

基于博弈论的容迟网络中布雷斯路由悖论研究

赵晨曦,王 杨,许闪闪,孟 丹,赵传信
(安徽师范大学 数学计算机科学学院,安徽 芜湖 241000)

摘 要:容忍网络中的布雷斯(Braess)路由悖论现象对于其网络拓扑设计的高效性和合理性的提升具有重要的意义。对容迟网络常用的路由算法进行了分析,得出链路状态路由算法克服了距离矢量路由算法收敛慢、容易成环的缺点;但此算法采用的是迪杰斯特拉(Dijkstra)算法,每次寻找的是最短路径,可能会在一定情况下出现悖论现象。因此通过博弈理论的基本原理分析了 Braess 悖论及其对偶形式的存在性,对应于上述算法中,在其他条件不变的前提下,增加网络负载权值,提高了选择过程的效率。最后,在仿真实验中随机建立容迟网络的路由节点,规定路由由节点间的距离、传输速度等权值,并收集多组基于不同网络特征参数的数据样本,编程来模拟路由算法并进行数据分析,进一步验证了 Braess 悖论在容迟网络路由算法中的存在。

关键词:容迟网络;链路状态路由算法;迪杰斯特拉算法;布雷斯悖论;博弈论

中图分类号:TP301

文献标识码:A

文章编号:1673-629X(2018)10-0059-05

doi:10.3969/j.issn.1673-629X.2018.10.012

Research on Braess's Routing Paradox in Delay Tolerant Networks Based on Game Theory

ZHAO Chen-xi, WANG Yang, XU Shan-shan, MENG Dan, ZHAO Chuan-xin
(School of Mathematics & Computer Science, Anhui Normal University, Wuhu 241000, China)

Abstract: The Braess's routing paradox in the tolerance network is of great significance to the efficiency and rationality of their network topology design. In this paper, we analyze the routing algorithms commonly used in networks, and it's concluded that the link-state routing algorithm overcomes the shortcoming of slow convergence and easy loop formation of the distance vector routing algorithm. However, this algorithm adopts the Dijkstra algorithm, which looks for the shortest path each time, and may encounter paradoxes under certain circumstances. Therefore, the existence of the Braess's paradox and its dual form is analyzed through the basic principle of the game theory, which corresponds to the above algorithm. Under other conditions unchanged, the weight of network load is increased to improve the efficiency of the selection process. Finally, in the simulation experiment, we randomly set up the routing node of the delay tolerant network, specify the distance between routing nodes, transmission speed and other weights, collect multiple sets of data samples based on the characteristics of different network parameters, and simulate the routing algorithm by programming and conduct the data analysis, which further validates the existence of Braess's paradox in the delay tolerant network routing algorithm.

Key words: delay tolerant network; link state routing algorithm; Dijkstra algorithm; Braess's paradox; game theory

0 引 言

容迟网络(delay tolerant network)是一种自组织网络,不需要在源节点和目的节点之间建立完整的通信路径,利用节点相遇实现网络通信,是无线网络研究领域的一个新兴热门方向。延迟容忍网络架构^[1]保证了异步消息在链路中断和传输节点资源有限情况下的可靠传输。其应用涵盖了因特网以外的许多通信网

络,如星际网络、乡村网络、战争网络、移动 AdHoc 网络和无线传感器网络等。人们对容迟网络的研究是希望在不稳定的动态变化情况下,网络可以提供足够品质的服务,对未来网络建设具有重要的研究意义。

在一个交通网络上增加一条路段反而使网络上的旅行时间增加了,而且所有出行者的通行时间都相应增加了,这一附加路段不但没有减少交通延滞,反而降

收稿日期:2017-11-27

修回日期:2018-03-20

网络出版时间:2018-05-28

基金项目:安徽省社科规划项目(AHSKY2017D42);安徽省重大人文社科基金项目(SK2014ZD033);安徽省优秀青年人才基金(2009SQZ127);蚌埠医学院科研基金(BY0839)

