

一种基于游标的多径流量分割算法

吴春明¹, 王保进², 陈均华¹, 姜 明³, 张 栋¹

(1. 浙江大学人工智能研究所, 浙江杭州 310027; 2. 国家数字交换系统工程技术研究中心, 河南郑州 450002;

3. 杭州电子科技大学计算机学院 浙江杭州 310018)

摘 要: 多径传输使用多条连接源节点和目的节点的路径进行传输, 在提高资源利用率的同时会引起包乱序问题, 并且存在路径之间的负载均衡问题. 本文提出了一种基于游标的流量分割算法, 游标是当前路径传输延迟与相邻包到达源节点的时间差之间的差值, 它作为选取路径的延时基线来保证包到达的有序性. 游标会随着路径不同或相邻包到达源节点的时间差不同而动态地滑动, 通过动态滑动游标使得尽可能多的路径可用来传输当前包, 从而很好地实现负载均衡. 仿真结果表明, 与已有的保证包有序的算法相比, 本算法使负载更加均衡.

关键词: 多径; 包有序; 负载均衡; 游标

中图分类号: TP393.01

文献标识码: A

文章编号: 0372-2112 (2010) 11-2550-05

A Traffic Splitting Algorithm Based on Nonius in Multi-Path

WU Chun-ming¹, WANG Bao-jin², CHEN Jun-hua¹, JIANG Ming³, ZHANG Dong¹

(1. AI Institute, Zhejiang University, Hangzhou, Zhejiang 310027, China;

2. National Digital Switch System Engineering and Technological R&D Center, Zhengzhou, Henan 450002 China;

3. College of Computer Science, Hangzhou Dianzi University, Hangzhou, Zhejiang 310018, China)

Abstract: Multi-path transmission, using several paths between the source and the destination, will cause packet reordering and present some problems about load balancing while achieving high bandwidth utilization. This paper proposes a traffic splitting algorithm based on nonius. We define the nonius as the deference between the transmission delay of the current path and the time between two successive packets. As a baseline of the delay, the nonius prevents the flow from packet reordering. The nonius can slide dynamically because of different path or the different time between two successive packets. The packet can be transmitted among as many paths as possible because the nonius slides dynamically; as a result, the load balancing can be achieved. The simulation results show that the proposed algorithm gains a prominent improvement in load balancing over previous algorithms, while without reordering is ensured.

Key words: multi-path; without reordering; load balancing; nonius

1 引言

随着网络技术的不断发展、新业务对网络传输性能的要求越来越高, 传统的单径传输由于只利用了多条并行链路中的一条, 不能很好地利用网络资源和降低拥塞; 相对于单径传输, 多径传输能提高传输的可靠性, 充分使用网络资源, 降低拥塞, 提高服务质量, 满足虚拟网构建的需要, 满足一体化逻辑承载网构建的需要^[1~4], 这些都促进了多径传输的研究与应用. 但是多径传输也会引起包乱序和路径之间负载均衡问题. 首先, 由于多径中每条路径的传输延时不一样, 使得到达目的节点的包可能乱序, 而对不同协议和应用将引发不同的问题. 譬如, TCP 将包乱序误认为是拥塞引起的, 发送端将重传并将发送窗口减半, 最终将影响到传输性能^[5,6]; 而

对于 VOIP 这样的 UDP 业务, 乱序将影响到服务质量^[5,7]. 其次, 由于包可以选取任一条路径进行传输, 每条路径所传输包的数量是不确定的, 这将导致各条路径负载不均衡. 因此, 保证包的有序性、同时又兼顾路径的负载均衡问题是多径传输研究的一个重要方面.

