网络流量控制系统在开源路由器上的实现

黄文焱,褚 伟
(合肥工业大学 计算机网络系统研究所,安徽 合肥 230009)

摘 要:路由器是接入网络的关键设备,为了防止网络流量过大而造成网络拥塞的状况,设计了一种在路由器上的网络流量控制系统。DD-WRT是一种开源的路由器固件,对DD-WRT重新定制开发,可以实现用户自己想要的一些功能。介绍了一种基于DD-WRT实现的网络流量统计和控制系统,介绍了系统捕包、流量统计和流量控制的一般流程,同时着重介绍了针对网络应用的流量统计和控制方法。系统采用嵌入式Linux编程,实现了在开源路由器中对网络流量进行控制的功能,并有效地防止了网络拥塞。

关键词:开源路由器;流量统计;流量控制;网络应用;网络拥塞
中图分类号:TP393.07 **文献标识码**:A **文章编号**:1673-629X(2010)08-0225-04

Realization of Network Traffic Control System in Open-Source Router

HUANG Wen-yan, CHU Wei
(Institute of Computer Network Systems, Hefei University of Technology, Hefei 230009, China)

Abstract: Router is the key equipment to access network, in order to prevent excessive network flow and network congestion, a network traffic control system on the router has been designed. DD-WRT is an open source firmware of router, you can re-develop it, and you can achieve some of the features you want. Present a network traffic statistics and control system based on DD-WRT, and describe the general flow of the sniffing, the statistics and the control of the system, at the same time highlight the traffic statistics and control methods for network applications. The system is programmed by embedded Linux, and it achieves the network traffic control functions in the open-source router, effectively prevent network congestion.

Key words: open source router; traffic statistics; traffic control; network applications; network congestion

0 引 言

网络的发展,目前主要是应用的多样化,对于网络流量控制和带宽管理提出了新的要求,从简单地针对IP或端口的带宽管理到针对不同应用,满足不同需求的流量控制。通过带宽管理来改善网络状况,可以采取扩大带宽容量和控制网络流量的方法。由于扩容的花费较大,因此,通过控制网络流量来改善网络状况,成为了人们研究的热点。

目前,在局域网接入互联网的方式中,接入路由器是其中的一种关键设备,所以在接入路由器中实现对局域网访问外网的流量进行控制是一种较为有效的方法。虽然商业路由器能够实现流量控制的基本功能,但固件的更新慢,可扩展性差,不能适应因网络应用不断丰富而对流量控制个性化实现的要求。而且,商业路由器源代码不公开,因此不能由第三方软件开发者开发特定软件。而目前出现了多种开源的路由器软件,可以运行在某些特定型号的路由器上。因此可以在开源路由器软件的基础上,根据需求实现流量控制功能的定制开发。

DD-WRT就是其中一种开放性的路由器固件,它其实就是一个供无线路由器使用的嵌入版Linux,它可以采用Broadcom公司CPU的小型无线路由器实现数千元的商用无线路由器功能,而且人们甚至可以自行编译程序,自由扩展无线路由器功能。

本系统在DD-WRT原有功能的基础上扩展开发了流量统计和控制的功能,通过查看网络流量状态,并对带宽进行分配来控制网络流量,从而有效地解决了网络拥塞的问题。

1 总体方案设计

现在一些小区、楼层、小的企事业单位,由于人们大量的下载上传以及P2P的普遍使用,并且缺乏一种

收稿日期:2009-12-16;**修回日期**:2010-03-16
基金项目:国家自然科学基金资助项目(90718037)
作者简介:黄文焱(1984-),男,福建上杭人,硕士,研究方向为计算机网络、嵌入式Linux;褚 伟,副教授,研究方向为计算机网络、嵌入式Linux。