作者简介:赵晨曦(1992-),女,硕士研究生,研究方向为社交网络;王 杨,教授,研究生导师,研究方向为网络与云计算。

网络出版地址:http://cnki.net/kcms/detail/61.1450.TP.20180525.1610.062.html

低了整个交通网络的服务水准,这种出力不讨好且与人们直观感受相悖的交通网络现象就是人们所说的布雷斯悖论现象。

文中简要介绍了延迟容忍网络的架构,分析了延迟容忍网络中常用的几种路由选择算法;然后依据博弈论的知识,分析 Braess 悖论的成因,建立相应模型;最后给出了仿真实验及结果分析。

1 延迟容忍网络的架构

由于主机和路由器的不断移动而出现了“受限网络”^[2],这就对现有的 Internet 体系结构和协议应用产生了挑战。因为在陆地移动网、军事无线自组织网等网络^[3-6]中,由于情况的特殊,经常会出现大的链路延迟、端对端无路由路径、缺少及时的能量补给和足够的存储能力等状况。

为了解决这个问题,可以选择两种方式:一种是在现有的 Internet 体系结构和协议应用下,提出一些“弥补”措施,例如“链路修正法”(link-repair approaches)和“网络特殊代理法”(network-specific proxies approaches)。前者是试图把网络中有问题的链路转化为可以适应 TCP/IP 的类似链路,尽力保持互联网端到端的可靠性和命运共享模式,而所有的路由器和端节点要执行 IP 协议;后者是把那些受限网络当作互联网的边缘,用特殊的代理方式连接互联网和受限网络。但是因为受限网络端到端之间的高延迟,低数据率,易断开,排队时间长,端系统寿命有限等特性,理所当然地想在这种网络上修改、加强协议来进行应用也是很复杂、难适应的;另一种方式就是提出一种新的专用于容迟网络的体系结构,它可以避免上述麻烦,很好地解决问题。

因为目前因特网互联主要还是依靠有线信道,因此 TCP/IP 协议被普遍应用。但是随着无线互联技术的发展,TCP 协议的劣势开始显露,主要还是由于 TCP 通信需要一段时间往返以建立连接,若传输的延迟超出了通信持续时间,那么应用层就没有数据可发送,其对数据丢失和网络拥塞处理的方式使 TCP 吞吐量随着往返延迟的增加而减少。所以研究者在应用层和传输层之间插入一个包裹来确保端到端可靠的数据传输服务,这个包裹层可以提供存储转发功能,可以很好地克服网络中断现象^[7-8]。

2 延迟容忍网络路由算法

2.1 源路由选择算法

源路由选择算法将一个分布式问题转化为集中式问题,算法中每个节点都保留着所有的全局状态信息,包括网络的拓扑结构和每条链路的状态信息。利用已

知信息,在源节点便可计算出全局的可行路径,然后沿此路径,用于通知中间节点前后节点信息的控制报文被发送出去,其中链路状态协议用来在每个节点更新全局状态。因此,源路由选择算法的概念十分简单且易于测试。但是对于小型网络,其开销尚可接受,而对于大型网络,每次为了保持全局信息的准确,就必须频繁地依次刷新,通信开销不小,实际可行性较低。

2.2 分布式路由选择算法

在分布式路由选择中,源节点和目的节点之间的各个中间节点都在进行路径的计算。节点之间交换控制报文,同时每个节点上的状态信息被集中用来进行路径寻找,大部分分布式路由选择算法都采用距离向量协议(或链路状态协议)以距离向量的形式在每个节点上保持全局状态。由于路径通过分布式的计算得出,因此路由选择的响应时间大大缩短,算法更加易于扩展。网络并行寻找多条路径,进而从中找出可行的一条,大大提高了成功的可能性。大部分现有的分布式路由选择算法都要求每个节点保存全局网络状态信息,并利用此状态逐跳进行路由选择。

因而,分布式路由选择相比源路由选择更能适应容迟网络多变的拓扑结构,但是因为利用了全局状态进行路由选择,所以或多或少也存在源路由算法的问题;而如果不需要任何全局状态的话,则必须传送更多的报文,通信量一样很大。此外,当不同节点上的全局状态不互相关联时,就有可能出现环路。

