Internet 路由器实时性的研究

梁建武** 陈语林

(中南大学信息科学与工程学院 长沙 410075)

[摘要] 介绍了路由器体系结构及数据交换方法，提出了队列分组、前馈监控和拥塞控制等综合改进方案。实验仿真结果表明，该方案对提高网络数据交换实时性和网络数据吞吐量有一定的实用价值，并能有效地遏制由局部端口阻塞蔓延到整个网络甚至崩溃等现象发生。

关 键 词 路由器；队列；前馈；拥塞；控制

中图分类号 TP393 06 文献标识码 A

Research on Real-Time Property of Internet Routers

Liang Jianwu Chen Yulin

(School of Information Science & Engineering, Central South University Changsha 410075)

Abstract This paper briefly talks about the router's architecture and the method of switching data and puts forward a comprehensive improving scheme of queue grouping, feedforward monitoring and congestion controlling. Emulation shows that the adoption of the scheme is of certain practical value to improve the real-time property of network data switching and raise the throughput capacity. It also effectively keeps the phenomena of spreading of local port block to the whole network within limits.

Key words router; queue; feedforward; congestion; control

网络中的数据流量迅猛增长使得主干网日益拥塞，新业务的涌现对网络提出更高的服务要求。而用于WAN连接、异种网互连、网络分段的路由器是网络构成的核心之一，路由器的体系结构、交换方式直接影响着数据传输的质量[1]。路由器的功能是数据转发和路径选择，如何设计高性能的路由器就成了许多研究机构和网络设备生产厂商重点研究的课题。为了满足Internet发展需求，主干网路由器就必须采取一定的策略来避免和控制网络拥塞，提高数据交换的实时性，从而保证网络通畅并提供一定的质量保证。本文针对路由器如何提高数据交换的实时性提出新的观点进行探讨。

1 路由器的具体实现

路由器体系结构主要是多个交换端口通过数据总线与共享内存、CPU相连。共享内存分为系统缓冲区和包缓冲区，系统缓冲区用于存储没有及时交换的数据，而包缓冲区用于存储最近发送到达的数据包。CPU的功能是为交换数据包选路，具体选路的依据是路由表和快速缓存。

当路由器交换端口在接收到数据包的情况下，首先将包存放在数据端口的缓存。如果路由器存在阻塞情况，端口数据缓存就要对输入数据进行排队，排队情况如图1所示。端口*A*队列中存在三个数据包，数据包1发往*C*，其他两个发往*B*，其排队方法一般是先入先出方式。

2002年6月12日收稿

* 国家自然科学基金资助项目，编号：60173041

** 男 40岁 硕士 高级工程师 主要从事网络安全理论与应用方面的研究

从以上路由器体系结构和路由实现的分析可知，提高路由器数据交换的能力有两种方法，一种是提高CPU的处理能力，另一种是选择最优的路由算法。为了不丢失数据，加大共享内存容量，该方法有悖于数据交换实时性的提高，况且容量是有限的。在CPU处理能力一定的情况下，就只有采用改进路由排队算法和监控路由器的通信流量的方法：当检测到拥塞发生的前兆就开始采取预防措施，力图把网络拥塞扼杀在萌芽状态，及将采取拥塞控制策略对付一旦发生的网络拥塞。

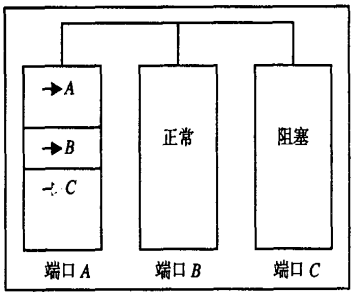

图1　路由器体系结构

2　TCP协议的自适应特性

为了更好地设计路由控制策略，必需了解网络TCP协议固有的拥塞控制的自适应特性。TCP拥塞控制是基于滑动窗口协议的，通过限制发送端向网络注入分组的速率而达到避免拥塞的目的。采用窗口的端口到端口的闭环控制方式，决定发送窗口大小的因素有两个：一是接收方所通告的窗口大小，二是发送端拥塞窗口的大小。发送窗口取两者中的较小者。在非拥塞状态下，拥塞窗口和接收窗口大小相等，一旦发现拥塞，TCP就把拥塞窗口减半，这样拥塞窗口呈几何级数减小，而发送方发放分组和重传分组速度也呈几何级数减小，直至停止状态，从而化解网络拥塞[2]。一旦拥塞结束，TCP又采取了一种算术级窗口恢复策略。

