计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2021.20200384
58(2):319-337,2021

面向跨数据中心网络的节点约束存储转发调度方法

林 霄[1] 姬 硕[1] 岳胜男[2] 孙卫强[2] 胡卫生[2]
[1](福州大学物理与信息工程学院 福州 350116)
[2](区域光纤通信网与新型光通信系统国家重点实验室(上海交通大学) 上海 200240)
(linxiaocer@fzu.edu.cn)

Node-Constraint Store-and-Forward Scheduling Method for Inter-Datacenter Networks

Lin Xiao[1], Ji Shuo[1], Yue Shengnan[2], Sun Weiqiang[2], and Hu Weisheng[2]
[1](*College of Physics and Information Engineering, Fuzhou University, Fuzhou* 350116)
[2](*State Key Laboratory of Advanced Optical Communication Systems and Networks (Shanghai Jiao Tong University), Shanghai* 200240)

Abstract Performing store-and-forward (SnF) using abundant storage resources inside datacenters has been proven to be effective in overcoming the challenges faced by inter-datacenter bulk transfers. Most prior studies attempt to fully leverage the network infrastructure and maximize the flexibility of the SnF scheme. Their proposed scheduling methods hence aim at a full storage placement where all network nodes (e.g., datacenters) are SnF-enabled and every node is taken into account in the scheduling process. However, the computational complexity of the prior methods exponentially increases with the network scale. As a result, the prior methods may become too complicated to implement for large-scale networks and online scheduling. In this work, based on the inter-datacenter optical network, SnF models are presented to quantify the impact of the number of SnF-enabled nodes on the performance and the complexity of the SnF scheduling problem. Our key findings show that taking a few SnF-enabled nodes into account in the scheduling process can provide high performance while maintaining low complexity under certain circumstances. It is unnecessary to take every node into account in the scheduling process. Therefore, a node-constraint SnF scheduling method is proposed, whose features are twofold: 1) by taking a portion of nodes into account, it reduces the complexity of the SnF scheduling problem; 2) by introducing a topology abstraction, it condenses the link states between the considered nodes and hence reduces the problem size, which improves its efficiency in solving the SnF scheduling problem. Simulations demonstrate that the proposed method outperforms the prior method in terms of blocking probability and computation time.

Key words big data transfers; inter-datacenter networks; wavelength routing; storage; scheduling method

收稿日期:2020-06-08;**修回日期**:2020-08-10

基金项目:国家自然科学基金青年科学基金项目(61901118);国家自然科学基金重点项目(61433009);上海交通大学区域光纤通信网与新型光通信系统国家重点实验室开放基金项目(2019GZKF03003)

This work was supported by the National Natural Science Foundation of China for Young Scientists (61901118), the Key Program of the National Natural Science Foundation of China (61433009), and the Open Foundation of the State Key Laboratory of Advanced Optical Communication Systems and Networks (2019GZKF03003).

摘 要 借助海量数据中心存储,通过存储转发(store-and-forward, SnF)调度大数据传输,已被证明能有效解决跨数据中心间大数据传输难题.然而,多数现有调度方法将数据途经的所有网络节点(例如数据中心)均纳入 SnF 调度决策,导致其计算复杂度过高,难以为大规模网络提供实时调度服务.针对跨数据中心光网络场景,给出 SnF 模型,量化分析存储节点数量对调度问题性能与复杂度的影响.研究表明:在一定条件下,无需将所有节点都纳入调度决策也可获得良好的调度性能.由此,提出了节点约束 SnF 调度方法.该方法的特点在于:1)仅将部分数据途经节点纳入调度决策,降低调度问题求解难度;2)引入拓扑抽象,将被选节点间链路状态压缩,缩小调度问题规模、提高算法求解效率.仿真结果表明:在阻塞率和算法计算时间方面,该方法优于现有调度方法.

关键词 大数据传输;跨数据中心网络;波长路由;存储;调度方法

中图法分类号 TP391

随着新兴在线应用与云服务的迅猛发展,跨数据中心的大数据传输需求正呈现井喷态势[1-2].目前典型的大数据传输应用,例如数据中心(datacenter, DC)备份、大科学计算等,其传输数据量多达数百太字节(TB),传输所需带宽高达数吉比特每秒(Gbps),传输时间甚至可持续数天[3-5].这对跨数据中心网络提出了前所未有的挑战.由于大数据传输时间长(数小时到数天),因此数据往往对传输延时不敏感,具有延迟容忍性(delay tolerant).为了应对上述挑战,许多研究工作尝试利用延迟容忍性,为大数据的传输与调度提供额外灵活性[3-9].

文献[3-9]的共同特征之一是采用端到端(end-to-end, E2E)连接传输大数据.然而,网络中带宽资源的使用情况在时间和空间上均呈现不均衡性,这使得在较大网络范围内进行 E2E 数据传输难以实现.例如,在跨时区的数据传输中,由于不同时区中网络出现带宽使用的峰谷时间不一致,跨多个时区的 E2E 高带宽通路难以实现.即便在同一个时区中,由于网络中各条链路带宽可用情况差异甚大,能够提供给 E2E 传输的时间窗口也很小,难以满足大数据传输要求[10].为承载持续上升的网络高峰期流量,即便网络在非高峰期有大量闲置带宽,数据中心运营商仍然必须不断从互联网服务提供商(Internet service provider, ISP)购买更多的带宽,或者持续升级其专用线路的带宽容量.

为了克服 E2E 传输面临的困境,DC 存储被引入数据传输路径.当网络流量进入高峰时段(例如正午),将延迟容忍的数据缓存于途经节点(例如 DC),从而避免大、小数据流的带宽竞争;当网络流量进入低谷时段(例如凌晨),充分利用大量闲置带宽资源继续传输数据,即通过存储转发(store-and-forward, SnF)"错峰传输"大数据,已被证明能有效解决跨数据中心间大数据传输难题[11].

本文以海量 DC 存储与光电路交换(optical circuit switching, OCS)的结合作为研究场景.一方面,OCS 可以为大数据提供高带宽、大容量、低开销的传输通道.另一方面,通过 SnF 实现数据分段传输,避免了传统 OCS 面临的 E2E 传输困境[11-15].然而,存储的引入将传统的路由问题变为更加复杂的 SnF 调度问题.求解该调度问题,不仅需要在空间上寻找传输路径,还需要在时间上规划何时传输、何时缓存.相比传统路由问题,SnF 调度问题求解难度大,计算复杂度高[10].此外,不恰当地调度存储、带宽资源,反而导致绕路、网络资源碎片化等问题,进而恶化网络性能[13].显然,SnF 效能的发挥取决于能否高效、合理求解 SnF 调度问题.

本质上,SnF 的灵活性取决于数据传输路径上的存储节点数.每个存储节点都为调度决策提供一个 SnF 选项.存储节点越多,SnF 调度方案就越灵活.目前大多数研究工作旨在最大限度地发挥 SnF 的灵活性,因此将数据传输路径上的所有节点均纳入 SnF 调度决策[10-21].当每个节点接收到数据后,其必须决定是否存储该数据、需要存储多长时间,以及应该以何种速率将该数据传输到下一节点.这导致调度问题的复杂度随传输路径跳数的增加呈指数增长[10,15].因此,在大规模网络、动态网络场景下,调度问题将变得难以求解.

在实际传输过程中,数据通常只需在部分途经节点,而非途经的所有节点,进行 SnF 即可满足传输需求[13,16].例如,文献[16]研究表明大多数请求通过 1,2 次 SnF 即可到达目的地.此外,在 OCS 网络中,每次 SnF 都需要进行昂贵的光电光(optical-to-electrical-to-optical, OEO)转换,这会带来额外的功耗和网络管控开销[11].由此,本文将探索如何将

部分途经节点而非所有节点纳入 SnF 调度决策,从而降低计算复杂度的方法.

本文的主要贡献有 3 个方面:

1) 首先借助理论分析模型,比较了 SnF 与 2 种典型 E2E 传输方式的调度性能与复杂度.研究表明,在某些情况下,即使使用单个存储节点传输方式,也可以获得与多存储节点传输方式近似的调度性能.

2) 进一步扩展理论分析模型,量化分析了参与调度决策的存储节点数量对 SnF 调度性能与复杂度的影响.研究表明,在调度过程中,仅将部分途经节点纳入调度问题决策,同时扩大时间维度的调度范围,不仅能获得更好调度性能,同时能有效降低计算复杂度.

3) 提出了节点约束 SnF 调度方法.该方法将部分数据途经节点纳入调度决策,同时根据所选节点进行拓扑抽象.给定相同的计算复杂度限制,该方法可以对更大时间范围内的网络状态进行搜索,比使用所有节点进行调度的现有调度方法作出更优的调度决策.仿真表明,该方法在阻塞率和运算时间方面优于现有调度方法.

1　相关工作

1.1　现有 SnF 调度方法

本文根据调度策略,将现有 SnF 调度方法分为 2 类:1)基于最大灵活性(max flexibility, MF)策略的调度方法,该策略旨在最大化 SnF 的调度灵活性,因此其将所有网络节点都纳入调度决策;2)基于节点约束(node constraint, NC)策略的调度方法,该策略旨在以牺牲一定的调度灵活性为代价简化调度问题,因此其仅将部分网络节点纳入调度决策.图 1 对现有 SnF 调度方法进行分类总结.

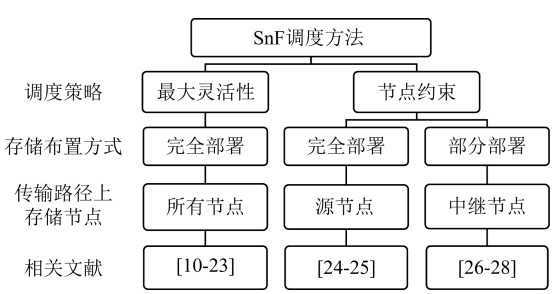


Fig. 1　Classification of existing SnF scheduling methods

图 1　已有 SnF 调度方法及其分类

1) 基于 MF 策略的 SnF 调度方法

多数学者旨在充分利用现有网络基础架构,最大程度地发挥 SnF 调度的灵活性,即采用基于 MF 的调度策略.这些学者针对所有网络节点均部署有存储的网络场景开展研究,即完全部署(full placement)网络场景[10-23].

在此基础之上,学者们提出的 SnF 调度方法会将数据传输路径上的所有节点均纳入调度决策[10-23].因此,每个节点都为 SnF 提供了潜在可能.在这些研究工作中,SnF 调度问题被建模为优化问题,例如线性规划问题和网络流量问题[10-13,18-21],并且采用经典的优化算法或启发式算法来求解问题,实现路由、调度与资源分配的最优解.现有以优化建模为基础的 SnF 调度方法复杂度较高,更适合解决小规模网络、静态网络场景下的离线调度优化问题.静态网络场景中,请求的数量和到达时间都是提前已知的.然而,文献[14]的研究表明,SnF 调度问题的规模会随着传输路径跳数的增加而呈指数增长.这意味着当调度问题规模较大时,问题可能难以求解.因此,上述 SnF 调度方法难以为大规模网络、动态网络场景提供实时调度.在动态网络场景中,请求是随机到达网络,请求的数量和到达时间都是随机的.一部分学者同样也针对完全部署网络场景开展研究.但是,他们对调度过程进行了部分限制[22-23].文献[22]旨在通过最少的 SnF 次数来完成数据传输.因此,文献[22]所提出的调度方法是采用贪婪算法,从允许一次 SnF 开始逐次递增 SnF 次数,直到找到可行的 SnF 方案为止.文献[23]旨在减轻存储系统的高速读/写负担,所以仅允许源节点缓存数据.然而,文献[22-23]均未研究存储节点数对 SnF 的性能与复杂度的影响.

2) 基于 NC 策略的 SnF 调度方法

采用基于 NC 的调度策略以牺牲一定的调度灵活性为代价,简化调度问题.其中,一部分学者针对完全部署网络场景开展研究[24-25];另一部分学者针对仅部分网络节点部署有存储的网络场景展开研究,即部分部署(partial placement)的网络场景[26-28].

在完全部署网络场景下,文献[24-25]比较了 SnF 和提前预约机制(advance reservation, AR).AR 可等价于仅有源节点具备存储功能的一个 SnF 特例.文献[24-25]研究表明,当网络负载较高时,与 SnF 相比,AR 所能带来的性能增益是有限的.