2.3 距离矢量路由选择算法

在距离矢量路由选择算法(distance vector routing algorithm)中,每个路由器都有一张以其他路由器为索引的向量表,表中包括每个目的地已知的最佳距离和路径。

设节点 X 的邻接点集合为 $T\{G_1, G_2, \dots, G_n\}$,其中 X 到 G_i 的代价为 $C(X - G_i)$, G_i 到 Y 的最小代价为 $C_{\min}(G_i - Y)$,则节点 X 到节点 Y 的最小代价为:

$$C_{\min}(X - Y) = \min \{C(X - G_i) + C_{\text{Least}}(G_i - Y)\}, \\ i = 1, 2, \dots, n \quad (1)$$

2.4 链路状态路由算法

链路状态路由算法(link state routing algorithm)概括起来有 4 步:发现本节点所有的邻居节点,计算开销;把收集到的交换信息合并为分组,并通知其他路由器;扩散发布链路分组;计算所有路由器的最短路径。这其实就是通常意义上的迪杰斯特拉(Dijkstra's algorithm)算法^[9]。

迪杰斯特拉算法是典型的单源最短路径算法,用于计算一个节点到其他所有节点的最短路径。该算法主要特点是以起始点为中心向外层层扩展,直到扩展到终点为止,又叫 SPF 算法。从某个源节点到目的节

点的最短路径就是所有到目的节点的路径中具有最小权值的那条。迪杰斯特拉算法一般的表述通常有两种方式,一种用永久和临时标号方式,一种是用 OPEN, CLOSE 表示的方式。

如图 1 所示,其最短路径计算过程为:从节点 A 开始(A 放入 S 集合),相邻节点为 B、C,其中 $A \rightarrow C$ 最短,C 加入 S 集合;C 相邻节点有 B、D、E,而 $A \rightarrow C \rightarrow B = 5$,比上面 $A \rightarrow B = 6$ 短, $A \rightarrow C \rightarrow D = 6$, $A \rightarrow C \rightarrow E = 7$,B 加入 S 集合,B 相邻节点有 D,而 $A \rightarrow C \rightarrow B \rightarrow D = 10$ 比上一步 $A \rightarrow C \rightarrow D = 6$ 长,所以 B 出 S 集合,D 进入 S 集,按照这些步骤,最后结果是 $A \rightarrow C \rightarrow D \rightarrow F = 9$ 。

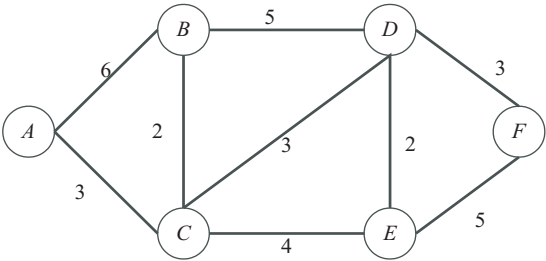


图 1 无向图

此算法对网络上的每个节点仅发送路由表中包含自身链路的那部分,克服了距离矢量路由算法收敛慢、容易成环的缺点。

3 布雷斯悖论博弈论分析

3.1 Braess 悖论产生原因

Braess 悖论现象是由 1968 年意大利数学家 Dietrich Braess 发现并提出的^[10],是交通网络均衡理论的典型案例^[11-12]。Braess 就满足 Wardrop 第一出行原则的用户平衡分配问题给出了一个实例,即在一个交通网络上增加一条路段反而使网络上的旅行时间 (travel time) 增加了,而且使所有出行者的旅行时间都增加了,这一附加路段不但没有减少交通延滞,反而降低了整个交通网络的服务水准 (level of service)。

例如图 2 中的交通网,有 2 000 辆车打算从起点通行至终点。起点到 A 的通行时间是路上车的数量除以 100,而起点到 B 的通行时间是固定的 25 min (另一条路时间相同)。若不存在近路,从起点到 A 到终点需要的时间: $T_{\text{start}} - A - \text{end} = \frac{A}{100} + 25$,而从起点到 B

