基于 Linux 内核的 BT 流量控制的原理与实现

吕效红

(黄山学院 网络中心,安徽 黄山 245041)

摘 要:利用 Linux 内核的网络流量控制机制,对 Linux 内核的 Netfilter/Iptables 技术和带宽控制技术进行分析,提高这种框架下对现有 BT 流量控制技术的准确性,通过分析 BT 协议和 BT 实际传输数据,找出 BT 传输过程中出现的各种特征字符串,从而利用这些特征字符串建立 Iptables 需要的识别特征规则库来提高识别准确性。实验证明该流量控制管理方案的有效性和实用性。

关键词:流量控制;netfilter/iptables; bt; Linux

中图分类号:TP393.08 **文献标识码**:A **文章编号**:1672-447X(2010)03-0028-04

1 引 言

近年来,随着大规模的存储和分布式系统的成熟,互联网上传输的不再只是简单的文本和图像。多媒体通信,能够集成文本,视频,音频,图像为一体,得到越来越广泛的应用。眼下,随着技术的发展,众多新型业务已经渗透到网络的各种主流应用之中。这些流量中 BitTorrent(BT)变得越来越流行,自从 2002 年 BitTorrent 技术产生以来,迅速成为互联网上最高效、便捷的下载工具。目前,根据主流运营商的统计,70%的带宽被 P2P 流量所占据,而这其中大部分是 BT 流量。其流量的迅速增长给网络运营与维护带来极大的压力。然而这也使得一些无关紧要的应用(如 BT 下载、BT 在线电影等)都来与一些关键性应用争夺有限的广域网资源,使得关键性应用无法高效率运行。正是基于这种"劣币驱逐良币"日趋严峻的现实,对 BT 流量进行有效的控制也变得日趋紧迫。

网络流量控制器的主要工作在于通过监视网络流量,并根据各种应用和服务的通信特点对网络数据包进行分类,限制相关应用的有效吞吐量从而提高网络利用率。目前,流量控制中许多研究工作将重点放在了对 Linux 流量整形算法的研究实现,及在其已有技术基础上的深入改进,代表性算法研究包括:HTB[1,2]、CBQ[3,4]、SFQ[5]等。利用 Netfilter/Iptables 模块实现简单的流量控制功能等领域上。而对于在这种框架下去针对具体的应用(如 BT 应用等)建立何种与之相匹配并且适合这种业务特点的规则库以及很好的结合 Linux 的内核机制却一直没有得到重视和很好的研究,特别是随着当今 BT 实现机制的多元化趋势,如何针对网络运行状况,组建一行之有效的 BT 流量控制管理机制,合理组织分配有限的网络资源,优化网络应用流量,提高关键业务的 QoS 已成为当下之需。

本文在对 Linux 流量控制机和 BT 通信协议机制的研究基础上,针对现在主流的 BT 会话实现的流程提取出识别的特征字符串,加入到 iptables 的匹配规则库中,提出一套控制 BT 流程的原理与实现方案,在不损害流控系统性能、保证关键业务畅通运行的情况下,有效的提高了对 BT 业务的可管理性。同时,本文通过实验分析证明其有效性和实用性。

收稿日期:2010-01-05

作者简介:吕效红(1982-),黄山学院现代教育技术中心助理实验师,研究方向为网络测试。

万方数据

2 Linux 流量控制机制

Linux 从 Kernel2.1.105 开始就提供了支持 QoS 的完整的、强大的带宽管理代码,它主要通过路由工具包(IPROUTE2)的流量控制命令(Traffic Control command)来进行带宽控制。这套灵活、功能强大的流量控制的机制是当网络节点发送数据包时,负责将等待发送的数据包按照一定的规则分类、排队及调度后再发送出去,其框架如图 1 所示。