在部分部署网络场景下[27],文献[26]针对 3 节点串联网络展开研究,其中仅有中继节点具备 SnF 能力.文献[26]将 SnF 调度问题建模为单跳、单路径传输问题,但并未考虑任何路由或调度问题.在文献

[27]中,当 E2E 光路无法建立时,调度方法将尝试从源节点建立光路到最近的存储节点,以便缓存数据,等待一段时间后再尝试建立从存储节点到目的地的光路.尽管数据传输路径可能会途经多个中继节点,但是只有一个中继节点是存储节点.这就极大限制了 SnF 的灵活性,以及所能带来的性能增益.此外,文献[27]将存储部署问题转化为设施选址问题(facility location problem),并通过拓扑中心性求解该问题.显然,存储部署主要取决于网络拓扑特征,但并未考虑任何调度复杂度问题.文献[28]旨在以最小的存储使用量获得最大网络数据传输能力.文献[28]将路由问题与存储使用问题转化为最大流问题(maximum flow problem),从而实现对上述 2 个问题的联合优化.文献[28]同样也没有考虑存储节点数量对于调度复杂度的影响.

3) 现有研究工作的总结

多数工作主要研究基于 MF 策略的 SnF 调度方法.这些工作所提出的调度方法更适合小规模网络、静态网络场景下的离线调度优化,难以为大规模网络、动态网络场景提供实时调度.另一部分工作研究如何使用源节点或部分中继节点进行调度,即采用基于 NC 策略的 SnF 调度方法.然而,这些研究工作旨在于减少存储部署或使用量,并未考虑存储节点数量对于调度性能和复杂度的影响.显然,调度方法的设计本质上是对性能与复杂度的折中.设计高效的 SnF 调度方法应当仔细考虑用于调度决策的存储节点数.然而,现有的研究工作尚未充分考虑该问题.

与现有基于 MF 策略的调度方法不同,本文所提出的新型调度方法将 NC 调度策略融入基于时移多层图(time-shifted multilayer graph, TS-MLG)[16]的路由调度,有效减少了调度决策过程需要的存储节点数量.在此基础之上,新型调度方法使用基于存储节点的拓扑抽象,在保证调度性能的同时减小调度问题规模、降低调度问题求解难度.现有基于 NC 策略的调度方法通常固定选择源节点或部分中继节点作为存储节点.与此不同,新型调度方法根据节点对之间路由跳数的不同,选择合适的存储节点数,获得了低复杂度、高调度性能,更适合为大规模网络、动态网络场景提供实时、高效的调度服务.

1.2 时移多层图

时移多层图[16]是一种面向 SnF 的路由调度框架,是由多个层组成的图,如图 2 所示.TS-MLG 中每层都是网络拓扑在某个时刻的快照,反映了当时的网络状态.TS-MLG 通过一系列拓扑快照,捕捉了随时间变化的网络状态.例如,TS-MLG 的最顶层表明时刻 t_0 链路 E—F 没有可用带宽.第 2 层表明时刻 t_1 链路 $E^{(1)}$—$F^{(1)}$ 有可用带宽.此外,TS-MLG 引入时间链路(temporal link)和空间链路(spatial link)的概念.请求穿越时间、空间链路分别表示数据缓存和数据传输.仅需对 TS-MLG 进行最短路由,即可获得一条"穿越时空的端到端"路径,同时实现空间路由与时间调度的融合调度、带宽与存储资源的联合分配,极大简化了 SnF 调度问题的求解过程.

假设传输请求 r 在时刻 t_0 到达网络,要求从节点 A 向节点 E 传输数据.然而,在时刻 t_0 无法进行 E2E 传输.但通过对图 2 所示的 TS-MLG 进行路由搜索(例如使用 Dijkstra 算法),即可找到路径 A—F—$F^{(1)}$—$E^{(1)}$,其中节点 F 是中继存储节点.显然,借助 TS-MLG,时空二维的 SnF 调度问题被转变为一维的路由问题.

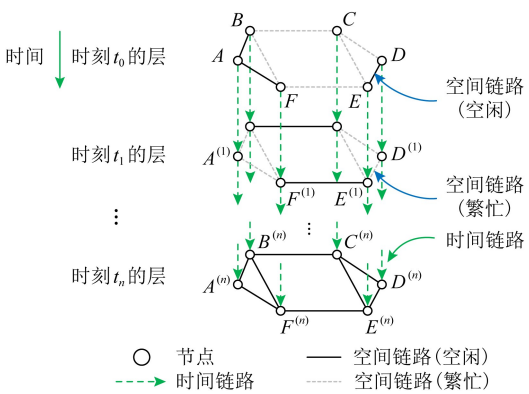


Fig. 2 Time-shifted multilayer graph (TS-MLG)

图 2 时移多层图(TS-MLG)

随着请求的到达和离开,网络状态将发生改变,TS-MLG 的层数也将发生动态改变.当网络产生新状态,新层将被添加到 TS-MLG.随着时间流逝,当现有网络状态超时,相应的层将被从 TS-MLG 中移除.

路由算法的计算复杂度通常取决于网络拓扑的规模.TS-MLG 的规模等于网络拓扑节点数乘以 TS-MLG 的层数.因此,TS-MLG 的层数很大程度上决定了对其进行路由搜索的计算复杂度.由于数据流的突发性可能会导致 TS-MLG 的层数出现短时激增,使得后续请求的路由复杂度陡然增大.为了限制复杂度,用于路由的层数必须予以限制.因此,请求是否能被网络接收取决于请求是否能在给定的 TS-MLG 层数内找到有效路径.

2　不同数据传输方式的比较研究

立即预约(immediate reservation, IR)和提前预约(advance reservation, AR)是 OCS 网络中 2 种典型 E2E 数据传输方式[29].本节首先简要介绍了 IR,AR,SnF 的基本原理,随后利用理论模型比较分析了 3 种数据传输方式的性能与复杂度.本文所涉及的主要符号及其含义如表 1 所示:

Table 1　Table of Notations

表 1　主要符号说明表

符号	定义
R	从源节点 s 到目的节点 d 的固定路由,其中 $R=\{s,i_1,i_2,\cdots,i_{N-2},d\}$
N	固定路由 R 的节点数量
L	TS-MLG 中可以用于调度的层数
l	TS-MLG 的第 l 层
$d^{(l)}$	TS-MLG 中位于第 l 层的目的节点 d
L_L	当层数为 L 时 TS-MLG 中层的集合,其中 $L_L=\{1,2,\cdots,l,\cdots,L\}$
p_b	请求在 1 条空间链路上找不到所需带宽的概率
p_s	请求在 1 条时间链路上找不到所需存储的概率
N_s	固定路由 R 的存储节点数量
L_s	NC 模型的层数
L_{L_s}	当层数为 L_s 时 TS-MLG 中层的集合,其中 $L_{L_s}=\{1,2,\cdots,l,\cdots,L_s\}$

在 IR 中,请求是否能被接收取决于当前网络带宽是否能够提供 E2E 传输.当请求到达网络时,网络调度器(scheduler)必须立即根据网络当前的带宽资源可用情况作出调度决策.在 AR 中,请求的传输可以被推迟,并等待可用的 E2E 连接出现再开始传输.源节点具备推迟请求传输的能力.因此,当请求到达网络时,网络调度器根据网络当前与未来的带宽资源可用情况作出调度决策.在 SnF 中,每个节点都可以进行 SnF.因此,当请求到达网络时,网络调度器根据网络当前与未来的带宽与存储资源可用情况作出调度决策.IR 与 AR 可等价为 SnF 在没有任何节点或仅有源节点具备存储能力时的 2 种特例.

2.1　建模分析

本文扩展了文献[14]的分析模型,比较 IR,AR,SnF.假设固定路由 $R=\{s,i_1,i_2,\cdots,i_{N-2},d\}$,其中 s 是源节点,d 是目的节点,N 表示固定路由的节点总数,且 $N\geqslant 2$.基于 TS-MLG 的 IR,AR,SnF 路由模型,如图 3 所示.假定 L 层可以用于调度.

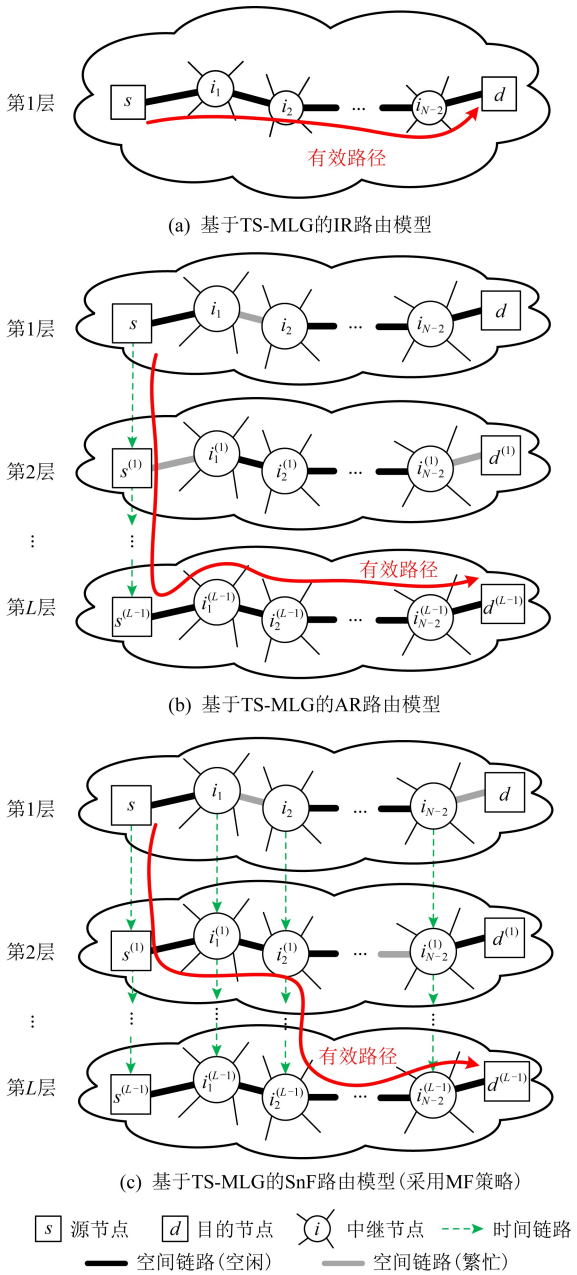


Fig. 3　Routing models of the three transfer manners

图 3　3 种传输方式的路由模型

IR 的 TS-MLG 仅由一层组成,如图 3(a)所示.这是因为 IR 中仅考虑了当前网络状态.在图 3(b)中,AR 的 TS-MLG 由 L 层组成,因为 AR 需要同时考虑当前和未来的网络状态.此外,仅源节点通过时间链路相互连接.在图 3(c)中,由于每个节点都可以进行 SnF,所以 TS-MLG 包含多层、多条时间链路.令 $d^{(l)}$ 表示 TS-MLG 中位于第 l 层的目的节点 d,其中 $l\in[1,L]$.在本文中,备选路径(alternate path)定义为 TS-MLG 中从节点 s 到节点 $d^{(l)}$ 的路径,其中 $l\in[1,L]$.备选路径并未考虑每条链路的

资源可用性.有效路径(viable path)定义为在每条空间或时间链路都具备所需带宽或存储资源的备选路径.

借助 TS-MLG,本文将调度问题转变为路由问题.求解路由问题的简单方法之一就是列举从 s 到 d 的所有备选路径,并从中搜寻有效路径.因此,备选路径数量决定了路由问题的计算复杂度.例如,在图 3(a)中,节点对(s,d)之间只有一条备选路径.而随着层数增加,备选路径可以到达不同 $d^{(l)}$.如图 3(b)(c)所示,(s,d)之间的备选路径数随层数增加而增加.$P_{(s,d)}(N,L)$定义为从 s 到 $d^{(L_L)}$ 的所有备选路径的总数,其中 $L_L=\{1,2,\cdots,l,\cdots,L\}$.本文采用 $P_{(s,d)}(N,L)$来衡量 SnF 调度问题的计算复杂度.

此外,SnF 的主要思想是采用存储置换带宽.因此,分析模型需要进一步考虑网络资源的可用性.预约失败率 $F_{(s,d)}(N,L)$定义为未能从数量为 $P_{(s,d)}(N,L)$的备选路径集合中找到有效路径的概率.令 p_b 表示请求未能在一条空间链路上找到所需带宽的概率;p_s 表示请求未能在一条时间链路上找到所需存储的概率.本文采用 $F_{(s,d)}(N,L)$来衡量 SnF 调度问题的潜在调度性能.

