一个基于 TCR 改进模型的网络流量管理系统①

汤　赟　李双庆　曹　伟 (重庆大学 计算机学院 重庆 400044)

摘　要： 网络流量管理技术是保证网络服务质量的重要前提。在研究网络流量识别与控制两种关键技术后，提出了一个实用的网络流量管理系统。在端口和特征值双层识别流量类型基础之上，针对 TCP Rate control(TCR)在往返时延的测量上难度大，精度低等不足，设计出了一种改进的流量控速模型，使得速率的控制更加有效和精确。最后在 Linux 的 netfilter 架构下实现了基于该模型的原型系统。通过实验测试验证了其有效性和准确性。

关键词： 流量识别；流量控制；流量整形；往返时延

A Network Traffic Management System Based on Improved TCR Model

TANG Yun, LI Shuang-Qing, CAO Wei

(Department of Computer Science, Chongqing University, Chongqing 400044, China)

Abstract: Network traffic management is a significant technology to ensure network QoS. At first, this paper analyzes the key technologies of network traffic management, including traffic identification technology and control technology. Then, a strategy of network traffic management system is proposed. To address the deficiencies of mensuration and precision of main network traffic rate control technology TCR, an improved traffic rate control model is proposed. This model makes the rate control more effective and accurate. Finally, a network traffic management prototype system is implemented based on Linux OS and netfilter firewall framework. Experimental test shows the effectiveness and the accuracy of this system.

Keywords: traffic identification; traffic control; traffic shaping; RTT

1　引言

随着 Internet 的广泛应用，越来越多关键业务的开展依赖于互联网络进行。尽管科技的发展使得互联网络的带宽得到不断扩充，但一些关键业务的网络服务质量仍然无法得到有效的保障。究其原因是现有互联网络是基于尽力传送(Best-Effort Service)的应用模式，缺乏对网络流量有效的识别和管理技术。一些非关键业务，如 P2P，流媒体等，消耗大量带宽资源，严重干扰了网络的正常运行。因此网络流量控制和带宽管理成了一个亟待解决的问题。而研究网络流量识别与控制技术，并将其运用于管理有限的网络带宽，最大化网络资源利用率，对于保障计算机网络服务质量有非常重要的意义[1]。目前，Packeteet[2]、Allot 等公司已在市场上推出相关网络流量管理产品，但由于高昂的价格，国内广大中小企业仍无力广泛部署。

本文从中小企业的实际需求出发，在开源平台下对网络流量管理领域中流量识别与流量控制两种关键技术进行了研究。采用了端口与特征签名值双层相结合的方法进行流量识别分类，提高了准确率以及识别的效率。在识别之后对流量的进一步控速上，提出了一种针对当前 TCR 技术缺陷之处改进的控速模型。接着综合流量识别技术和控速模型提出了一套网络流量管理架构。并在 Linux 操作系统及 netfilter[3]防火墙框架等开源工具基础上实现了该网络流量管理系统。

图 1 为该管理系统架构图，整个系统分为用户空间层次和系统空间层次。其中识别分类和实时控制模块完全工作在内核区，端口更新、特征值扩展以及流量监控模块工作在用户区。在数据包的采集之后，通

