

基于路径可行距离的流量调整算法

关礼安 汪斌强

(国家数字交换系统工程技术研究中心 郑州 450002)

摘 要 针对现有多路径流量调整算法的负载不均衡问题,提出一种改进的流量调整算法(IAH, Improved Adjustment Heuristic)。它将反映网络均衡性的全网信息和局部信息的结合使用,根据路由信息中路径可行距离的变化,遵循距离远的路径能力差和距离近的路径能力强的原则来调整流量,使多路径中流量与相应的可行距离匹配。仿真结果表明,IAH 算法比 AH 算法更晚出现丢包,其丢包率性能整体上要优于 AH 算法,即 IAH 算法的负载均衡性优于 AH 算法。

关键词 负载均衡,多径,可行距离,流量调整

中图法分类号 TP393 **文献标识码** A

Adjusting Traffic over Multi-paths Based on Feasible Distance of Path

GUAN Li-an WANG Bin-qiang

(China National Digital Switching System Engineering and Technological Research Center, Zhengzhou 450002, China)

Abstract For eliminating the problem of poor load balancing which is resulted from implementing current adjusting traffic algorithms over multi-paths, IAH(Improved Adjustment Heuristic) was proposed. IAH utilizes the information of whole network and nodes which show the state of loads on the links. Based on the change of feasible distance of paths in routing information and the principle that the capability of short path is better than one long path, IAH adjusts traffic over multi-paths, transfers traffic from links with heavy load to light ones and make their traffic respectively match with their feasible distance. Simulation results show that the time of losing packets in implementing IAH is later than AH and the losing packet rate of IAH is better than AH in most time. Namely, the load balancing performance of IAH exceeds one of AH.

Keywords Load balancing, Multi-path, Feasible distance, Adjusting traffic

1 引言

随着因特网的飞速发展,各种业务对网络资源的需求急速增加。当前大多数域内路由协议总是倾向于选取处理能力强的节点和链路来计算节点到目的地的最短路径,某些链路被多条传输路径同时使用,容易引发网络传输拥塞。为解决单径路由带来的传输拥塞问题,人们提出了多径路由的概念,并使用流量调整算法在多条可行路径间分配流量来提高网络的均衡性,以充分地利用网络资源。在实际应用中,多路径间流量调整的原则是:均衡各路径的负载和流的保序。但这两个原则互相冲突,需要相应的折中来满足业务需求。逐分组轮转算法^[1]在分组长度恒定时可以获得最优的负载均衡,但分组乱序严重;分组不恒定时,两个目标都不能实现。贪婪算法^[2]在分组到达后,比较每跳链路的负载计数器,将分组向负载轻的链路转发,但存在计数器溢出和乱序问题。直接哈希算法^[3]将标识流的五元组作为哈希函数的输入来选择转发链路,可以实现分组流的严格保序,但负载均衡较差。基于表的

哈希算法^[3]将直接哈希算法中分组流与多路径间的映射关系进行人为设定,虽然改善了负载均衡性且无分组乱序,但不能很好处理动态网络流量。映射表动态重配哈希算法^[4]每隔一个固定时间片,根据记录的路径流量信息将映射表重新配置,流量负载计数器清零,只要选择适合的时间片长度就能达到很好的负载均衡且无分组乱序,但链路的负载信息是局部网络信息,不能反映后续链路的负载情况,会出现前通后堵的情况。Srinivas^[5]给出了一种分级流量均衡机制,即将流量均衡分为两部分:首先考虑全网时延信息,计算得到多条路径间初始分配流量,在重新计算路由完成前,根据局部信息(变化的可行距离)在多条路径间调整流量-AH 算法。该算法的流量调整充分考虑了全网信息和局部信息,能够很好地均衡网络流量且基于流操作,避免了乱序问题,且性能要优于上述几种算法,但其局部信息利用不充分,只是简单地按照一定比例转发,均衡性没有达到最优。

因此,针对上述问题,本文提出一种能充分利用局部信息的改进 AH 算法。该算法能根据网络局部信息调整流量,使

到稿日期:2010-01-27 返修日期:2010-04-12 本文受国家 863 计划项目(2008AA01A323, 2008AA01Z214),国家 973 计划项目(2007CB307102)资助。

关礼安(1979—),男,博士生,主要研究方向为宽带信息网络和新型路由技术等, E-mail: storm1995@163.com; 汪斌强(1963—),男,教授,主要研究方向为下一代网络和可重构路由器单元等。