在图 3(a)中,IR 的 TS-MLG 仅有 1 层,因此,$P^{IR}_{(s,d)}=1$.由于 $N-1$ 个空间链路串联连接,所以如果 $N-1$ 个空间链路中有任何一条链路不能提供所需的带宽,请求就将被阻塞.由此可得:

$$F^{IR}_{(s,d)}(N)=1-(1-p_b)^{N-1}. \tag{1}$$

在图 3(b)中,AR 的 TS-MLG 中共有 L 层,因此,$P^{AR}_{(s,d)}(L)=L$.由于 L 条备选路径共享部分时间链路,它们的预约失败率应该是相关的.为简化问题,假设每条备选路径的时间链路都是相互独立的.因此,仅当 L 条备选路径都无法提供可用资源时,请求才会被阻塞.由此可得:

$$F^{AR}_{(s,d)}(N,L)=\prod_{l=1}^{L}[1-(1-p_s)^{l-1}(1-p_b)^{N-1}]. \tag{2}$$

在图 3(c)中,SnF 的 TS-MLG 中共有 L 层,且每个节点均有时间链路链接.$P^{SnF}_{(s,d)}(N,L)$和 $F^{SnF}_{(s,d)}(N,L)$表达式可参见文献[12].

N 和 L 决定了 TS-MLG 的规模.增加 N 表示选择一条更长的路由传输数据,而增加 L 表示扩展时间维度的调度范围(例如允许更长的数据缓存时间).

2.2 调度性能与复杂度分析

本文借助 $P_{(s,d)}(N,L)$和 $F_{(s,d)}(N,L)$对 3 种传输方式的复杂度和调度性能进行了比较研究.

给定 N 和 L,表 2 比较了 3 种传输方式所能提供的备选路径数量,即 $P_{(s,d)}(N,L)$.$P^{IR}_{(s,d)}$ 与 N 和 L 无关,始终保持为 1.$P^{AR}_{(s,d)}$ 随 L 线性增加,但与 N 无关.因此,表 2 中 IR 和 AR 的结果没有变化.$P^{SnF}_{(s,d)}$ 随着 L 增大而增大.N 值越大,$P^{SnF}_{(s,d)}$ 的增幅越大.相比于 IR 和 AR,SnF 能够提供更多备选路径,请求因此也更容易通过 SnF 找到有效路径.但是,数量庞大的备选路径集合也给路由搜索造成了沉重的计算负担.

Table 2 Complexity Comparison Based on $P_{(s,d)}(N,L)$

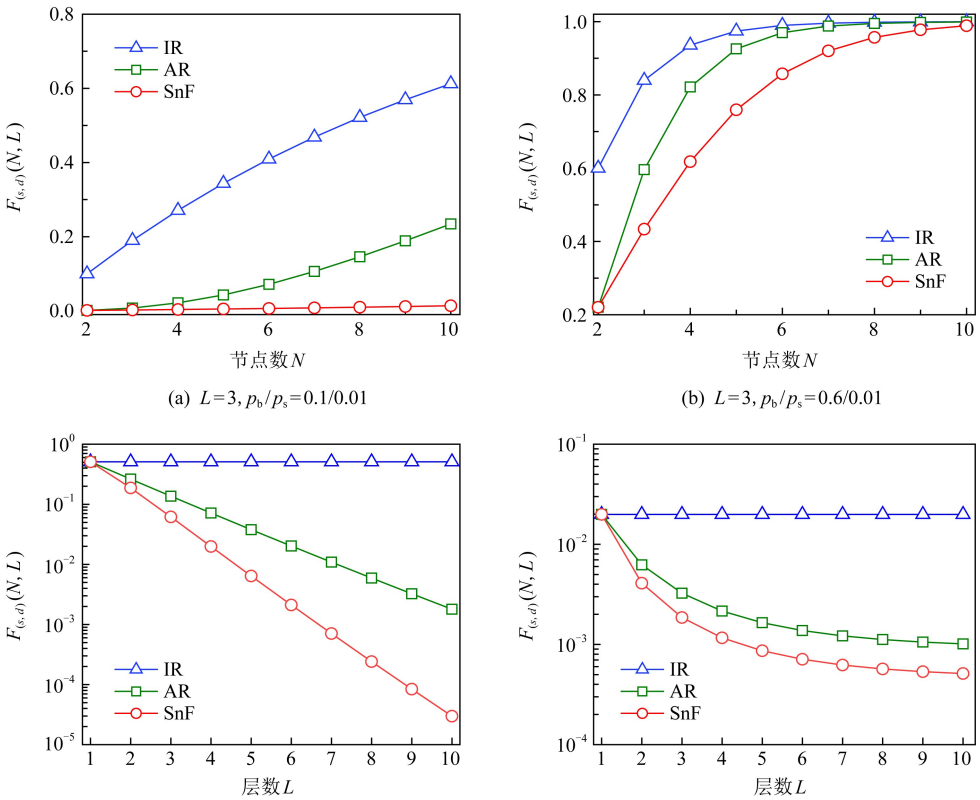
表 2 基于 $P_{(s,d)}(N,L)$的复杂度比较

N	传输方式	$L=1$	$L=2$	$L=3$	$L=4$	$L=5$
3	IR	1	1	1	1	1
	AR	1	2	3	4	5
	SnF	1	3	6	10	15
5	IR	1	1	1	1	1
	AR	1	2	3	4	5
	SnF	1	5	15	35	70

给定 $N,L,p_b/p_s$,图 4 比较了 3 种传输方式的预约失败率,即 $F_{(s,d)}(N,L)$.图 4(a)~(c)假设 $p_s \ll p_b$.因为在典型的跨数据中心网络中带宽资源较为稀缺,而存储资源较为丰富.在某些城域网应用场景中,例如传统通信机房 DC 化(center office re-architected as datacenter, CORD)和面向边缘计算的微型 DC,有限的存储资源可能不足以满足大数据传输需求.因此,在这些网络场景中,p_b 和 p_s 的取值值得未来继续研究.

图 4(a)(b)表明 3 种传输方式下 $F_{(s,d)}(N,L)$均随 N 增大而增大.请求更难在较长的路径上找到可用资源.$F^{SnF}_{(s,d)}$ 明显低于其他 2 种传输方式.当 $N\in[2,6]$且 $p_b=0.1$ 时,$F^{AR}_{(s,d)}$ 与 $F^{SnF}_{(s,d)}$ 的差距较小.随着 N 增大,两者差距逐渐增大.此外,p_b 的数值越大,请求找到所需带宽的难度就越大,$F_{(s,d)}(N,L)$随 N 增大的幅度也越大.在图 4(c)中,当 $p_b/p_s=0.3/0.01$ 时,$F^{AR}_{(s,d)}$ 和 $F^{SnF}_{(s,d)}$ 均随 L 的增大而减小,而 $F^{IR}_{(s,d)}$ 始终保持恒定.$F^{AR}_{(s,d)}$ 与 $F^{SnF}_{(s,d)}$ 的差距随着 L 的增大而扩大.

之前的研究均考虑 $p_s \ll p_b$ 的情况.为了不失一般性,本节进一步考虑 $p_s \gg p_b$ 的情况.图 4(d)假设 $p_b/p_s=0.01/0.3$.此时,带宽资源充足,但是存储资源稀缺.此时,增大 L 难以有效降低 $F^{AR}_{(s,d)}$ 与 $F^{SnF}_{(s,d)}$.



(a) $L=3, p_b/p_s=0.1/0.01$　(b) $L=3, p_b/p_s=0.6/0.01$

(c) $N=3, p_b/p_s=0.3/0.01$　(d) $N=3, p_b/p_s=0.01/0.3$

Fig. 4　Performance comparison based on $F_{(s,d)}(N,L)$

图 4　基于 $F_{(s,d)}(N,L)$的调度性能比较

2.3　分析与总结

借助复杂度与调度性能分析模型,本节比较了IR,AR,SnF 这 3 种数据传输方式,主要发现如下:

1) 虽然 IR 和 AR 的复杂度较低,但对于大规模网络、动态场景,其性能可能不足.SnF 的性能优于 IR 和 AR,但代价是复杂度较高.

2) 当 N,L,p_b 较小时,AR 和 SnF 之间的性能差距较小.这表明,在这种情况下,使用 1 个存储节点参与调度即可获得较好性能,而无需多个存储节点参与调度.但随着 L 的增大,AR 和 SnF 的性能差距急剧扩大.这表明,当允许调度器在更大的时间范围内进行资源调度时,SnF 比 AR 更有优势.

3) 相比 IR 和 AR,SnF 具备的高性能与高复杂度源于其能够使用更多的存储节点参与调度决策.因此,有必要进一步研究存储节点数对 SnF 的影响.

3　存储节点数量对 SnF 的影响

本节首先扩展分析模型以量化存储节点数对 SnF 的影响;然后将 MF 调度策略与 NC 调度策略进行比较研究.

3.1　建模分析

假设固定路由 $R=\{s,i_1,i_2,\cdots,i_{N-2},d\}$.令 N_s 表示 R 上的存储节点数.目的节点不能用于数据缓存.基于 NC 调度策略的 SnF 模型,简称 NC 模型,如图 5 所示,其中 R 上的前 N_s 个节点是存储节点,

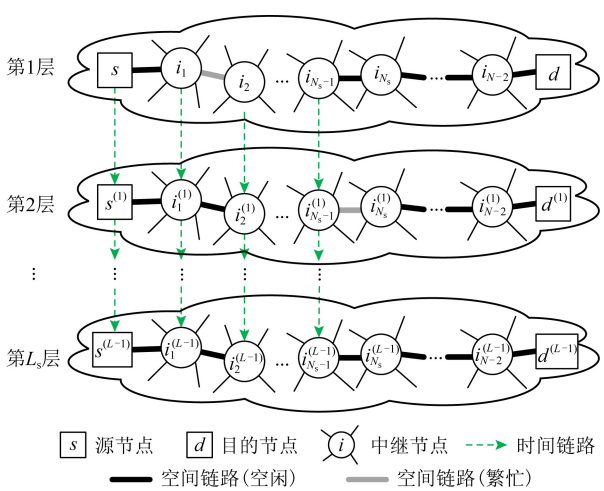


Fig. 5　SnF model based on NC scheduling strategy (N_s storage nodes and L_s layers, $1<N_s<N-1$)

图 5　基于 NC 调度策略的 SnF 模型(N_s 个存储节点和 L_s 层,$1<N_s<N-1$)

L_s 层可以用于调度.基于 MF 调度策略的 SnF 模型,简称 MF 模型,如图 3(c)所示.L 和 L_s 分别表示 MF 模型和 NC 模型的层数.

当 $N_s=1$ 且源节点是存储节点时,NC 模型等价于图 3(b)中所示的 AR 模型.当 $N_s=N-1$ 时,NC 模型等价于 MF 模型,如图 3(c)所示.当 $1<N_s<N-1$ 时,存在多种存储节点部署方案,如图 5 所示.

$P^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)$ 定义为从 s 到 $d^{(L_{L_s})}$ 的备选路径总数,其中 $L_{L_s}=\{1,2,\cdots,l,\cdots,L_s\}$.$P^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)$ 与存储节点部署方案无关.这是因为只有存储节点连接时间链路.对于存储节点,其路由选择要么是其下游相邻节点(即下一跳空间链路),要么是未来的自己(即下一跳时间链路);而对于非存储节点,其路由选择仅限于下游相邻节点.因此,NC 模型可以等效于规模较小的 MF 模型.进一步说,有 N 个节点,N_s 个存储节点和 L_s 层的 NC 模型等效于具有 N_s+1 个节点和 L_s 层的 MF 模型(其中 $L=L_s$).由此可得:

$$P^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)=P^{\mathrm{SnF}}_{(s,d)}(N_s+1,L_s). \quad (3)$$

给定目的节点 $d^{(l)}$,从 s 到 $d^{(l)}$ 的所有备选路径都具有相同的空间跳数(即 $N-1$)和时间跳数(即 $l-1$),并且与具体存储节点部署方案无关.由于每条时空链路的 p_b 和 p_s 设为相互独立,因此这些备选路径也具有相同的预约失败率,且与存储节点部署方案无关.

