1, C_2, \dots, C_M\}$ 表示, 其处理能力为 $\Omega = \{\Omega_1, \Omega_2, \dots, \Omega_M\}$, 对控制器设置相应冗余因子 $\beta_m \in (0, 1)$ 以预留部分处理能力预防过载^[13]。LEO 集合表示为 $S = \{S_1, S_2, \dots, S_N\}$ 。两节点间最短路径表示为 d_{mn} 。在卫星网络中网络流量可以表示为时间的函数, S_n 在时间 t 内的流请求速率可以表示为 $\lambda(t)_n$ 。矩阵 X 描述 SDSN 控制域中从控制器和 LEO 的控制关系, 其中元素取值见式(1)。

$$x(t)_{mn} = \begin{cases} 1, & \text{时间 } t \text{ 内 } C_m \text{ 控制 } S_n \\ 0, & \text{其他} \end{cases} \quad (1)$$

每个 LEO 仅与一个从控制器建立控制关系。因此整个控制域可划分为 M 个控制子域: Domain $m_1 \sim$ Domain m_M 。

现有 OpenFlow 交换机均为多核交换机, 且 OpenDaylight、Ryu 等控制器均采用多线程处理方式, 因此, 可将 LEO 与从控制器间的通信过程近似为 $G/M/k$ 排队模型, 即消息到达为一般过程、控制器对于流请求的处理为马尔科夫过程、从控制器线程数为 k 。在此基础上, 可计算相关参数。

2.1 LEO 到从控制器时延

所有与 C_m 相连的 LEO 的流请求速率之和即为 C_m 在时间 t 内需要处理的流请求, 记为 $L(t)_m$, 见式(2), 其中各 LEO 交换机之间的流请求相互独立。

$$L(t)_m = \sum_{n=1}^N \lambda(t)_n x(t)_{mn} \quad (2)$$

由 Little 原理可得流请求排队时延 $Q(t)_m$, 见式(3)。

$$Q(t)_m = \sum_{n=1}^N \frac{1}{\Omega(t)_m - \lambda(t)_{nm}} \quad (3)$$

式中: $\Omega(t)_m$ 表示时间 t 内从控制器 C_m 的可用处理能力, 且 $0 < \Omega(t)_m < \Omega_m$; $\lambda(t)_{nm}$ 表示 S_n 在时间 t 内对 C_m 的流请求速率。

C_m 的平均处理时间为 $P(t)_m$, 满足:

$$P(t)_m = \sum_{n=1}^N \frac{\Omega(t)_m \Omega(t)_m}{\Omega(t)_m - t L(t)_m} \quad (4)$$

卫星网络覆盖范围大, 路径传播时延也是控制子域规划的重要指标。传播时延 $T(t)_m$ 满足式(5):

$$T(t)_m = \frac{\overline{d(t)_m}}{c} \quad (5)$$

式中: c 表示光速; $\overline{d(t)_m}$ 表示子控制域内 LEO 到从控制器的平均距离。

$$\overline{d(t)_m} = \frac{\sum_{n=1}^N x(t)_{nm} d_{nm}}{\sum_{n=1}^N x(t)_{nm}} \quad (6)$$

则控制子域内 LEO 到从控制器平均时延:

$$AT(t)_m = Q(t)_m + P(t)_m + T(t)_m \quad (7)$$

控制域平均时延为 $AT(t)$:

$$AT(t) = \frac{1}{\sum_{m=1}^M L(t)_m} \cdot \left(\sum_{m=1}^M (Q(t)_m + P(t)_m + T(t)_m) L(t)_m \right) \quad (8)$$

2.2 网络开销

SDN 架构下无法匹配流表的数据包需上交从控制器处理, 且从控制器需要进行周期性通信以更新控制逻辑, 保持控制域内网络视图。因此网络开销主要可分为转发开销和状态同步开销 2 个主要部分。转发开销 P_T 主要为从控制器处理 LEO 的 Packet-in 包并下发流表所带来的网络开销, 见式(9)。状态同步开销 P_S 见式(10)。

$$P_T = v_R \sum_{n \in N} \sum_{m \in M} \lambda(t)_n d_{nm} x(t)_{nm} \quad (9)$$