规范的管理机制，所以经常导致网络流量激增，甚至拥塞。因此针对这种情况，设计一种能查看网络流量状态，并对其进行有效控制的基于开源路由器的流量控制系统。

系统主要分成三个部分设计：数据包捕包模块、流量统计模块和流量控制模块。为了有效捕获数据包，捕包模块采用 pf_ring 的环形缓冲区捕包机制。流量统计功能可按 IP 地址、端口号和应用协议三种方案进行统计。当网络发生拥塞时，在流量控制模块中动态地调用 TC(Traffic Control)来对网络流量进行控制。系统的基本流程如图 1 所示。

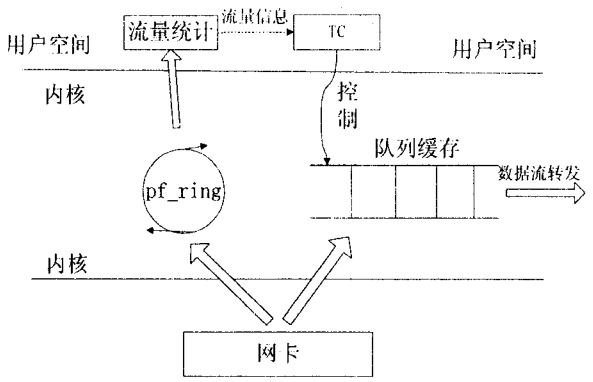

图 1 总体流程

2 关键技术

2.1 pf_ring 捕包机制

针对传统 libpcap[1]捕包处理时间长、效率低的缺点，采用了一种新的基于环形缓冲区的套接字模型 pf_ring[2]。它的主要工作原理如下所述：

采用 PF_RING 技术，操作系统将包采用 DMA 方式拷贝到内核缓冲区的环形队列中，再把网卡缓冲区中收到的这些包丢弃，不把它们传入内核协议栈缓存区中进行排队。环形缓存区在每个套接字被建立时分配，直到套接字关闭时，环形缓存区才被释放。每次由网卡缓存区拷入内核环形缓存区时，不会进行分配和去配的操作，而是新到的包按环形的方式，将原有的包覆盖。PF_RING 技术还提供了对 MMAP 技术的支持。该技术将用户应用程序空间映射到内核缓存区，从而省略了将数据从内核缓存区向用户缓存区的拷贝操作。这样可以节省一次拷贝所占用的系统资源和缩短包处理的时间，从而提高了捕包的效率。

2.2 流量统计方案

(1)按照 IP 地址汇聚。

系统对源和目的 IP 地址进行流量汇聚，统计出各 IP 地址单位时间内的上行流量和下行流量。这种统计方式能反映出本地各主机、各网段的网络负载情况，系统可以此为依据，进行路由调整及流量控制。

(2)按照端口号汇聚。

TCP/UDP 的端口号代表网络上的不同应用(HTTP、FTP、P2P 等)[3]，按照单位时间内访问的 TCP/UDP 端口来统计流量，可以查看各端口的流量分布情况，当网络出现异常时，可按照端口号来进行流量控制。

(3)按照网络应用汇聚。

常规的网络应用一般通过常用的端口就可以识别[4]，并进行流量统计，然而 P2P 技术不断地发展演进，其拓扑结构从最初的集中式发展到纯分布式再到目前的混和式架构，其端口特性也由最初的固定端口发展到随机动态端口再到伪装端口[5]。所以，系统对应用进行统计，主要是对 P2P 应用[6]的解析统计。

对 P2P 应用的解析，首先是经过以太网解析，获取 IP 数据包，然后再通过网络层和传输层解析，得到源目的 IP 地址、源目的端口、传输层协议类型和完整的 payload 信息。由于 P2P 协议一般动态使用非知名端口进行通信，因此仅仅根据端口来检测 P2P 流是不准确的，必须进行应用层协议识别。在应用层识别过程中，根据协议格式以及消息里的特征字符串，采用深度包检测(DPI)方法[7]来识别 P2P 业务。深度包检测时采用模式匹配方法查找特征字符串。例如，可以根据特征字符串“0x13 BitTorrent protocol”检测出 BitTorrent 协议数据包，根据特征字符串“0xe319010000”检测出 eDonkey2000 协议数据包[8]。根据网络应用统计流量就是要按照这些特征字符串来进行流量汇聚。同时，解析 P2P 协议，也为按应用进行流量控制提供了一种可靠的手段。