① 收稿时间:2009-09-17;收到修改稿时间:2009-11-21

过识分类模块和控制模块之间的接口调用，实现网络流量的管理控制。

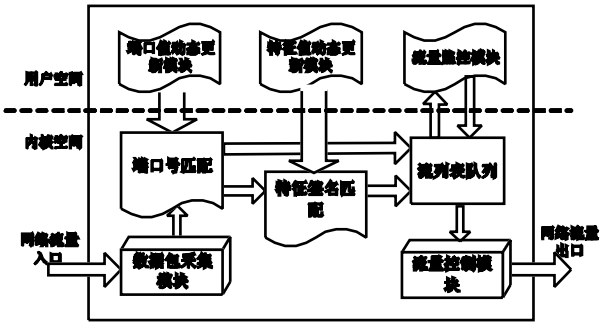

图 1　网络流量管理系统架构

最后给出的实验测试证明了该架构在识别和控制网络流量上能达到预期效果，从而能很好的保证测试环境下网络的性能和服务质量。

2　网络流量管理关键技术

2.1 流量识别分类技术

近年来，流量的识别分类在学术和应用研究领域备受重视，已形成一个相对独立的研究领域。流量识别通常可基于数据包的端口号、应用层特征签名以及流量行为特征进行。基于端口号的识别根据数据包头部的端口字段值，依照在 IANA[4]中注册的端口号来识别各种应用程序。基于应用层特征签名的识别指针对应用层报文进行深度内容检查，判断其中是否含有特定应用程序的签名值；基于流量行为的识别指通过统计流量行为特征来识别不同的流量类型。

端口识别实现简单，计算资源开销小。但由于下述原因，端口识别越来越多的受到限制：(1)不是所有协议都在 IANA 中注册使用的端口。例如 BT 等 P2P 协议。并且某些注册的端口号同时被多个应用程序所使用。例如 888 端口同时被 accessbuider 和 CDDBP 所使用。(2)在某些情况下，通信双方的端口是动态分配的。例如 FTP 的数据传输端口就是在控制流中协商的。(3)一些应用协议伪装成常见的端口号，以绕过防火墙的封堵。所以仅仅使用基于端口号的识别技术并不能有效对流量进行识别分类。另一方面，基于流量行为特征的识别当前并不成熟，在实际应用中存在实时性差，识别精度不足的问题。所以本文采用基于端口识别和应用层特征签名识别[5]相结合的流量识别技术。

表 1　几种应用较广泛的通信协议端口号以及特征签名

网络协议	特征字符串	传输层协议	协议端口
Http	"HTTP/1.1"	TCP	80
Fastrack	"Get /.hash"	TCP	1214
	0x270000002980	UDP	
Gnutella	"GNUT","GIV"	TCP	6346-6889
	"GND"	UDP	
BitTorrent	0x13Bit	TCP	6881-6889
eDonkey2000	0x31900000	TCP/UDP	4661-4665
	0xc53f010000		
Direct Connect	"$MyN","$Dir"	TCP	411-412

具体的实现过程如下：在 linux 下利用 netfilter 防火墙框架，在 IP 协议栈的 NF_IP_PRE_ROUTING 函数挂载点上，挂载流量识别模块。截取进入协议栈 IP 层的数据包进行包头检测。首先与常用协议(HTTP，FTP，DNS 等)的默认端口号进行匹配。如果成功，打上所属协议的标签，否则继续特征签名的匹配。在根据某条流的前几个数据包识别了该条流的应用类型之后，随后的数据包依据五元组规则<源端口,目的端口，源 IP,目的 IP,传输层协议>直接进行归流处理，将结果归入对应的流列表队列，交给下一步的流量控制模块进行流量的控制和整形。

该识别策略利用端口号匹配，减少了后续特征字段匹配模块的数据输入，提高了处理速度。同时其原理简单，易于工程实现。此外，目前在我国国内占主导地位的两种 P2P 协议[6]BitTorrent 和 eMule 均开放源代码，有条件采集到他们的特征字段，因为认为该识别策略有较好的完整性；在准确性方面，根据前文分析，特征字段匹配的准确性非常高，而 HTTP 和电子邮件等传统应用至今都仍使用固定端口号，所以通常情况下其准确性也能得到保障。

2.2 流量控制技术 TCP Rate Control

在对网络流量识别分类以后，还须对已分类的各种流量进行控速整形，使各类流量能够按照用户策略进行带宽分配及限速整形，从而保证关键流量的带宽资源，确保网络服务质量。

流量控制技术通常分为限速与整形两部分：流量限速通过管理各类流量的缓冲队列、链路带宽等网络资源，使不同的流量类型可根据用户策略获得不同的带宽值；流量整形是调整数据传输的平均速率，使同类应用中各条流获得平等的发送机会，避免突发性通