3　路由器实时监控方案

3.1　对数据流分队列

为了保证对于不同特性的数据流在实时响应方面具有公平性，则要对数据流按定制方式进行分队列(例如 TCP、UDP、SNA、IPX)，以便对每个流区别对待，原因是，前面提到的具有TCP协议流有自适应特性。当网络发生拥塞时，具有TCP协议的源对分组丢失十分敏感，它会降低发送速率，而其他对TCP不友好的流如UDP流对丢失根本无所谓，拥塞时不会降低发送速率，使拥塞不能解除，结果造成TCP源始终得不到应有的带宽，从而使公平性得不到保障。网络资源的不公平性，反过来加剧拥塞，甚至可能导致拥塞崩溃[3]。如果经路由器的数据流分别排队，就能很好地解决网络总体吞吐量的实时性和瓶颈链路带宽分享的公平性，同时也能有效地隔离具有侵略性的源。

3.2　前馈监控

路由器一般采用“去尾”的丢弃方式，其最大优点在于实施简单，但这种方式将拥塞控制的所有责任都推给网络边缘，没有考虑被丢弃包的重要程度。若采用一种前馈控制的方式，也就是时刻监视穿越路由器的数据流量，每当检测到拥塞发生的前兆就开始采取预防措施，力图把网络拥塞扼杀在萌芽状态。

这种前馈控制方式是监控数据包的排队长度及数据流量的变化率(即长度L对时间t的微分)$\mathrm{d}L/\mathrm{d}t$。一旦发现拥塞迫近，或出现数据流量突发，就按一定的概率丢弃进入路由器的数据包，这样就可以及早通知源端减小拥塞窗口，以减少进入网络的数据量，避免了必须等到队列完全占满才被迫丢弃所有到达的数据包。

该控制方式包含监控拥塞前兆和丢弃数据包的时间。采用原则是计算队列的平均长度

$$L=(1-W)L+WS_L$$

式中　$W\in(0,1)$，S_L为实测长度(每次一个新数据包到达路由器时测量的)，先定义两个阈值$L_{\min}$和$L_{\max}$，当一个数据包到达路由器时，若$L\leqslant L_{\min}$，将此包排队；若$L_{\min}<L<L_{\max}$，则计算丢弃概率P，并以此概率P丢弃此包；若$L\geqslant L_{\max}$，则丢弃此包。丢弃概率P不仅是L的函数，同时也是L变化率的函数，即

$$T_P=\max P(L-L_{\min})/(L_{\max}-L_{\min})+C\times \mathrm{d}L/\mathrm{d}t$$

$$P=T_P/(1-N\times T_P)$$

式中　T_P是P的中间变量，N则表示L在两阈值之间有多少数据包排队，C为常数，$\max P$为T_P的上限值。以上前馈控制是针对每个数据队列控制的，所以丢弃原则还要综合考虑各队列的情况。当缓冲区满时，超额使用缓冲区的连接很可能有多个，究竟丢弃哪个队列的分组以腾出空间给新到达的分组仍然有待选择。一

种方法是从那些超额使用缓冲区空间的连接的队列中随机选择一个，如果所有流都是TCP流，这种方式很好，但如果存在某些对分组不负责的流，这个方案就不理想，故综合采取以下两种方案：1) 严厉地惩罚那些持续超额使用缓冲区分配和侵占过多带宽的连接；2) 应该丢弃分组队列中最长的队列。