按照流分割粒度不同, 目前流量分割与选路常用的方法有: 按包分割、按流分割、按最大最小路径传输延时之差 (Max-Min Delay Difference, MMDD) 分割. 按包进行流量分割^[8]时, 源节点根据各条路径的负载均衡情况选择相应的路径来传输每一个包, 分割粒度为一个包, 从而使各路径间的负载不均衡度不大于一个包的大小, 缺点是会引起乱序. 与按包进行流量分割相反, 按流进行流量分割^[9]时, 同一流中的所有包在同一条路径上传输, 分割粒度为一个流, 保证了包的按序到达, 但由于流

的大小不同,各路径间的负载均衡性得不到保证.综合了按包分割的细粒度和按流分割的包有序性,文[10]提出了按最大最小路径传输延时之差进行流量分割的方法 FLARE(Flowlet Aware Routing Engine),FLARE 将单个流切分成多个分片(flowlet),每个分片按流量均衡算法分配到相应路径,分割粒度为一个分片,由于分片的字节数小于等于所属流的字节数,因此较好地保证了路径间的负载均衡;由于一个分片中的包走同一路径且相邻 flowlet 的间隔大于 MMDD,因此保证了包的有序性.FLARE 是一种较好的流量分割方法^[11],但 FLARE 并不能很好地实现负载均衡,当同一流中的包到达间隔较小而 MMDD 较大时,这种方法的性能就接近按流分割,而流量很大的流通常包到达间隔较小(从后面的 FLARE 分片实验结果可以看出).

为解决 MMDD 过大带来的负载不均衡问题,本文充分利用各条路径间的传输延时差,把同一流中相邻包到达源节点的时间间隔补偿到路径延时上,通过选取游标右边的路径进行传输来保证包的有序性,通过动态地滑动游标使尽可能多的路径可选来均衡负载.

2 游标

为了使源节点的每个包都有机会选择不同的传输路径以实现负载均衡,本文采用了游标的思想来分割流和选路.

2.1 基本概念的定义

在介绍基于游标的选路思想之前先定义几个本文将涉及到的基本概念.

定义 1 (流量分割与选路单元).要实现多径传输,就需要将流量进行分割和选择传输路径,流量分割和选路单元如图 1 所示,源节点将待发送的包按一定算法分配到相应的路径上进行传输,从而有效地满足了负载均衡和使得包按序到达目的节点.

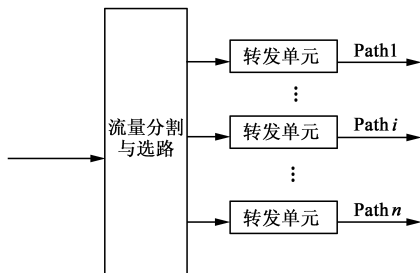


图1 流量分割与选路单元

定义 2 (当前包与前一个包).当前包 p_{curr} 与前一个包 p_{prev} 是指同一个流中相继到达源节点的两个包.

定义 3 (游标).如图 2 所示,设 Path i 是 p_{prev} 所走的路径, D_i 是 Path i 的传输延时, $t_{current}$ 为当前时间, t_{prev_packet} 是 p_{prev} 到达源节点的时间,则 $(t_{current} - t_{prev_packet})$

是该流相邻两个包到达源节点的时间间隔,令 $D_{nonius} = (D_i - (t_{current} - t_{prev_packet}))$,则 D_{nonius} 随着 D_i 和 $(t_{current} - t_{prev_packet})$ 的变化而在传输延时轴上动态滑动,就象游标卡尺上的游标,因此本文称 D_{nonius} 为游标.

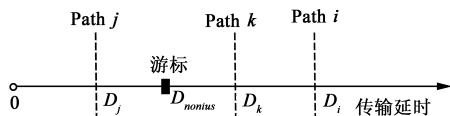


图2 路径传输延时和游标

定义 4 (配额、期望配额和实际配额).配额(Quota)是流量在各条路径中分配的比例,在 n 条路径中,各条路径相应配额一起构成配额向量 $(q_1, q_2, \dots, q_n)$,并且 $\sum_{i=1}^n q_i = 1$.期望配额(Expect Quota, EQ)是希望分配到各条路径的配额,期望配额的选取根据实际情况进行选取,如:当流量在各时间段有明显的特征时,可以按时间段来设置相应的配额;当希望充分体现实际的负载均衡时,可以将感知到的各条路径可用资源以一定的函数映射到配额,使各路径的资源得到充分利用.实际配额(Real Quota, RQ)是该条路径实际传输的流量与所有并行路径传输的总流量之比.