信量导致的拥塞问题。两部分都有相应的算法，如令牌桶算法、随机早期丢弃算法等。而 TCP Rate Control(TCR)[7]是目前应用较广泛的一种综合控速技术。TCR 通过窗口控制和 ACK 控制两方面进行控速，前者应用于流量控制，而后者应用于流量整形[8]。一个 TCR 速率控制模型如图 2 所示：

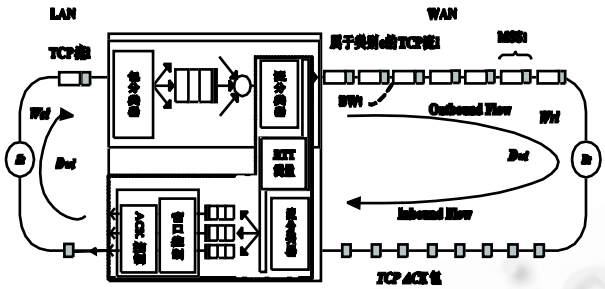

图 2　TCR 控制模型

BW_C：某应用类 c 的带宽；BW_i：类下某条 c 流 i 的带宽(BW_c/N)；MSS_i：流 i 的最大报文长度；Dw_i 时延；BDP_i：流 i 的带宽时延积。

2.2.1 窗口控制

假设类别 c 有 n 条网络流，分配带宽为 BW_c。在网络流 i 达到额定带宽的情况下，其带宽大小为：

$$BW_i = \frac{Byte\ Sent}{Time\ Interval} = \frac{TCP\ window}{RTT} = \frac{Byte\ in\ WAN}{Round\ Trip\ WAN\ Delay} = \frac{W_{wi} \times MSS_i}{D_{wi}} \quad (1)$$

在网络流 i 超出额定带宽的情况下，为使流 i 达到额定带宽，可限制其在测得的 D_{wi} 时间内只发送的 W_{wi} 个数据包，这时，由上式可得：

$$W_{wi} = \frac{BDP_i}{MSS_i} = \frac{BW_i \times D_{wi}}{MSS_i} \quad (2)$$

为了使网络流 i 达到额定带宽 $BW_i(=BW_c/n)$，窗口控制利用 TCP 中的序号和确认号，周期性的检测数据包在网络上的往返时延 D_{wi}，使得每个 ACK 包中的接收窗口大小限定为 BDP_i 大小(W_{wi} 个数据包)，这样在 D_{wi} 的时间里，网络流 i 将填充满从发送方到接收方的 WAN 管道，从而保证网络流 i 不会超出额定带宽。

2.2.2 ACK 控制

控速之后为了将 W_{wi} 个数据包在 D_{wi} 的时间里平滑的发送到网络中去，实现网络流的整形，要对到达发送方的 ACK 进行队列缓存，每隔△i 时间释放一个 ACK 包。由于进行了窗口控制，发送方每收到一个 ACK 包，只能紧接着发出一个数据包到网络中，实现额定带宽下的流量整形。由 ACK 控制的原理知：

$$\Delta i = \frac{D_{wi}}{W_{wi}} = \frac{MSS_i}{BW_i} \quad (3)$$

这样，设定一个计时器之后，在△i 时间间隔里，每个网络流只需释放1个ACK包就可以实现对流量的整形。

3　改进的TCR控速模型

3.1 对 TCR 带宽时延积的计算

TCR 的控速中，两个关键技术是窗口控制和 ACK 控制[9]。但由于 TCR 在流控中的关键步骤是需要对网络流进行往返时延 RTT 的测量，而每条流的往返时延不仅和流经网络的拓扑结构有关，还和当前网络环境有关，这使得 RTT 的值 DW_i 不仅测量难度大，而且精度低[10]。因此可能造成多个网络流实际带宽超出(低于)额定带宽，大大的影响了 TCR 的控速效果。

