一种在 MPLS 网络中支持 QoS 的流量工程算法

崔丙锋 杨 震 丁 炜
(北京邮电大学宽带通信网实验室 北京 100876)

摘 要：该文提出了一种改善 QoS 的队列调谐算法(Queue Tuning Algorithm：QTA)，该算法应用于多协议标签交换(MPLS)流量工程。该算法通过把不同类别的业务，如传统的尽力而为(Best-Effort)业务和具有 QoS 需求的业务，映射到不同队列，然后根据链路的状况以及业务的 QoS 需求来进行队列调谐。QTA 在实现负载均衡的同时又支持业务流的 QoS，优化网络资源的利用。仿真表明该算法在 MPLS 网络中运行良好。

关键词：多标签协议交换，QoS，队列调谐算法
中图分类号：TP393.4 **文献标识码**：A **文章编号**：1005-5896(2005)08-1324-04

A Traffic Engineering Algorithm Supporting QoS in MPLS Networks

Cui Bing-feng Yang Zhen Ding Wei
(*Broadband Communication Networks Lab, BUPT, Beijing* 100876, *China*)

Abstract In this paper, a Queue Tuning Algorithm (QTA) supporting QoS for traffic engineering in Multiprotocol Label Switching(MPLS) networks is proposed. The algorithm aims at mapping different traffics, consisting of both best-effort traffic and traffic with QoS requirement, onto different queues while preserving the order of packets within a flow in traffic. The algorithm supports QoS while realizing load balancing to optimize network resource utilization. Simulation indicates that the algorithm works well in MPLS networks.

Key words Multiprotocol label switching, QoS, Queue tuning algorithm

1 引言

随着 Internet 的高速发展，用户对业务的 QoS 要求也越来越高。传统基于尽力而为提供服务的 Internet 架构已经逐渐不能满足现实发展的要求。因此，运营商不得不面对如何才能高效地提供各种不同 QoS 业务以满足不同需求这样的难题。Internet 流量工程(Traffic Engineering，TE)就是为解决这一难题而提出的一项关键技术。根据 IETF 的定义，Internet 流量工程是指能够处理 IP 网络性能评估以及性能优化的一种网络工程[1]。支持显示路由的多协议标签交换(MultiProtocol Label Switching，MPLS)技术的出现及其日益成熟，为流量工程的实现提供了一种基本的机制[2]，也使得更加高效的 Internet 流量工程成为可能。

MPLS 网络中，为了保证链路的可靠性，在入口 LSR (Label Switching Router)和出口 LSR 之间通常要建立多条并行的 LSP(Label Switched Path)。为了实现高效地利用多条 LSP，Elwalid 以及 Widjaja 等人首次提出了 MPLS 自适应流量工程(MPLS Adaptive Traffic Engineering，MATE)的概念[3]。整个 MATE 架构如图 1 所示。但是 MATE 没有考虑各个数据流的不同特点以及 QoS 需求。如何充分利用多条 LSP 以满足不同业务 QoS 的需求成为一个难题。采用传统的负载均衡算法，虽然可以充分利用多条并行 LSP，但传统的负载均衡算法只是根据流的数量来分配 LSP，完全不考虑流的特点，因此也就不能满足各种流的 QoS 需求。本文正是在 MATE 架构基础上，提出一种队列调谐算法(Queue Tuning Algorithm，QTA)。QTA 根据 LSP 的不同状态动态地分配流量以提高 LSP 的利用率，实现了负载均衡；同时将不同的队列映射到不同的 LSP，满足不同流的 QoS 需求。

2 相关研究

如图 1 所示，整个 MATE 架构可以分成 3 部分：流量分割、流量分配以及测量分析。流量分割模块根据一定的规则将到达的数据包分成不同的部分；而流量分配模块则是根据测量分析的结果，按照一定的规则将这些分割好的数据包分配到各个 LSP 上；测量分析模块主要完成对各个 LSP 参数(如：时延、带宽等)的测量。由此可以看出流量分割是整个系统的第一步，也是比较关键的一部分。在流量分割中有许