2.3 带宽分配机制

Linux 内核提供了强大的带宽管理代码，它主要使用规则过滤工具 netfilter/iptables 和路由工具包的流量控制命令 TC 相结合的方式来进行带宽控制[9]。

netfilter/iptables IP 信息包过滤系统实际上是由两个组件 netfilter 和 iptables 组成的。netfilter 组件被称为内核空间(Kernel Space)，是内核的一部分，主要由一些信息包过滤表组成，这些表中包含内核用来控制信息包过滤处理的规则集。iptables 组件则是一种规则过滤工具，它称为用户空间(User Space)，主要用于插入、修改和除去信息包过滤表中的规则。

TC 是 Linux 环境下一种功能强大的网络流量控制软件，它可以分为三个部分：队列策略(Queue Discipline)、分类器(Classifier)和过滤器(Filter)。队列策略实质是一些算法，控制如何处理进入队列的报文。队列策略算法主要有 FIFO(先进先出)、RED(随机早期

探测)、CBQ (类基队列)和 HTB(层次令牌桶)等[10]。过滤器按照过滤条件,将数据报进行分类处理。一般来说,数据报的处理步骤如下:队列策略对数据报文进行调度,过滤器根据报文信息来决定把它放入到哪一个类中。在不同的类中,每个类也包含一个队列策略,同样进行调度、分类,将报文按照既定的规则排序发送出去。

3　系统实现

3.1　捕包模块

模块采用 pf_ring 套接字的方式,用户层通过调用 socket(PF_RING,SOCK_RAW, tons(ETH_P_ALL))建立一个 PF_RING 类型的 socket ,并返回一个套接字描述符。接着调用 bind(fd,(struct sockaddr *)&sa,sizeof(sa))将 socket 绑定到本地 IP 和端口。socket 和 bind 实际上分别调用了 ring_create 和 ring_bind,而这两个函数的作用就是为套接字创建一个环形缓冲区,然后将其绑定到一个设备上。通过建立的 PF_RING 套接字就可以进行数据传输了。由于用户空间采用了直接访问内核空间的环形缓冲区的方式,所以效率比原来的 libpcap 有了明显提高。之所以称之为环形缓冲区是因为在连续的内存中有一个 FlowSlotInfo 结构[11],该结构中包含了描述环状缓存的基本信息,插入删除都是循环操作的。当有数据包被网卡接收时,通过 add_skb_to_ring 来实现将 sk_buff 插入到环形缓冲区,从 skb->data 中读取一个数据包头结构,然后使用 memcpy 直接将 insert_slot 处内存覆盖。

3.2　流量统计模块

通过捕包模块获取数据包及其信息。通过 IP 地址,统计出各 IP 以及各 IP 网段的网络数据流量,并统计出总的上行和下行流量。通过端口,统计出 TCP/UDP 源和目的端口的流量。通过应用层 payload 字段的字符串特征符来统计相应网络应用的流量。当总体流量超过一定的限定值,而引起网络拥塞时,根据具体情况,选择以上流量统计中的一种作为依据,动态地调用 TC,对网络流量进行控制。

3.3　流量控制模块

流量控制的基本步骤为:(1)针对网卡建立一个队列;(2)取出数据包源/目的地址、源/目的端口、协议类型五元组信息;(3)通过 DPI 对到达的数据包进行识别;(4)根据用户要求,使用 netfilter/iptables 匹配功能,为匹配到的数据包打上 mark 值;(5)在这个建立的队列上再建立分类;(6)设置过滤器为每一分类建立选路,使数据流进入相应队列,并分配带宽;(7)数据包调度。