定义 5 (负载失衡度).负载失衡度(Expect-Real Quota Difference, ERQD)是用来衡量负载均衡性的指标,值越大说明均衡性越差,用各条路径的 EQ 与 RQ 之差的绝对值之和 $\sum_{i=1}^n |\Delta q_i|$ 来求得,其中 $\Delta q_i = \frac{\Delta S_i}{S}$, ΔS_i 为路径 i 中期望分配的流量与实际路径中传输的流量之差, S 为并行路径的总流量, ΔS_i 越大表明路径 i 可用带宽越大,所以优先选取 ΔS 大的路径来保证负载尽可能的均衡.

2.2 基于游标的选路思想

在多径传输时,源节点需要为待发送的包选择传输路径.使用游标选取传输路径的主要思想是,根据 p_{curr} 所走路径的传输延时大于 p_{prev} 所走路径的传输延时将不会引起乱序的特点,充分利用各条路径与传输 p_{prev} 的路径之间的传输延时差,把同一流中相邻包到达源节点的时间间隔补偿到路径传输延时上,传输 p_{prev} 的路径的传输延时经过时间间隔补偿变成游标.每次选路前将游标滑动到新的传输延时点,游标右边的路径构成的一个路径子集,选取该路径子集中的路径传输包不会引起乱序(定理 1).再从路径子集中选取可用带宽 ΔS 最大的路径来传输该包,从而使游标右边路径的负载失衡度最小,通过游标的动态滑动,最终达到所有路径间的负载均衡(定理 2).

采用基于游标的选路方法,只有当 p_{prev} 走的是传输延时最长的路径且游标右边只有这条路径时,流才不

会被分片,否则流就可以被分片以实现游标右边各条路径间的负载均衡,这样流被分割成很小的分片后选择在游标右边的路径中可用带宽 ΔS 最大的路径上传输,从而能实现更好的负载均衡。

定理 1 选择游标右边的任何一条路径来传输包不会引起乱序。

证明:如图 3 所示,左边为源节点,右边为目的节点,设 p_{prev} 到达源节点时间是 t_{prev_packet} ,它选择路径 i 进行传输,已知路径 i 的传输延时是 D_i ,则 p_{prev} 到达目的节点的时间是 $t_{prev_packet} + D_i$;同样,设 p_{curr} 到达源节点时间是 $t_{current}$,它在传输延迟大于游标的路径中按负载均衡算法选取路径 k 进行传输,已知路径 k 的传输延迟是 D_k ,则 p_{curr} 到达目的节点的时间是 $t_{current} + D_k$,从下面的推理可知同一流的所有包是按序到达目的节点的:

选择传输延时大于游标的路径,即 $D_k > D_{nonius}$

$$\Rightarrow D_k > D_i - \Delta D_{cur_prev}$$

$$\Rightarrow D_k > D_i - (t_{current} - t_{prev_packet})$$

$$\Rightarrow t_{current} + D_k > t_{prev_packet} + D_i$$

$\Rightarrow p_{curr}$ 比 p_{prev} 更晚到达目的节点,即包按序到达

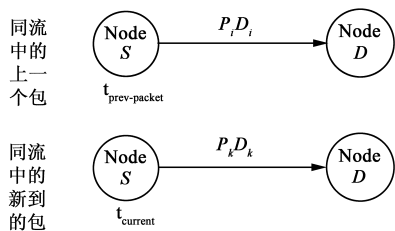


图3 同一流中相邻包的传输

定理 2 动态滑动游标能实现负载在所有路径间均衡。

证明:如图 2 所示,游标左右滑动时,越往右边的路径位于游标右边的次数越多,进入可选路径子集的概率也就越大,被选中用来传输包的机会也就越大,这将使得偏右边的路径过载而偏左边的路径负载不足.这个结果又将反作用于游标,在过多选取靠右边路径后,越左边的路径的可用带宽 ΔS 就越大,因此会选取游标右边离游标近的路径,在图 2 中即为 $D \leq D_i$ 的路径 k ,这样游标就会逐渐向左端滑动,使得左边的路径也有同样的机会被选取,从而很好地实现负载在所有路径间均衡。