2004-03-05 收到，2004-09-16 改回
国家自然科学基金(20020013011)和教育部博士专项基金(20020013011)资助课题

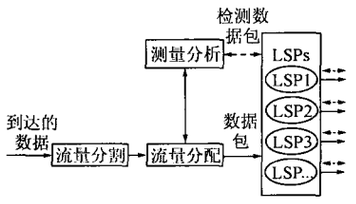

图1　MPLS 自适应流量工程整体架构

多基本的要求是必须要考虑的[4]。由于流量分割是对每一个数据包都要进行，而且一般是根据数据包的头来进行分割，因此必须考虑由此带来的资源消耗。同时，流量分割算法必须要保证同一 TCP 数据流的报文不会产生乱序。由于 TCP 的拥塞控制机制，报文乱序会导致整个系统吞吐量下降。

文献[4]提出并分析了多种流量分割的算法，并指出在各种流量分割算法中，基于 Hash 的流量分割算法具有最好的性能，特别是采用五元组(源 IP、目的 IP、源端口、目的端口以及协议 ID)作为输入的 Hash 具有更好的性能。在各种 Hash 分割方案中，基于 CRC16 的 Hash 方案虽然比较复杂，但是目前的高速网络已经在硬件上实现了 CRC16，所以这种方案已经得到应用。文献[4]采用实际的网络跟踪数据对基于 CRC16 的 Hash 性能进行了深入的分析。在基于 CRC16 的 Hash 方案中，流量分割模块采用上述的五元组作为输入进行 CRC16 运算，然后再根据链路的数目 N 进行取模运算。整个 Hash 函数可以表示为

$$H(\cdot)=\mathrm{CRC16}(\mathrm{srcIP}\cdot\mathrm{destIP}\cdot\mathrm{srcPort}\cdot\mathrm{destPort}\cdot\mathrm{protocolID})\ \mathrm{Mod}\ N \quad (1)$$

上述所介绍的各种方案都是仅仅从负载均衡的角度来考虑对流量进行分割，都没有考虑每个数据流的具体特征。只是从数据流的数量角度来实现分割，没有考虑数据流的 QoS 需求。因此，上述的各种流量分割算法都不能满足数据流的 QoS 需求。

流量分配是图 1 所示结构中的另一个重要模块。理想的流量分配模型如图 2 所示。假设到达的流量是无限可分的，链路的数目是 N。第 i 条链路的容量是 μ_i，$S_i(\tau,t)$ 是链路 i 在 $[\tau,t]$ 时间段内转发的流量。从理想负载均衡系统的观点来看，流量分配应该根据各条链路的容量按比例地分配流量。因此在任一时间段 $[\tau,t]$ 内，理想流量分配模型分配流量应该满足如下公式：

$$[S_i(\tau,t)]/[S_j(\tau,t)]=(\mu_i)/(\mu_j) \quad (2)$$

实际的流量分配是不可能严格按照式(2)分配的。因为流量至少是以一个数据包为单位的，不是无限可分的。但理想流量分配模型是本文 QTA 算法的基础，并采用基于 CRC16 的 Hash 分割方案。

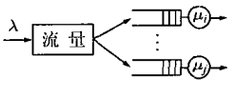

图2　流量分配参考模型

3　队列调谐算法

3.1 算法描述

QTA 的目的是在实现负载均衡的同时又能满足不同数据流的 QoS 需求。QTA 只运行在多协议标签交换网络中的边缘路由器上，对中间路由器不需要作任何的修改，而且 QTA 也不需要了解中间路由器的任何信息。在详细描述 QTA 之前，本文认为对于 QTA 来说，两种类型的流量与一种类型的流量有着本质区别，而对于两种类型以上的流量与两种类型的流量没本质区别。因此为了简单起见，本文假设有两种类型的流量：尽力而为(Best Effort, BE)的流量和快速转发(Expedited Forwarding, EF)的流量。

整个 QTA 实现的流程如图 3 所示。其基本思路是对流量进行分类，根据基于 CRC16 的 Hash 结果放入不同的队列，然后动态改变队列到 LSP 的映射，实现负载均衡以及满足 QoS 的需求。QTA 的实现具体流程如图 4 所示。整个算法运行步骤如下：