式中: v_R 为从控制器轮询 LEO 的平均速率。

$$P_S = v_S \sum_{n, m \in M} d_{nm} x(t)_{nm} \quad (10)$$

式中: v_S 为从控制器状态信息的平均传播速率。网络开销即为转发开销与状态同步开销之和。

$$TL(t) = P_T + P_S \quad (11)$$

2.3 目标函数

为了合理地配属从控制器和 LEO 之间的匹配关系, 尽可能减少从控制器和子域内 LEO 之间的平均时延且控制网络开销。因此, 得到如下目标函数:

目标函数及约束条件:

$$\min Object = [y AT(t) + (1-y) TL(t)] \quad (12)$$

$$\forall m, L(t)_m \leq \beta_m \cdot \Omega(t)_m \quad (13)$$

$$\forall n, \sum_{m=1}^M x(t)_{nm} = 1 \quad (14)$$

式中: $y \in (0, 1)$, 用于调节平均时延和网络开销对于控制域规划的影响。

上述约束条件中, 式(13)表示网络中没有控制

器过载的情况发生。式(14)表示所有 LEO 都只能连接到唯一一个从控制器。

3 算法设计

3.1 匹配机制设计

定义 1 匹配。即 LEO 与从控制器建立控制关系。例如 S_n 若受 C_m 控制, 则 LEO 与 C_m 完成匹配。记为 (C_m, S_n) 。

定义 2 从控制器匹配列表。从控制器将控制域内所有 LEO 按照 $d_{nm}\lambda(t)_n$ 升序排列, 并构造匹配列表 $\Gamma(C_m)=\{S_n, \dots\}$, 在从控制器未过载的情况下优先与匹配列表中排序靠前的 LEO 建立匹配。

定义 3 优选。若 S_n 在从控制器匹配列表 $\Gamma(C_m)=\{S_n, \dots\}$ 中排在首位, 则 S_n 是 C_m 的优选 LEO, 记为 $S_n \leftarrow \Gamma(C_m)$ 。

3.2 基于模拟退火的多控制域均衡划分算法

MDBPSA 算法的基本流程为:首先收集网络历史状态信息,由于卫星轨道的周期性,我们能够获得所需信息作为划域依据。匹配过程中从控制器在处理能力允许的情况下,优先与优选 LEO 进行匹配。针对栅格化的卫星网络使用鲁棒性强的模拟退火算法对从控制器与 LEO 的匹配进行优化,寻求使目标函数最小的匹配方式。不断重复该过程,直至网络中不存在未得到匹配的 LEO, 得到 SDSN 多域划分。

表 1 基于模拟退火的多域均衡规划算法

输入: SDSN 网络 $G=(V, E)$
LEO 流请求速率: $\lambda(t)_n$
从控制器可用处理能力: $\Omega(t)_m$
从控制器冗余因子: β_m
输出: SDSN 多域划分 X
1: 收集网络状态信息, 得到 $\lambda(t)_n, \Omega(t)_m, \Omega_m$, 对从控制器构建匹配列表 $\Gamma(C_m)=\{S_n, \dots\}$
2: while (存在未匹配的交换机)
3: 选取从控制器满足 $L(t)_m < \beta_m \Omega(t)_m$ 且 $\Omega(t)_m > 0$ 及相应 $S_n \leftarrow \Gamma(C_m), (C_p, S_q)$
4: 若无满足条件从控制器则跳出循环
5: 使用模拟退火算法对目标函数进行优化
6: 输出一组匹配结果 (C_p, S_q)
7: $\Gamma(C_m) = \Gamma(C_m) \setminus \{S_q\}$
8: end while
9: 将未匹配的 LEO 与从控制器按 $\Omega(t)_m$ 。降序排列的顺序建立匹配
10: 得到 SDSN 多域划分 X

建立匹配时, LEO 向所有从控制器提交匹配申

请,各从控制器匹配列表中元素相同,只有排序的差异。在匹配过程中若存在从控制器不满足 $L(t)_m > \beta_m \Omega(t)_m | \Omega(t)_m \leq 0$, 则采用模拟退火算法为不满足条件的从控制器建立匹配,并将建立了匹配的 LEO 从匹配列表中剔除。若所有从控制器均满足 $L(t)_m > \beta_m \Omega(t)_m | \Omega(t)_m \leq 0$, 则放宽匹配条件,为仍未得到匹配的 LEO 选择可用处理能力最大或过载程度最轻的从控制器建立匹配,直至匹配列表为空集。综上,在整个匹配过程中不会遗漏任何 LEO,能够为每个 LEO 选取合适的从控制器建立匹配关系。