局部流量分配达到最优。本文首先分析了 AH 算法及其对网络流量的影响,并与最优流量分配进行了比较。之后给出了改进 AH 算法的描述,并通过仿真对其性能进行了验证。

2 AH 算法介绍

本文中,将 IP 网络模拟为一个简单的加权有向图 $G=(N,L)$,这里 N 为网络中的节点(路由器)集合, L 为网络中的链路集合, N^i 代表节点 i 的邻居节点集合。每条链路为双连通且双向花费可能不同。

Gallager 提出的最小时延路由算法^[6],是一种最优多径路由算法。根据 Gallager 的理论,最小时延路由问题就是在满足式(1)的条件下,计算节点 i 到达目的节点 j 的路由参数集 $\{\phi_{jk}^i\}$,可用下一跳集 S_j^i 和到目的节点的可行距离 D_j^i :

$$\begin{cases} D_j^i = \sum_{k \in N^i} \phi_{jk}^i (D_j^k + l_{ik}) \\ S_j^i = \{k | \phi_{jk}^i > 0 \cap k \in N^i\} \\ D_j^i = D_j^k + l_{ik}, D_j^k < D_j^i + l_{ik} \end{cases} \quad (1)$$

式中, l_{ik} 为链路 (i,k) 的花费。

该算法的执行是基于输入流量和网络拓扑相对恒定的假设,这些假设在实际应用中无法得到满足;同时,该算法计算路由参数的复杂度较高,目的节点需要频繁触发计算过程来调整各个节点的路由参数,这些路由参数的扩散也受到网路直径和拓扑稳定性等因素的影响。为了实际应用上述算法,文献[5]对其进行了近似,提出了一种采用不同状态信息来计算路由的算法,亦即使用长间隔信息(一定时间内认为相对稳定)计算到达目的节点的多个无环路径,这些路径在较长时间内不再重新计算;启发式算法利用短间隔信息(局部随时变化)调整路由参数。因此,该算法与单径路由算法具有相同的收敛速度。同时,启发式算法也能快速地调整局部流量,即 $\{\phi_{jk}^i\}$, S_j^i 和 D_j^i 满足式(2)所示条件:

$$\begin{cases} D_j^i = \min\{D_j^k + l_{ik} | k \in N^i\} \\ S_j^i = \{k | D_j^k < D_j^i \cap k \in N^i\} \\ \phi_{jk}^i = \Psi(k, A_j^i, B_j^i), k \in N^i \end{cases} \quad (2)$$

式中, $A_j^i = \{D_j^k + l_{ik} | k \in N^i\}$, $B_j^i = \{\phi_{jk}^i | k \in N^i\}$, Ψ 为生成路由参数的启发式函数。

上述等式说明,近似的多径路由算法必须先建立无环路径,再通过启发式算法在路径间分割流量。

一般来讲,启发式函数 Ψ 是满足上述条件的任何函数。但对于实际网络,要求 Ψ 能很好地均衡流量。假设网络各节点都能获得正确的网络信息,多径路由算法已经根据这些信息计算出多条路径;启发式算法执行过程中,多径路由算法不重新计算路由。

启发式函数实现两个功能:流量分割和流量调整。当重新计算得到 S_j^i 时,需要执行流量分割。文献[5]中提出一种基于流的 IH(Initial Heuristic)算法来分割流量,此时的启发式函数 Ψ 只与可用下一跳的距离有关,即 $\phi_{jk}^i = \Psi(k, \{D_j^k + l_{ik} | k \in N^i\})$ 。距离小的下一跳,传输能力较强,可承载较多的流量。反之,距离大的下一跳可承载流量较少。因此,当下一跳多于一个,距离大的下一跳分割得到的流量小于距离小的下一跳。随着网络局部状态的变化, S_j^i 上的流量不再符合距离比例,需要执行流量调整算法。文献[5]给出了一种 AH 算法来调整局部流量,此时的启发式函数 Ψ 与目前可用下一跳

上的流量和距离有关。该算法将流量从链路距离大的链路切换到链路距离最小的链路。上述算法在一定条件下,可以使各下一跳链路中的流量相对均衡。但是,将其他链路的冗余流量全部切换到距离最短的链路上,又将引起新的流量不均衡。同时,调整后的各下一跳流量,其比例与距离比例仍存在一定差距,因此引出下面的分析。

3 改进的 AH 算法

在上述假设成立的情况下,距离下最优路由参数^[5]为:

$$\bar{\phi}_{jk}^i = \frac{1 - \frac{D_{jk}^i}{\sum_{m \in S_j^i} D_{jm}^i}}{n - 1} \quad (3)$$