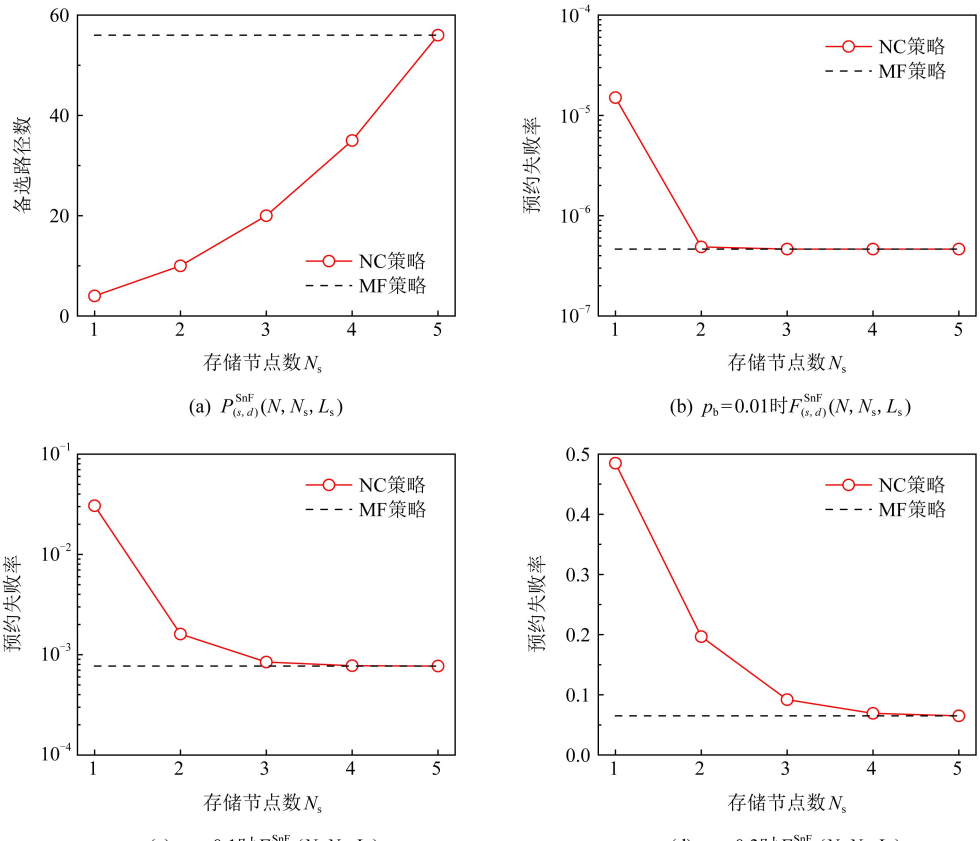
$F^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)$ 定义为请求未能从数量为 $P^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)$ 的备选路径集合中找到任何有效路径的概率.为简单起见,考虑了图 5 所示的存储节点部署方案,由此可得:

$$F^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)=\begin{cases}\prod_{l=1}^{L_s}[1-(1-p_s)^{L_s-1}(1-p_b)\times \\ \quad (1-F^{\mathrm{SnF}}_{(i_1,d)}(N-1,N_s-1,l)), & N_s>1.\\ F^{\mathrm{AR}}_{(s,d)}(N,L_s), & N_s=1.\end{cases} \quad (4)$$

详细证明参见附录 A.

3.2 调度性能和复杂度分析

本节研究 $P^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)$ 和 $F^{\mathrm{SnF}}_{(s,d)}(N,N_s,$



(a) $P^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)$ (b) p_b=0.01时$F^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)$

(c) p_b=0.1时$F^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)$ (d) p_b=0.3时$F^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)$

注:假设 $N=6$, $N_s=L_s=4$ and $p_s=0.01$

Fig. 6 $P^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)$ and $F^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)$

图 6 备选路径数 $P^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)$ 和预约失败率 $F^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)$

L_s)的性质,并对 MF 模型与 NC 模型进行比较研究.为了更好理解 N_s 的影响,引入了 2 个指标:性能比 R_p 和复杂度比 R_c,具体定义为:

$$R_p=F^{SnF}_{(s,d)}(N,L)/F^{SnF}_{(s,d)}(N,N_s,L_s),$$

$$R_c=P^{SnF}_{(s,d)}(N,N_s,L_s)/P^{SnF}_{(s,d)}(N,L).$$

首先研究 $L_s=L$ 情况下 N_s 对 SnF 的影响.假设 $N=6$,$L_s=L=4$,$p_s=0.01$.在图 6(a)中,$P^{SnF}_{(s,d)}(N,N_s,L_s)$随着 N_s 的增加而增加.存储节点越多,NC 模型能提供的备选路径就越多.图 6(b)~(d)表明,$F^{SnF}_{(s,d)}(N,N_s,L_s)$随着 N_s 增加而减小,而且 p_b 越小,$F^{SnF}_{(s,d)}(N,N_s,L_s)$下降的越快.在图 6(b)中,给定 $p_b=0.01$,即使 $N_s=2$,$F^{SnF}_{(s,d)}(N,N_s,L_s)$与 $F^{SnF}_{(s,d)}(N,L)$也非常接近.

表 3 表明,R_p 和 R_c 均随 N_s 增加而增加.随着 p_b 减小,R_p 的增幅更加明显.这表明,获得指定性能所需的 N_s 值随着 p_b 的减小而减小.例如,当 $p_b=0.3$ 时,至少需要 4 个存储节点($N_s=4$)才能实现 $R_p>0.9$.但是,当 p_b 减小到 0.1 时,只需要 3 个存储节点($N_s=3$)就能实现 $R_p>0.9$.同时,当 N_s 从 4 降低到 3 时,R_c 从 0.625 降低到 0.357.与 MF 模型相比,当 $p_b=0.1$ 时,NC 模型仅需要 35.7%的原始复杂度,即可获得超过 90%的原始性能.这意味着仅将部分存储节点纳入调度,不仅可以获得理想的性能,而且能有效减轻调度过程的计算负担.当 $N=10$ 时所得结果与当 $N=6$ 时所得结果的变化趋势相似.

Table 3　R_c and R_p when $L_s=L$ and $p_s=0.01$

表 3　在 $L_s=L$,$p_s=0.01$ 情况下 R_c 和 R_p

N_s	$L_s=L=4$,$N=6$			$L_s=L=4$,$N=10$		
	R_c	R_p		R_c	R_p	
		$p_b=0.1$	$p_b=0.3$		$p_b=0.1$	$p_b=0.3$
1	0.071	0.025	0.134	0.018	0.011	0.198
2	0.178	0.479	0.331	0.045	0.112	0.250
3	0.357	0.911	0.708	0.091	0.527	0.353
4	0.625	0.991	0.941	0.159	0.818	0.522
5	1	1	1	0.255	0.940	0.716

尽管 R_p 和 R_c 均随 N_s 增加,但 R_p 和 R_c 之间存在间距,该间距随 N_s 而变化.因此存储节点的最佳数量应当使得 R_p 最大化而 R_c 最小化.但是,由于 $1\leqslant N_s\leqslant N-1$,所以该间距变化始终受限.本节继续探索如何进一步扩大该间距,突破上述限制的方法.

继续研究 $L_s>L$ 的情况,假设 $N=10$,$L=4$,$p_s=0.01$.本节将研究 L_s 变化如何影响 R_p 和 R_c,结果如表 4 所示.给定 $L_s=L=4$,当 N_s 从 2 增大到 4 时,R_c 从 0.045 增加到 0.159.同时,当 $p_b=0.1$ 时,R_p 从 0.112 增加到 0.818;而当 $p_b=0.3$ 时,R_p 从 0.250 增加到 0.522,如表 4 第 1 行所示.可见,当 $p_b=0.1$ 时,增加 N_s 可以将 R_p 增加到 0.818,但代价是 $R_c=0.159$.相比之下,给定 $L=4$ 且 N_s 保持 2,当 L_s 从 4 增大到 7 时,R_c 从 0.045 增加到 0.127;同时,当 $p_b=0.1$ 时,R_p 从 0.112 增加到 29.117;而当 $p_b=0.3$ 时,R_p 从 0.25 增加到 0.488,如表 4 第 4 列所示.显然,与增大 N_s 相比,增大 L_s 不仅可以提高 R_p,还能保持较低的 R_c.

Table 4　R_c and R_p when $L_s\geqslant L$ and $p_s=0.01$

表 4　在 $L_s\geqslant L$,$p_s=0.01$ 情况下 R_c 和 R_p

L_s	$L=4$,$N_s=2$,$N=10$			$L=4$,$N_s=4$,$N=10$		
	R_c	R_p		R_c	R_p	
		$p_b=0.1$	$p_b=0.3$		$p_b=0.1$	$p_b=0.3$
4	0.045	0.112	0.250	0.159	0.818	0.522
5	0.068	0.652	0.302	0.318	6.755	1.196
6	0.095	4.243	0.378	0.573	51.998	3.250
7	0.127	29.117	0.488	0.955	374.488	9.619

3.3　分析与总结

借助分析模型,本节研究了存储节点数量对于 SnF 调度性能与复杂度的影响,主要发现如下:

1) 在时间维度的调度范围不变的前提下(即 $L_s=L$),仅将传输路径途经的部分节点而非所有节点纳入调度决策,虽然可以减轻调度过程的计算负担,但是调度性能也会随之降低.

2) 当 $p_s\ll p_b$ 时,扩展时间维度的调度范围比使用更多存储节点参与调度更有助于提高调度性能.但是,扩展时间调度范围将导致请求经历更长延迟.

3) 当 $N_s<N$ 和 $L_s>L$ 时,NC 策略的调度性能不仅可能超过 MF 策略,而且能保持较低的复杂度.换而言之,仅将传输路径沿途的部分节点而非所有节点纳入调度决策,有可能同时实现高性能、低复杂度.

4) 设计高效的 SnF 调度方法应当联合优化 N_s 和 L_s,以到达性能与复杂度的最佳折中.

4　节点约束存储转发调度方法

4.1　网络模型与主要假设

本文以波分复用(wavelength-division multiplexing, WDM)网络为基础架构的跨数据中心网络

作为研究场景.借助基于软件定义网络(software-defined networking, SDN)的传输感知优化技术[30],网络运营商可以将光网络基础设施与DC资源有效整合,以实现跨DC的大数据传输.光纤中的带宽资源按照波长通道分配.每个DC的光交换平台均具备OEO转换和波长转换能力.DC可以将大数据缓存于存储集群.存储集群具备绕行企业级防火墙的能力,从而为大数据传输提供高速通道[31-32].如图7所示,当WDM2较为繁忙时,E2E传输难以保障.来自DC1的大数据通过WDM1传输并存储于DC2的存储群集.当WDM2较为空闲时,大数据通过WDM2传输到DC3.假设每个请求占用一个波长进行数据传输.与传输延迟相比,数据传播延迟、网络处理开销(例如光网络重构所需的时间)可以忽略不计[5,33].假设存储集群的数据读/写速度等于单波长的传输容量.文献[34]指出大数据适宜使用专门OCS资源提供传输服务,所以本文假设部分网络资源专用于承载大数据流量.

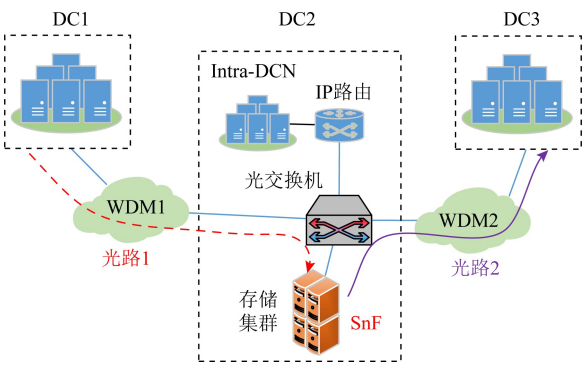


Fig. 7 SnF scheme for inter-DCNs

图7 面向跨数据中心网络的SnF传输方案

4.2 算法研究

受第3节的研究启发,本文将NC调度策略与拓扑抽象融入基于TS-MLG的路由调度,从而实现$N_s<N$与$L_s>L$的组合.由此,本文提出了节点约束SnF(node-constraint SnF, NC-SnF)调度方法.

NC-SnF方法的主要思想源于2方面.首先,在请求到达网络时,NC-SnF方法仅将部分传输路径途经节点纳入调度决策.其次,根据所涉节点,NC-SnF方法通过将TS-MLG中连接这些节点的空间链路合并,并对具有相同网络状态的层压缩,从而实现拓扑抽象.一方面,NC-SnF方法缩小了TS-MLG规模,从而减轻了调度过程的计算负担;另一方面,当可用于路由的层数有限时,相比没有拓扑抽象的传统调度方法,NC-SnF方法可以搜索时间上距离当前时刻更远的层.换言之,给定相同计算复杂度上界,相比于传统调度方法,NC-SnF方法本质上能够为请求提供更大的时间调度范围.因此,NC-SnF方法在降低计算复杂度的同时保持良好的调度性能.

NC-SnF方法分为5个步骤.每个节点对(i,j)均已预先计算K条最短路由,并将备选路由存储在集合$R_{(i,j)}$.假设请求r的源-目的节点对为(s,d).

步骤1. 算法1第②行从集合$R_{(s,d)}$中选择一条从s到d的固定路由R_k,其中$R_k\in R_{(s,d)}$且$|R_k|=N_k$.

