文章编号:1674-2869(2013)09-0082-05

校园网多径混合路由算法

徐 方
(湖北工程学院现代教育技术中心,湖北 孝感 432000)

摘 要:针对目前校园网路由算法中最小生成树的计算和最短路径的生成存在速度慢和效率低的问题,提出了一种多径混合路由算法.结合了静态路由算法和动态路由算法的优点,减少了计算最短路径树时的总执行时间,当网络中链路有新的权重变化时,它使用多径信息来创建最短路径树,并且能够根据网络中链路权重变化的位置来决定使用静态路由算法或者是动态路由算法.与现有的迪杰斯特拉(Dijkstra)算法、动态Dijkstra算法和混合最短路径树算法进行了对比实验,结果表明多径混合路由算法降低了最小生成树的计算时间.在校园网中使用多径混合路由算法可以加快了网络路由的收敛,提高了网络的性能.

关键词:网络路由;动态路由;校园网络;混合路由
中图分类号:TP393 **文献标识码**:A **doi**:10.3969/j.issn.1674-2869.2013.09.017

0 引 言

随着校园网应用的不断深入,校园网中不仅有传统的网络(WEB)服务、文件传输协议(File Transfer Protlcol,以下简称:FTP)服务、域名解析系统(Donain Name System,以下简称:DNS)服务、E-Mail服务等,而且大量的新型的网络应用也不断增加,如远程教学、网络课程、视频点播以及迅速增长的对等网络(Peer-to-Peer,以下简称:P2P)应用,这些使得校园网络内承载了大量的视频和语音的流量.这些应用对网络的性能提出了更高的要求,特别是对路由协议的收敛时间.开放最短路径优先协议(Open Shortest Path First,以下简称:OSPF)是一种基于区域实现的、建立在链路状态(Link State)算法和最短路径优先(Shortest Path First,以下简称:SPF)算法基础之上的内部网关动态路由协议[1],也是校园网内应用最广泛的协议.开放最短路径优先协议是最常用的自治系统内部的路由协议,如何在其中设计高速的路由算法就显得更加重要.在使用OSPF协议的网络中,当遇到突发事件网络的拓扑发生变化时,网络路由算法就开始重新计算并更新路由表.例如,如果网络中有一个链路发生了故障,则路由协议会重新计算最短路径.在这种情况下,传统的OSPF使用迪杰斯特拉(Dijkstra)算法进行最短路径的计算[2].然而,当网络中链路的权重有新的变化时,网络会使用Dijkstra算法在每个节点上进行重复的大量计算和不必要的修正,而不关注链路权重的变化在网络中发生的位置.因此,这可能会导致整个路由表频繁更新,从而使网络变得不稳定[3].

为了解决上述问题,可以使用动态路由算法来计算最短路径.目前有很多研究者正在研究动态路由算法,其中动态Dijkstra算法[4]和可靠的动态最短路径(Reliability of Dynamic Shortest Path,以下简称:RDSP)[5]算法是动态路由算法的典型代表.与静态路由算法相比,动态路由算法可能在某一个节点上需要更多的计算时间,然而,动态路由算法可能会减少总的计算时间,因为它可以减少计算的节点的数目.也就是说,当一个网络中一些链接有新的权重时,动态路由算法利用变化的链路找到受影响的节点,只需要重新计算这些受影响节点的最短路径.因此,动态路由算法可以比静态路由算法需要更少的时间来计算最短路径.然而,动态的路由算法所需要的计算时间依赖于权重变化链路在网络中的位置.

混合式路由算法可以结合动态路由算法和静态路由算法的优点,它的路由收敛的计算时间会更少.目前有些研究者提出了混合式路由的想法,使用混合最短路径树(Hybrid Shortest Path Tree,以下简称:HSPT)[2]可以有效地计算最短路径,其中静态和动态的算法运行的次数是非常关

收稿日期:2013-06-08
基金项目:湖北工程学院自然科学基金(Z2012006)
作者简介:徐 方(1981-),男,湖北洪湖人,实验师,硕士.研究方向:计算机网络、信息系统集成.

键的因素.本文提出了一种多径混合路由算法,当一些链路的权重发生变化时,它使用多径信息来创建最短路径,如果找不到多条路径,则使用混合路由算法计算最短路径.为了评价所提出的算法,将其与静态的 Dijkstra、动态 Dijkstra 算法和 HSPT 算法进行了比较.