式中, $n = |S_j^i|$ 为下一跳数。

定义路由参数差异绝对值之和来反映实际路由参数与最优路由参数的背离程度,即:

$$\mu = \sum_{k=1}^n |\phi_{jk}^i - \bar{\phi}_{jk}^i| \quad (4)$$

显而易见,在一条链路上,实际路由参数与理想路由参数差异越大,上述参数也越大,即流量均衡性越差。流量调整前的距离和路由参数分别为 $\{D_{jk}^i\}$ 和 $\{\phi_{jk}^i\}$,改变后的距离为 $\{\bar{D}_{jk}^i\}$,进行流量调整得到的路由参数为 $\{\bar{\phi}_{jk}^i\}$ 。

定理 通过 AH 算法进行流量调整前的路由参数差异绝对值和 $\mu \neq 0$; AH 算法并不能稳定地将路由参数调整到最优,即 $\mu \geq 0$ 。

证明:(1)在执行流量调整算法前,假设流量已发生变化,下一跳链路实际转发流量比例与该链路路由参数不一致,即流量调整前的 $\mu > 0$ 。

(2)采用 AH 算法进行流量调整后,得到:

$$\mu = \sum_{k=1}^n |\bar{\phi}_{jk}^i - \bar{\phi}_{jk}^i| \quad (5)$$

式中, $\bar{\phi}_{jk}^i$ 为最优路由参数。由 μ 的定义和绝对值的性质可知,流量调整后的路由参数差异绝对值和 $\mu \geq 0$ 。证毕。

由上述定理可知, AH 算法并不能稳定地将局部流量调整到最优。下面通过一个例子来说明 AH 算法在某些情况下不能将流量参数调整到最优。实例 1 如图 1 所示。

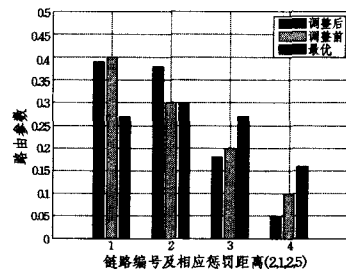


图 1 一个 AH 算法实例

如图 1 所示,执行 AH 算法后,调整后的路由参数并没有达到最优。相反,某些链路的路由参数与最优相差更大,链路 4 的路由参数相差值由调整前的 0.06 增大到 0.11。

因此,针对上述 AH 算法不能稳定地将路由参数调整到最优的问题,本文提出一种 IAH 算法。它充分考虑了各条链路流量的差异,将各链路路由参数调整到最优。算法伪代码如下。

IAH algorithm

输入:每个下一跳的可行距离

输出:调整后路由参数

$\tilde{D}_{jk}^i \leftarrow \tilde{D}_{jk}^i + l_k^i$; //得到每个下一跳的可行距离//
 $1 - \frac{\tilde{D}_{jk}^i}{\sum_{m \in S_j^i} \tilde{D}_{jm}^i}$
 $\tilde{\phi}_{jk}^i \leftarrow \frac{m \in S_j^i}{(|S_j^i| - 1)}$; //计算每个下一跳的最优路由参数//
for $k \in S_j^i$
 $\Delta_k = \phi_{jk}^i - \tilde{\phi}_{jk}^i$; //计算原有参数与最优参数的差//
endfor
array $\{\Delta_k\} \leftarrow \{\Delta_k | k \in S_j^i\}$ 将 $\{\Delta_k\}$ 从大到小排列//
for $k' \leq |S_j^i|$
if $\Delta_{k'} \geq 0$
 $\tilde{\phi}_{jk}^i \leftarrow \phi_{jk}^i - \Delta_{k'}$
elseif $\Delta_{k'} < 0$
 $\tilde{\phi}_{jk}^i \leftarrow \phi_{jk}^i + \frac{|\Delta_{k'}|}{\sum_{\Delta_{k'} < 0} |\Delta_{k'}|} \times \sum_{\Delta_{k'} > 0} |\Delta_{k'}|$
endif
endfor
endfor

执行上述算法,可以使各个下一跳链路的路由参数达到最优,可通过路由参数差异绝对值之和的定义来证明其正确性。算法对于某一目的节点的执行复杂度为 $O(N^i)$;全网执行时,其执行复杂度为 $O(N)$ 。

