

基于区分服务的主动队列管理算法研究

张 振,周井泉

(南京邮电大学 电子科学与工程学院,江苏 南京 210003)

摘 要:分析了几种主动队列管理算法。RIO算法是用于支持区分服务确保转发逐跳行为的主动队列管理算法,它是对RED算法的简单扩充,但是该算法的性能对配置参数敏感。PI算法是基于控制论的主动队列管理算法,具有队列长度抖动小的特点。PIP算法是PI算法的改进,比PI算法具有更快的收敛速度。为了更好地满足AF PHB的要求,基于PIP算法,结合三色标记器的功能,提出一个新的主动队列管理算法PIPGYR(PIP with Green&Yellow&Red)。通过仿真验证,该算法队列长度抖动小,同时能够保护高优先级分组。

关键词:主动队列管理;区分服务;确保转发

中图分类号:TP393

文献标识码:A

文章编号:1673-629X(2011)02-0109-03

Research on DiffServ-Based Active Queue Management Algorithm

ZHANG Zhen, ZHOU Jing-quan

(College of Electronic Science and Engineering, Nanjing University
of Posts and Telecommunications, Nanjing 210003, China)

Abstract: Analyze some major active queue management (AQM) algorithm. RIO is the active queue management (AQM) algorithm for DS (Differentiated Services) AF PHB (Assured Forwarding Per Hop Behavior), which is based on random early detection. As the performance of RED is sensitive to configuration parameters, the performance of RIO (RED with IN and OUT) is also sensitive to configuration parameters. PI (Proportional Integral) is the AQM algorithm based on control theory and has smaller queue length oscillation than RED. PIP improves the convergence speed of PI. To meet with requirements of AF PHB, design a new AQM for DiffServ AF PHB based on PIP, cooperating with three colors maker which is called PIPGYR (PIP with Green&Yellow&Red). The simulation results indicate that this algorithm has smaller queue oscillation than RIO and can protect high-priority packets.

Key words: active queue management; differentiated services; assured forwarding

0 引言

在因特网协议服务质量(IP QoS, Internet Protocol Quality of Service)呼声越来越高的今天,实现QoS的重要机制之一是主动队列管理(AQM, Active Queue Management)技术。作为基于路由器拥塞控制^[1]的关键技术,AQM与TCP端到端的拥塞控制相结合是解决目前Internet拥塞控制问题的一个主要途径。因为AQM既可以通过维持稳定且大小适当的队列长度来减小丢包率和端到端延迟,提高吞吐量等支持IP QoS,又可以通过对不同业务采用不同的AQM机制来实现区分服务(DiffServ)。

DiffServ网络体系结构^[2]是支持下一代网络服务质量的基础。DiffServ定义了一个相对简单且粒度较粗的控制网络,它将相同特性应用需求的若干业务流聚集在一起,为整个汇聚流^[3]服务,具有很好的扩展性,从而能够为大型网络提供QoS保障。

IETF区分服务工作组定义了两类PHB:EF PHB^[4]和AF PHB^[5]。AF PHB为IP分组提供4组不同的转发特征,对应4种不同数量的转发资源(如缓冲区和带宽等),并且为每个分组指派三个不同的丢弃优先级。AF类逐跳行为的特点是在总的流量不超过预设速率的前提下以更大的可能性转发分组,即使在网络拥塞的情况下,也能保证用户仍然享有预约的带宽。AF PHB内有四个独立的AF类^[6],提供的都是保证最小带宽的服务。在AF中,每类AF_i($i = 1 \sim 4$)从低到高有三个丢弃优先级DP0、DP1、DP2。按双速率三色标记器taTCM进行标记^[7]时,当业务流到达率小于承诺的目标速率时,其中的包将标记为DP0(绿色)——最低丢弃速率时;若小于峰值目标速率大于承

收稿日期:2010-06-08;修回日期:2010-09-12

基金项目:国家“863”计划基金资助项目(2009AA01Z202);江苏省科技支撑项目(BE2008134)

作者简介:张 振(1984-),女,湖南长沙人,硕士,主要研究方向为可信网络的QoS路由技术;周井泉,教授,硕士生导师,研究方向为研究通信网络中所涉及的优化、信息管理和控制问题。

