一种P2P流量监控系统的设计及实现

徐 鹤[1],王汝传[1,2]
(1.南京邮电大学 计算机学院,江苏 南京 210003;
2.南京大学 计算机软件新技术国家重点实验室,江苏 南京 210093)

摘 要:当前P2P流量已经成为互联网流量的重要组成部分,针对互联网中存在大量的P2P业务流量影响了互联网关键业务的应用,设计了一种P2P流量监控系统。该系统采用分布式检测技术,通过网络抓包工具Analyzer分析的P2P业务流量特征,对P2P流量进行检测,并在网卡驱动NDIS层实现P2P流量控制。其监控策略由系统管理员制定,而策略的具体实施是在各个终端的网卡驱动层,通过实验验证了该方案的可行性。实验结果显示该系统检测精度高和扩展性强,易于在现有网络中部署实现。

关键词:P2P;流量检测;流量控制;控制策略
中图分类号:TP393 **文献标识码**:A **文章编号**:1673-629X(2009)10-0006-05

Design and Implementation of P2P Traffic Monitoring System

XU He[1],WANG Ru-chuan[1,2]
(1.College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003,China;
2.State Key Laboratory for Novel Software Technology at Nanjing University, Nanjing 210093,China)

Abstract:The extensive use of P2P (Peer-to-Peer) network technology has a serious impact on the existing network. In this paper, design a P2P traffic monitoring system, which uses distributed detection technology, through network packet capture tool-Analyzer analysis the P2P services traffic character to detect P2P flow, and realized P2P traffic control in the network driver layer-NDIS. The control strategy developed by the system administrator and implemented by the various terminals in the network driver layer. Through experimental ensuring the system is feasibility, and the experimental result shows that the designed system having a good scalability and high detection, and it is easy realized in the existing network.

Key words:Peer-to-Peer;traffic detection;traffic control;control strategy

0 引 言

20世纪90年代后期,P2P网络技术[1]被广泛采用,基于P2P技术的应用层出不穷。P2P的传播和使用使得P2P流量占据了当前网络相当大部分的Internet流量,对现有网络业务的正常使用产生了严重的影响。目前P2P技术主要应用到以下几个领域:提供文件和其它内容共享的P2P网络;挖掘P2P对等计算能力和存储共享能力;基于P2P方式的协同处理与服务共享平台,例如JXTA、Magi、Groove、.NET My Service等;即时通讯交流和语音通信软件,包括ICQ、OICQ、Yahoo Messenger、Skype等;视频共享系统,如基于P2P的VOD和IPTV等;安全的P2P通讯与信息共享,例如CliqueNet、Crowds、Onion Routing等。其中P2P内容共享系统对网络带宽的占用度最大,尤其是P2P文件共享系统。互联网上P2P用户的总数庞大到数以百万计,其累计业务量已占互联网业务的总量,造成了网络带宽的巨大消耗,妨碍了正常的网络业务的开展和关键应用的普及,同时通过并不安全的网络环境获得的应用程序和P2P协议给网络安全带来了极大的隐患。由于P2P技术在互联网上的突飞猛进普及和发展,成为网络资源的最大消耗者,已经超过了Web、E-mail、FTP等的数据流量,成为网络的主要负担,甚至引起网络拥塞,影响和降低其它业务的性能。据相

收稿日期:2009-02-01;**修回日期**:2009-05-10
基金项目:国家自然科学基金(60573141,60773041);江苏省自然科学基金(BK2008451);国家高科技863项目(2007AA01Z404,2007AA01Z478);南京市高科技项目(2007软资127);现代通信国家重点实验室基金(9140C1105040805);江苏省博士后基金(0801019C);江苏高校科技创新计划项目(CX08B-085Z,CX08B-086Z)
作者简介:徐 鹤(1985-),男,安徽巢湖人,博士研究生,研究方向为网络测量与安全、多媒体技术和计算机软件在通信中的应用;王汝传,教授,博士生导师,研究方向为计算机软件、计算机通信、信息安全、无线传感器网络、移动Agent技术和虚拟现实技术等。