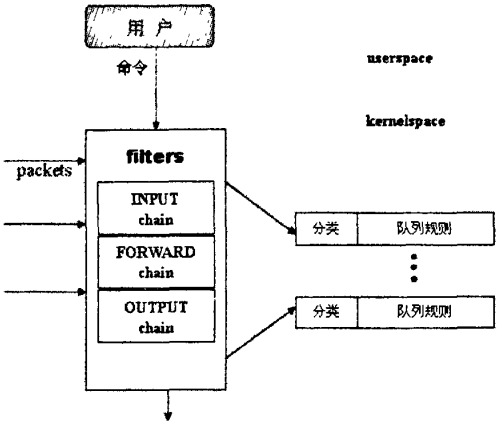

图 1　Linux 流量控制框架图

由图 1 可知,Linux 流量控制分为两部分,即用户空间和内核空间。用户空间用于存放和运行人机交互的用户程序,而内核空间存放和运行特定队列操作的核心代码。

Linux 内核模块将不同标签的数据包进行区分,送往不同队列进行调度整形,实现区分服务。而实现这一系列功能的元素(除去 Policies)大体上可分为如下 3 大类:[6]

1.分类 classes

2.队列规则 queuing disciplines

3.过滤器 filters

在实施流量控制过程中,需要根据数据包标签,将不同的包发往合适的分类,这一过程就叫做包分类。在 Linux 流量控制模块中,包分类是通过过滤器完成的。一个过滤器包含若干匹配项,如果符合匹配项,就按此过滤器分类。而过滤器中存放的规则就是识别上层应用的规则库 (如识别 BT 流量的特征规则库)。

内核中关于流量控制的代码主要在目录/net/sched 下。内核流量控制与用户空间的接口都定义在 include/linux/pkt_cls.h,include/linux/pkt_sched.h 文件中。一些只供内核用的内联函数和声明被定义在 include/net/pkt_cls.h 和 include/net/pkt_sched.h 文件中。

3 BT 流量识别规则库的生成

BT 协议的开放性,使得我们可以结合实际 BT 交互过程,深入分析 BT 协议本身。本文采取以下原则来选择 BT 协议的特征字串:在 BT 交互过程中必定会出现具有稳定形态的字串,优先选择会重复出现的特征字串,同时为了不对识别效率造成过大影响,特征字串越短越好。由于特征字串是包含在具体的数据包中的,因而本文将包含特征字串的数据包称为特征包,选择特征字串实际就是选择特征包。

BT 协议交互分为两部分:客户端之间的交互、客户端与服务器之间的交互。其中客户端与服务器之间的交互数据量小,本文将不研究对这部分数据的识别。在建立对等连接过程时,发送的握手信息包中包含的特征字串就是"19BitTorrentProtocol"。此后的互传比特域信息阶段,重复的数据传输阶段等各个阶段中所发送的信息包中都包含有特定意义的字串。

表 1 中列出了各阶段在各种情况下发送的信息包中所包含的特征字串。大部分的信息包都有固定的大小,通过研究发现,信息包的大小也可能会被不同的 BT 传输软件所改变,而前 4 个字节组成的整数和后一个字节表示的字符的组成可识别这些信息包的特征。因而本文就以信息包中前 4 个字节组成的整数和后 1 个字节表示的字符为 1 个特征字串来识别具体的信息包。

表 1　BT 包有效载荷特征表

包的标识	包大小	前四个字节内容	后一个字节内容
Request	17	13	6
Have	9	5	4
Choke	5	1	0
Unchoke	5	1	1
Interest	5	1	2
Not-Interest	5	1	3
Bitfield	不定	不定	5
Piece	片的大小+13	片的大小+9	7
Cancel	17	13	8
握手信息包		前 20 个字节为"19BitTorrent protocol"	

在考虑从各种信息包中选择用于识别 BT 数据的特征包时,从表 1 中可以看到,Bitfiled 信息包中前 4 个字节的特征不定,且其在整个 BT 数据交

万方数据

互过程中也不会重复出现，所以不考虑将其作为识别 BT 数据的特征包。Piece 信息包包含的是客户所需的实际文件数据，BT 协议将文件分成固定大小的片，封装在 Piece 信息包中进行传输，由于此信息包一般会大于 MTU，在链路层会进行分片处理，因而单个 Piece 信息包的特征字串不能完整的识别整个 Piece 信息包，故亦不考虑将其作为识别 BT 数据的特征包。Have，Unchoke，Interest3 个信息包所具有的特征确定，前 4 个字节和后 1 个字节有确定的数值，且在 BT 数据传输过程中会重复出现，很好的满足了选择特征字串的要求，因而将这 3 个包确定为所需的特征包。此外，Choke 信息包是在网络状态不好时发出，Not-Interest 信息包是在一方不需要另一方数据时发出，Cancel 信息包则是在传输将要结束时发出，且都具有固定的格式，因而可以将它们也作为识别 BT 数据的特征包，以提高识别的准确性。这样所有这些信息包的特征都是由 5 个字节组成，对识别效率的影响有限。同时由于可识别的特征数据包的增加，和特征数据包在传输过程中的重复出现将使得对 BT 数据识别的准确性上升，避免出现一旦没有捕获握手信息包就无法识别接下来的 BT 数据的情况发生。