步骤2. 算法1第③行根据存储节点选择方案,在R_k上选择的N_s个节点用于SnF调度.令α表示R_k上可用于调度节点数占节点总数的百分比,其中$0<\alpha\leqslant 1$.$N_s=[(N_k-1)\times\alpha]$.集合$S=\{s,I_1,I_2,\cdots,I_i,\cdots,I_{N_s-1},d\}$,其中$I_i$表示被选存储节点,$I_i\in R_k$且$|S|=N_s+1$.集合$S$表示$R_k$被存储节点划分为$N_s$个片段.集合$S$的第1个节点是$s$,因为已有研究表明,在多数调度过程中,源节点是首个发生SnF的节点[16,24-25].为简单起见,本文假设其他被选节点等间隔分布于R_k.网络级存储节点选择或部署方案是一个有趣但复杂的问题,值得在未来深入研究.

步骤3. 算法1第④行根据R_k将原始TS-MLG图G缩减为规模较小的图G'.具体而言,图G'仅包含R_k中的节点以及连接这些节点的链路.

步骤4. 算法1第⑤行根据集合S运行算法2对图G'进行拓扑抽象,以获得抽象压缩后的子图G''.

步骤5. 算法1第⑥~⑪行在图G''上运行广度优先搜索(breadth-first search, BFS)算法,寻找到有效路径,即$Path$.如果BFS算法未能在R_k上找到任何有效路径,则NC-SnF方法将使用下一条路由R_{k+1}重新运行算法.如果在K条预选路由上都未找到有效路径,则请求r将被阻塞.

算法1. NC-SnF调度方法.

输入:$r=\{s,d\}$,图G,K,$R_{(s,d)}$,α;

输出:$Path$和$Find$.

① for $k=1;k\leqslant K;k++$ do

② 选择一条从s到d的路由R_k,其中$R_k\in R_{(s,d)}$且$|R_k|=N_k$;

③ 根据存储节点选择方案在R_k中选择N_s个节点,得到路径片段集合$S=\{s,I_1,I_2,\cdots,I_i,\cdots,I_{N_s-1},d\}$,其中$|S|=N_s+1$,$I_i\in R_k$和$N_s=[(N_k-1)\alpha]$;

④ 根据R_k将图G缩减为其子图G';

⑤　运行算法 2,根据集合 S 对图 G' 进行拓扑抽象,并获得抽象压缩后子图 G'';

⑥　在图 G'' 上运行 BFS 算法,寻找有效路径 $Path$;

⑦　if 找到 $Path$ then

⑧　　return $Path$ 和 $Find$=True;

⑨　end if

⑩ end for

⑪ 未找到任何有效路径,return $Find$=False.

算法 2 的主要思想是根据集合 S 实现拓扑抽象.该算法主要包括 3 个步骤:

步骤 1. 算法 2 第①行创建 1 个辅助图 $G''=(V'',E'')$,其中 $V''=S$ 和 $E''=\varnothing$.

步骤 2. 算法 2 第②～⑧行将图 G' 中每个路径片段内的空间链路状态合并为图 G'' 中的逻辑链路.L_R 表示用于路由的层数.L_{L_R} 表示这些层的集合,$L_{L_R}=\{l_1,l_2,\cdots,l_j,\cdots,l_{L_R}\}$.对于图 G' 中第 l_j 层的每个路径片段 $\{I_i,I_{i+1}\}$,算法 2 第④～⑥行找到在 $\{I_i,I_{i+1}\}$ 内具有最小剩余带宽的空间链路 e_i,将 e_i 添加到逻辑链路 $\langle I_i,I_{i+1}\rangle$;同时将图 G' 中属于节点 I_i 和节点 I_{i+1} 的时间链路添加到图 G''.

步骤 3. 算法 2 第⑨～⑭行将图 G' 中的冗余层压缩.具体而言,如果 TS-MLG 中相邻 2 层的网络状态相同,则存在冗余状态,可将其压缩为一层.

算法 2. 基于存储节点的拓扑抽象算法.

输入:图 G',L_R,S;

输出:图 G''.

① 创建辅助图 $G''=(V'',E'')$,其中 $V''=S$ 和 $E''=\varnothing$;

② for 图 G' 中所有层 $L_{L_R}=\{l_1,l_2,\cdots,l_j,\cdots,l_{L_R}\}$ do

③　for 图 G' 中第 l_j 层的每一个路径片段 $\{I_i,I_{i+1}\}$ do

④　　在图 G' 中第 l_j 层中找到从节点 I_i 到节点 I_{i+1} 具有最小剩余带宽的空间链路 e_i;

⑤　　将 e_i 添加到图 G'' 中第 l_j 层的逻辑链路 $\langle I_i,I_{i+1}\rangle$;

⑥　　将连接图 G' 中的节点 I_i 和节点 I_{i+1} 的时间链路添加到图 G'';

⑦　end for

⑧ end for

⑨ for 图 G'' 中的所有层 $L_{L_R}=\{l_1,l_2,\cdots,l_j,\cdots,l_{L_R}\}$ do

⑩　if $l_{j+1}==l_j$ then

⑪　　从图 G'' 移除层 l_{j+1};

⑫　end if

⑬ end for

⑭ return 图 G''.

4.3　算法运行示例

本节将通过图 8 和图 9 展示 NC-SnF 方法是如何减小 TS-MLG 的规模.原始 TS-MLG 图 G 如图 2 所示.图 G 由多层组成,每层包含 6 个节点.假设请求 r 需要从节点 A 向节点 D 传输数据.路由 $R_k=\{A,B,C,D\}$ 被用于请求 r 的传输.首先,根据 R_k 将图 G 缩减为子图 G',如图 8(a)所示.图 G' 由 4 层组成,每层包含 4 个节点.假设 $\alpha=0.4$,可得 $N_s=2$.根据 4.2 节所述存储节点选择方案,选择节点 A 与节点 C 参与调度,即 $S=\{A,C,D\}$.因此,省略连接节点 B 的时间链路,如图 8(b)所示.

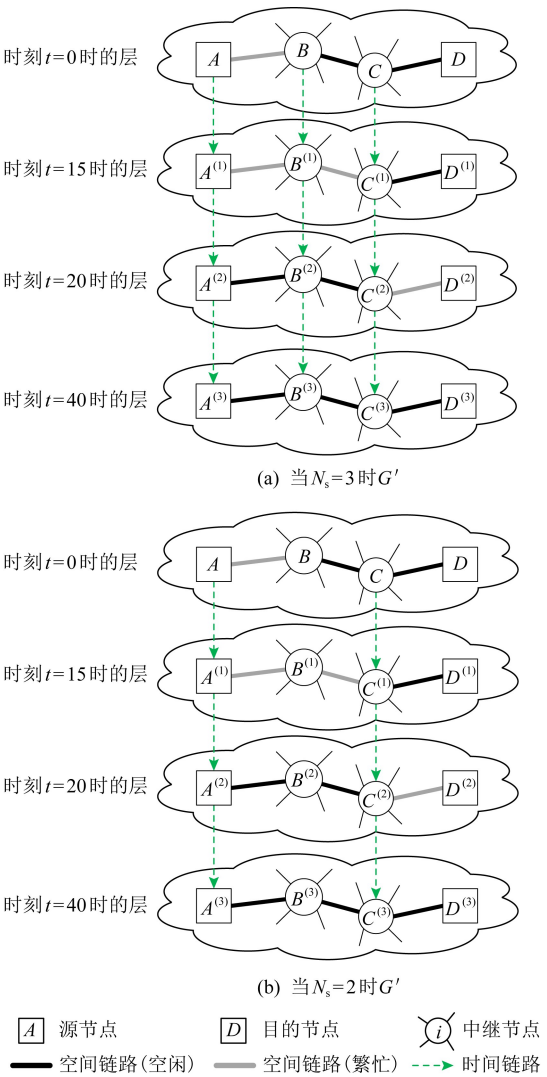


Fig. 8　Comparison of the reduced graph G' with different N_s

图 8　不同 N_s 下简化图 G' 比较

随后,运行算法 2 合并节点 A 和节点 C 之间的空间链路,生成逻辑链路〈A,C〉.合并图 G'' 由 4 层组成,每层包含 3 个节点,如图 9(a)所示.图 9(a)中每一层逻辑链路〈A,C〉表示在该时刻从节点 A 到节点 C 的最小剩余带宽.例如,在图 8(b)中,空间链路 A—B 在 $t=0$ 时剩余带宽为 0,而空间链路 B—C 在 $t=0$ 时剩余带宽为 1.因此,在图 9(a)中,逻辑链路〈A,C〉的带宽在 $t=0$ 时为 0.

在图 9(a)中,图 G' 中 $t=0$ 和 $t=15$ 的层表示相同的网络状态,即逻辑链路〈A,C〉繁忙.这些层不仅无法提供更多有用信息,还会给搜索带来额外的计算负担,因此是冗余层.算法 2 将 $t=15$ 的层移除,得到压缩图 G'',如图 9(b)所示.显然,通过运行 NC-SnF 方法,可以极大减小 TS-MLG 的规模.

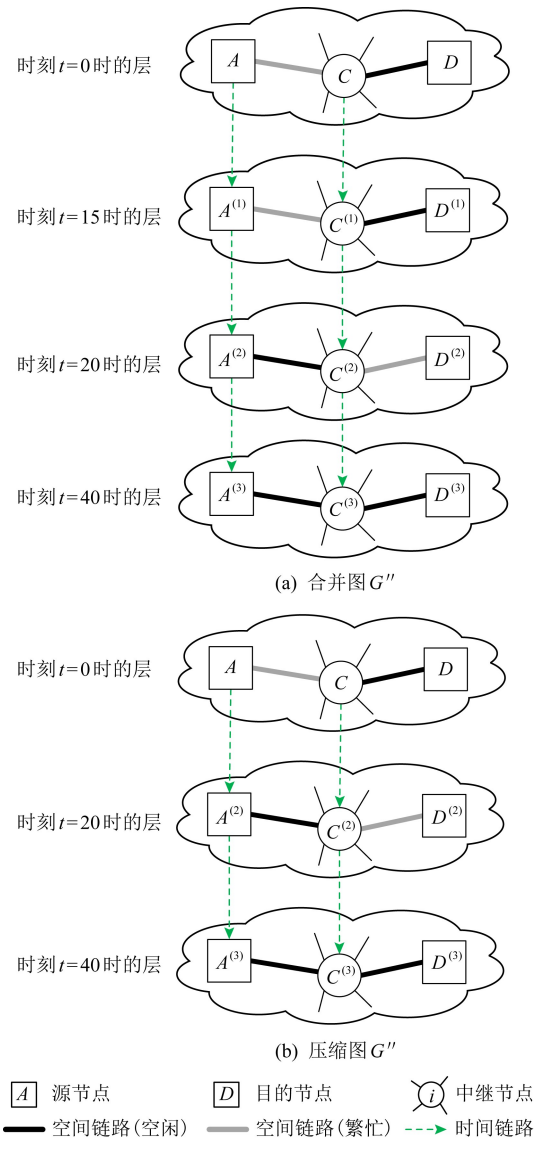


Fig. 9 The merged graph G'' vs the condensed graph G''

图 9 合并图 G'' 和压缩图 G'' 比较

4.4 资源预约窗口研究

为了限制计算复杂度,请求只能在给定层数(即 L_R)内搜索有效路径.L_R 的值决定了时间维度的调度搜索范围.因此,算法 2 合并、压缩链路与层之后,时间调度范围会相应地发生改变.为了对此展开研究,资源预约窗口定义为最顶层与第 L_R 层(即可以用于路由的最后一层)之间的时间间隔.资源预约窗口的大小与 L_R、层与层之间的时间间隔均有关.

本节比较了图 8(a)所示的图 G' 的资源预约窗口与图 9(b)所示压缩后 G'' 的资源预约窗口.假设 $L_R=3$.图 10 展示了不同图的资源预约窗口.在图 10(a)中,G' 的资源预约窗口是最顶层和第 3 层之间的时间间隔,即 $t=0$ 和 $t=20$ 的层间距.因此,图 10(a)中的窗口大小为 20.在图 10(b)中,G'' 的资源预约窗口同样也是最顶层和第 3 层之间的时间间隔.