以下是对 BT 流量控制[12]进行设置的一个例子。

iptables 设置:

iptables -A PREROUTING -t mangle -p tcp -m BT -j MARK --set-mark 1 //将 BT 连接标记为"1"

TC 设置:

tc qdisc add dev eth0 handle 1:0 root htb //创建 HTB 的根队列策略

tc class add dev eth0 parent 1:0 classid 1:1 htb rate 95000kbps ceil 95000kbps //设置主类带宽

tc class add dev eth0 parent 1:1 classid 1:2 htb rate 10kbps ceil 80kbps prio 3 //建立一个 BT 子类,设置带宽在 10k 到 80k 之间

tc qdisc add dev eth0 parent 1:2 handle 1:2:1 pfifo //对 BT 子类建立 FIFO 队列策略

tc filter add dev eth0 parent 1:0 protocol ip prio 100 handle 1 fw classid 1:2 //设置过滤器,将标记为"1"的 BT 包送到 1:2 这个类中

采用 HTB 规则为以太网卡绑定一个主队列,并创建根分类,然后可以选择通过 IP 地址、端口号或网络应用分出子类,之后结合 iptables 的 mark 值进行过滤,数据流进入相应分类的队列。但是为了保证网络用户的基本使用,系统对一些常规的业务直接输出,而不作控制。流量控制流程图如图 2 所示。

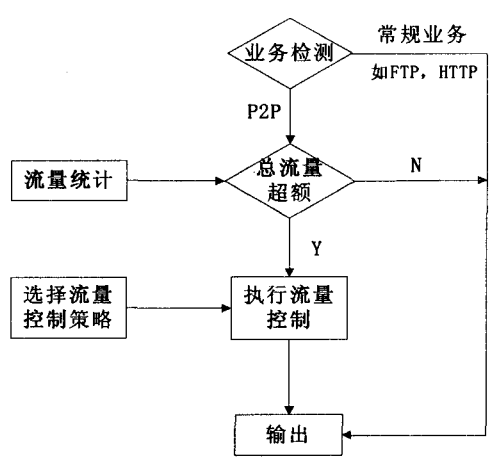

图 2　流量控制流程图

4　DD-WRT 的流量控制功能定制

定制步骤如下:

1) 下载固件解压和压缩工具的源代码 firmware_mod_tools.tar.gz,解压并编译这个工具;

2) 下载固件 dd-wrt.v24_std_generic.bin,使用 firmware_mod_tools 解压,生成目录 dd-wrt;

$ ./extract_firmware.sh dd-wrt.v24_std_

generic.bin dd－wrt/

3) 修改 dd－wrt 固件，将流量控制程序添加进 dd－wrt 目录中；

在目录 dd－wrt 中有两个目录文件，一个是 image_parts，一个是 rootfs。其中 image_parts 中保存的是固件的引导内核，rootfs 中保存的是固件中的文件。直接在 roots 中添加流量控制程序。