3 基于游标流量分割算法及其稳定性分析

基于游标的流量分割(Traffic Splitting Based on Nonius, TSNB)算法维护路径信息表和游标表,分别保存各条路径的相关信息和游标的相关信息.其中,路径信息包含分配给该路径的流量、期望配额和传输延时,游标

表包含流标识、该流前一个包所走路径和该流前一个包到达的时间。

当流的 $(t_{current} - t_{prev_packet})$ 大于 $MMDD$ 时,选取任何一条路径来传输该流的包都不会乱序,所以为了降低算法的空间复杂度和时间复杂度,将定期删除游标表中 $(t_{current} - t_{prev_packet}) > MMDD$ 的项。

为了保证包的有序性,可用路径只能在传输延迟大于游标的路径中选取;同时,为了实现负载尽可能的均衡,需要在选路前将游标移动到新的位置,并且选取可用路径中剩余带宽 ΔS 最大的路径,TSBN 算法的流程图如图 4 所示。

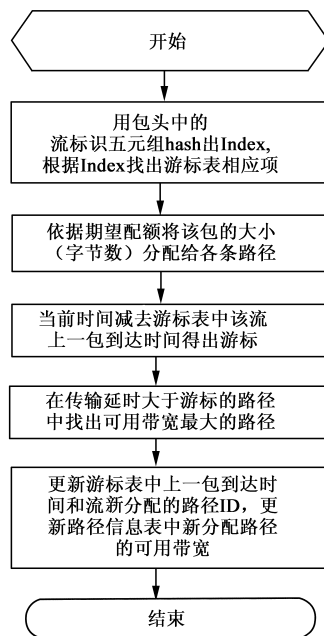


图4 TSNB算法处理流程

在多径路由网络中,用户都倾向于选择拥塞较小的路径,这与拥塞控制算法相互作用就会导致流量在路径间抖动,从而引起不稳定^[12].与这种由用户各自选择路径的分散式管理不同,TSBN 算法在源节点对相关路径上的流量进行集中式管理,按负载均衡度将流量分配到相应路径,不会出现流量在路径间抖动情况,因此 TSNB 算法是稳定的。

4 算法的实验仿真与分析

本文采用跟踪驱动模拟方法进行实验,实验所用 trace 文件采集自现实的骨干链路。

4.1 实验场景

实验拓扑结构如图 5 所示,源节点和目的节点之间有 5 条并行的路径,在传输数据前对各条路径进行资源预留,因此各条路径传输延迟抖动忽略不计.设路径的传输延时依次为 $(20\text{ms}, 25\text{ms}, 30\text{ms}, 35\text{ms}, 45\text{ms})$, $MMDD$ 为 25ms ^[10],流量在各路径上平均分配,各为 20% 的配额。

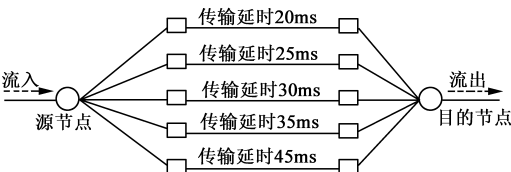


图5 实验拓扑结构

4.2 实验方法

向量(源 IP,目的 IP,源端口,目的端口,协议类型)唯一标识一个流,但由于将整个向量作一个流标识符进行处理非常复杂,实验中采用 CRC16 对向量进行 hash^[13],将 hash 结果用于标识流。

4.3 实验结果分析

使用 TSBN 算法时,包到达间隔不受 *MMDD* 约束,即使同一流中相邻两个包到达间隔无穷小,比该流前一个包传输延时大的路径还是始终位于游标右边,而新到的包选取游标右边的路径不会引起乱序;而使用 FLARE 算法时,只有包到达间隔大于 *MMDD* 时才会对流进行分隔,所以 TSBN 算法对流的分割粒度比 FLARE 算法更细。从图 6 中可以看出,无论是小流还是大流,TSBN 算法都比 FLARE 算法更多次地分割流,即流分割的粒度更细,并且随着流大小的增加,TSBN 算法对流的分割次数呈现出上升趋势,而 FLARE 算法在流量大于 13MB 后对流的分割次数呈下降趋势。流分割的粒度越细,就越能实现负载均衡^[10],因此 TSBN 算法比 FLARE 算法能更好地实现多径中的负载均衡。