再加上普遍使用的握手信息包特征，从而确定以表 1 中除 Bitfield 和 Piece 信息包外的所有数据包为特征包，其所具有的特征字串为用于识别 BT 数据的特征字串。

4 Linux 内核对 BT 流量控制算法的设计

可以通过上述 BT 通信流程的分析进行监测，获得 BT 的业务流信息。下面设计出动态的业务流配结构和相应的流识别算法识别出整个 BT 通信流量。整个流程的算法如下：

1.对捕获的数据包提取报头的元组信息，搜索 Hash 表，若与 Hash 表中某个流结点元组信息匹配，则该数据包为这个 BT 会话流的数据包，此时标记该数据包(nfmark=1)返回 1；否则 goto2。

2.根据上节 BT 建立会话流程的分析，对符合特征字符串的 TCP 连接包确定是 BT 的请求报文。若是 goto3，否则 goto7。

3.根据上节是 BT 的应答报文的特征分析，对符合特征字符串的 TCP 连接包确定是 BT 的应答报文。若是则提取此时通信双方的 IP，将流(source_ip，dest_ip，TCP)标记为建立连接的会话，goto4；否则 goto7。

4.根据上节的分析，当源 IP 是 3 中 source_ip 和目的 IP 是 3 中 dest_ip 并且符合特征字符串的数据包确认是建立数据通道的数据包，此时通信双方的端口就是动态会话的数据传送端口，由于可能有多个端口对，因此此时提取出的元组信息可能有多个(source_ip，source_port，dest_ip，dest_port，TCP)，将其全部标记为动态会话的流元组信息。goto6；否则 goto5。

5.根据上节的分析，当源 IP 是 3 中 source_ip 和目的 IP 是 3 中 dest_ip 并且特征值是结束会话的数据包，此时删除 Hash 表源 IP 是 source_ip 和目的 IP 是 dest_ip 的所有元组信息，goto7。

6. 将流信息(source_ip，source_port，dest_ip，dest_port，TCP)写入 Hash 表；返回 1。

7.返回 0；分析下一个数据包。

下面给出了专门用于存放元组的流结构 voip_flow。

```
struct bt_flow
{int source_ip; //源 IP
int dest_ip; //目的 IP
unsigned short int source_port; //源端口
unsigned short int dest_port;  //目的端口
int protocol;  //协议类型
struct timeval update_time; //更新时间
int pktNumber; //匹配包的个数
int type; //业务类型，保留成员以便扩展
struct voip_flow *nextPtr; //指向下一个流信息结点
};
```

在实现了对 BT 数据包的识别之后，我们就可以通过 Linux 的带宽控制工具 TC 来实现对 BT 应用的流量控制，将命令写成脚本文件在系统启动时自动加载。其中关键的命令如下：

tc qdisc del dev eth1 root 2>/dev/null// 清除网卡上的所有 QoS 规则

tc qdisc add dev eth1 root handle 20: htb default 20// 定义新建下载最顶层根用户规则

tc class add dev eth1 parent 20: classid 20:1 htb rate 94000kbps ceil 940000kbps // 定义 10:1 总下载带宽 94M

tc class add dev eth1 parent 20:1 classid 20:10 htb rate 40kbps ceil 100kbps prio 0// 建立一个通道，最小保证带宽 40K，最大占用带宽 100K。

tc qdisc add dev eth1 parent 20:10 handle 201: pfifo

tc filter add dev eth1 parent 20: protocol ip prio 100 handle 2010 fw classid 20:10 // 建立队

万方数据

列策略和过滤

iptables -F -t mangle

/sbin/iptables -t mangle -A POSTROUTING -m ip2p -bi -j MARK --set-mark 2010 将 bt 应用的数据打上标记并丢到相应的通道