4 仿真与分析

下面的分析中,验证了上述多径流量调整算法的性能。仿真中选取的网络拓扑如图2所示。

图2中, $R_0 \sim R_8$ 为核心网骨干路由器,每个核心路由器连接一个接入路由器 r 、一个服务器 S 和一个客户主机 C 。连接方式如图中所示,客户主机向其他非直连服务器请求服务。仿真平台为SSFnet,多径生成由OSPF协议完成。设定仿真时间为100s。具体参数配置如下:与接入路由器相连的链路均为100Mbps,骨干网中链路带宽均为10Mbps且权值相同。 $S_1 \sim S_8$ 服务器为不同步,同时开始发包,发送分组长度在[40,1500]字节范围内的非定长包,每个服务器发包速率为[发包速率, 发包速率 $\times (1 + 10\%)$]的均匀分布。在实验中,设置缓存大小为16kB,各链路时延设置为0.1ms。对丢包率性能的比较结果如图3所示。

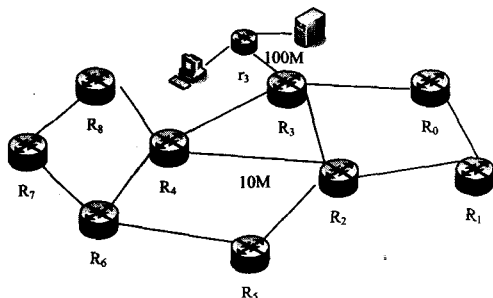


图2 实验网络拓扑

从图3可以看出,随着发包速率的增长,单径路由首先出现丢包,这也是单径路由由网络资源利用不均的体现。发包速率进一步提高,AH算法在发包速率达到0.3Mbps时开始丢包。因此可知,在不丢包的情况下,基于多径的AH算法由于较为充分地使用网络资源,网络吞吐量一般为单径路由的

1.5倍。随着发包速率的进一步提高,IAH算法也开始出现丢包,原因是受网络拓扑的限制,所有下一跳链路中的流量均超出链路的容量,IAH算法也无法完成流量调整。但IH算法较AH算法出现丢包晚,其网络吞吐性能要优于后者。两种多径流量调整算法的丢包率在开始丢包阶段增长较快,之后进入相对稳定的增长阶段,但IAH算法的丢包率性能始终优于AH算法。发包率达到1.6Mbps后,IAH算法的丢包率增加变快,这是由于IAH算法也是基于流的调整,由于流的粒度较大,反而使最优调整对流量均衡性影响变大。但总体来说,IAH算法要优于AH算法。

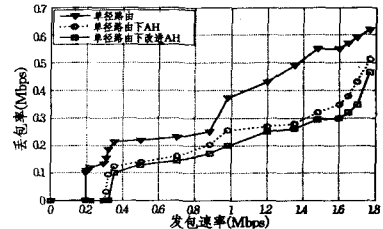


图3 丢包率性能比较

结束语 为了解决多路径间负载均衡问题,提出了很多算法。但多数算法都是基于局部信息来调整网络流量。AH算法利用由全网信息得到的多路径局部调整流量,在一定条件下可以使各下一跳链路中的流量相对均衡,但是,将其他链路的冗余流量全部切换到距离最短的链路上,又将引起新的流量不均衡。同时,调整后的各下一跳流量,其比例与距离比例仍存在一定差距。基于此,本文提出了一种IAH算法,即将流量按可行距离比例分配,距离比例被严格地用于流量调整,这符合网络实际能力,能够获得比AH算法更好的性能。仿真结果表明,IAH算法比AH算法更晚出现丢包,吞吐量也大于AH算法,IAH算法整体上要优于AH算法。

参考文献

- [1] Hahne E L. Routing-robin Scheduling for Max-min Fairness in Data Networks[J]. IEEE Journal on Selected Areas in Communications, 1991, 9(7): 1024-1039
- [2] Shi H, Sethu H. Greedy Fair Queuing: A Goal-oriented Strategy for Fair Real-time Packet Scheduling[C] // Real-time Systems Symposium, Cancun, Mexico: Institute of Electrical and Electronics Engineers Inc, 2003: 345-356
- [3] Cao Z, Wang Z, Zegura E. Performance of hashing-based schemes for internet load balancing[C] // Proceedings of IEEE INFOCOM, NJ: IEEE, 2000: 332-341
- [4] Chim T W, Yeung K L. Traffic distribution over Equal-cost-multi-paths[C] // Proceedings of IEEE International Conference on Communications, NJ: IEEE, 2004: 1207-1211
- [5] Srinivas V. Multipath Routing Mechanisms for Traffic Engineering and Quality of Service in the Internet[D]. California, University of California Santa Cruz, 2001
- [6] Gallager R G. A Minimum Delay Routing Algorithm Using Distributed Computation[J]. IEEE Transactions on Communications, 1977, 25(1): 73-85