到终点需要的时间: $T_{\text{start}} - B - \text{end} = \frac{B}{100} + 25$ 。

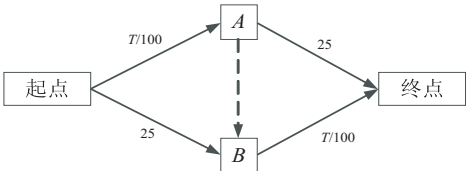


图 2 Braess 悖论示例

如果其中某条路的通过时间更短,是不可以达到纳什均衡 (Nash equilibrium) 的,因为任何一个理性的司机都会选择更短的路。因为共 2 000 辆车,可得 $A + B = 2000$,解出: $A = B = 1000$ 。这样每条路的通过时间都是 $T = \frac{1000}{100} + 25 = 35 \text{ min}$ 。

现在假设有了一条近路 A 至 B (通过时间接近于 0),在此情况下所有司机都会选择从起点到 A 到 B 这条线路,因为就算所有的车都走这条路,通过时间也不超过 20 min,小于起点到 B 的 25 min。到达 A 之后,所有司机都会选择从用接近 0 的时间行驶到 B 再到终点,因为就算所有的车都走这条路,通过时间也不超过 20 min,小于 A 到终点的 25 min。这样所有车的通过时间是 $T_{\text{all}} = \frac{2000}{100} + \frac{2000}{100} = 40 \text{ min}$,比不存在近道的时候还多了 5 min。因为没有司机愿意切换到别的路上去,所以走原先的路线 (起点至 A 至终点,起点至 B 至终点) 的时间都变成了 45 min。如果大家都约定好不走近路,那么都可以节约 5 min 的时间。但是,由于单个司机总是能从走近道上获益,所以这种约定是不稳定的,于是 Braess 悖论便出现了。

由于非合作网络中的纳什平衡点不在帕累托边界 (Pareto boundary) 上,Braess 悖论现象中出现效益负增长。这种情况下,存在一种非平衡的流量分布,使网络相对于平衡流量分布时某些用户的出行时间缩短,同时其他用户的出行时间也不会增加^[13-16]。从博弈论的角度分析,这是一个典型的“囚徒困境”,即每个博弈方都以其从起点到终点所需的行程时间最小为原则,在选择路径的时候不考虑其选择对其他驾车者的影响。博弈双方都追求个人利益最大化的结果是:每个博弈方的状况恶化,与此同时,使整个系统的效率降低。

依据上述分析,如果在容迟网络路由选择中可能出现布雷斯悖论现象,那么此路由选择方法中就必须出现追求自我利益最大化的选择策略,从而陷入上述的“囚徒困境”。

回顾第 2 节的延迟容忍网络路由算法可以发现,在链路状态路由算法的运算过程中,因为普遍采用的是 Dijkstra 算法,每次寻找的是最短路径,因此可能会在一定情况下出现悖论现象。

3.2 Braess 悖论对偶模型

要证明 Braess 悖论的存在,可从其对偶形式入手,证明 Braess 悖论的对偶形式存在即可。

Braess 悖论的对偶形式为:在其他条件不变的前提下,增加车流量,系统总通行时间减少,与预期相反。对应于容迟网络路由算法中,即可理解为在其他条件

不变的前提下,增加网络负载权值,不但没有增加路由选择所需的总时间,反而提高了选择过程的效率。

4 仿真实验

4.1 节点仿真

先在 Matlab 中随机建立容迟网络的路由节点,规定路由由节点间的距离、传输速度等权值。容迟网络的拓扑结构采用网络拓扑随机生成算法,程序中的参数包括区域边长、节点个数、网络特征参数等。各参数和作用如表 1 所示。

表 1 仿真程序参数列表

参数		意义
输入参数	Length	正方形区域边长
	NodeNum	路由节点个数
	Scale	边比例
	Density	边密度
	GraPrint	网络拓扑图显示参数,默认 1 显示
输出参数	ParPrint	节点参数显示参数,默认 1 显示
	NodeInf	存储路由由节点序号,横纵坐标的矩阵
	Dist	存储有向边的邻接矩阵
	Cost	存储边消耗的邻接矩阵
	Delay	存储边时延的邻接矩阵
Loss		丢包率