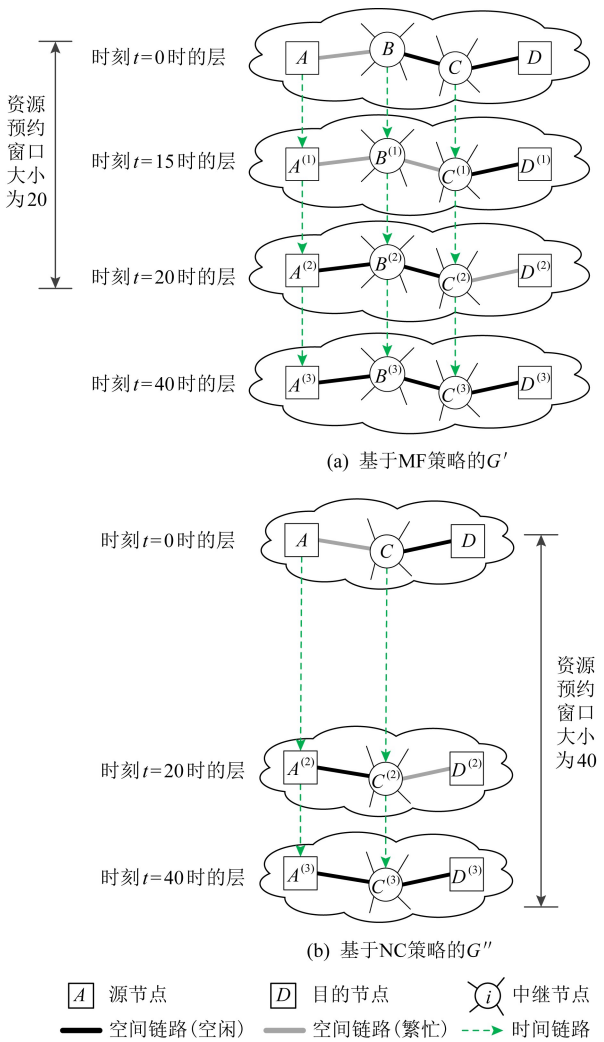


Fig. 10 Comparison of the reservation windows in the graphs with different scheduling strategies ($L_R=3$)

图 10 不同策略下资源预约窗口比较($L_R=3$)

但是,由于 $t=15$ 的层被移除,第 3 层不再是 $t=20$ 的层,而是 $t=40$ 的层.因此,图 10(b)的窗口大小为 40.假设请求 r 在 $t=0$ 到达网络,需要从节点 A 向节点 D 传输数据.请求 r 无法在图 10(a)所示的窗口内找到有效路径,因此 r 被阻塞.相反,在图 10(b)所示的窗口内,有效路径 $A—A^{(2)}—C^{(2)}—C^{(3)}—D^{(3)}$ 可用于传输请求 r.

简而言之,给定相同计算复杂度限制,与传统调度方法相比,NC-SnF 方法能够提供更大的资源预约窗口,因此有助于降低请求阻塞率.

4.5　计算复杂度研究

TS-MLG 的规模决定了路由算法的计算复杂度.文献[16]给出的复杂度为 $O((V\times L_R)^2)$,V 表示网络拓扑中的节点总数.NC-SnF 方法使用 BFS 算法对压缩后的 TS-MLG 图 G''进行有效路径搜索.BFS 算法的复杂度为 $O(V''+E'')$,V''表示图 G''的总节点数,E''表示图 G''的总边数.在最坏的情况下,$V''=L_R\times N_s$;而 $E''=(L_R-1)\times N_s+(N_s-1)\times L_R$.因此,NC-SnF 方法的复杂度为 $O(K\times L_R\times N_s)$.

5　结果与讨论

本节模拟动态网络环境,比较 NC-SnF 方法与传统的 SnF 调度方法[12](即 MF-SnF 方法).

本节采用了 4.1 节的主要假设,并在研究中使用美国国家科学基金会网络(National Science Foundation Network, NSFNET)拓扑.为简单起见,放宽了存储容量限制.在调度过程中不考虑存储容量制约,网络调度器仅根据请求是否能在给定层数(即 L_R)内找到有效路径,决定请求是否被接受.假设请求到达是独立的,并在所有节点对之中均匀分布;请求的到达服从到达率为 λ 的泊松分布;请求的持续时间服从服务率为 μ 的负指数分布.网络负载 $\rho=\lambda/\mu$.链路容量,即每条链路的波长数,用 w 表示.每个节点对之间 3 条最短备用路由,即 $K=3$.传输路径中参与调度的节点数百分比为 α.当 $\alpha=1$ 时,传输路径中的所有节点均参与调度.此时,NC-SnF 方法等效于 MF-SnF 方法.为简单起见,假设 $\alpha=0.4$ 和 $\alpha=0.6$.每次仿真实验产生 500 000 个请求,独立重复 20 次并取平均值.

5.1　网络性能研究

本节首先研究阻塞率(blocking probability)如何随 ρ 增大而改变.可以通过增加 λ 或减小 μ 来增大 ρ.由于 2 种情况下所得结果相似,因此在以下仿真实验中 $\lambda=1$,通过改变 μ 来改变 ρ.

仿真结果如图 11 所示.当 ρ 从 10 增加到 60 时,阻塞率增加.在图 11(a)中,给定 $w=4$,$L_R=4$,当 $\rho=10$ 时,NC-SnF 方法所得阻塞率为 0,而 MF-SnF 方法所得阻塞率为 5.05×10^{-6}.随着 ρ 的增加,NC-SnF 方法开始出现阻塞.α 值越大,阻塞率的增幅越显著.图 11(b)中结果遵循相似的趋势,但是请求阻塞出现在较大的 ρ 值上,这是由于图 11(b)中的 w 比图 11(a)中的 w 大.

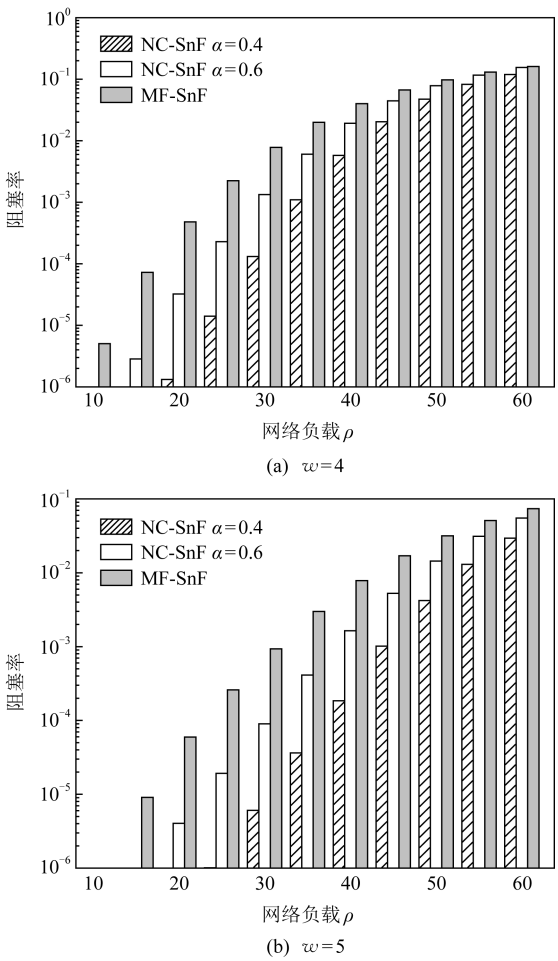


Fig. 11　Blocking probability under various ρ ($L_R=4$)

图 11　不同网络负载 ρ 下的阻塞率($L_R=4$)

直观上,存储节点数越少,阻塞率越高.然而,图 11 却显示相反的结果,即存储节点数越少,NC-SnF 方法获得的阻塞率反而越低.为了理解该结果,本节将研究重点放在图 11(a),研究其他网络性能指标如何随 ρ 变化,结果如图 12 所示.活跃请求(active request)定义为已被网络接受但尚未完成传输的请求.请求缓存率(ratio of stored requests)定义为缓存请求数与请求总数的比率.延迟定义为从请求到达网络直到请求结束传输的时间间隔.

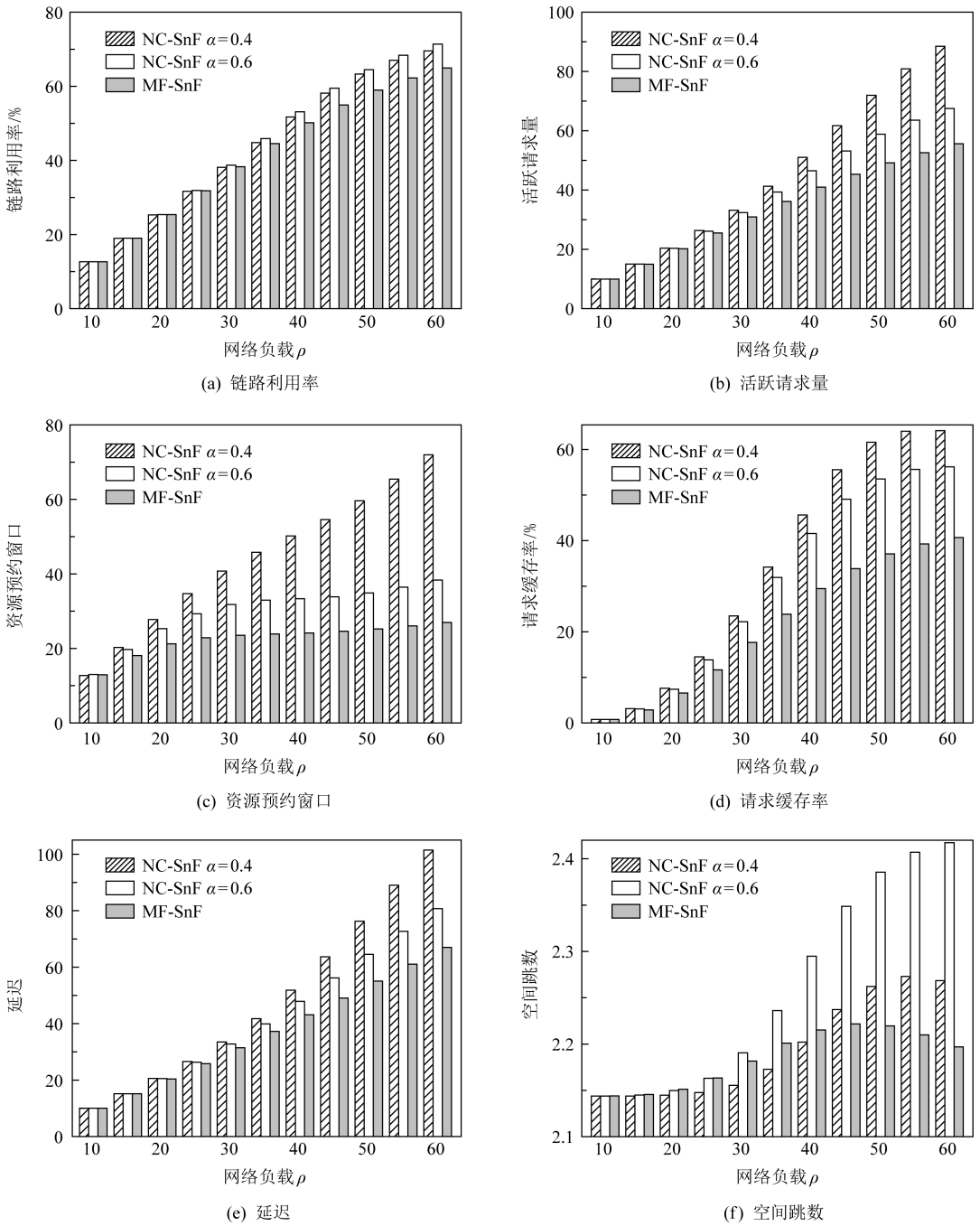


Fig. 12 Network performance under various ρ ($w=4$, $L_R=4$)

图 12 不同网络负载 ρ 下的网络性能($w=4$, $L_R=4$)