1　问题的提出

网络路由的收敛是指在最佳路径的判断上所有路由器达成一致的过程.当出现某些网络事件引起路由发生故障或不可达时,路由器就会发出路由更新的消息.遍及整个网络的路由更新会引发路由器重新计算最佳路径,所有的路由器最终形成一个公认的最佳路径[6].然而,网络中链路状态的变化会导致每个路由器重新计算自己的最短生成树(Shortest Path Tree,以下简称:SPT)并完全更新相应设备的路由表.但是研究发现,在很多情况下,旧的 SPT 和新的 SPT 几乎没有什么变化.这种现象的出现,主要原因是在较大范围的路由区域内,少数几条链路状态的变化对 SPT 的生成影响不大.所以,研究一种新的动态更新 SPT 的算法来处理网络链路的变化是很有必要的.

Dijkstra 算法是一个静态的路由算法,被广泛用于在网络路由中.然而,这个被充分研究的静态路由算法在某些场合下也是低效率的,比如在网络中只有一小部分的链路状态发生变化而需要更新时.这是因为此时,该算法会在所有节点上重新计算新的最短路径,而不再利用原来的最短路径信息.这一过程对于一个路由器来说影响是非常大的,因为它需要占用大量的 CPU 资源.并且这个过程使用 Dijkstra 算法计算最短路径需要一个较长的时间,同时存在一个路由不稳定的时间段.在该时间段内,路由表不能真实反映实际的网络拓扑结构,直到算法运行完成才能得到真实的网络拓扑,在这个时间段内网络就可能会出现丢包的现象[7].

动态路由算法优于静态路由算法,它通过利用大部分原有路由表中的信息,只需要重新计算受影响节点的最短路径并更新路由表,因此它们可以减少计算时间.然而,动态路由算法所需要的计算时间取决于变化的链路的位置.也就是说,当根节点附近有一些链路的权重有变化时,路由的收敛需要较长的时间,这是因为此时大量的节点会受到影响而不得不重新计算最短路径.相反,如果在终端节点附近的一些链接有一个新的权重,计算时间则是较短的,因为此时只有较少节点受影响.因此,动态路由算法在计算时间上并不是一定优于静态路由算法,因为它受到上述因素的影响,这是网络路由中的一个不确定性问题.

目前一些研究者提出了混合路由算法的思想,它结合了静态路由算法和动态路由算法的优点.混合路由算法比静态或者动态的算法更高效.但是,如果混合路由算法使用多径信息,它的性能会提高很多.也就是说,多径混合路由算法比混合路由算法具有更少的计算时间.

2　多径混合路由算法

本文提出的多径混合路由算法利用多径信息减少了总的最短路径树的执行时间.该算法被称为多路径混合的最短路径树(Multi-path Hybrid Shortest Path Tree,MHSPT),是在 HSPT 算法的基础上提出来的.为了使 HSPT 有效地计算最短路径,应用静态算法和动态算法运行的次数是关键的因素.当根节点附近的链路的权重变化时,网络中受影响的节点数目会很大,适合使用静态路由算法来计算最短路径.此时动态路由算法不存在优势,因为此时根节点附近的很多节点都要重新计算路由.在这种情况下,静态路由算法是一个快速的计算方法.如果此时使用动态路由算法,每个节点计算最短路径就需要更多的计算时间,从而增加了总的计算时间.然而,当终端节点附近的链路权重变化时,适合使用动态路由算法来计算最短路径.动态路由算法适合于这种场合是因为此时终端节点附近只有少量的节点需要重新计算路径.在这种情况下,使用动态路由算法仅仅需要重新计算被影响的少数节点的最短路径信息,而不需要像静态路由算法那样需要计算所有节点最短路径信息,所以此时动态路由算法收敛快于静态路由算法.此外,MHSPT 使用了多径信息,多径信息的获取方法参照文献[8].如果其他的最小权重路径被发现,该算法使用多路径信息,可以减少计算时间.如果其他的最低成本的路径都没有找到,该算法可以使用的多径信息减少的受影响的节点数目.MHSPT 算法的伪代码如图 1 所示.