诸目标速率时,标记为 DP1(黄色);大于峰值目标速率时,标记为 DP2(红色)—最高丢弃优先级。在网络资源不足时,先丢弃红颜色包,再丢弃黄颜色包,最后才丢弃绿颜色包。

PI^[8]算法以控制论为基础,报文的丢失概率是瞬时队列长度的函数。PI 算法的队列长度抖动比较小,但其收敛速率较慢,但可以通过 PIP^[9]算法反馈补偿改善。中文引入了主动队列管理的概念,分析、讨论了 RIO,PIPIO 两种算法的实现原理。为了满足 AF PHB 对不同优先级别包的处理要求,着重研究了 PIP 与三色标记器相结合的技术,提出了 PIPGYR 算法的改性思想,并给出了实现。通过仿真验证,该算法队列长度抖动小,同时能够保护高优先级^[9]分组。

1 主动队列管理算法

1.1 RIO 算法

RIO 主动队列管理算法^[10]的基本思想是:边界路由器监视每个进入网络的用户数据流,根据它们的服务规格对包进行标识,超出预约带宽的流量标为 OUT(out of profile),预约带宽以内的流量标为 IN(in profile)。

RIO 算法如下:

(1)对于每个到达分组:

如果标记为 IN 包,则计算 IN 包的平均队列长度 avg_in ;

如果标记为 OUT 包,则计算 OUT 包的平均队列长度 avg_out 。

(2)比较平均队列长度和阈值的关系:

如果标记为 IN 包,当 $avg_in < min_in$ 时,则不丢弃报文;当 $min_in < avg_in < max_in$ 时,以 $p_{drop-in}$ 概率丢弃报文;当 $avg_in > max_in$ 时,则以概率 1 丢弃报文;

如果标记为 OUT 包,当 $avg_out < min_out$ 时,不丢弃报文;当 $min_out < avg_out < max_out$ 时,报文以 $p_{drop-out}$ 概率丢弃;当 $avg_out > max_out$ 时,以概率 1 丢弃报文;并且 OUT 包的丢包概率与总平均队长 avg_Q 有关,而 IN 包的丢包率与 IN 包的平均队长 avg_in 有关,同时 max_in_p 小于 max_out_p ,所以如果 RIO 检测到早期拥塞时,首先丢弃 OUT 包,当 OUT 包全丢弃时,如果继续拥塞,才开始丢弃 IN 包。从而在一定程度上保护了 IN 包。

RIO 主动队列管理算法虽然稳定性及鲁棒性较好,能有效地控制网络上的拥塞,但是 RIO 算法在不正确的参数配置时,会引起带宽利用率降低和平均队列长度的抖动增大;在配置的参数固定时,OUT 参数和 IN 参数之间的关系不好确定,不适应网络环境的动态变化。

1.2 PIPIO 算法

PIP 算法的报文丢弃概率函数如下:

$$p(k) = p(k-1) + \frac{\delta t}{T}(q(k) - q_0) + \left[\frac{\tau}{T} - k_h\right][q(k) - q(k-1)] \quad (1)$$

上式中, q_0 是目标队列长度, τ, T, k_h 是参数配置。S 为采样周期, p 在 $K * S$ 时刻进行更新。

PIPIO 算法的思想是:在计算总的报文丢弃概率后,再丢弃 IN/OUT 报文的丢弃概率,从而保护了 IN 报文。

PIPIO 算法由两个独立的算法组成:第一部分为 PIP 计算总的报文丢失概率,第二部分为 LDP(Loss Probably Divider)计算报文 OUT 和 IN 的丢弃概率。

PIPIO 算法如下:

当一个报文到达时,如果其为 OUT 报文,更新为 OUT 报文与其对应的到达速率 $rate_{out}$;否则,更新为 IN 报文与其对应的到达速率 $rate_{in}$;由 PIP 模块计算 p ;由 LDP 模块计算 IN 和 OUT 报文的丢弃概率 p_{in} 、 p_{out} ;如果到达报文为 OUT,以概率 p_{out} 丢弃该报文;如果到达报文为 IN,以概率 p_{in} 丢弃该报文,否则报文进入 FIFO 队列。

变量: $rate_{in}/rate_{out}$:IN/OUT 报文到达速率; p :报文的丢失概率;IN 和 OUT 报文的到达速率由时间滑动窗口方法^[11]计算。LDP 求报文 OUT 和 IN 的丢失概率:

令 $rate_{in} + rate_{out} = rate_w$, 则

$$\frac{rate_{in}}{rate_w} p_{in} + \frac{rate_{out}}{rate_w} p_{out} = p \quad (2)$$

其中: $0 \leq p_{in} \leq p_{out} \leq 1$

为了保护 IN 报文:

当 $p \cdot (rate_{in} + rate_{out}) / rate_{out} < 1$ 时,

$$p_{in}/p_{out} = 0$$

当 $p \cdot (rate_{in} + rate_{out}) / rate_{out} \geq 1$ 时,

$$p_{in}/p_{out} = p_i$$

计算时首先令 $p_{in} = 0$,然后根据式(2)计算 p_{out} ,如果 $p_{out} > 1$,则 $p_{out} = 1$,然后再根据式(2)计算 p_{in} 。

PIPIO 算法参数配置简单,配置参数与 PIP 算法相同;队列长度抖动相对小,随着负载在一定的范围内变化,它不会出现大幅度的波动。

1.3 主动队列管理算法 PIPGYR

PIPGYR 算法的算法组成是:第一部分由 PIP 计算总的报文丢失概率,第二部分由 LDP(Loss Probably Divider)计算 Green、Yellow、Red 报文的丢弃概率。

PIPGYR 算法如下:

当一个报文到达时:

如果为 Green 报文

更新 Green 报文到达速率 $rate_G$;
如果为 Yellow 报文
更新 Yellow 报文到达速率 $rate_Y$;
否则为 Red 报文
更新 Red 报文到达速率 $rate_R$ 。
PIP 模块计算总的丢失概率 p ;
LDP 模块计算 Green 报文、Yellow 报文和 Red 报文的丢失概率 p_G 、 p_Y 、 p_R ;
如果(为 Green 报文)
以概率 p_G 丢弃该报文;
如果(为 Yellow 报文)
以概率 p_Y 丢弃该报文;
如果(为 Red 报文)
以概率 p_R 丢弃该报文;
否则报文进入 FIFO 队列。
变量:
 $rate_G/rate_Y/rate_R$: Green/Yellow/Red 报文到达速率;
 p : 报文(Green 或 Yellow 或 Red)的丢失概率;
 $p_G/p_Y/p_R$: Green/Yellow/Red 报文的丢失概率。
算法中,Green 或 Yellow 或 Red 报文的到达速率使用时间滑动窗口方法^[11]估计。

LDP 由以下方程组求 Green、Yellow、Red 报文的丢失概率:
令 $rate_G + rate_Y + rate_R = rate_w$,则:
$$\frac{rate_G}{rate_w}p_G + \frac{rate_Y}{rate_w}p_Y + \frac{rate_R}{rate_w}p_R = p$$

其中 $0 \leq p_G \leq p_Y \leq p_R \leq 1$
计算时首先令 $p_G = 0, p_Y = 0$,计算 p_R ,如果 $p_R > 1$,则 $p_R = 1$,再令 $p_G = 0, p_R = 1$,计算 p_Y ,如果 $p_Y > 1$,则 $p_Y = 1$,则令 $p_R = 1, p_Y = 1$,计算 p_G 。

PIPGYR 算法参数配置较简单,随着负载的变化队列长度不会出现太大幅度的波动,能更好地适应区分服务网络的 QoS 要求。

2 仿真分析

ns-2 模拟器比较 PIPGYR 和 RIO 的性能,模拟拓扑验证在流量动态变化时算法的性能,如图 1 所示。
图中,发送源节点为 $S1 \sim Sn, S1' \sim Sm'$,边界节点为 $E1, E2, E3, E4$,接收节点为 $R1 \sim Rn, R1' \sim Rm'$,目标队列长度为 50 个报文,发送源节点至边界节点 $E1, E2$ 的延迟是在 $[0ms \sim 20ms]$ 间均匀分布的随机延迟。表 1 是 RIO 参数配置。

实验 1 资源充足时:
 $0s \sim 300s$: $n(100)$ 个 FTP 源进行传输,订购 30 % ($E1$ 为 $S1 \sim S100$ 预约 3Mbps 带宽); $150s \sim 300s$:

$m(100)$ 个 FTP 源加入传输,订购 30 % ($E2$ 为 $S1' \sim S50'$ 预约 3Mbps 带宽);总的订购带宽为 6Mbps (60 %)。图 2 为 RIO 和 PIPGYR 的队列长度分布情况示意图。

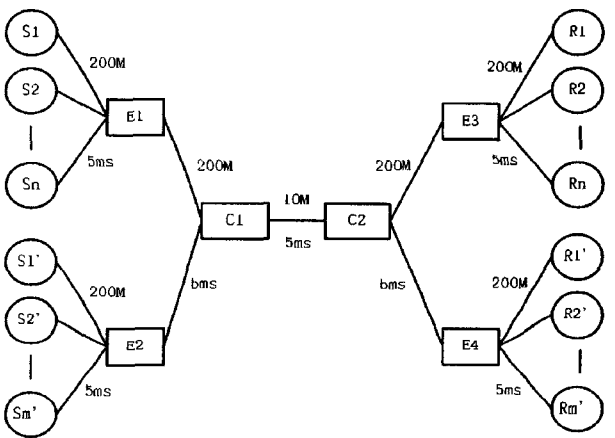


图 1 网络拓扑结构图

表 1 RIO 参数设置

	优先级 0	优先级 1	优先级 2
Min_{TH}	10	20	40
Max_{TH}	20	40	80
Max_p	0.20	0.10	0.02
W_q	0.002	0.002	0.002

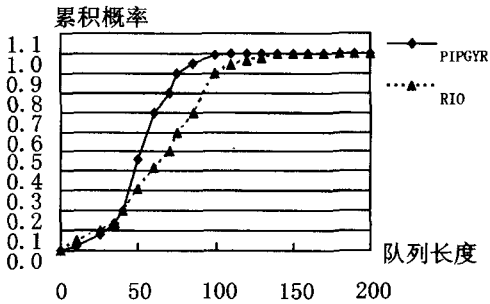


图 2 PIPGYR 和 RIO 队列长度比较

由图 2 可见,PIPGYR 比 RIO 的平均队列长度小, RIO 的平均队列长度大于 50,而 PIPGYR 的平均队列长度为 50。可知,PIPGYR 相对 RIO 算法的队列长度抖动小。

实验 2 资源不足时:
 $0s \sim 300s$: $n(100)$ 个 FTP 源进行传输,订购 60 % ($E1$ 为 $S1 \sim S100$ 预约 6Mbps 带宽); $150s \sim 300s$: $m(100)$ 个 FTP 源加入传输,订购 60 % ($E2$ 为 $S1' \sim S100'$ 预约 6Mbps 带宽);总订购带宽为 12Mbps (120 %)。PIPGYR 和 RIO 队列长度比较见图 3。

由图 3 可知 PIPGYR 和 RIO 的队列长度抖动。在资源不足时,RIO 的队列长度增加,因为 RIO 算法中 $max_ou \leq tmin_in$,可知,在资源不足和资源充足时, RIO 算法的平均队列长度稳定值不同。

办法来解决,如果冲突的非功能需求优先权相同则可以通过与用户协商进行解决。

4 结束语

软件非功能特性的处理长期以来困扰研究人员和开发人员,成为软件工程领域一个迫切解决的问题,文中通过利用面向方面的思想,对软件进行关注点的分离,对分离处理的关注点利用了著名的 NFR 框架进行描述处理,通过这些方法在一定程度上很好的解决了软件非功能特性的处理。后期工作是利用 UML Profile 扩展机制对 NFR 框架的描述方法进行补充^[8],使其图形化的同时不失语义的精确性,也可以对其进行必要的定量描述;非功能需求的形式化描述也是本文后期的主要研究工作。

参考文献:

- [1] GRADECKI J D, LESIECKI N. 精通 AspectJ [M]. 王欣轩, 吴东升, 等译. 北京:清华大学出版社, 2005.
- [2] Filman R E. 面向方面软件开发 [M]. 莫 倩等译. 北京:机械工业出版社, 2006.
- [3] 龙湘明, 孙其博, 苏 森. 使用 UML 扩展机制描述软件非功能需求的新方法 [J]. 计算机工程, 2005, 31(14): 60-63.
- [4] CHUNG L, NIXON B A, YU E, et al. Non-Functional Re-

quirements in Software Engineering [M]. Norwell: Kluwer Academic Publishers, 2000.