5 实验验证与分析

本文在校园内部机房搭建网络实验环境，如图 2 所示。

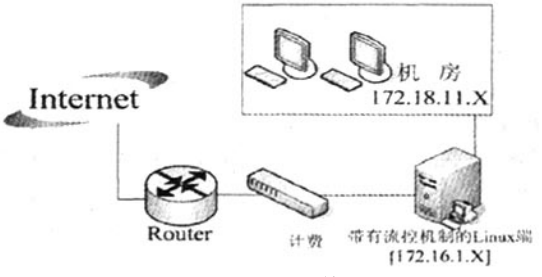

图 2 实验环境图

实验将 Linux 控制机 (Intel 至强 X5300、8GRAM、双千兆网卡)以串联的方式接入网络，局域网中其他机器均通过 Linux 控制机与外界联网。通过 MRTG 软件采集路由与计费系统间端口数据，监测所有出入境流量。为了测试整个策略库的有效性和实用性，实验从实际流量的监控来进行评估。

运行单策略限制整个局域网的流量，实验效果图如图 3 所示。

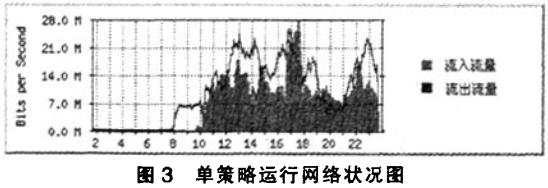

图 3 单策略运行网络状况图

初始 18:00 时，机房多台计算机在进行 BT 下载，此时路由流出流量介于 14-21MB/s,即 BT 下载流量速率始终介于 14-21MB/s 之间，19:00 对 BT 限流，速率明显降至 7MB/s 以下，后 21:00 取消策略限制，流量恢复。

通过上述实验结果分析，网络流量始终处于期望的受控状态，整个策略库表现出良好的管理精度。由于整个策略设计采取分而治之的思想，避免了在对制定一个较大网络的流量控制方案时容易出现的混乱状况，有效提高了带宽管理的简洁性和易用性。

参考文献:

[1]Ivancic, D. Hadjina, N. Basch, D. Analysis of precision of the HTB packet scheduler[C]. Electromagnetics and Communications, 2005. ICECom 2005. 18th International Conference, 2005:1-4.

[2]Kevin Kaichuan He. Why and How to Use Netlink Socket [J]. Linux Journal, 2005,130:124-137.

[3]M. Devera. Hierarchical Token Bucket Theory[EB/OL].http://luxik.cdi.cz/-devik/qos/htb,2003.S Analysis of protocols: Review of traffic scheduler features on general purpose platforms[J].ACM SIGCOMM Computer Communication Review, 2001, 31(2):473-489.

[4]Semenov, V.K, Filippov, T.V Polyakov, Yu.A. Likharev, K. K. SFQ balanced comparators at a finite sampling rate[J]. Superconductivity IEEE Transactions,1997,7(2):3617-3621.

[5]ALMESBERGER W. Linux Network traffic control-Implementation Overview [R].Technical Report SSC/1998/037, November 1998:12-30.

[6]Rober Love. Linux 内核设计与实现[M]. 北京:机械工业出版社, 2006: 12-90.

[7]L. Zhang Virtual Clock.A New Traffic Control Algorithm for Packet—Switched Networks [J].ACM Transaction on Computer Systems,2005,9(2):101-124.

[8]SmarTraining.Red Hat Linux9 网络服务[M].北京:机械工业出版社,2005:412-440.

[9]W.Richard Stevens, Stephen A.Rago. UNIX 环境高级编程[M].北京:人民邮电出版社,2006:507-540.

责任编辑:胡德明

Theory and Implementation of BT Flow Control Based on Linux Kernel

Lv Xiaohong

(Network Center, Huangshan University, Huangshan245041, China)

Abstract: Using the Linux Kernel-based network flow control mechanism, this paper probes into the Netfilter/Iptables and bandwidth control technology in Linux Kernel. Through an analysis of BT protocol and BT actual transmission, various tagged strings in BT transmission are picked out to create a tag-identifying rule database needed for Iptables to improve the accuracy of recognition and the accuracy of the existing BT flow control technology within the framework. The paper finally demonstrates with experiments the efficiency and feasibility of the flow control.