此程序可以随意控制网络参数,生成指定数量的路由节点,每个节点的参数可以显示或者输出到文件。仿真在 Intel Pentium Dual-Core 1.86 GHz CPU、内存 3 G 的计算机上进行。

4.2 算法实现

根据仿真得到的随机节点各项参数,通过输出到文件收集了多组基于不同网络特征参数的数据样本,应用这些数据样本,可以编程来模拟路由算法以进行分析。

分析程序使用 Codeblocks 10.05 编译运行,gcc 版本 4.6.1。程序中的参数解释见表 2。

表 2 程序参数列表

参数	意义
L	节点有效传输范围
Speed	节点之间传输速度
T	节点向指定节点传输的时间
coefficient	阻塞系数
totalFlow	网络总流量
totalTime	传输总时间
N	路径数目

表 3 是两组运行结果的比较。

表 3 运行结果

通路总数	网络总流量	传输总时间
156	12.856 7	11.596 4
156	12.862 8	11.672 8

4.3 结果分析

对于 Length = 10, NodeNum = 30, Scale = 10 的随机拓扑网络,其运行结果如表 4 所示。

表 4 运行结果

Length	NodeNum	Scale	totalFlow	totalTime
10	30	10	128 567	116 728
10	30	10	128 628	115 964
10	30	10	153 455	134 763
10	30	10	154 567	133 678
10	30	10	228 410	183 852
10	30	10	228 523	184 224
10	30	10	253 465	203 245
10	30	10	259 673	215 543

取表 4 的第一和第二行数据样本,网络总流量 $128\,567 < 128\,628$,而总时间 $116\,728 > 115\,964$ 。从实验结果可以得出,网络流量虽然在增加,但并不是严格地呈上升趋势,相反某些相邻点之间呈现了下降趋势。所以悖论的对偶问题成立,悖论现象出现,因而悖论在此也是成立的。

5 结束语

对容迟网络的路由选择算法进行了分析,通过深入研究布雷斯悖论的理论出现原因,找到了可能的出现场景,并通过仿真和程序验证了猜想,得出容迟网络路由中存在布雷斯悖论现象的结论。但是不能忽视的是布雷斯及其对偶形式存在一个严格的假设前提:博弈方在选择路径前完全了解网络信息。这在实际中是不可能实现的。因此,此研究在产生条件、参数影响、分析方法等方面仍有进一步深入探讨的空间。

参考文献:

[1] HUI Pan,CROWCROFT J,YONEKI E. BUBBLE rap:social-based forwarding in delay-tolerant networks[C]//Proceedings of the 9th ACM international symposium on mobile ad hoc networking and computing. Hong Kong, China:ACM,2008:241-250.

[2] KAWECKI M,SCHOENEICH R O. Mobility-based routing algorithm in delay tolerant networks[J]. EURASIP Journal on Wireless Communications & Networking,2016(1):1-9.

[3] CHOUDHARY N,SINGH V. Performance enhancement of routing protocol for longer life time of WSN[J]. International Journal of Computer Applications,2015,109(11):16-19.

[4] ZENG Deze, GUO Song, HU Jiankun. Reliable bulk – data dissemination in delay tolerant networks[J]. IEEE Transactions on Parallel & Distributed Systems, 2014, 25(8): 2180–2189.

[5] CAO Wei, CAO Guohong, PORTA T L, et al. On exploiting transient social contact patterns for data forwarding in delay – tolerant networks[J]. IEEE Transactions on Mobile Computing, 2013, 12(1): 151–165.

[6] MENG Tong, WU Fan, YANG Zheng, et al. Spatial reusability – aware routing in multi – hop wireless networks[J]. IEEE Transactions on Computers, 2016, 65(1): 244–255.

[7] WANG Ruhai, WEI Zhiguo, ZHANG Qinyu, et al. LTP aggregation of DTN bundles in space communications[J]. IEEE Transactions on Aerospace & Electronic Systems, 2013, 49(3): 1677–1691.