关统计,迄今为止,P2P 业务共占所有宽带数据吞吐量的 80% 以上。因此,在许多情况下需要有特别的技术来检测 P2P 流量。在 P2P 应用早期阶段,由于 P2P 协议使用特定的 TCP 或 UDP 端口,P2P 流量很容易识别。然而,很快 P2P 应用能够使用动态变化的端口来躲避识别。而现在,随着 P2P 技术的发展,P2P 应用使用加密技术使得原有的检测方法很难再识别。

目前 P2P 流量识别与控制技术[2]研究已经成为当前 P2P 技术研究的一个重要方向。由于 P2P 应用协议的多样性、自定义性,造成了对该部分流量难以统计和控制。文中设计了一种 P2P 流量监控系统,该系统采用分布式检测技术,策略由系统管理员制定,而策略的具体实施是在各个终端。该系统与现有技术相比,可以避免单点监控系统性能瓶颈,且能够根据要求制定策略,具有良好的可扩展性。

1　P2P 流量监控系统模型

1.1　系统的网络结构图

本监控系统可以应用于校园网内 P2P 流量的监控。其运行的网络结构如图 1 所示。其中,管理服务器需要安装运行本系统监控中心管理程序;终端需要安装运行终端应用程序和 NDIS 驱动程序[3]。监控中心管理程序主要负责控制策略的管理(见第 2.1 节)、流量统计和日志管理等功能。终端根据具体的控制策略进行流量检测和控制。

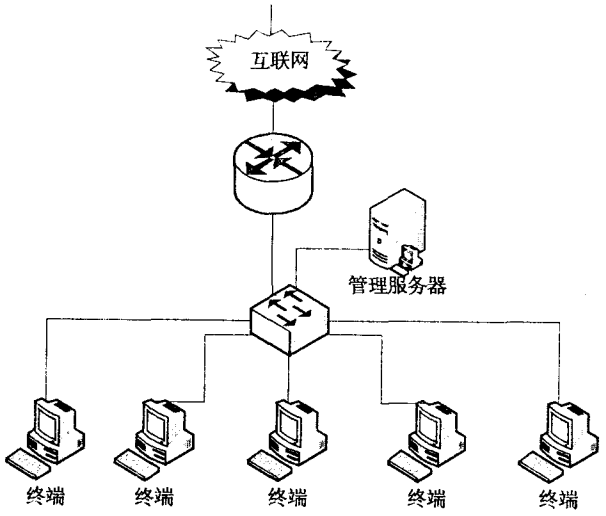

图 1　P2P 流量监控系统网络结构图

1.2　系统的体系结构

整个系统分为应用层、服务层、连接层和 NDIS 层,层次结构如图 2 所示。

应用层:包括流量统计、控制策略和日志管理模块。流量统计模块收集终端监控系统汇报的关于 P2P 流量的统计数据(如识别的 P2P 数据包大小、类型、连接信息等)。控制策略模块主要提供控制策略的管理。日志管理模块提供监控系统的一切操作及数据记录。

服务层:包括 UI 界面层和 Web Service 服务层。方便管理员查询网络使用状态信息。

连接层:包括流量监视、流量匹配和流量控制模块。可以根据策略要求进行流量监控,实现 Winsock 层流量监控,如在此层进行 TCP/IP 连接数限制和连接带宽限制等。