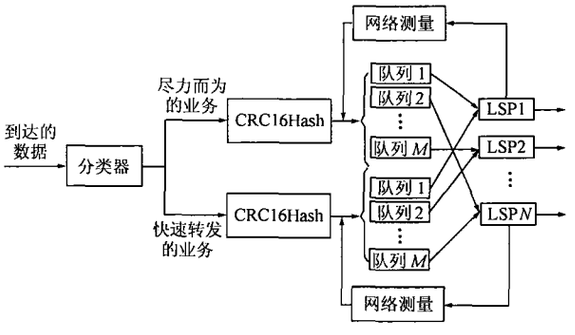

图3　QTA 算法实现架构

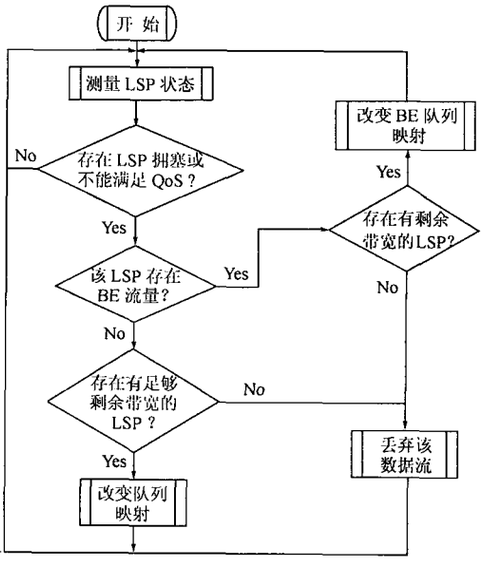

图4　QTA 实现流程

第 1 步　在对各条 LSP 状态一无所知的情况下，按照传统 CRC16 的 Hash 结果进行流量分配。

第 2 步　入口 LSR 向各条 LSP 发送检测分组，检测分组从出口 LSR 返回后计算各条 LSP 的时延，同时对各条 LSP 的利用率进行测量。如果不存在链路参数超出预定阈值的 LSP，则等待一个测量周期后继续第 2 步。

第 3 步　找出一条参数超出预定阈值最多的 LSP，检查该条 LSP 上是否存在 BE 类业务。因为 QTA 原则上尽量不转移 EF 类的业务，以避免对 EF 类业务产生时延抖动。如存在 BE 类业务，检查是否存在有足够剩余带宽的 LSP。如存在则改变一个 BE 队列的映射，把该队列上的 BE 类业务转移到该 LSP 上。如不存在有足够剩余带宽的 LSP，则该 LSP 上某个 BE 队列的数据会被丢弃。如果不存在 BE 类业务，使用同样的方式对 EF 类业务进行转移。

第 4 步　等待一个测量周期以后，重新回到第 2 步。

3.2 算法分析

假设 BE 队列和 EF 队列的个数都为 M(可以不一样)，LSP 的个数为 N。从上面的分析可以看出队列调谐的颗粒度由 $\alpha = N/M$ 来决定。M 越大，调谐的颗粒度越细腻，同时达到稳定所需时间越长。因此一般采用 α 在 0.2～0.5 之间。对 LSP 参数的测量一般是测量时延。文献[3]指出在链路各种参数的测量中，时延是最容易测量，而且测量结果是最可靠的。但是为了尽量避免测量结果的随机性，可以采用文献[5]给出的一种加权低通滤波器算法进行测量。同时，测量周期决定了 QTA 的反应时间。通过上面算法描述可以看出，QTA 首先采用 CRC16 的 Hash 对流量进行分割，因此能够保证数据报文的次序，实现传统的流量均衡，充分利用多条 LSP。QTA 还能够根据不同数据流的 QoS 需求，动态调整队列映射，实现数据流的 QoS 需求。同时由于其仅仅在入口和出口 LSR 上运行，因此对整个骨干网络不需要做任何修改。