[8] MING Zhongxing, XU Mingwei, WANG Dan. Age – based cooperative caching in information – centric networking[C]//International conference on computer communication and networks. Shanghai, China: IEEE, 2012: 1–8.

[9] CHANG M, PENG S, LIAW J. Deferred – query: an efficient approach for some problems on interval graphs[J]. Networks, 2015, 34: 1–10.

[10] WITTHAUT D, TIMME M. Nonlocal failures in complex supply networks by single link additions[J]. European Physical Journal B, 2013, 86: 377.

[11] TAGUCHI A. Braess’ paradox in a two – terminal transportation network[J]. Journal of the Operations Research Society of Japan, 2017, 25(4): 376–389.

[12] XU Meng, WANG Guangmin, GRANT – MULLER S, et al. Joint road toll pricing and capacity development in discrete transport network design problem[J]. Transportation, 2017, 44(4): 731–752.

[13] 刘巍, 曾庆山. 基于复杂网络的 Braess 悖论现象[J]. 计算机工程与设计, 2015, 36(4): 1098–1102.

[14] 董顺珍. 蜈蚣博弈悖论理性人的认知分析[J]. 长江丛刊, 2017(18): 148.

[15] 余孝军, 黄海军. 交通网络效率的度量 and 元件重要性的计算方法[J]. 系统工程理论与实践, 2012, 32(7): 1546–1552.

[16] 傅白白, 刘法胜. 管理中的 Nash 平衡与 Braess 悖论现象[J]. 运筹与管理, 2004, 13(1): 150–155.

(上接第 58 页)

[9] 董红斌, 滕旭阳, 杨雪. 一种基于关联信息熵度量的特征选择方法[J]. 计算机研究与发展, 2016, 53(8): 1684–1695.

[10] 王康, 李东静, 陈海光. 分布式存储系统中改进的一致性哈希算法[J]. 计算机技术与发展, 2016, 26(7): 24–29.

[11] 李庆华, 李肯立, 蒋盛益, 等. 背包问题的最优并行算法[J]. 软件学报, 2003, 14(5): 891–896.

[12] 陈慧南. 算法设计与分析: C++ 语言描述[M]. 北京: 电子工业出版社, 2006: 45–47.

[13] COOPER B F, SILBERSTEIN A, TAM E, et al. Benchmarking cloud serving systems with YCSB[C]//ACM symposium on cloud computing. Indianapolis, Indiana, USA: ACM, 2010: 143–154.

[14] XU Yuehai, FRACHTENBERG E, JIANG Song. Building a high – performance key – value cache as an energy – efficient appliance[J]. Performance Evaluation, 2014, 79: 24–37.

[15] PAIVA J, RUIVO P, ROMANO P, et al. AUTOPLACER: scalable self – tuning data placement in distributed key – value stores[C]//International conference on autonomic computing. New York: ACM, 2014.

[16] MIRANDA A, EFFERT S, KANG Y, et al. Random slicing: efficient and scalable data placement for large – scale storage systems[J]. ACM Transactions on Storage, 2014, 10(3): 9.

[17] DAS S, NISHIMURA S, AGRAWAL D, et al. Albatross: lightweight elasticity in shared storage databases for the cloud using live data migration[J]. Proceedings of the VLDB Endowment, 2011, 4(8): 494–505.

[18] ZHANG Jiao, REN Fengyuan, LIN Chuang. Delay guaranteed live migration of virtual machines[C]//Proceedings of IEEE INFOCOM. Toronto, ON, Canada: IEEE, 2014: 574–582.

[19] LIU Haikun, HE Bingsheng. Vmbuddies: coordinating live migration of multi – tier applications in cloud environments[J]. IEEE Transactions on Parallel & Distributed Systems, 2015, 26(4): 1192–1205.

[20] BASIN D, BORTNIKOV E, GOLAN – GUETA G, et al. KiWi: a key – value map for scalable real – time analytics[C]//ACM SIGPLAN symposium on principles and practice of parallel programming. Austin, Texas, USA: ACM, 2017: 357–369.