NDIS 层:包括流量监视、流量匹配和流量控制。可以根据策略要求进行流量监控,实现内核层流量监控,如可以根据要求进行流量限制和发包速率限制等。

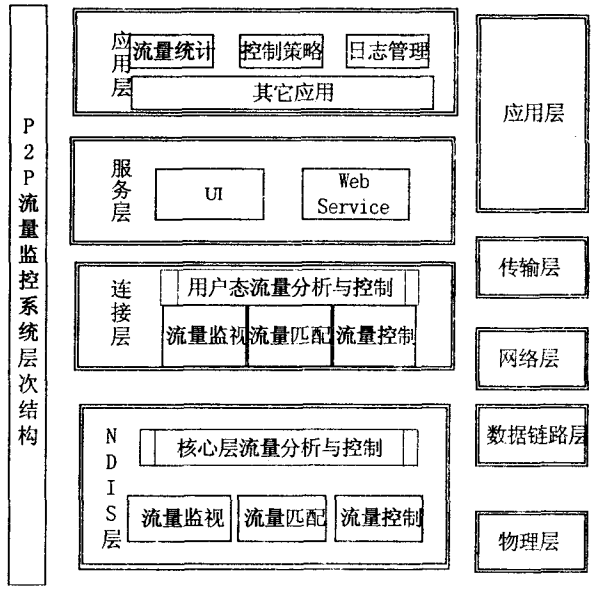

图 2　P2P 流量监控系统体系结构图

2　P2P 流量监控系统的实现

本系统主要分为管理服务端、终端应用程序和终端 NDIS 驱动程序。系统各部分的功能在第 1.1 节已经说明。

2.1　控制策略的管理

在 P2P 流量监控系统中,控制策略是根据每个终端分别制定的。每个终端都有自己的默认的策略信息库,当需要改变控制策略时,管理员改变策略信息并发送改变的控制策略信息给终端。控制策略的管理分为策略的生成、分发、执行和存储。

策略生成:控制策略是由管理服务器端的管理员根据一定格式要求制定。

策略分发:当终端启动网络时登陆管理服务器,管理服务器发给终端控制策略信息。

策略执行:根据相应的控制策略对终端的数据包进行检测与控制。策略的执行是在每个终端上,且每个终端的策略可能不同。

策略存储:控制策略在终端和管理服务器端各有一份。终端的策略信息加密存储在文件中,防止终端用户非法篡改;服务器端的策略信息存放在专门的策略信息库中,且与每个终端一一对应。

2.2 系统的实现

2.2.1 NDIS 技术介绍

NDIS(Network Driver Interface Specification)是 Microsoft 和 3Com 公司开发的网络驱动程序接口规范的简称,它支持如下三种类型的网络驱动程序:微端口驱动程序(Miniport Driver)、中间层驱动程序(Intermediate Driver)和协议驱动程序(Protocol Driver)。其中中间层驱动介于协议层驱动和微端口驱动之间,其功能非常强大,可以提供多种服务,能够截获所有的网络数据包(以太帧),过滤微端口驱动程序,实现特定的协议或其他诸如数据包加密、认证等功能。在 NDIS 中间层捕获网络数据包的方法结构规范,功能强大。文中的系统实现核心部分就是采用中间层驱动捕获网络数据包,对其进行分析判断是否是 P2P 数据包。

下面首先对微软提供的一个中间层驱动范例 Passthru 进行分析,然后讨论如何实现文中提出的流量监控系统。

2.2.2 *微软的中间层驱动范例 Passthru 分析*

Passthru 是 Microsoft 在其推出的免费驱动开发工具(Driver Development Kits,DDK)中提供的一个中间层驱动源代码范例。它提供了中间层驱动程序的总体框架,可以帮助开发人员迅速掌握中间层驱动程序的实现方法。Passthru 工作在网络层和媒体接入控制层之间,实现了对网络数据包的底层截获。只要在 Passthru 基础上进行适当的修改,就可以实现各种各样的应用。

下面具体介绍与本系统实现密切相关的 Passthru 发送、接受数据流程,以及驱动程序与应用程序通信的基本内容。

2.2.2.1 Passthru 发送数据流程

图 3 为 Passthru 发送数据流程图。

(1)Protocol driver 调用 NdisSend 向下层发送数据报文。

(2)Passthru 的 MPSend/MPSendPacket 例程根据上层传下来的数据报文分配 MyPacket,调用 NdisSend 发送到下层。如果返回 pending,就在 PtSendComplete 中释放 MyPacket;否则就在本函数中紧接着释放 MyPacket。

(3)当下层 miniport driver 发送完成 MyPacket 以后,会调用 NdisMSendComplete。

(4)NDIS 接着调用 Passthru 的 PtSendComplete,在这个函数里,应该释放 MyPacket,并且通知上层 Protocol driver 去释放它们的 packet。