Key Words: flow control; netfilter/iptables; bit torrent; Linux

万方数据

万方数据 WANFANG DATA 文献链接

基于Linux内核的BT流量控制的原理与实现

作者： 吕效红， Lv Xiaohong
作者单位： 黄山学院,网络中心,安徽,黄山,245041
刊名： 黄山学院学报
英文刊名： JOURNAL OF HUANGSHAN UNIVERSITY
年，卷(期)： 2010，12(3)
被引用次数： 0次

参考文献(9条)

1. Ivancic,D. Hadjina,N. Basch,D Analysis of precision of the HTB packet scheduler 2005

2. Kevin Kaichuan He Why and How to Use Netlink Socket 2005

3. M.Devera Hierarchical Token Bucket Theory 2003

4. Semenov,V.K. Filippov,T.V. Polyakov,Yu.A. Likharev,K.K SFQ balanced comparators at a finite sampling rate 1997(2)

5. ALMESBERGER W Linux Network traffic control-Implementation Overview[Technical Report SSC/1998/037] 1998

6. Rober Love Linux内核设计与实现 2006

7. L.Zhang Virtual Clock A New Traffic Control Algorithm for Packet-Switched Networks 2005(2)

8. SmarTraining Red Hat Linux9网络服务 2005

9. W.Richard Stevens. Stephen A.Rago UNIX环境高级编程 2006

相似文献(10条)

1. 期刊论文 杨虎. 张大方. 谢鲲. 雷渊明. 何施茗. YANG Hu. ZHANG Da-fang. XIE Kun. LEI Yuan-ming. HE Shi-ming Netfilter/Iptables框架下基于TCP滑动窗口的串行流量控制算法 -计算机工程与科学2009,31(10)

传统的基于流量整形的流量控制算法通常需要建立对应的队列模型,实施起来极为复杂,而且所有数据包都要进入整形队列,加大了网络延时.本文从TCP协议拥塞控制和数据包组包机制出发,提出了基于TCP滑动窗口的串行流量控制算法,通过改变TCP发送端窗口的大小来达到流量控制的目的.本文在Linux内核Netfilter/iptables框架中实现了该流量控制方法,在部署的网络环境中,比较了不同参数设置下的算法效果.与CBQ算法相比,该方法降低了数据包在队列中排队整形的延时.

2. 会议论文 杨虎. 张大方. 谢鲲. 雷渊明. 何施茗 Netfilter/Iptables框架下基于TCP滑动窗口的串行流量控制算法 2009

传统的基于流量整形的流量控制算法通常需要建立对应的队列模型,实施起来极为复杂,而且所有数据包都要进入整形队列,加大了网络延时.本文从TCP协议拥塞控制和数据包组包机制出发,提出了基于TCP滑动窗口的串行流量控制算法,通过改变TCP发送端窗口的大小来达到流量控制的目的.本文在Linux内核Netfilter/iptables框架中实现了该流量控制方法,在部署的网络环境中,比较了不同参数设置下的算法效果.与CBQ算法相比,该方法降低了数据包在队列中排队整形的延时.

3. 期刊论文 徐苏磊. 梁伟. XU Su-lei. LIANG Wei 基于Netfilter/Iptables内核扩展的P2P流量管理 -计算机技术与发展2010,20(6)

为了缓解P2P流量对网络造成的带宽影响,合理利用网络资源,准确识别和测量P2P流量,才可以更好地保障网络的QoS.而传统上按照端口方式来识别P2P流量,随着P2P应用的发展,这种方法已经不能满足对P2P流量管理的需要.介绍了P2P应用及其优缺点,分析了Netfilter和Iptables架构的实现机制和扩展技术,以及P2P协议的特征.阐述了如何利用Netfilter/Iptables框架进行内核扩展来实现P2P流量识别与管理,通过实验进行了验证,并且对实验的结果进行了简单分析与总结,从分析的结果来看,明显在对P2P流量识别和管理上有所提高.