3.3 阻塞控制

由前面CISCO公司的路由器具体排队方法图1可知，在端口C处于阻塞状态时，发送端口C的数据包就只好等待端口C正常后再发送，但由于发送C端口的数据排在队的首位，排在该数据包之后发往B端口数据无法发送，由于端口B处于正常状态，所以数据应当可以正常地发送到B端口，存在这种情况会导致交换机由局部端口阻塞繁殖蔓延为整个网络范围的数据阻塞，严重影响网络实时数据的吞吐量。

针对以上情况，一旦发生阻塞时，在端口内部应建立新的队列，存储发送处于阻塞状态端口的数据包，并设置阻塞端口标志和阻塞时间标志，一旦端口阻塞情况停止，转入正常队列，或者阻塞端口时间过长，丢弃该数据包，该程序框图如图2所示。

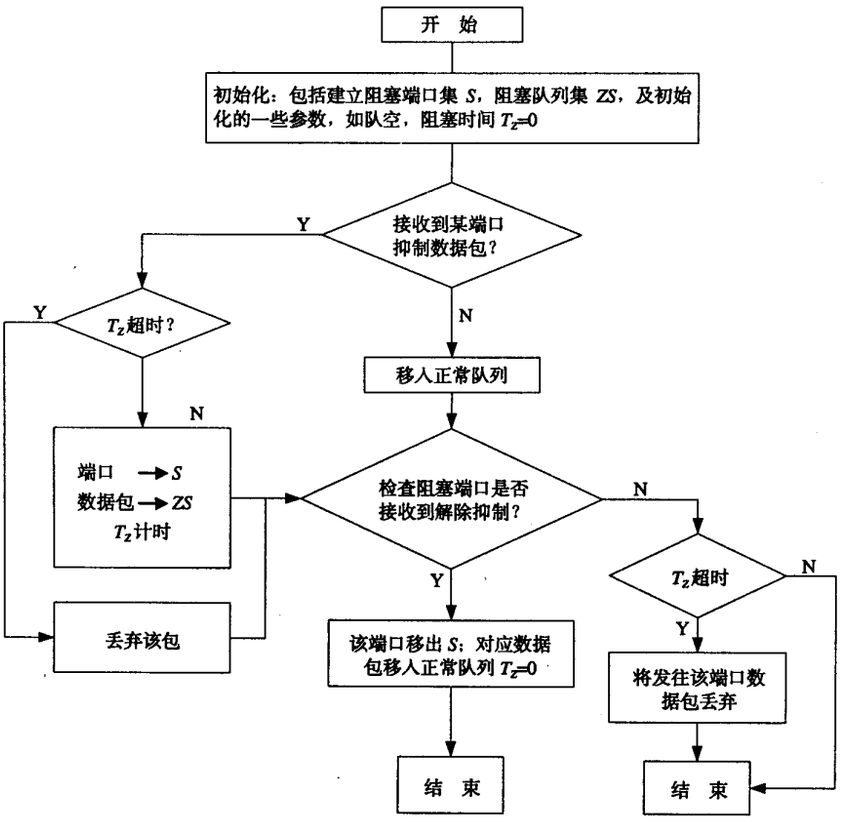

图2 阻塞控制程序框图

在图2中T_Z为端口阻塞持续的时间集($Z = A$，B，C，…)，如T_C表示端口C阻塞持续的时间，该时间视具体情况而定，也可以采用自适应的原则自行整定优化，最终达到惩罚那些不负责任的数据流，缩短在网上滞留的时间，保证网络畅通。

4 实验仿真

在NT局域网上，采用Visual C++ 6.0作为开发平台，使用WinSock插件采用各种路由算法对某一台机器进行访问，计算并观察每次访问传送时间、信息返回时间，以及在规定的时间[0，t]内网络故障发生情况。在此基础上计算回应率数学期望、可靠性PR(t)、平均无故障性能(MPTF)以及阻塞率等参数值，再通过计

(下转第450页)