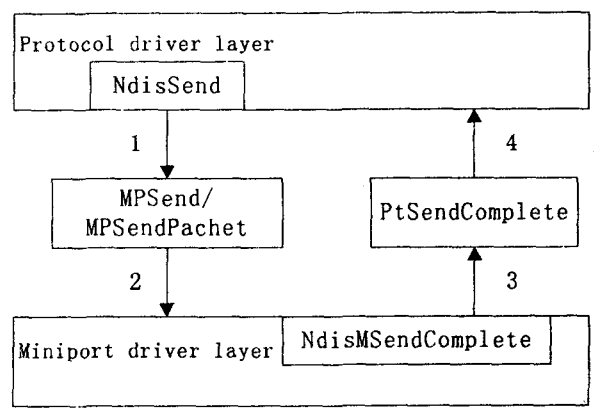

图 3 Passthru 发送数据流程图

2.2.2.2 Passthru 接收数据流程

图 4 为 Passthru 接收数据流程图。

(1)底层驱动使用 NdisMIndicateReceive / NdisMEthIndicateReceive 通知上层已经收到数据报文。

(2)在 PtReceive 中如果通过 NdisGetReceivedPacket 得到了一个完整的 packet,就分配自己的 MyPacket,根据底下传上来的 packet 设置 MyPacket,然后调用 NdisMIndicateReceivePacket 通知 NDIS,NDIS 会接着调用上层协议驱动的相应 PtReceive 例程。如果此时 MyPacket 的 status 是 NDIS_STATUS_RESOU-

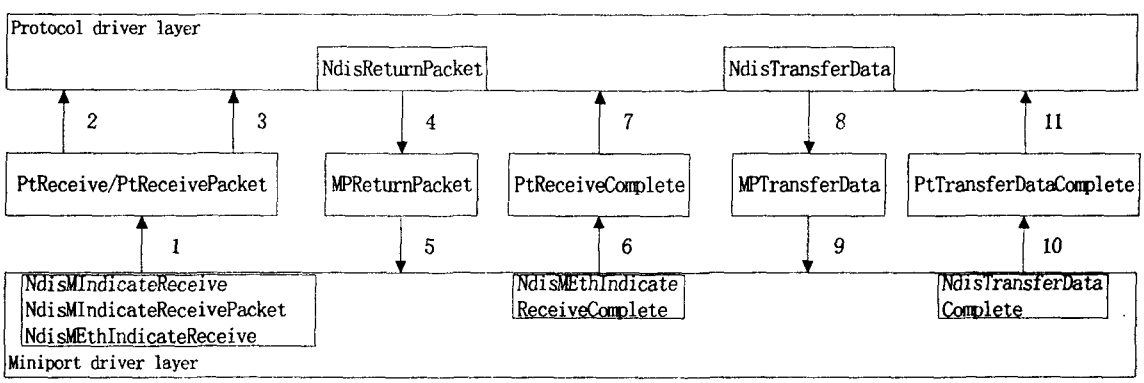

图 4 Passthru 接收数据流程图

RCES,就在本函数中释放分配的 MyPacket;否则在上层发送(4)的时候,在 MPReturnPacket 中释放 MyPacket。

(3)在 PtReceive 中如果通过 NdisGetReceivedPacket不能得到一个完整的 packet,那就直接调用 NdisMEthIndicateReceive 等函数通知 NDIS。

(4)当上层协议驱动得到了一个完整的数据报文并且处理完毕以后,它会调用 NdisReturnPacket,然后 NDIS 会调用 MPReturnPacket。

(5)在 MPReturnPacket 中,释放自己分配的 MyPacket,然后同样向下层调用 NdisReturnPacket。下层会释放他们自己分配的 packet。

(6)如果(3)发生,当底层 miniport 驱动收到了一个完整的数据报文,它会调用 NdisMEthIndicateReceiveComplete,然后 NDIS 会调用 PtReceiveComplete。

(7)PtReceiveComplete 同样会调用 NdisMEthIndicateReceiveComplete,通知 NDIS"我们已经收到了完整的报文"。

(8)当上层协议驱动得知底层已经收到了完整的数据报文以后,可能会调用 NdisTransferData,要求下层把剩余的数据传上来。

(9)(8)的调用会导致 NDIS 调用 MPTransferData 例程。在 MPTransferData 中,做同样的调用 NdisTransferData。注意该函数的返回值:如果返回 success,说明剩余的数据立刻就传上来了,此时会立即返回,(10)、(11)两步骤就不会调用;如果返回 pending,表明底层在此阻塞,底层会在稍后的时候调用(10)。