多路径混合路由算法分为如下步骤进行.首先,使用多径信息找出最短路径.如果发现了一个最短路径,就把它归入最短路径树中;如果没有找到多路径,就开始启用混合路由算法.第二步,为了决定使用哪一种路由算法,需要使用深度优先搜索(DFS)来计算整个网络的深度.第三步,当某条链路的权重发生变化,并且发现它在网络中的

```
步骤1：利用多径信息
通过使用多径信息查找最短路径
if 最短路径找到 then
    更新路由表
  else
    go to 步骤2
endif
步骤2：查找到整个网络的深度
使用深度优先搜索（DFS）找到网络的深度ND
D=40%ND
for 所有节点 do
    查找链路开销发生变化的链路
    使用DFS计算链路的深度LD
      if LD<D then
          go to 步骤3
      else
          go to 步骤4
      endif
endfor
步骤3：静态路由算法
使用静态路由算法计算最短路径
更新路由表
步骤4：动态路由算法
使用动态路由算法计算最短路径
更新路由表
```

图1 MHSPT算法的伪代码

Fig. 1 Pseudo code for the MHSPT

深度小于40%时，就使用静态路由算法来计算最短路径. 当变化权重的链路的深度大于等于40%时，使用动态路由算法来计算最短路径.

3 性能分析

为了研究本文提出的路由算法MHSPT的性能，使用C++语言编写了相应的算法的仿真程序，将本文提出的MHSPT算法性能与Dijkstra、动态Dijkstra(D-Dijkstra)和HSPT算法的性能进行了比较. 在仿真实验中，使用如表1所示的输入参数：节点的数量、链路权重的变化率、链路权重的偏差.

表1 实验输入参数

Table 1 Input parameters in experiment

参数	参数取值
节点的数量	60,120,180, 240,300,360
链路权重的变化率/%	100, 300
链路权重的偏差	6, 18

节点的数目表示在整个网络中的节点的数目，节点数从60到360能充分表达节点规模从小到大时路由协议的性能. 链路权重的变化率是新链路权重与旧链路权重的比值，根据网络中链路的实际变化率统计，100%和300%是比较常见的情况. 在本实验中，链路权重平均值设为12. 链路权重的偏差是当前链路权重与链路权重平均值之间的差异，在此选择的偏差值为6和18，分别在平均值的之下和之上，与平均值的距离相等，也比较符合网络中的实际情况. 在相同环境的网络中，使用上述几种算法计算最短路径，然后把各种算法所需要的总的计算时间进行比较.

图2表明了上述4种路由算法随着节点数量和链路权重变化率的变化计算最短路由总计算时间变化的情况. 在图2(a)中链路权重的变化率为100%，图2(b)中链路权重的变化率是300%. 随着节点数量的增加，计算最短路径的时间会随之增加. 从图2(a)可以看出：当节点规模增加时MHSPT计算最短路径树的时间接近于HSPT. 同样在图2(b)中也可以看到类似的情况. 如果链路权重变化率较低，计算最短路径的时间会呈线性增长. 当链路权重的变化率较高时，MHSPT算法的性能表现会更好.

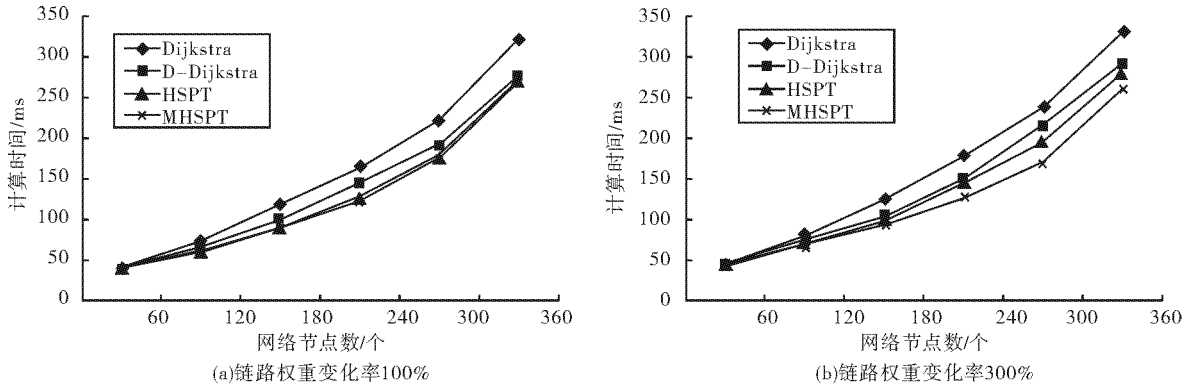

图2 总的计算时间随着节点数量和链路权重变化率的变化情况

Fig. 2 Total computation time with change of the number of nodes and the changed rate of link weights