4. 学位论文 罗聪 基于Linux内核扩展模块的P2P流量控制研究 2009

随着计算机技术尤其是网络技术的不断发展，越来越多的业务的处理都更加依赖于网络，对网络带宽的要求也越来越高，但是网络上许多于业务无关的网络流量却在吞噬着带宽，使正常的网络运行受到很大的影响。尤其是现在由于P2P技术的快速发展，占用了大量带宽，从而导致很多业务因为带宽资源被占而无法进行。

传统的P2P流量检测和控制工具一般是通过源/目的IP地址、MAC地址、TCP/UDP端口等进行检测和控制，缺少对其应用层流量分析，但是现在很多的P2P软件都使用了动态端口，或者是伪装成了HTTP流量，利用传统的检测和控制就很难对P2P流量进行检测和控制。而在本文则中充分利用了Linux2.4内核防火墙框架Netfilter/Iptables的可扩展性，并结合linux流量整行工具对P2P流量进行识别和控制，使网络资源能合理分配。

本文中作者提出了一种基于Linux内核扩展模块的P2P数据识别和控制的方法，首先在网络上抓包，并且对常见的P2P协议进行分析，通过Linux系统防火墙框架Netfilter连接跟踪机制来跟踪网络上的数据包，并根据数据包的的特点来判断这个包所属的连接类型。作者详细分析Linux Netfilter/Iptables框架并通过P2P协议分析的结果对其进行扩展，然后根据各种P2P协议的应用层的特征来对P2P流量进行识别，然后在Linux内核中设置一些过滤规则来管理P2P的流量。利用这种方法可是使得网络中的P2P流量完全被丢弃，同时，Linux的服务质量的合理分配资源的P2P网络的手段来限制使用带宽，可以使用网络性能得到大幅度的提高。

本文开发的基于Linux内核扩展模块的P2P流量控制系统，主要采用的是对Iptables中表规则的设置，同时还结合了带宽管理中的队列规则。基于应用层的特点来分析和识别各种P2P流量，可以将一些网络管理员不想要的P2P下载流量完全禁止，也可以利用Linux流量控制技术将P2P下载的流量控制在

一个保证网络稳定运行的范围内。本文为网络管理者提供了一个很好的流量监控的解决方案。

5.期刊论文 李勇.周健.邵东轶.LI Yong.ZHOU Jian.SHAO Dong-yi 一种基于Netfilter的BitTorrent流量控制方法 -计算机安全2008(4)

BitTorrent是目前互联网上广泛使用的一种基于P2P的文件共享协议.它使用了动态端口,这给BitTorrent流量控制带来了很大的困难.该文在分析BitTorrent协议基础上,给出了一种基于应用层特征匹配的BitTorrent流量控制方法.首先提取出BitTorrent流的应用层特征,利用Linux 的Netfilter/lptabIes扩展架构实现数据包的应用层特征匹配.并利用TC实现对它的流量控制.

6.期刊论文 郑伟发.杨创新 基于Netfilter/Iptables和TC的带宽管理设计与实现 -华南金融电脑2009,17(4)

随着网络应用的迅速发展,带宽管理日益成为网络管理不得不面对的问题.Linuxnetfilter/iptables结合TC(Linux TKacfic Control)工具实现了一种性价比较高的带宽管理方案,成为很多网络管理员的重要选择.本文首先介绍Linux内核防火墙netfilter/iptables的实现原理,然后介绍TC工具,最后结合实例详细说明如何利用iptables工具和TC工具实现网络流量控制.

7.学位论文 梁泠霞 基于QoS带宽管理技术的P2P流量控制研究及其实现 2006

在互联网飞速发展的今天,企业、学校等各种机构越来越多的关键业务依赖于互联网,但是许多与工作事务无关的网络应用占去了绝大部分的带宽,导致关键业务响应时间过长,严重影响了正常网络应用的运行.特别是现在的网络应用更为复杂,P2P等下载软件无限制地占用带宽资源,它们使用动态端口,同时将本身的流量伪装成为HTTP流量,很难被防火墙、路由器以及其他的过滤设备发现.传统的网络级防火墙(包过滤)都是工作在OSI模型的第2、3或者4层,通过基于数据包的源/目的IP地址、MAC地址、TCP/UDP端口等进行过滤,缺少对应用层(OSI模型的第7层)流量的分析,因此也无法识别P2P下载的流量.在本文中,提出了一种基于Linux系统防火墙框架Netfilter连接跟踪机制的P2P数据识别和控制的方法.连接跟踪系统负责跟踪所有的连接并且判断每个经过的包属于哪个连接.通过扩展Netfilter/Iptables框架,可以根据应用层数据识别P2P连接,并可以通过设置适合的防火墙过滤规则来管理P2P流量.此方法可以完全禁止不想要的P2P流量,还可以和Linux QoS工具一起使用以限制P2P的带宽占用,合理分配网络资源,从而提高网络性能.本文开发的P2P流量控制系统,基于应用层分析技术识别各种P2P流量,可以限制或者禁止P2P下载,也可以利用Linux流量控制技术将P2P下载的流量控制在某个范围内.总之,本文对于网络中间存在的P2P下载提供了很好的监视和控制解决方案,使用的应用层分析技术也可让网络管理者对网络资源的应用分布情况得到充分的了解和认知.