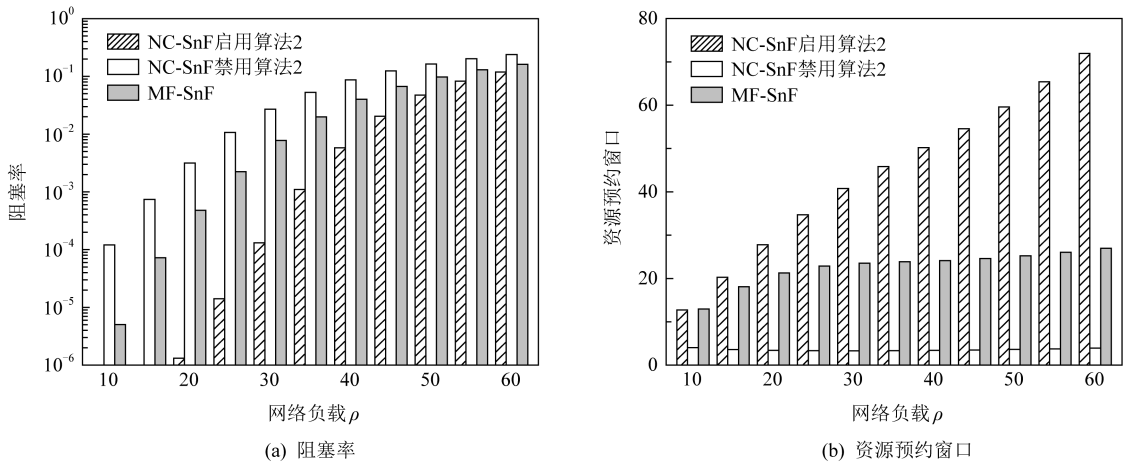
在图 12(a)中,当 ρ 从 10 增加到 30 时,NC-SnF 方法和 MF-SnF 方法的链路利用率相似.当 ρ 超过 35 时,$\alpha=0.6$ 的 NC-SnF 方法所得链路利用率明显高于其他方法,而 MF-SnF 方法的链路利用率低于其他方法.相反,在图 12(b)中,当 $\rho>30$ 时,$\alpha=0.4$ 的 NC-SnF 方法的活跃请求量最多.这表明,当网络负载为中等或更高时,采用 $\alpha=0.4$ 的 NC-SnF 方法,网络可以同时容纳更多请求.α 值越小,网络中容纳的请求越多.这是因为 $\alpha=0.4$ 的 NC-SnF 方法能比其他方法提供更大的资源预约窗口,如图 12(c)所示.α 值越小,存储节点越少,合并的空间链路就越多.这也增加了在 TS-MLG 中找到冗余层的机会.给定 L_R 值,在压缩更多冗余层的情况下,层与层的时间间距变大,NC-SnF 方法也就可以在更大的时间范围内搜索可用资源.因此,α 的值越小,越多请求可以通过 SnF 到达目的节点,所以请求缓存

率也越高,如图 12(d)所示.简言之,得益于拓扑抽象,请求使用 NC-SnF 方法比使用 MF-SnF 方法更容易被传输.然而,随着资源预约窗口增大,请求也将经历更长的延迟,如图 12(e)所示.

随着资源预约窗口的扩大,更多的请求可以通过 SnF 选择更短的路由,而不是通过较长的路由绕行.图 12(f)表明成功传输请求的平均空间跳数.与 $\alpha=0.6$ 相比,$\alpha=0.4$ 的 NC-SnF 方法产生的请求空间跳数更短.当 $\rho\in[45,60]$时,与 NC-SnF 方法相比,MF-SnF 方法产生的请求空间跳数更短.这是因为 MF-SnF 方法的阻塞率高于 NC-SnF 方法.对于需要通过较长的路由绕行才能完成传输的请求,MF-SnF 方法难以满足这些请求的需求.随着这类请求被阻塞,MF-SnF 方法产生的平均空跳明显短于 NC-SnF 方法.

5.2　基于存储节点的拓扑抽象算法研究

得益于拓扑抽象算法(即算法 2),空间链路和冗余层被合并和压缩,使得 NC-SnF 方法能够提供更大的资源预约窗口.为了验证算法 2 的作用,本节比较了原始 NC-SnF 方法和没有拓扑抽象功能的 NC-SnF 方法(即简化 NC-SnF 方法).在简化 NC-SnF 方法中,算法 2 被禁用.因此,BFS 算法直接对简化图 G'而非压缩图 G''进行有效路径搜索.此处,$w=4$,$L_R=4$.仿真结果如图 13 所示:



(a) 阻塞率　(b) 资源预约窗口

Fig. 13　The original NC-SnF method vs the NC-SnF method disabling the topology abstraction ($\alpha=0.4$)

图 13　原始 NC-SnF 方法与禁用拓扑抽象功能的 NC-SnF 方法比较($\alpha=0.4$)

给定 $\alpha=0.4$,原始 NC-SnF 方法的阻塞率优于简化 NC-SnF 方法,如图 13(a)所示.这是因为原始 NC-SnF 方法中的资源预约窗口随 ρ 增大而增大,而简化 NC-SnF 方法中的资源预约窗口几乎保持恒定,如图 13(b)所示.

随后,继续对比简化 NC-SnF 方法与 MF-SnF 方法,如图 13(a)所示.MF-SnF 方法优于简化 NC-SnF 方法.这是因为一方面没有了拓扑抽象功能,简化 NC-SnF 方法的资源预约窗口明显小于 MF-SnF 方法;另一方面 MF-SnF 方法能够比简化 NC-SnF 方法使用更多存储节点,因此 MF-SnF 方法的调度更加灵活.

5.3　算法计算时间研究

本节研究了 TS-MLG 的节点数和层数是如何影响 NC-SnF 方法的算法计算时间.本节使用链路密度为 0.6 的随机生成拓扑进行仿真实验.链路密度定义为网络拓扑中任意 2 个节点之间存在边的概率.V 表示网络拓扑的节点数.

表 5 和表 6 描绘了不同调度方法对给定的 TS-MLG 进行一次路由搜索的平均计算时间.在表 5 中,

Table 5　Computation Time Under Various V ($L_R=10$)

表 5　不同拓扑节点数 V 下算法计算时间($L_R=10$)

调度方法	计算时间/ms		
	$V=10$	$V=50$	$V=100$
NC-SnF $\alpha=0.4$	0.79	2.14	6.27
NC-SnF $\alpha=0.6$	0.92	3.22	11.54
MF-SnF	0.98	5.53	26.38

Table 6　Computation Time Under Various L_R ($V=10$)

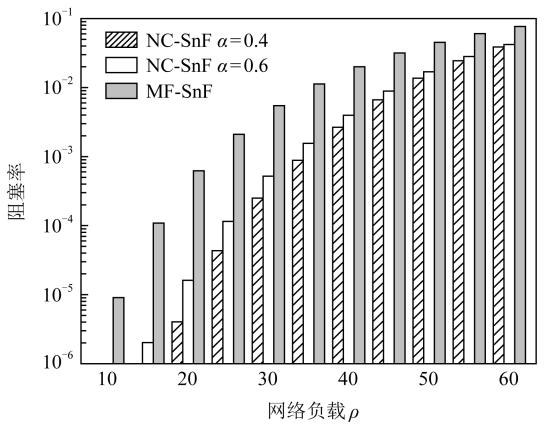
表 6　不同层数 L_R 下算法计算时间($V=10$)

调度方法	计算时间/ms		
	$L_R=10$	$L_R=50$	$L_R=100$
NC-SnF $\alpha=0.4$	0.79	8.26	47.45
NC-SnF $\alpha=0.6$	0.92	15.46	110.97
MF-SnF	0.98	24.72	196.91

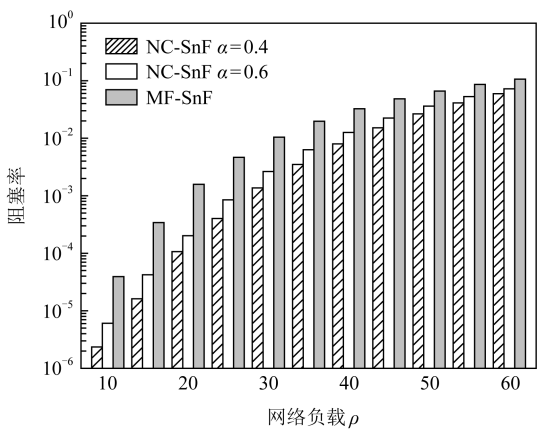
$L_R=10$.计算时间随 V 的增加而增加.在表 6 中,$V=10$.计算时间随着 L_R 的增加而增加.表 5 和表 6 表明,α 值越小,计算时间的增幅越小.这是因为当 α 值较小时,需要搜索的 TS-MLG 规模也相应减小.

5.4 不同网络拓扑研究

本节继续研究了不同的网络拓扑中阻塞率随 ρ 的变化情况.选取 19 个节点 39 条链路的泛欧洲全光网(optical pan-european network, OPEN)和 24 个节点 43 条链路的美国骨干网络(US backbone network, USNET)作为研究对象.此处,$w=4$,$L_R=4$.图 14(a)(b)分别描述了在 OPEN 和 USNET 中不同 ρ 下的阻塞率.图 14 的结果与图 11 的结果类似.因此,在不同的网络拓扑中,NC-SnF 方法的阻塞性能均优于 MF-SnF 方法.然而,图 14 中 NC-SnF 调度方法获得的阻塞率略高于图 11.例如,当 $\alpha=0.4$ 和 $\rho=20$ 时,在 NSFNET 中 NC-SnF 方法获得的阻塞率为 1.32×10^{-6};而在 OPEN 和 USNET 中 NC-SnF 方法获得的阻塞率分别为 4.04×10^{-6} 和 1.07×10^{-4}.这是因为 OPEN 和 USNET 的拓扑规模大于 NSFNET.因此,OPEN 和 USNET 中给定节点对之间的路由跳数多于 NSFNET.给定相同的 α,NC-SnF 方法在 OPEN 和 USNET 中每条路由上选定的存储节点数多于 NSFNET.根据 4.3 节讨论可知,借助拓扑抽象算法,选定的存储节点数越少,所获得的压缩子图规模就越小,相应的资源预约窗口也就越大.因此,相比 OPEN 和 USNET,NSFNET 中 NC-SnF 方法可获得更大的资源预约窗口,进而获得更低的阻塞率.



(a) OPEN



(b) USNET

Fig. 14 Blocking probability in different topologies

图 14 不同网络拓扑下的阻塞率

6 总 结

本文将 SnF 与 IR 和 AR 进行了对比.研究表明,选择合适数量的存储节点用于调度决策,实际上是调度性能与复杂度之间的折中问题.为此,本文提出了分析模型,揭示了存储节点数对 SnF 复杂度与性能的影响.研究发现,在一定条件下,NC 调度策略能够在降低复杂度的同时获得比传统 MF 调度策略更好的调度性能.

受此启发,本文提出 NC-SnF 调度方法,只将传输路径上的部分节点纳入调度决策.同时,NC-SnF 方法引入了基于存储节点的拓扑抽象.与传统的 MF-SnF 方法相比,给定相同的计算复杂度界限,NC-SnF 方法可以获得更大的时间调度范围.仿真结果表明,与 MF-SnF 方法相比,NC-SnF 方法所需的计算时间更短、获得的阻塞率更低.

尽管本论文的研究工作围绕基于 OCS 的跨数据中心网络开展,但是研究结果对于虚电路交换网络、带宽可管理的分组交换网络等类似网络同样具有借鉴意义.本文的理论分析模型主要针对固定路由场景,而继续将研究场景扩展到通用网络场景,探索网络级存储选择与部署方案,是未来研究方向.

参 考 文 献

[1] Lu Xingjian, Kong Fanxin, Liu Xue, et al. Bulk savings for bulk transfers: Minimizing the energy-cost for geo-distributed data centers [J]. IEEE Transactions on Cloud Computing, 2020, 8(1): 73-85

[2] Zeng Gaoxiong, Hu Shuihai, Zhang Junxue, et al. Transport protocols for data center networks: A survey [J]. Journal of Computer Research and Development, 2020, 57(1): 74-84 (in Chinese)
(曾高雄, 胡水海, 张骏雪, 等. 数据中心网络传输协议综述 [J]. 计算机研究与发展, 2020, 57(1): 74-84)

[3] Garcia-Dorado J L, Rao S G. Cost-awaremulti data-center bulk transfers in the cloud from a customer-side perspective [J]. IEEE Transactions on Cloud Computing, 2019, 7(1): 34-47

[4] Luo Long, Kong Yijing, Noormohammadpour M, et al. Deadline-aware fast one-to-many bulk transfers over inter-datacenter networks [J/OL]. IEEE Transactions on Cloud Computing, 2019 [2020-06-07]. https://doi.org/10.1109/TCC.2019.2935435

[5] Yang Zhenjie, Cui Yong, Wang Xin, et al. Cost-efficient scheduling of bulk transfers in inter-datacenter WANs [J]. IEEE/ACM Transactions on Networking, 2019, 27(5): 1973-1986

[6] Hu Zhiyao, Li Dongsheng, Li Ziyang. Recent advances in datacenter flow scheduling [J]. Journal of Computer Research and Development, 2018, 55(9): 1920-1930 (in Chinese)
(胡智尧, 李东升, 李紫阳. 数据中心网络流调度技术前沿进展[J]. 计算机研究与发展, 2018, 55(9): 1920-1930)

[7] Li Wenxin, Zhou Xiaobo, Li Keqiu, et al. Trafficshaper: Shaping interdatacenter traffic to reduce the transmission cost [J]. IEEE/ACM Transactions on Networking, 2018, 26(3): 1193-1206