我们知道 TCR 对往返时延值 DW_i 的测量是为了计算出网络流 i 的时延带宽乘积 $BDPi$。

$$BDP_i = BW_i \times D_{Wi} \quad (4)$$

经过变换，可得：

$$D_{Wi} = \frac{BDPi}{BW_i} \quad (5)$$

而往返时延是随着网络环境的不同而改变，不会因为当前对这条网络流进行控制而改变。同时因为控制的原因，必将减少分组进入网络的数量，以达到预期控制的带宽大小，因此控制网络流的时延带宽乘积不会造成网络的拥塞，导致对往返时延的影响。

考虑将某条网络流 i 控制前后的带宽进行比较。假设控制前网络流 i 的带宽为 BW_i，时延带宽乘积为 BDP_i，控制后的带宽为 BW_i'，时延带宽乘积为 BDP_i'。由以上分析可得出：

$$D_{Wi} = \frac{BDP_i}{BW_i} = \frac{BDP_i'}{BW_i'} \quad (6)$$

经过变换可以得出：

$$BDP_i' = \frac{BDP_i \times BW_i'}{BW_i} \quad (7)$$

由此公式我们可知，为了得到额定带宽所需的时延带宽乘机，我们需要知道控制前网络流 i 的带宽和其对应的时延带宽乘积。因此在改进的 TCR 模型中我

们需要对 BDP 和流带宽进行测量。

3.2 BDP 和流带宽的测量

1) 对网络流 BDP 的测量相对于往返时延 RTT 的测量要简单许多，其过程如下：

a 记录数据包发送序号：每当一条网络流的数据包进入网络之前，流量控制器就将数据包的发送序号更新到一个变量中。

b 进行 BDP 计算：当该网络流的一个 ACK 包从网络中进入流量控制器时，得到的 ACK 包确认号是目的端对其对应的数据包的确认。这意味着从这个 ACK 包对应的数据包发送，到该 ACK 包的接收时间之间的间隔，即是 TCR 模型中的往返时延 Dw。那确认号与记录的数据包发送序号的差值，就是该网络流在往返时延 D_{wi} 之内已进入网络的字节数，即 BDPi 大小。图 3 描述了 BDPi 的计算。

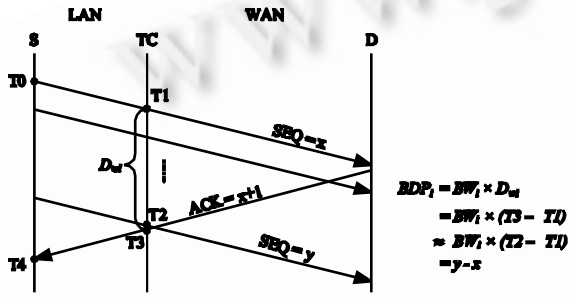

图 3　BDPi 的计算

由于在控制器中对发送序号的及时更新，这意味着最后更新的时间点 T2 与 ACK 包到达的时间点 T3 十分接近。于是从 T1 到 T3 时间内发送端发送到网络中的字节数可认为是确认号与最新更新的发送序号的差值，即 BDPi。另外为减少因为流量突发引起测量误差，对测量的 BDP 值应进行加权平均，即：

$$BDP(n) = \theta \times BDP(n-1) + (1-\theta) \times BDP(n) \qquad (8)$$

其中 $0 < \theta < 1$

2) 对流带宽的测量同样非常简单。首先对某类网络流量设定一个定时器，启动该定时器的同时维护一个变量，以记录每条网络流额定时间内字节数的增量。计算在额定时间内发送的字节数即是该流的流带宽值。