4 仿真与性能评估

4.1 仿真环境建立

4.1.1 实验工具

实验利用 STK11.01 工具进行卫星轨道的相关仿真并借助 Matlab2016b 工具对算法进行仿真和分析实验结果。

4.1.2 拓扑选择

实验拓扑以 IRIDIUM^[18] 进行调整得到,如表 2 所示。这样可使得时间片划分个数少于 644 个,最短持续时间大于 8 s, 平均持续时间大于 133.17 s^[19], 为控制域规划留出了充足的时间。

表 2 卫星轨道参数设定

参数名称	数值
高度/km	893.8
倾角/(°)	86.4
轨道周期/h	12/7
卫星个数	66
轨道个数	6
每颗卫星的轨道内链路数	固定 2
每颗卫星的轨道间链路数	动态 0~2

4.1.3 仿真参数设定

本文仿真参数设定如下:差异化的选取 LEO 流请求速率 $\lambda(t)_n$ 的值, 将取值范围设定为 100~500 kB/s 以模拟卫星网络控制流量特征。从控制器处理能力均为 8 MB。设定从控制器冗余因子为 0.9, 轮询 LEO 的平均速率为 $v_R=10$ kB/s, 进行状态同步的传播速率 $v_S=1$ kB/s。y 值设定为 0.5。

4.2 仿真结果分析

4.2.1 算法运算时间

为了缩短模拟退火算法运算时长以适应卫星网络拓扑变化, MDBPSA 算法设计了迭代跳出机制以缩短算法运行时间。模拟退火算法的特点是能够依概率接受较差的解, 避免陷入局部最优。因此设置

计数器记录新解被拒绝的次数,若连续 k 次被拒绝,则提前结束本次迭代。 k 与目标函数值及算法运算时间的关系如图 1 所示,将相同条件下重复实验 50 次的最优值作为最终结果。图中红色虚线为不设置迭代跳出机制时 $Object$ 极限值,此时的运算时间为 10.27 s。

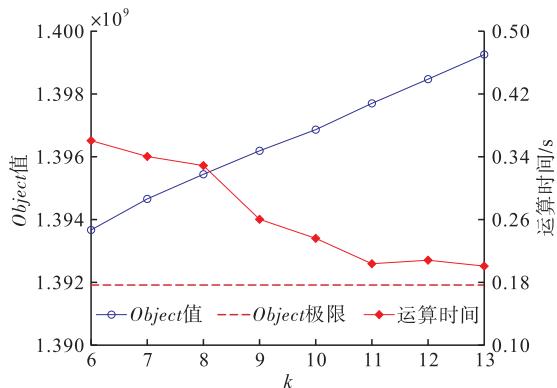


图 1 $Object$ 值、运算时间与 k 的关系

由图 1 可以看出, $k=11$ 时能够在保证求解结果良好的情况下尽可能缩短算法运算时间。此时算法运算时间的累积分布函数如图 2 所示,可知运算时间基本保持在 0.56 s 之内,较时间片最短持续时间小很多。迭代跳出机制能够明显加快算法运算速度,使得算法能够适应卫星网络拓扑的快速动态变化。