4) 重新打包 dd－wrt 固件，将其保存到 new_ddwrt 中；

$ ./build_firmware.sh new_ddwrt/ dd－wrt/

5) 将定制好的 DD－WRT 固件下载到路由器中，刷新路由器固件；

6) 重新启动路由器就可以实现路由器的新功能了。

5 实验结果分析

在实验中，根据流量统计数据，分别画出控制前和控制后的总体流量、P2P 流量和 HTTP 流量曲线图，如图 3 和图 4 所示。

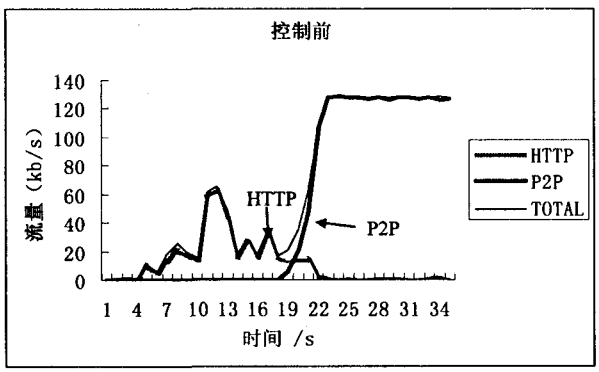

图 3 控制前流量统计图像

根据控制前后图像对比，系统有效地控制了网络数据流量，并且保证了常规数据流的通信。

6 结束语

本系统通过对 DD－WRT 的重新定制，添加了流量统计和流量控制功能。系统的主要创新点在于：添加了对网络应用流量进行统计和控制的功能，这在一定程度上有效地控制了异常流量的发生，并且保证了常规流量的正常通信。

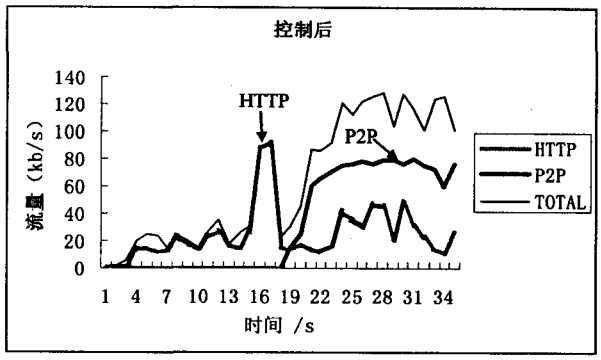

图 4 控制后流量统计图像

参考文献：

[1] 刘文涛.网络安全开发包详解[M]. 北京：电子工业出版社，2005.

[2] Wood P. Libpcap－mmap, Los Alomos National Labs[EB/OL].2006－04.http://public. lanl. gov/ cpw.

[3] Stevens W R. TCP/IP 详解卷 1：协议[M].范建华，译.北京：机械工业出版社，2006.

[4] 吴 敏，王汝传.基于主机的 P2P 流量检测与控制方案[J].计算机技术与发展，2009，19(10)：26－29.

[5] 李江涛，姜永铃.P2P 流量识别与管理技术[J].电信科学，2005，49(3)：57－61.

[6] Thomas, Karagiannis, Broido A, et al. Transport Layer Identification of P2P Traffic[C]// International Measurement Conference. Laormina, Italy:[s.n.], 2004.

[7] 於时才，安凌鹏.协议分析与深度包检测相结合的入侵防御系统[J].微计算机信息，2009(7－3)：67－69.

[8] 陈 亮.基于特征串的应用层协议识别[J].计算机工程与应用，2006，42(24)：16－19.

[9] 姚伯威，田 珂.Linux2.4 下的带宽(流量)控制功能[J].计算机应用，2001(4)：16－17.

[10] 高 杰，沈 军.基于下一代流量控制机制 TCNG 的带宽管理实现[J].微计算机信息，2006，22(4－3)：146－148.

[11] Corbet J，Rubini A，Kroah－Hartman G. Linux Device Drivers[M].USA:O'Reilly，2005.

[12] 李 勇. 一种基于 Netfilter 的 BitTorrent 流量控制方法[J]. 计算机安全，2008(4)：65－69.

(上接第 224 页)

[7] Priestley M. Practical Object－Oriented Design with UML[M].北京：清华大学出版社，2005.

[8] 罗时飞.精通 Spring[M].北京：电子工业出版社，2008.

[9] 孙卫琴.精通 Struts：基于 MVC 的 Java Web 设计与开发[M].北京：电子工业出版社，2007.

[10] 周志刚，徐 芳，肖晓华，等 . 应用 Struts 框架开发管理信息系统的研究[J]. 河南理工大学学报，2006(5)：415－419.

[11] 董崇杰，傅秀芬，王凤梅，等.基于 J2EE 公安厅审计系统的设计与实现[J].计算机技术与发展，2009，19(9)：246－249.

[12] Ambler S W. Mapping Objects to Relational Databases：O/R Mapping In Detail[EB/OL]. 2004.http://www. agiledata.org/essays/mappingObjects.html.