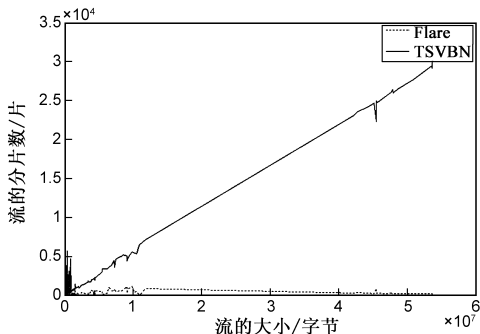


图6 流的大小与分片数

表 1 分片到达率

Trace	到达率(次/秒)	
	FLARE	TSBN
CENIC-I	4932.33	19836.25
CESCA-I	57001.8	88576.9
NCAR-I	12919.12	227880.22

每秒内分片到达次数越多,负载在路径间均衡次数也就越多,路径间的负载也就越均衡。从表 1 可以看出,对于实验所用的三个 Trace

数据,TSBN 的分片到达率都高于 FLARE 的分片到达率,即 TSBN 能使负载更加均衡。

分片的累积分布函数(Cumulative Distribution Function, CDF)是分片按字节数从小到大排序后,分片字节数累计占总流量的比例,是评价流分割粒度的一个直

接指标。对比图 7 中的(a)和(b),TSBN 算法下 65% 的流量被分割成不大于 1.5kB 的分片传送,而 FLARE 算法下不大于 74kB 的分片只占传输流量的 53%,这也进一步表明了 TSBN 算法将流分得更多更细,能更好地实现负载均衡。

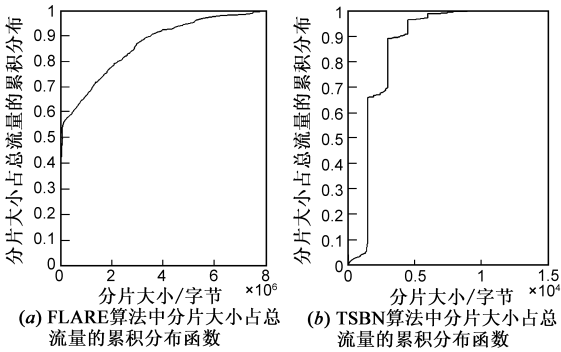


图7

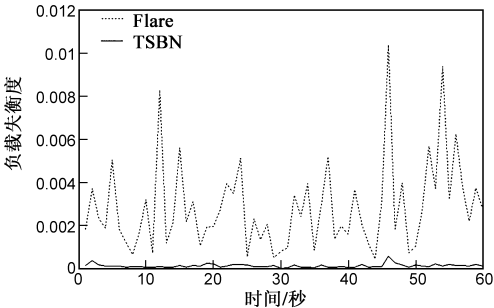


图8 负载失衡度

TSBN 算法在每个包到来时都会游标右边的路径中进行负载均衡,而 FLARE 算法只有当相邻包发送间隔大于 *MMDD* 时才会各路径中进行负载均衡,因此 TSBN 算法能更好地实现负载均衡。负载是否均衡,各路径负载失衡度是一个直接的评价指标,从图 8 可以看出,每个时刻 TSBN 算法的负载失衡度都小于 FLARE 算法的负载失衡度,这也表明 TSBN 算法的负载均衡性更好。

类似于 FLARE 算法,TSBN 算法定时清除游标表中包到达间隔大于 *MMDD* 的项,所以两种算法下表大小相同。实验中流信息表最大为 296 项,只需要不到 4kB 的空间来保存游标表和路径信息表,因此 TSBN 算法的额外开销很小。