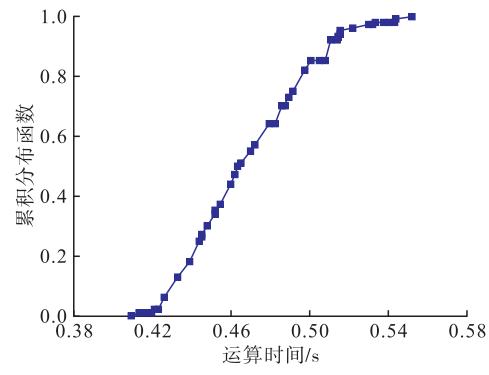


图 2 算法运算时间

4.2.2 子域划分

为了展示 MDBPSA 算法的控制器负载均衡性能,后续实验将地面网络中较为成熟的 AMDA 方法^[5]及文献[12]划域思想(下文简称为 ILP 方法)作为对比。AMDA 算法按照负载自适应方式为控制器动态分配交换机完成多域构建;ILP 方法力求使网络流建立时延最小。以控制域网络拓扑为仿真对象,其中共包含 25 个 LEO 节点及 40 条星间链路。假设网络中所有节点间统计相互独立且都具备部署从控制器的能力,将其中 4 个节点设置为从控制器节点,使用不同方法对上述网络拓扑进行子域划分,结果如图 3 所示。图中以不同形状区分 LEO 节点及从控制器节点,C₁~C₄ 所管理的控制子域 Domain m₁~Domain m₄ 分别以白色、浅绿色、深绿色、黑色表示。

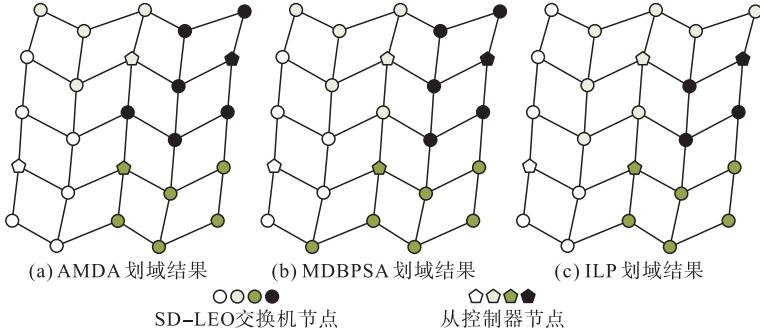


图 3 多域规划结果

3 种划域方式控制子域负载对比见图 4。

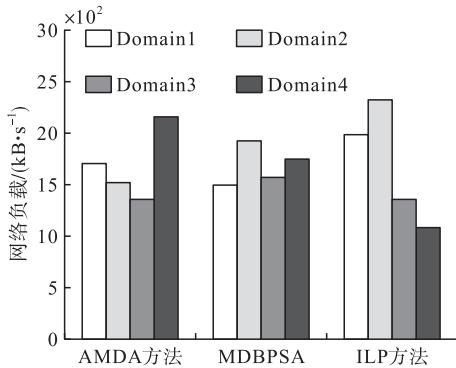


图 4 网络负载对比

从图 4 可以看出 ILP 方法从控制器负载差异

最大,这是由于该方法将流建立时延作为优化目标,仅考虑传播时延而忽略了控制器处理能力限制,这也说明了在卫星网络中部署控制器时不考虑处理时延和排队时延的做法不够合理。AMDA 方法和 MDBPSA 均实现了 LEO 与从控制器的均衡连接。相较而言,MDBPSA 加入了对处理时延的考虑,且网络时延由各控制子域时延按负载加权得到,加强了对网络负载的限制,故能取得更好的负载均衡性能。以从控制器负标准差作为衡量划域均衡性的指标,可得到如下结果:AMDA 方法 297.039、MDBPSA 算法 168.158、ILP 方法 492.446,可以看出 MDBPSA 算法负载均衡性明显优于其他算法,较 AMDA 方

法提升 43.39%，较 ILP 方法提升 65.85%。

4.2.3 LEO 到控制器平均时延及网络开销

在实验 2 所得子域划分的基础上，本实验对 3 种方法 LEO 到从控制器时延进行对比。实验中假设数据包在星间链路传播时速率稳定可靠且丢包率为零。图 5 为 3 种算法的控制域平均时延情况。