虽然如此计算出的流带宽是一定时间间隔内的平均值，但是在定时器设置的时间间隔较小的情况下，该测量值可视为当前网络流的实时带宽。

在测量到的流带宽超过额定带宽的情况下，流量控制器启动窗口控制和 ACK 控制。根据公式(7)计算出额定的 BDP。然后设定每个 ACK 包中的接收窗口大小限定为计算出的 BDP 大小，并按照?i=MSSi/BWi 的时间间隔释放 ACK 包，这样在网络流的往返时延的时间里，数据包将平滑的填充从发送方到接收方的 WAN 管道，从而达到用户设定的额定带宽值，保证网络服务质量[11]。

4　网络流量管理系统的测试

为了验证基于论文所提出的识别策略以及控速模型的网络流量管理系统的有效性和准确性，本文对该原型系统做了功能与性能两方面的测试。基于最常用的 32 位 Intel X86 硬件平台、以开源 Linux 作为软件平台，在 Netfilter 防火墙框架下实现该原型系统。该系统被部署在实验室内网和连接外网的路由器之间，即 LAN/WAN 出口路由器之后，对进/出内网的流量进行控制。

4.1 功能测试

在一条带宽为 10M 的链路上，开启局域网内对外的 BitTorrent、HTTP 和 FTP 这三种分别代表 P2P，Web 浏览和文件传输的典型应用流量。记录网络流量管理系统开启前的流量类型和带宽消耗情况。先后开启基于 TCR 流量管理系统以及本文所提出的基于 TCR 改进模型的流量管理系统。记录各个时间间隔内每种流量的识别控制情况。

1) 在测试流量识别的同时，使用 Sniffer 抓包工具捕获流量，根据系统识别出的应用流量与该应用实际传输流量的对比，计算出系统对该应用的流量识别率，通过手工分析获得系统对该应用连接的识别率。其部分识别结果如表 2 所示。

表 2　流量识别结果分析

应用名	实际流量(Byte)	识别出的流量(Byte)	流量识别率	实际连接数	识别出的连接数	连接识别率
BitTorrent	2103746041	2038529913	96.9%	7631	7096	93%
HTTP	5367851	5367851	100%	871	871	100%
FTP	32765983	32765983	100%	513	513	100%

运行结果显示，本策略能够较准确的识别出常见应用流量。在对于 BT 流量的识别上，经研究发现，为防止上文提到的基于应用层特征签名的流量

识别，一些 BT 客户端软件采用了一定的对抗技术。例如 BitSpirit 采用一种“扩展握手协议”的方法。其实现是在含有 BT 特征签名的 TCP 握手信息前加入了一个 HTTP 包头，BitComet 则是将握手信息加密扰乱。这两种方法可以在一定程度上突破基于 BT 协议特征签名的流量识别，所以实验数据中的 BT 流量识别率并未达到 100%。但由于这些方法局限于同一种客户端软件之间才能互联，启用该选项后，资源很少，下载速度较慢，因此使用者较少。

2) 在开启管理系统前后，分别定时记录各种应用的速率，用于进行实验对比。其中 BT 流量的控速管理如图 4 所示，其他类型流量类似。

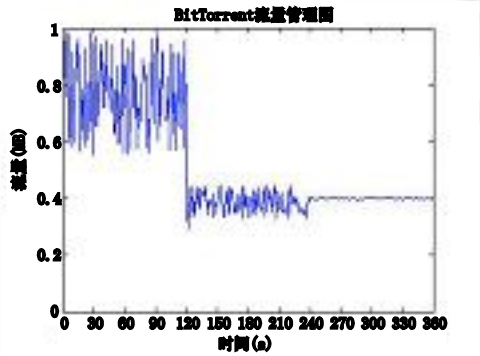

图 4　BitTorrent 流量管理

在前 120 秒内，在未对流量进行控制之前，可以看出 BT 下载所占用的带宽资源较多，且突发性强。在开启基于 TCR 的流量控制之后 120 秒内，我们分配 0.4M 的额定带宽，带宽消耗和流量突发性问题得到了一定的改善。但带宽精确度在几十 KB 的范围内波动较大。在最后一个阶段，使用了基于改进 TCR 模型的流量管理系统之后，波动范围极其微弱，大概在 2K 至 3K 范围内，能很好的满足事先设定的带宽精确度，对网络流量的控制起到了很好的效果。同时也能很好的解决流量的突发性问题。