图3是上述4种不同路由算法随着节点数量和链路权重偏差的变化计算最短路由总计算时间变化的情况，图3(a)中链路权重的偏差是6，图3(b)中链路权重偏差是18；链路权重偏差的平均值是12. 从图3中可以看出随着节点的数量的增加和链路权重偏差的变化，MHSPT的性能优于Dijkstra、D-Dijkstra和HSPT算法.

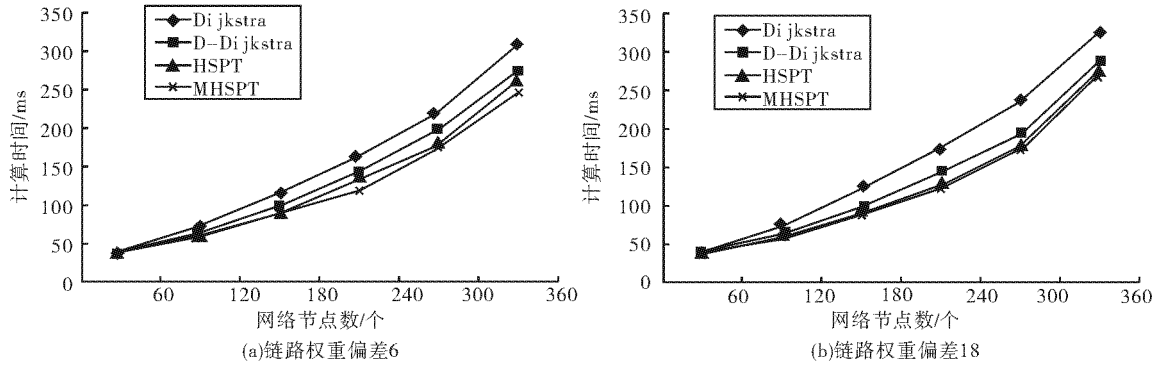

图 3　总的计算时间随着节点的数量和链路权重偏差变化的情况

Fig. 3　Total computation time with change of the number of nodes and deviation of link weights

从图 3 可以看出:当节点规模不断增大时,MHSPT 算法中最小生成树的计算时间小于其它 3 种算法,具有明显的优势;随着节点链路权重偏差的增加,MHSPT 计算最短路径的总时间有所增加,但也同样少于其它算法. 上述实验结果表明,无论是从节点规模变化、链路权重变化率的变化,还是链路权重偏差的变化等方面比较,MHSPT 算法都优于目前其它先进的算法.

4　结　语

由于教育信息化的不断深入发展,校园网上的应用系统越来越丰富,校园网承载的业务量也越来越大. 此外,大量的实时业务对网络的服务质量提出了更高的要求. 因此,非常有必要在校园网络路由协议中寻求一种快速生成最小生成树和最短路径的方法. 本文结合静态路由算法和动态路由算法的优点,并利用多径信息提出了一种混合路由算法. 对比实验表明,该算法降低了最短路径的计算时间,加快了网络的收敛过程. 然而,本算法也存在一些不足的方面,如算法在节点上的计算复杂性可能会高于上述其它算法,这可能会增加网络节点负载和能耗,不过这一点对于以有线网络为主的校园网来说影响并不大. 下一步的研究工作是将本文提出的算法与实际校园网络紧密结合,进一步对所提出的算法进行完善.

致　谢

感谢湖北工程学院对本研究的资金资助!

参考文献:

[1] Wang F Y, Wu S F. On the vulnerabilities and protection of OSPF routing protocol [C]// Proceedings of the International Conference on Computer Communications and Networks, Washington, DC, USA: IEEE Computer Society, 1998:148-152.