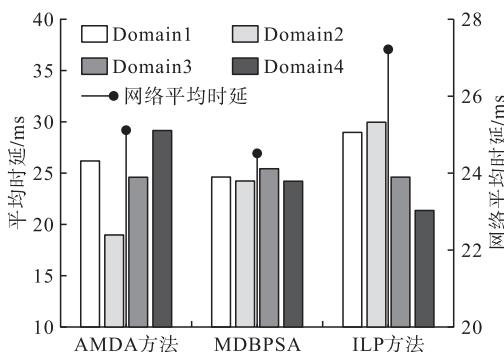


图 5 平均时延对比

从图 5 可以看出，MDBPSA 算法各子域的平均时延相差不大，而 AMDA 方法中 Domain4 和 ILP 算法 Domain2 的平均时延很大。这是因为 AMDA 作为一种弹性规划方式，虽然能够快速的均衡从控制器负载，但从控制器与 LEO 之间的连接关系并不稳定，不稳定的网络关系导致部分从控制器平均时延增大。在更为复杂的网络环境下这种现象更加明显，甚至可能引起跨域通信问题使网络时延急剧增加。而 ILP 算法缺乏对控制器处理时延、排队时延的考虑，导致时延性能不够理想。综合来看，MDBPSA 算法将排队时延和处理时延纳入考虑范围后能够明显平衡各控制子域时延，使网络平均时延低于 AMDA 方法且较 ILP 方法降低约 10%。

图 6 为 3 种方法的网络开销情况。

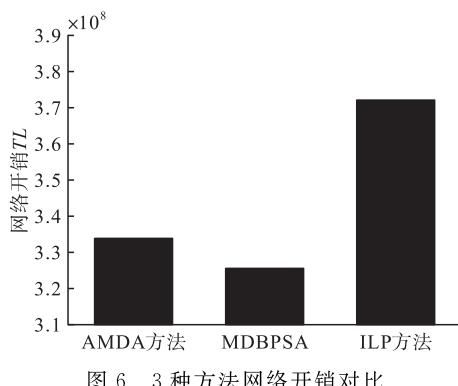


图 6 3 种方法网络开销对比

由图 6 可以看出 MDBPSA 算法在具有较好性能的同时，网络开销也低于对比算法。这是由于 AMDA 方法、ILP 方法分别注重负载和流建立时延而对控制器到交换机距离缺乏关注故而性能并不理想。通过对算法的分析可知，MDBPSA 算法在进行划域时对节点间距离与流量负载情况均进行了考

虑，而网络开销主要与节点间距离及流量负载情况有关，故而 MDBPSA 算法能够使网络开销较 ILP 方法下降 12.52%。

5 结语

针对现有控制域划分算法时延考虑不全面的问题，本文在建立时延模型时加入排队时延和处理时延，并设计了 MDBPSA 算法对网络控制域进行规划，可得以下结论：

1) 迭代跳出机制能有效缩短算法求解时间，使算法对控制域内的子域划分问题求解时间在 0.53 s 之内，能够较好地适应卫星网络环境；

2) 算法具有良好的负载均衡性能，较 AMDA 方法提升 43.39%，较 ILP 方法提升 65.85%；

3) MDBPSA 算法的时延性能明显优于对比算法，且构建的网络连接关系较为稳定，虽然缺乏对流量抖动的敏感，但节省了不必要的网络开销。

未来将在下列方面作进一步改进：

1) 从网络节点可靠性和负载等角度加强从控制器部署选址问题的研究；

2) 将时间片变化时交换机迁移所产生的开销纳入考虑范围。

参考文献

- [1] 杨泽卫, 李呈. 重构网络: SDN 架构与实现 [M]. 北京: 电子工业出版社, 2017: 2-3.
- [2] BERDE P, GEROAL M, HART J, et al. ONOS: Towards an Open, Distributed SDN OS[C]//Proceedings of the ACM SIGCOMM 2014 Workshop on Hot Topics in Software Defined Networking. Chicago: ACM, 2014: 1-6.
- [3] WANG G, ZHAO Y, HUANG J, et al. A K-means-Based Network Partition Algorithm for Controller Placement in Software Defined Network[C]//IEEE International Conference on Communications. Malaysia: IEEE, 2016: 1-6.
- [4] 张创, 谷建华. 基于虚拟机迁移的虚拟机集群负载均衡策略研究[J]. 微电子学与计算机, 2014, 31(4): 85-88.
- [5] XIAO P, LI Z Y, GUO S, et al. A K Self-Adaptive SDN Controller Placement for Wide Area Networks [J]. Frontiers of Information Technology & Electronic Engineering, 2016, 17(7): 620-633.
- [6] 胡涛, 张建辉, 孔维功, 等. SDN 中基于双向匹配的多控制器动态部署算法[J]. 通信学报, 2018, 39(1): 159-169.