8.期刊论文 周华先.王伟平.ZHOU Hua-xian.WANG Wei-ping 基于Linux下L7-filter模块的P2P流量控制 -湖南科技学院学报2008,29(4)

本文分析了Linux Netfilter/Iptables架构下L7-filter的功能、工作原理和实现机制,以及P2P协议的特征.通过编写相匹配模板文件,扩展防火墙的规则集,从而实现P2P流量控制的目的,而且可以根据不断出现的P2P业务来更新模板文件,具有很好的扩充性能.

9.期刊论文 陈海军.李仁发.杨磊.CHEN Haijun.LI Renfa.YANG Lei 基于Linux内核扩展模块的P2P流量控制 -计算机工程2007,33(1)

分析了Linux Netfilter/Iptables架构的实现机制和扩展技术,分析了P2P协议的特征,通过扩展Linux内核库,利用共享库实现用户数据空间与内核空间的数据交互,扩展防火墙的规则集,从而实现P2P流量控制的方法,而且可以根据不断出现的P2P业务更新规则集,具有很好的扩充性能.

10.学位论文 苟继军 分布式并行流量控制技术的研究与实现 2007

随着Internet应用需求的快速增长，用户对带宽的需求不断增加，同时对网络可靠性的要求也愈来愈强。然而ISP提供的单一网络出口可能在带宽及可靠性方面不能满足用户需求；同时为了满足更多用户对于高稳定性、高可靠性、高性能低成本路由器的需求，分布式并行多出口路由器(Distributed&parallel multi portrouter，DPMPR)系统应运而生。

分布式并行多出口路由器结合了分布式并行技术和广域网多出口技术，分布式并行技术使本系统具有高稳定性、高扩展性、高可靠性；多出口技术的应用不仅增加了内部网络的出口带宽，更增强了内部网络的可靠性，提高了服务质量水平。

本文首先介绍了分布式并行路由器系统的开发背景，项目的研究内容和本文的主要工作内容。

然后讨论了系统涉及到的相关技术、概念和原理。介绍了分布式并行技术的相关概念，分布式系统的发展过程、研究现状。分析研究了TCP/IP协议的层次结构和Linux系统对TCP/IP协议的具体实现，QoS技术的相关概念以及Linux系统为保证服务质量采用的QoS技术，详细描述了Netfilter/Iptables的原理及其实现过程。

其次提出了分布式并行路由器系统的流量控制方案。该方案采用分布式并行技术，在系统范围内实现对用户流量的动态控制。具体策略如下：(一)基于用户IP带宽的流量控制，该策略可以精确控制所有用户的网络流量；(二)对所有用户设置优先级，周期性统计系统当前活动用户的优先级总数，根据系统出口带宽值计算出IP对应可用的带宽值，此方案可充分利用网络出口的带宽，在线用户数量较少的时候可以获得较大的可用带宽，用户较多的时候也可以保证所有用户都得到一定的服务质量保证；(三)综合策略一、二，对用户分配权值和带宽值，然后根据系统总流量情况分别执行不同策略：当系统流量小于40％时，对所有IP不设流量限制，转发所有数据包；当系统流量小于60％时，按照策略一执行，当系统流量大于60％时，按照策略二执行。该方案针对性强，可以给用户提供更好的服务质量。

最后验证了流量控制模块功能的正确性、稳定性和可靠性。

本文链接：http://d.wanfangdata.com.cn/Periodical_hsxyxb201003010.aspx
授权使用：黄山学院学报(qkhsxy)，授权号：833f634a-2eb4-4ebf-b8ec-9ebd00ad0360
下载时间：2011年4月6日