(10)当底层 miniport 驱动做好了一个完整的 packet,它会调用 NdisTransferDataComplete。

(11)同样在 PtTransferDataComplete 中,会作出同样的调用。

2.2.2.3　驱动程序和应用程序之间的通信

监控系统对截获的网络数据包进行分析,根据控制策略规则对数据包进行处理,并且将这些事件记入日志。由于控制规则设置和日志的记录一般是由一个具有人机界面的应用程序来完成,因此,要实现本系统还必须实现驱动程序和应用程序的信息交互。

驱动程序和应用程序的通信之间的通信包括两个方面:

(1)应用程序传送数据给 Passthru 驱动程序。

这部分实现比较容易,应用程序通过 CreateFile()函数获取设备驱动程序的句柄后,就可以使用如 DeviceIoControl()、ReadFile()或 WriteFile()这样的 Win32 函数来实现与 Passthru 驱动程序之间的通信。在本监控系统中使用 DeviceIoControl()通信。

(2)Passthru 驱动程序传送数据给应用程序。

当驱动程序按要求识别出相关 P2P 数据包后需要把一些信息(如数据包大小、源 IP 和目的 IP、源端口和目的端口等)送到应用程序并做日志记录。

在本监控系统中使用事件方式来实现驱动向应用程序的消息发送来通知应用程序有数据要传送。在事件方式下,应用程序先创建事件,然后将事件句柄传给驱动程序,应用程序将事件句柄传给驱动程序,可以使用(1)所介绍的 DeviceIocontrol()的方法。接着创建一个监听线程,等待事件的有信号状态。驱动程序获得该事件的句柄后,使用 ObReferenceObjectByHandle 来引用该事件对象,将它转换成能够使用的事件指针,并存储起来以便后面使用。当驱动程序有事件告诉应用程序时就通过使用 KeSetEvent()函数将事件设置为有信号状态,这样应用程序的监听线程马上收到消息并采取相应的消息处理。

2.2.3　系统监控终端的实现

监控终端分为终端应用控制程序和 NDIS 驱动程序。

终端应用控制程序负责与 NDIS 驱动交互信息,并与管理服务器通信,获取控制策略信息和向服务器汇报终端的监控数据。

在 NDIS 驱动程序中截获到网络数据包后,需要根据 P2P 应用协议通信时使用的协议特征对数据包进行处理。在实现本系统时,通过 Analyzer 网络抓包工具分析了 BitTorrent[4],eDonkey[5]和 PPLive[6]这几种 P2P 软件的通信协议特征(如表 1 所示)。监控系统在 NDIS 驱动中截获数据包后,结合这些 P2P 软件的协议特征进行分析,如果是 P2P 通信数据包则依据控制策略对其进行处理,并把处理结果实时传送给终端应用控制程序统计并做日志记录。

表 1　P2P 协议特征[7~10]

P2P 应用程序	使用传输层协议	特征
BitTorrent	TCP	从第 1 个字节起为 \ x13bittorrent| bitTorrent| torrentportal
eDonkey	TCP	第 1 个字节为 \ x e3 或 \ x c5
eDonkey	UDP	第 1 个字节为 \ x e4
PPLive	UDP	前 2 个字节为 \ x e903

2.2.4　实验结果和分析

在实验室搭配了如图 1 所示的网络环境下,把检测软件的客户端安装在终端,终端只运行 PPLive 软件,软件检测结果如图 5 所示。

通过实验的结果可以看出,该监控系统可以有效

地检测和控制网络中的 P2P 流量，且检测精度高、性能好，同时该系统网络架构可以应于多种网络环境，系统具有良好的可扩展性。

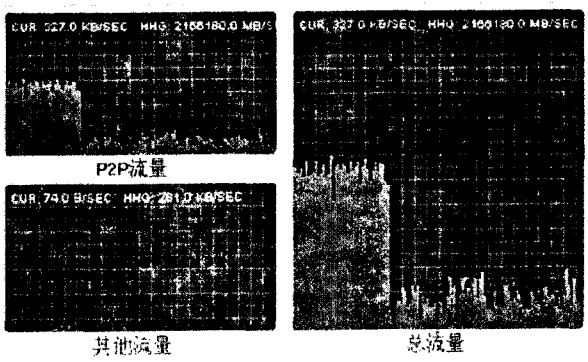

图 5 监控系统检测出的 P2P 流量

3 结束语

近年来 P2P 技术得到飞速发展，同时给网络带来了很大的负担，对 P2P 流量的识别和控制成为当前 P2P 技术研究一大热点。文中设计的 P2P 流量监控系统，采用分布式监控技术，控制策略由系统管理员根据要求制定，而控制策略的具体实施在各个终端上。该系统的实现与现有技术相比，可以避免单点监控系统性能瓶颈，能够根据要求制定控制策略，具有良好的可扩展性。