(下转第 52 页)

- LFSR Architectures Based on Improved State-Space Transformations [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2017, 25 (3): 1159-1163.
- [10] 王杰. NAND Flash 主控中 BCH 编译码器的研究和 ASIC 实现[D]. 杭州:浙江大学, 2010.
- [11] 毛旭富. BCH 编解码器在 NAND Flash 主控中的研究与优化实现[D]. 上海:上海交通大学, 2013.
- [12] CHOI H, LIU W, SUNG W. VLSI Implementation of BCH Error Correction for Multilevel Cell NAND Flash Memory[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2010, 18(5): 843-847.
- [13] BELLORADOJ, KAVČIĆ A. Low-Complexity Soft-Decoding Algorithms for Reed-Solomon Codes-part I: An Algebraic Soft-in Hard-Out Chase Decoder [J]. IEEE Transactions on Information Theory, 2010, 56 (3): 945-959.
- [14] 徐富新, 刘应, 刘雁群, 等. 模式可配置的 NAND Flash 纠错系统设计与实现[J]. 中南大学学报(自然科学版). 2013, 44(5): 1918-1925.
- [15] PAAR C. A New Architecture for a Parallel Finite Field Multiplier with Low Complexity Based on Composite Fields[J]. IEEE Transactions on Computers, 1996, 45(7): 856-861.
- [16] ZHANG M, WU F, XIE C, et al. A Novel Optimization Algorithm for Chien Search of BCH Codes in NAND Flash Memory Devices[C]//IEEE International Conference on Networking, Architecture and Storage (Nas 2015). Boston: IEEE, 2015:106-111.

(编辑:徐敏)

(上接第 38 页)

- [7] 杨力, 孔志翔, 石怀峰. 软件定义空间信息网络多控制器动态部署策略[J]. 计算机工程, 2018, 44(10): 64-69.
- [8] WU S, LIU X F, CHEN Q, et al. Update Method for Controller Placement Problem in Software-Defined Satellite Networking[C]//28th International Conference on Computer Communication and Networks (ICCCN). Valencia, Spanish:[s. n.] 2019: 1-7.
- [9] XU S, WANG X W, GAO B Y, et al. Controller Placement in Software-Defined Satellite Networks [C]//2018 International Conference on Mobile Ad-hoc and Sensor Networks. Shenyang, China: [s. n.], 2018: 146-151.
- [10] HU H F, ZHANG S S, TANG B H. LEO Software Defined Networking Based on Onboard Controller [C]//2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP). California, USA: IEEE, 2018: 1068-1071.
- [11] WU S, CHEN X Q, YANG L, et al. Dynamic and Static Controller Placement in Software-Defined Satellite Networking[J]. Acta Astronautica, 2018, 152: 1-10.
- [12] Papa A, Cola T D, Vizarreta P, et al. Dynamic SDN Controller Placement in a LEO Constellation Satellite Network [C]//2018 IEEE Global Communications Conference (GLOBECOM). Abu Dhabi, UAE: IEEE, 2018: 9-13.
- [13] 刘治国, 卢美玲, 李慧, 等. 基于 SDN 的卫星网络多控制器部署方法研究[J]. 计算机仿真, 2020, 37(4): 62-66, 97.
- [14] XIA D, ZHENG X, DUAN P, et al. Ground-Station Based Software-Defined LEO Satellite Networks[C]//2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS). Tianjin, China: IEEE, 2019: 687-694.
- [15] LING T, LIU L, ZHENG C, et al. An Optimized Forward Scheduling Algorithm in a Software-Defined Satellite Network[C]//2019 IEEE 17th International Conference on Software Engineering Research, Management and Applications (SERA). Lagos, Nigeria: IEEE, 2019: 27-32.
- [16] XU S, WANG X W, HUANG M, et al. Software Defined Next Generation Satellite Networks: Architecture, Challenges, and Solutions[J]. IEEE Access, 2018, 6: 4027-4041.
- [17] BI Y G, HAN G J, XU S, et al. Software Defined Space-Terrestrial Integrated Networks: Architecture, Challenges, and Solutions[J]. IEEE Network, 2019, 33(1): 22-28.
- [18] LEOPLOD R J, MILLER A. The Iridium Communications System[J]. IEEE Potentials, 1993, 12(2): 6-9.
- [19] LONG F. Satellite Network Robust QoS-Aware Routing [M]. Berlin: Springer, 2014:41-74.

(编辑:韩茜,徐楠楠)