- [5] LIU XIAOMEI, LIU SHULIN, ZHENG XIAOJUAN. Adapting the NFR Framework to Aspectual Use-case Driven Approach [C] // 2009 Seventh ACIS International Conference on Software Engineering Research, Management and Applications, 2009. USA: IEEE, 2009: 209-214.
- [6] 郑旭飞. 一种面向方面的非功能需求框架 AONFRF 建模研究 [D]. 重庆: 西南师范大学, 2005.
- [7] 杨放春, 龙湘明. 软件非功能属性研究 [J]. 北京邮电大学学报, 2004, 27(3): 1-12.
- [8] CHUNG L, MYLOPOULOS J, NIXON B. Representing and Using Non-Functional Requirements: A Process-Oriented Approach [J]. IEEE Transactions on Software Engineering, 1992, 18(6): 483-497.
- [9] FILMAN R E, ELRAD T, CLARKE S, et al. 面向方面的软件开发 [M]. 莫 倩, 王 恺, 刘冬梅, 等译. 北京: 机械工业出版社, 2005.
- [10] 何明昕. 关注点分离在计算思维和软件工程中的方法论意义 [J]. 计算机科学, 2009, 36(4): 60-63.
- [11] JACOBSON I, NG PAN-WEI. AOSD 中文版——基于用例的面向方面软件开发 [M]. 徐 锋译. 北京: 电子工业出版社, 2005.
- [12] 张琳琳, 应 时, 赵 楷, 等. 一种建模软件体系结构非功能属性的方法 [J]. 计算机科学, 2009, 36(7): 92-96.

(上接第 111 页)

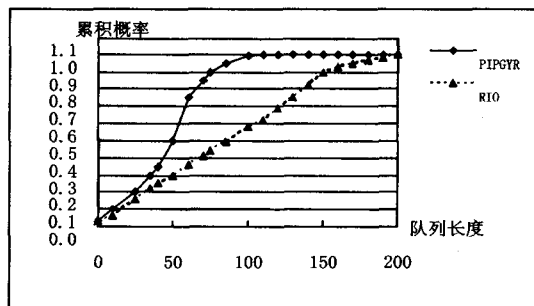


图3 PIPGYR 和 RIO 队列长度比较

3 结束语

文中结合 PIP 算法和三色标记器的功能,提出了 PIPGYR 算法,仿真分析表明: PIPGYR 算法的队列长度比较稳定,平均分组丢失率低,同时能够更好地保护高优先级报文。

参考文献:

- [1] 张明杰, 朱培栋, 卢锡城. DSOCC: 面向区分服务的拥塞控制 [J]. 国防科技大学学报, 2004, 26(3): 29-33.
- [2] 章 森, 吴建平, 林 闯. P2I: 一种新的主动队列管理算

法 [J]. 计算机学报, 2003, 26(10): 1288-1294.

- [3] 林 闯, 单志广, 盛立杰. 区分服务及其几个热点问题的研究 [J]. 计算机学报, 2000, 23(4): 419-433.
- [4] 李向丽, 陈 圆, 邱保志. 基于区分服务 AF PHB 的主动队列管理算法机制 [J]. 郑州大学信息工程学报, 2004, 22(2): 13-16.
- [5] 钱光明. 基于业务的多优先级队列区别服务方案 [J]. 计算机工程与应用, 2006, 42(10): 118-120.
- [6] 徐昌彪, 鲜永菊. 计算机网络中的拥塞控制与流量控制 [M]. 北京: 人民邮电出版社, 2007.
- [7] Blake S, Black D, Carlson M, et al. An Architecture for Differentiated Services [S]. RFC 2475, 1998.
- [8] Jacobson V, Nichols K. An Expedited Forwarding PHB [S]. RFC 2598, 1999.
- [9] Heinanen J, Baker F, Weiss W, et al. Assured Forwarding PHB [S]. RFC 2597, 1999.
- [10] Clark D, Fang W. Explicit Allocation of Best Packet Delivery Service [J]. IEEE/ACM Trans. on Networking, 1998, 6(4): 362-373.
- [11] Heinanen J, Baker F, Weiss, et al. Assured Forwarding PHB group [S]. RFC 2597, 1999.