3.3 仿真分析

在仿真中采用下面几种指标来衡量 QTA 的性能：(1)排队时延，在实际的应用系统中，排队时延随时间的变化而变化，它是网络中影响数据流 QoS 的一个重要因素。(2)队列长度，队列长度直接反映了负载均衡的效果。(3)LSP 利用率，利用率反映当前 LSP 是否发生拥塞。仿真所用网络拓扑如图 5 所示。$S_0, S_1, \cdots, S_{19}$ 为 20 个数据源，D_0, D_1 为目的节点。本文认为仿真中使用两条 LSP 与只有一条 LSP 有着本质区别，而两条以上 LSP 与两条 LSP 并没有本质区别。因此如图 5 所示仿真假定在 LSR1 和 LSR5 之间有两条 LSP：LSP1，LSP2。它们是通过相关协议(标签发布协议、受限标签发布协议或扩展资源预留协议等)已经成功建立。LSP1 的容量为 20Mbps，LSP2 的容量为 40Mbps。S_i 数据源产生 BE 和 EF 两种类型的数据，并且数据包长度服从均值为 93byte 的指数分布。D_0，D_1 负责接收数据。LSR1 上到达的数据包随时间分布如图 6 所示。仿真持续时间为 5s。

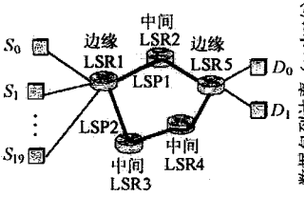

图 5　仿真网络拓扑图

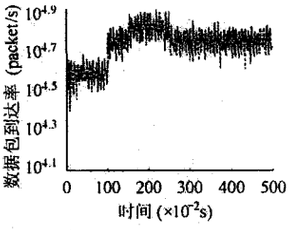

图 6　LSR1 上数据包到达率分布示意图

为了更好地理解 QTA 算法，首先描述在 LSR1 上没有 QTA 情况下，而是运行传统的基于 CRC16 的 Hash 负载均衡算法的情况。仿真条件与上述相同。从图 6 中可以看出，数据包从 1s 开始有明显的增长。我们假设增加的数据包属于 LSP1 上的数据流。例如，在 1s 时刻 S_0 和 D_0 之间有一视频流开始传送，而根据五元组计算的结果，该数据流使用的是 LSP1 链路。因此，在这种情况下，根据 CRC16 的 Hash 负载均衡算法，新增的数据将被分配到 LSP1 链路上。图 7 反映了 LSP 的利用率。从图 7 中可以看出，从 1s 开始 LSP1 出现了拥塞现象，而 LSP2 依然有充足的剩余带宽。同样的情况也可以从队列长度和队列时延上看出来。图 8 反映的是运行传统 CRC16 的 Hash 负载均衡情况下的两条 LSP 的平均队列长度，而图 9 反映的则是相同情况下的队列时延。从上面的仿真结果可以看出，传统基于 CRC16 的 Hash 负载算法均不能解决上述出现的问题，而且更不可能考虑数据流量的 QoS。