4.2 性能测试

性能测试中，在不同网络流量压力及吞吐量下，选取数据包平均时延，丢包率和重传率等性能参数来对 TCR 改进模型进行测试。测试结果见表 3。

综上所述，基于论文所提出的流量管理原型系统在带宽为 10MBps 的链路上，具有可被接受的丢包率、时延以及重传率。能够较好的对网络流量进行管理控制，体现了该改进模型的有效性和准确性。

表 3　不同吞吐量下丢包率测试结果

性能 / 输出速率	平均时延（ms）	丢包率（%）	重传率（%）
20KB/s	150.65	5.40	6.08
50KB/s	69.01	3.14	3.57
100KB/s	24.95	1.63	1.76
500KB/s	10.44	0.85	0.90
1000KB/s	6.12	0.32	0.34
2000KB/s	5.37	0.25	0.26

5　结语

本文首先从实际应用的角度说明了网络流量管理的重要性。随后通过对网络流量管理领域中流量识别分类和流量控制两种关键技术的研究，设计了一套流量识别策略以及基于 TCR 的流量控速改进模型。最后综合两种技术提出了一种网络流量管理架构，并对基于此架构的原型系统进行了功能和性能方面的实验测试。测试表明系统能够高效准确的识别不同种类的常见流量类型，并依照用户设定的带宽精确控速以及流量的整形。论文以当前最常用的 32 位 Intel X86 作为硬件平台、以开源 Linux 作为软件平台，在 Netfilter 防火墙框架下实现该原型系统。整个系统具有成本低、易使用、可扩展等优点，十分适合应用于中小企业接入网层面的网络流量管理。

参考文献

1 张信明,陈国良,等. Internet QoS 控制机制综述.计算机科学, 2002,(3):20－23.

2 Packeteer, Strategies for Managing Application Traffic [2009-07-12]. http://www.packeteer.com

3 Gheorhteh L. Designing and Implementing Linux Firewalls and QoS using netfilter, iproute2, NAT, and L7-filter. Packt Publishing Ltd. 2006,6(3):6－8,11.

4 IANA [2009-09-12]. http://www.iana.org/ assignments/port-numbers

5 Moore A, Konstantina P. Toward the Accurate Identification of Network Application. Passive & Active Measurement Workshop 2005,3:2－4,8.

6 CacheLogic Research. The True Picture of P2P File Sharing[2009-09-15]. http://www.cachelogic.com/ home/

(下转第 95 页)

(上接第 20 页)

pages/ research/ p2p2004.php

7 Wei HY, Lin YD. Assessing and Improving TCP Rate Shaping over Edge Gateways. IEEE Communications Surveys and Tutorials, 2005.4:240－248.

8 Sun YS, Lee C, Berry R, Haddad AH. An Application of the Control Theoretic Modeling for a Scalable TCP ACK Pacer. Proc. of the 2004 American Control Conference, 2004,(5):40－48.

9 CaserriC, Meo M. A new approach to model the stationary behavior of TCP connection. IEEE INFOCOM, 2000,7(5):143－149.

10 Sivaraman V, Fabio M, Gerla CM. Traffic Shaping for End-to-End Delay Guarantees with EDF Scheduling. IEEE/ACM Transactions. 2000. 10－18.

11 Dimitrios Stiliadiss, Anujan Varma.Latency-Rate Servers: A General Model for Analysis of Traffic Scheduling Algorithms. IEEE/ACM Transactions on Networking, 1998,(8):432－438.

12 Wehrle L, Paehlke F. The Linux Network Architecture: Design and Implementation of Network Protocols in the Linux kernel. Pearson-Prentice Hall, 2005.