[2] Calduwel Newton P, Arockiam L, Kim T H. An Efficient Hybrid Path Selection Algorithm for an Integrated Network Environment[J]. International Journal of Database Theory and Application, 2009, 2(1): 31-38.

[3] Ameen, S Y, Ibrahimi I A. MANET Routing Protocols Performance Evaluation with TCP Taho, Reno and New-Reno[J]. International Journal of u-and e-Service, Science and Technology, 2011, 4(1): 37-49.

[4] 刘代波,侯孟书,武泽旭,等. 一种高效的最短路径树动态更新算法[J]. 计算机科学,2011, 38(7): 96-99.
Dai-Bo Liu, Meng-Shu Hou, Ze-Xu WU, et al. Efficient Dynamic Algorithm for Computation of Shortest Path Tree[J]. Computer Science, 2011, 38(7): 96-99. (in Chinese)

[5] Cho T H, Kim J W, Kim B J, et al. A Study on Shortest Path Decision Algorithm for Improving the Reliability of Dynamic Routing Algorithm [J]. Journal of the Korean Institute of Information Scientists and Engineers, 2011, 38: 450-459.

[6] 余雪勇,卞乃猛,唐家益,等. 基于 OSPF 协议的快速动态路由算法研究[J]. 重庆科技学院学报,2008, 10(3): 74-77.
YU Xue-yong, BIAN Nai-meng, TANG Jia-yi, et al. Research on the Algorithm of Fast Dynamic Routing Based on OSPF[J]. Journal of Chongqing University of Science and Technology: Natural Sciences Edition, 2008, 10(3): 74-77. (in Chinese)

[7] Jedari B, Goudarzi R, Dehghan M. Efficient Routing using Partitive Clustering Algorithms in Ferry-based Delay Tolerant Networks[J]. International Journal of Future Generation Communication and Networking, 2009, 2(4): 1-14.

[8] Eramo V, Listanti M, Cianfrani A. Design and Evaluation of a New Multi-Path Incremental Routing Algorithm on Software Routers[J]. IEEE Transactions on Network

and service management, 2008, 5(4): 188-203.

Algorithm of multi-path hybrid routing in campus network

XU Fang

(Technique Center of Modern Education, Hubei Engineering University, Xiaogan 432000, China)

Abstract: A kind of high-efficiency multi-path hybrid routing algorithm was presented to address both the calculation of the minimum spanning tree and the generation of the shortest path in campus. The total computation time of shortest path tree was reduced by using the advantages of both static routing algorithm and dynamic routing algorithm. The multi-path information was used to create the shortest path tree when some links had new weights. The decision of using static routing algorithm or dynamic routing algorithm was made by the location of the links that had new weights. Comparisons with other routing algorithms, such as Dijkstra algorithm, Dynamic Dijkstra algorithm, and HSPT algorithm, demonstrate that the multi-path hybrid routing algorithm provides better performance in the calculation of minimum spanning tree as the execution time decreases. Therefore, the convergence time of network routing is accelerated by using the proposed algorithm, which provides better performance for campus network.

Key words: network routing; dynamic routing; campus network; hybrid routing

本文编辑:苗 变

☆

(上接第81页)

Community decomposition algorithm based on set expansion

LI Yuan-yuan[1,2], ***YU Wang-jian***[1], ***HE Min-hua***[1,2]

(1. School of Science, Wuhan Institute of Technology, Wuhan 430074, China;
2. Hubei Province Key Laboratory of Intelligent Robot, Wuhan 430074, China)

Abstract: Community structure is one of important characteristics in complex network, seeking communities has very important meaning in the analysis of structure and function of the whole network. On the basis of some classical complex network structure partition algorithms, a new community structure partition algorithm based on set expansion was proposed. In this work, we defined an indicator, which is the ratio of the number of edges internal and external community. In the proposed algorithm, the set was originally composed of two adjacent nodes, indicator was recalculated if a neighbor node joined the set. Set was expanded when the indicator decreased. A community was detected until there was no new neighbor node to join. Other communities in the network were detected by repeating this process. This algorithm was implemented to detect communities in two classical networks. Experimental results show that this algorithm can reasonably detect communities with little computation and high efficiency. Moreover, the proposed algorithm can be generalized to other large-scale complex networks.

Key words: complex network; community structure; set expansion

本文编辑:苗 变