从状态0出发，只能转到状态1，故右端前一项中仅有 $j=1$ 这一项，即

$$\int_0^t E\{N(t)\mid Z_1=j,T_1=u,Z_o=0\}\mathrm{d}Q_{0j}(u)=Q_{01}(t)*M_1(t)$$

右端第二项中，在 $Z_0=0$ 和 $T_1>t$ 条件下，在(0,t]内系统一直停留在正常状态，故 $E\{N(t)\mid Z_0=0,T_1>t\}=0$。其余各式类似可得，对式(7)两端取拉普拉斯-斯梯阶变换，再由托贝尔定理得系统的稳态故障频度为

$$M=\lim_{t\to\infty}\frac{M_i(t)}{t}=\frac{(m_1+m_2)(\hat{F}_1(m_2)+\hat{F}_2(m_1)-\hat{F}_1(m_2)\hat{F}_2(m_1))}{a+b+\hat{F}_1(m_2)+\hat{F}_2(m_1)-\hat{F}_1(m_2)\hat{F}_2(m_1)}$$

与初始状态 i 无关。

本文研究工作得到了电子科技大学青年基金(No.YF021102)资助，在此表示感谢。

参 考 文 献

[1] 曹晋华, 程 侃. 可靠性数学引论[M]. 北京: 科学出版社, 1986

[2] 程 侃. 寿命分布类与可靠性数学理论[M]. 北京: 科学出版社, 1999

[3] Widder D V. The laplace transform[M]. princeton University Press, 1941

[4] 唐应辉. 可修排队系统中可靠性指标的分解[J]. 电子学报, 1996, 24(11): 18-21

编 辑 孙晓丹

(上接第432页)

算平均值并与其他路由算法的阻塞率相比较，得到了如图3所示的仿真结果[4]。图中 P_{BL}、P_{BQ}、P_{NQ}、P_{NL} 分别表示宽带连接一般路由算法、队列分组路由算法、前馈监控路由算法、拥塞控制算法等情况下的阻塞率。仿真结果表明，每改善一项性能后，网络总的阻塞率减少了，并能较好地改善网络实时数据的吞吐量。

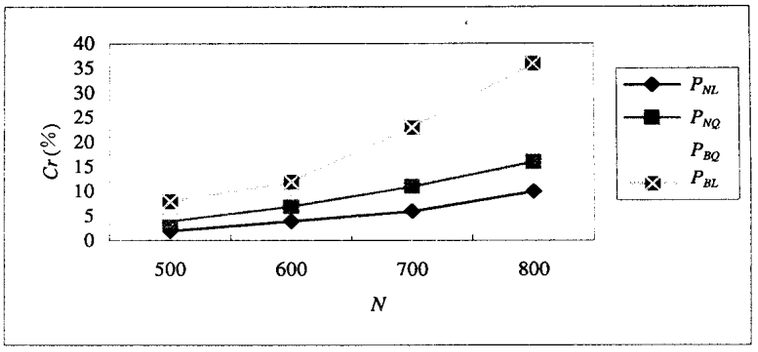

图3 连接阻塞率比较图 (横轴N表示申请连接数，纵轴C_r表示连接阻塞率)

5 结 束 语

综上所述，在对路由器设计时，除了硬件设计优化、多CPU并行处理、交换式背板方式等外，在对路由调度算法方面，为充分考虑网络数据吞吐量的实时性，避免网络阻塞，保证网络正常运行，以下三点考虑具有一定实用价值：1) 队列分组技术，为保证网络各流的数据交换实时性和公平性；2) 前馈监控，提前采取预防措施，力图把网络拥塞扼杀在萌芽状态；3) 拥塞控制，保证局部端口拥塞不致蔓延整个网络，甚至导致拥塞崩溃。

参 考 文 献

[1] 邓志成，周 旗. 一种公平接入的QoS路由算法[J]. 计算机学报，2000，23(6)：667-670

[2] Mathis M,Mahdavi J. TCP Selective Acknowledgment Options[C].RFC 2018, 1996

[3] 周明天，汪文勇. TCP/IP网络原理与技术[M]. 北京：清华大学出版社, 1993

[4] 陈语林，梁建武，曹 刚. 网络服务的公平性分析及性能改善[J]. 电子科技大学学报, 2002, 31(4)：409-412

编 辑 刘文珍