[8] Noormohammadpour M, Raghavendra C S, Kandula S, et al. Quickcast: Fast and efficient inter-datacenter transfers using forwarding tree cohorts [C] //Proc of IEEE Conf on Computer Communications (INFOCOM). Piscataway, NJ: IEEE, 2018: 225-233

[9] Hu Wenbo, Liu Jiang, Huang Tao, et al. A completion time-based flow scheduling for inter-data center traffic optimization [J]. IEEE Access, 2018,6: 26181-26193

[10] Wu Yu, Zhang Zhizhong, Wu Chuan, et al. Orchestrating bulk data transfers across geo-distributed datacenters [J]. IEEE Transactions on Cloud Computing, 2017, 5(1): 112-125

[11] Lu Ping, Zhu Zuqing. Data-oriented task scheduling in fixed- and flexible-grid multilayer inter-DC optical networks: A comparison study [J]. Journal of Lightwave Technology, 2017, 35(24): 5335-5346

[12] Sun Chao, Guo Wei, Liu Zhe, et al. Performance analysis of storage-based routing for circuit-switched networks [J]. Journal of Optical Communications and Networking, 2016, 8(5): 282-289

[13] Wang Yiwen, Su Sen, Liu A X, et al. Multiple bulk data transfers scheduling among datacenters [J]. Computer Networks, 2014, 68: 123-137

[14] Lin Xiao, Wang Xiaoyu, Yue Shengnan, et al. Design of an SnF scheduling method for bulk data transfers over inter-datacenter WANs [C] //Proc of the 20th Int Conf on High Performance Switching and Routing (IEEE HPSR). Piscataway, NJ: IEEE, 2019: 1-8

[15] Lin Xiao, Sun Weiqiang, Wang Xiaoyu, et al. Time-space decoupled SnF scheduling of bulk transfers across inter-datacenter optical networks [J]. IEEE Access, 2020, 8: 24829-24846

[16] Lin Xiao, Sun Weiqiang, Veeraraghavan M, et al. Time-shifted multilayer graph: A routing framework for bulk data transfer in optical circuit-switched networks with assistive storage [J]. IEEE/OSA Journal of Optical Communications and Networking, 2016, 8(3): 162-174

[17] Patel A, Tacca M, Jue J P. Time-shift circuit switching [C] //Proc of Optical Fiber Communication Conf/National Fiber Optic Engineers Conf (OFC/NFOEC). Washington, DC: OSA, 2008: 1841-1843

[18] Laoutaris N, Sirivianos M, Yang Xiaoyuan, et al. Inter-datacenter bulk transfers with NetStitcher [J]. ACM SIGCOMM Computer Communication Review, 2011, 41(4): 74-85

[19] Li Yangyang, Wang Hongbo, Zhang Peng, et al. D4D: Inter-datacenter bulk transfers with ISP friendliness [C] //Proc of Int Conf on Cluster Computing. Piscataway, NJ: IEEE, 2012: 597-600

[20] Feng Yuan, Li Baochun, Li Bo. Postcard: Minimizing costs on inter-datacenter traffic with store-and-forward [C] //Proc of Int Conf on Distributed Computing Systems Workshops. Piscataway, NJ: IEEE, 2012: 43-50

[21] Chhabra P, Erramilli V, Laoutaris N, et al. Algorithms for constrained bulk-transfer of delay-tolerant data [C] //Proc of Int Conf on Communications. Piscataway, NJ: IEEE, 2010: 1-5

[22] Feng Da, Sun Weiqiang, Hu Weisheng. Joint provisioning of lightpaths and storage in store-and-transfer wavelength-division multiplexing networks [J]. IEEE/OSA Journal of Optical Communications and Networking, 2017, 9(3): 218-233

[23] Feng Da, Sun Weiqiang, Zhang Xiaojian, et al. Dimensioning of the store-and-transfer WDM network with limited node storage under the sliding scheduled traffic model [J]. IEEE/OSA Journal of Optical Communications and Networking, 2017, 9(4): 275-290

[24] Patel A N, Zhu Yi, Jue J P. Routing and horizon scheduling for time-shift advance reservation [C] //Proc of Optical Fiber Communication Conf and National Fiber Optic Engineers Conf (OFC/NFOEC). Washington, DC: OSA, 2009: 1632-1634

[25] Patel A N, Zhu Yi, She Qingya, et al. Routing and scheduling for time-shift advance reservation [C] //Proc of the 18th Int Conf on Computer Communications and Networks (ICCCN). Piscataway, NJ: IEEE, 2009: 1-6

[26] Laoutaris N, Smaragdakis G, Stanojevic R, et al. Delay-tolerant bulk data transfers on the Internet [J]. IEEE/ACM Transactions on Networking, 2013, 21(6): 1852-1865

[27] Lee C, Rhee J K. Efficient design and scalable control for store-and-forward capable optical transport networks [J]. Journal of Optical Communications and Networking, 2017, 9(8): 699-710

[28] Iosifidis G, Koutsopoulos I, Smaragdakis G. Distributed storage control algorithms for dynamic networks [J]. IEEE/ACM Transactions on Networking, 2017, 25(3): 1359-1372

[29] Charbonneau N, Vokkarane V M. A survey of advance reservation routing and wavelength assignment in wavelength-routed WDM networks [J]. IEEE Communications Surveys and Tutorials, 2012, 14(4): 1037-1064

[30] He Rongxi, Lei Tianying, Lin Ziwei. Multi-constrained energy-saving routing algorithm in software-defined data center networks [J]. Journal of Computer Research and Development, 2019, 56(6): 1219-1230(in Chinese)
(何荣希, 雷田颖, 林子薇. 软件定义数据中心网络多约束节能路由算法[J]. 计算机研究与发展, 2019, 56(6): 1219-1230)

[31] Dart E, Rotman L, Tierney B, et al. The science DMZ: A network design pattern for data-intensive science [J]. Scientific Programming, 2014, 22(2): 173-185

[32] Wang Xiaoyu, Lin Xiao, Sun Weiqiang, et al. Comparison of two sharing modes for a proposed optical enterprise-access SDN architecture [C] //Proc of the 28th Int Telecommunication Networks and Applications Conf (ITNAC). Piscataway, NJ: IEEE, 2018: 427-434

[33] Yao Jingjing, Lu Ping, Gong Long, et al. On fast and coordinated data backup in geo-distributed optical inter-datacenter networks [J]. Lightwave Technology, 2015, 33(14): 3005-3015

[34] Castillo C, Rouskas G N, Harfoush K. On the design of online scheduling algorithms for advance reservations and QoS in grids [C] //Proc of Int Parallel and Distributed Processing Symp. Piscataway, NJ: IEEE, 2007: 1-10

Lin Xiao, born in 1988. PhD, assistant professor. His main research interests include big data network, intelligent optical network and edge computing.

林 霄, 1988 年生. 博士, 助理教授. 主要研究方向为大数据网络、智能光网络和边缘计算.

Ji Shuo, born in 1997. Master candidate. Her main research interests include big data network and intelligent optical network. (201127035@fzu.edu.cn)

姬 硕, 1997 年生. 硕士研究生. 主要研究方向为大数据网络、智能光网络.

Yue Shengnan, born in 1994. PhD candidate. Her main research interests include bulk data transfer and delay-tolerant networks. (shnyue@sjtu.edu.cn)

岳胜男, 1994 年生. 博士研究生. 主要研究方向为大数据传输和延迟容忍网络.

Sun Weiqiang, born in 1976. PhD, professor, PhD supervisor. His main research interests include big data network, information communication network, network optimization, and network performance evaluation. (sunwq@sjtu.edu.cn)

孙卫强, 1976 年生. 博士, 教授, 博士生导师. 主要研究方向为大数据网络、信息通信网络、网络优化和网络性能评估.

Hu Weisheng, born in 1957. PhD, professor, PhD supervisor. His main research interests include the next generation optical access network, optical switching, and optical network. (wshu@sjtu.edu.cn)

胡卫生, 1957 年生. 博士, 教授, 博士生导师. 主要研究方向为下一代光接入网、光交换和光网络.

附录 A

当 $N_s>1$ 时, $F^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)$ 的 TS-MLG 可以分解为 L_s 个不同的 $F^{\mathrm{SnF}}_{(i_1,d)}(N-1,N_s-1,l)$, 其中 $l\in[1,L_s]$. 在第 l 种情况中, 所有备选路径在到达节点 i_1 之前都要经过 L_s-l 条时间链路和一条空间链路. 随后, 这些备选路径将在 $F^{\mathrm{SnF}}_{(i_1,d)}(N-1,N_s-1,l)$ 的 TS-MLG 中继续分散, 如图 A1 所示. L_s-l 条时间链路、1 条空间链路和 $F^{\mathrm{SnF}}_{(i_1,d)}(N-1,N_s-1,l)$ 的 TS-MLG 属于串联关系. 因此, 第 l 种情况的预约失败概率为 $1-(1-p_s)^{L_s-l}(1-p_b)\times(1-F^{\mathrm{SnF}}_{(i_1,d)}(N-1,N_s-1,l))$. 假设这 L_s 种情况都是互相独立的, 即可得到 $F^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)$ 的下界, 因此有 $F^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)=\prod_{l=1}^{l=L_s}[1-(1-p_s)^{L_s-l}\times(1-p_b)(1-F^{\mathrm{SnF}}_{(i_1,d)}(N-1,N_s-1,l))]$.

当 $N_s=1$ 时, SnF 等价于 AR. 因此 $F^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)$ 的 TS-MLG 相当于图 3(b)中的 TS-MLG, 由此可得 $F^{\mathrm{SnF}}_{(s,d)}(N,N_s,L_s)=F^{\mathrm{AR}}_{(s,d)}(N,L_s)$.

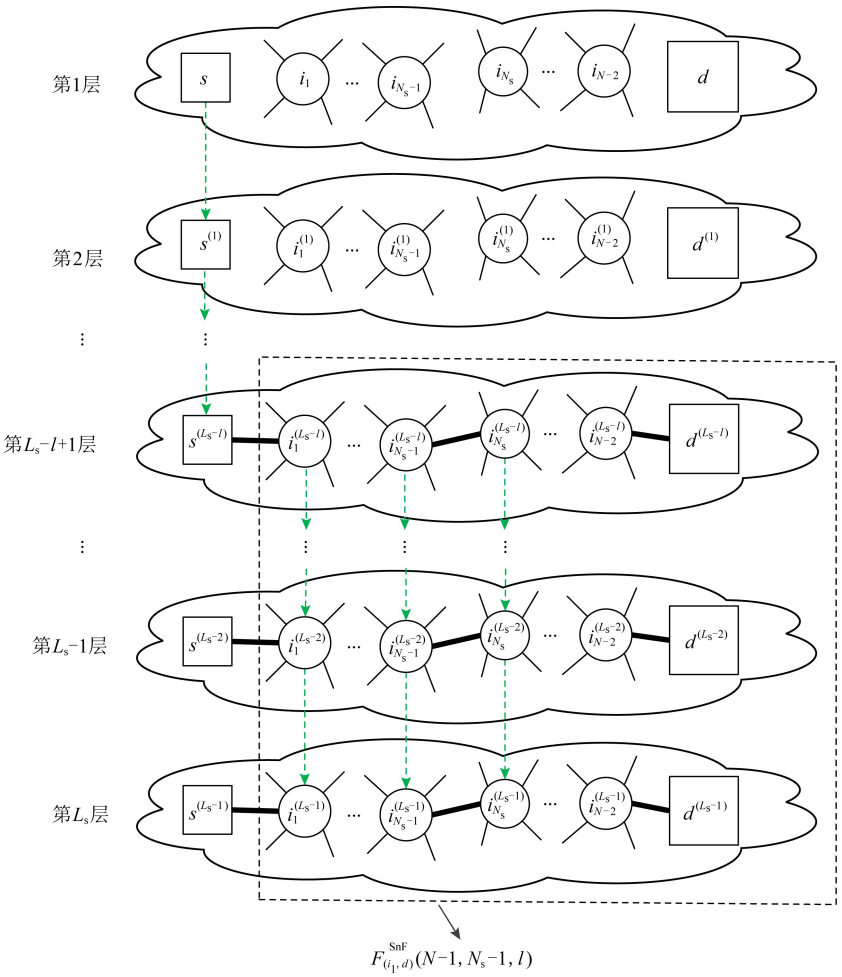


Fig. A1　Routing model of $F^{SnF}_{(i_1,d)}(N-1,N_s-1,l)$ when $N_s>1$

图 A1　当 $N_s>1$ 时 $F^{SnF}_{(i_1,d)}(N-1,N_s-1,l)$的路由模型