但是当前的系统还有缺点：基于 NDIS 的中间层驱动程序与操作系统结合的很紧密，进一步的工作是对其改进实现使其对不同操作系统兼容。此外，本系统只能够实现对有特定的 P2P 通信特征字的 P2P 流量进行监控，因此对未知协议的 P2P 流量识别算法研究也是下一步工作的重点。

参考文献：

[1] 吴国庆. 对等网络技术研究[J]，计算机技术与发展，2008,18(7):100－103.

[2] 蒋海明，张剑英，王青青，等. P2P 流量检测与分析[J]. 计算机技术与发展，2008,18(7):74－76.

[3] 郭兴阳，高 峰，唐朝京. 一种 NDIS 中间层数据包过滤方法[J]. 计算机工程，2004,30(17):102－103.

[4] Bittorrent[EB/OL]. 2008. http://www.bitcomet.com.

[5] EDonkey[EB/OL]. 2008. http://www.edonkey2000.cn.

[6] PPLive[EB/OL]. 2008. http://www.pplive.com.

[7] Cheng W Q, Gong J, Ding W. Identifying BT－like P2P Traffic by the Discreteness of Remote Hosts[C]//32nd IEEE Conference on Local Computer Networks. Washington, USA: IEEE Computer Society, 2007:237－238.

[8] Zhang Qi, Piumatti M, Singhal S K. Private Peer－to－Peer Overlay for Real－Time Monitoring of a Deployed Internet－Scale Peer－to－Peer Overlay[C]//Seventh IEEE International Conference on Peer－to－Peer Computing. Galway, Ireland: IEEE Computer Society, 2007: 235－236.

[9] Schollmeier R, Dr. Schollmeier I G. Why Peer－to－Peer (P2P) Does Scale: An Analysis of P2P Traffic Patterns[C]// Proceedings of the Second International Conference on Peer－to－Peer Computing (P2P'02). Linköping, Sweden: IEEE Press, 2002:112－119.

[10] Guo Lei, Chen Songqing, Zhen Xiao, A Performance Study of BitTorrent－like Peer－to－Peer Systems[J]. IEEE Journal on Selected Areas in Communications, 2007, 25(1): 155－169.

(上接第 5 页)

类别的股票信息归并到一起集中显示，这样让使用者更清晰地掌握各支股票的信息，能很大程度上协助进行相关分析。

参考文献：

[1] 梁 循. 数据挖掘－建模、算法、应用和系统[J]. 计算机技术与发展，2006,16(1):1－4.

[2] 梁 循，陈 华，杨 健，等. 基于互联网股市信息量和神经网络的股价波动率预测[J]. 金融科技，2005,5(109):92－96.

[3] Han J, Kamber M. 数据挖掘原理与技术[M]. 北京：机械工业出版社，2001.

[4] 杨 健. 股票市场技术分析手册[M]. 北京：中国宇航出版社，2002.

[5] 梁 循. 数据挖掘算法与应用[M]. 北京：北京大学出版社，2006.

[6] Hagan M T, Demuth H B, Beale M H. Neural Network Design [M]. [s.l.]: PWS Publishing Company, 1995.

[7] Osler C, Chang K. Head, shoulders: Not just a flaky pattern [M]. Staff Report No. 4, Federal Reserve Bank of New York, 1995.

[8] Leigh W, Paz N, Purvis R. Market timing: a test of a charting heuristic[J]. Economics Letters, 2002, 77(1): 55－63.

[9] Leigh W, Modani N, Hightower R. A computational implementation of stock charting: abrupt volume increase as signal for movement in New York Stock Exchange Composite Index [J]. Decision Support Systems, 2004, 37(4): 515－530.

[10] Lo A, Mamaysky H, Wang J. Foundations of technical analysis: computational algorithms, statistical inference, and empirical implementation[J]. Journal of Finance, 2000, 55(4): 1705－1765.