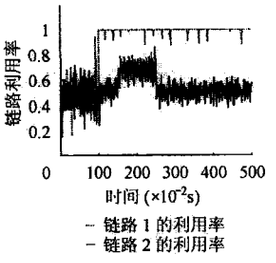

图 7　运行传统 CRC16Hash 情况下的 LSP 利用率

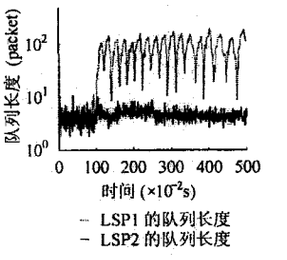

图 8　运行传统 CRC16Hash 情况下的队列平均长度

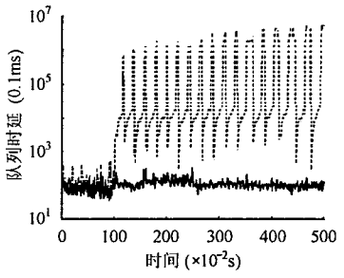

图 9　运行传统 CRC16Hash 情况下的时延

QTA 算法不仅实现负载均衡，解决上述问题，而且通过动态改变队列映射，满足数据流的 QoS 需求。仿真假设如果 EF 的队列时延大于 30ms 或者任一条 LSP 的利用率大于 0.85，QTA 就会启动。在这里的两个阈值是可以改变的，可以由 ISP 预先定义。仿真所用的其它条件如上面所述。图 10，图 11，图 12 显示了 QTA 是如何工作的。当流量在 1s 之后突然增加，网络测量模块会很快检测到这种变化(检测所需时间与检测周期密切相关[3])。当 QTA 发现 LSP1 的链路利用率已经超过预先定义的阈值时，便改变队列映射，将 LSP1 上的 BE 流量转移到 LSP2 上。但是随着越来越多的 BE 流量从 LSP1 上转移到 LSP2 上，LSP2 的链路利用率超出了其预先定义的阈值，这在图 10 中大约 1.5s 处可以看到。同样，网络测量模块也检测到这种情况，QTA 又开始把 LSP2 上的 BE 流量转移到 LSP1 上。最终，QTA 达到了新的平衡，系统达到新的稳定状态。图 11 和图 12 分别给出队列平均长度和队列平均时延。在图 12 中可以看出，在 1~1.8s 时间段内，LSP1 上的 BE 数据流的队列时延为 0，这说明在这段时间内，LSP 上已经不存在 BE 数据流。也就是说，为了满足 EF 数据流的 QoS 需求，QTA 将所有的 BE 数据流从 LSP1 转移到 LSP2 上。通过与传统基于 CRC16 的 Hash 负载均衡算法相比较，可以看出 QTA 具有明显的优势。

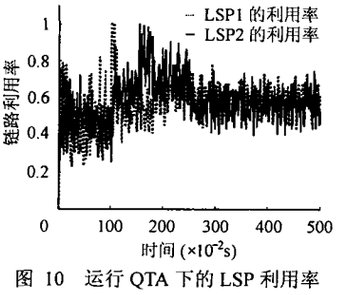

图 10　运行 QTA 下的 LSP 利用率

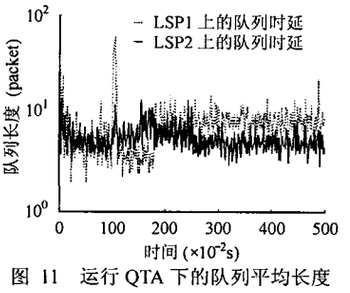

图 11　运行 QTA 下的队列平均长度

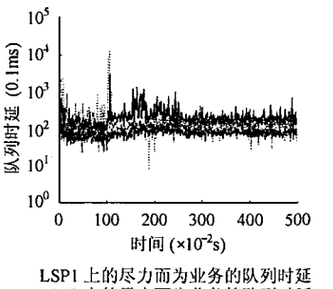

图 12　运行 QTA 下的平均队列时延

4　结束语

本文提出了一种支持 QoS 需求的动态负载均衡算法，它不仅能够完成传统负载均衡算法所实现的功能，而且更好地支持数据流的 QoS 需求。仿真也表明 QTA 算法比传统的负载均衡算法具有明显的优势。QTA 不需要对中间路由器上的协议软件作任何修改，仅在边缘路由器上运行，因此简单易于实现。但是本文的仿真还没有过多考虑实际中的运行环境，下一步的工作是在实际的网络环境中检验 QTA 算法，进一步完善该算法。

参 考 文 献

[1] Awduche D, Elwalidn A, Widjaja I, Xiao X. A framework for Internet traffic engineering. Internet draft.

[2] <draft-ietf-tewg-framework-01.txt>, Mar. 2000.

[3] Rosen E, Viswanathan A, Callon R. Multiprotocol Label Switching Architecture. IETF RFC3031, January 2001.

[4] Elwalid A, Jin C, Low S, Widjaja I. MATE: MPLS adaptive traffic engineering. Proc. INFOCOM'01, Anchorage, Alaska, USA, Apr. 2001: 1300 – 1309.

[5] Cao Zhiruo, Wang Zheng, Zegura Ellen. Performance of hashing-based schemes for Internet load balancing. Proceedings of IEEE/INFOCOM, Tel Aviv, Israel, March 26-30, 2000: 332 – 341.

[6] Gao Deyun, Shu Yantai, Liu Shuo, Yang W W. Delay-based adaptive load balancing in MPLS networks. Proceedings of International Conference on Communications, New York, NY, USA, April28-May2, 2002: 1184 – 1188.

崔丙锋：男，1976 年生，博士生，研究领域包括流量工程及其关键技术、移动 IP 与 MPLS 融合技术等.

杨　震：男，1976 年生，博士生，研究领域包括 MPLS 关键技术、高速路由与交换技术等.

丁　炜：男，1935 年生，教授，博士生导师，研究领域为宽带通信网.