5 结束语

本文引入游标来保证包的有序性,通过选取游标右边可用带宽最大的路径实现游标右边各条路径间的负载均衡,通过滑动游标来实现所有路径间的负载均衡。仿真实验将在实际网络中监测得到的 trace 文件作为 TSBN 算法的输入,验证了 TSBN 算法能以小的开销将流分为更多分片,且将能流量以接近约定的配额分配到各条路径。

参考文献:

- [1] Umar Javed, Martin Suchara, Jiayue He, Jennifer Rexford. Multipath protocol for delay-sensitive traffic [A]. Communication Systems and Networks and Workshops (COMSNETS 2009) [C]. Bangalore: IEEE Press, 2009. 1 – 8.
- [2] 牛志升, 段翔, 刘进. MPLS 网络中保证服务质量的多径路由选择策略[J]. 电子学报, 2001, 29(12): 1638 – 1641.
Niu Zhi-sheng, Duan Xiang, Liu Jin. A QoS-guaranteed multipath routing policy for MPLS networks [J]. Acta Electronica Sinica, 2001, 29(12): 1638 – 1641. (in Chinese)
- [3] Minlan Yu, Yung Yi, Jennifer Rexford, Mung Chiang. Rethinking virtual network embedding: Substrate support for path splitting and migration [J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 17 – 29.
- [4] 王浩学, 汪斌强, 于婧, 姜明. 一体化承载网络体系架构研究[J]. 计算机学报, 2009, 32(3): 371 – 376.
Wang Hao-xue, Wang Bin-qiang, Yu Jing, Jiang Ming. Research on architecture of universal carrying network [J]. Chinese Journal of Computers, 2009, 32(3): 371 – 376. (in Chinese)
- [5] Arthur CM, Lehan A, Harle D. Keeping order: Determining the effect of TCP packet reordering [A]. Third International Conference on Networking and Services (ICNS) [C]. Athens: IEEE Press, 2007. 116 – 116.
- [6] Weiguang Shi, Lukas Kencl. Sequence-preserving adaptive load balancers [A]. Proceedings of the 2006 ACM/IEEE symposium on Architecture for Networking and Communications Systems [C]. New York: ACM Press, 2006. 143 – 152.
- [7] Michael Laor, Lior Gendel, et al. The effect of packet reordering in a backbone link on application throughput [J]. IEEE Network, 2002, 16(5): 28 – 36.
- [8] M Shreedhar, George Varghese. Efficient fair queuing using deficit round-robin [J]. IEEE/ACM Transactions on Networking (TON), 1996, 4(3): 375-385.
- [9] Brice Augustin, Xavier Cuvelier, et al. Avoiding traceroute anomalies with Paris traceroute [A]. Internet Measuremet Conference [C]. New York: ACM Press, 2006. 153 – 158.
- [10] Srikanth Kandula, Dina Katabi, Shantanu Sinha, Arthur Berger. Dynamic load balancing without packet reordering [J]. ACM SIGCOMM Computer Communication Review, 2007, 37(2): 51 – 62.
- [11] Jiayue He, Jennifer Rexford. Toward internet-wide multipath routing [J]. IEEE Network, 2008, 22(2): 16 – 21.
- [12] Thomas Voice. Stability of multi-path dual congestion control algorithms [J]. IEEE/ACM Transactions on Networking (TON), 2007, 15(6): 1231 – 1239.
- [13] Zhiruo Cao, Zheng Wang, Ellen Zegura. Performance of hashing-based schemes for Internet load balancing [A]. INFOCOM 2000, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings, IEEE [C]. Tel Aviv: IEEE Press, 2000. 332 – 341.

作者简介:



吴春明 男, 1967 年生于浙江萧山, 博士, 浙江大学计算机学院教授、博士生导师, 主要从事人工智能、可重构网络技术、新一代高可信网络等方面的科学研究工作。

E-mail: wuchunming@cs.zju.edu.cn



王保进 男, 1974 年生于河北保定, 博士, 工作于国家数字交换系统工程技术研究中心, 主要研究领域为路由交换技术, 嵌入式系统实时系统。