基于 Netfilter/Iptables 内核扩展的 P2P 流量管理

徐苏磊,梁 伟
(南京邮电大学 计算机学院,江苏 南京 210003)

摘 要: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。

关键词:P2P;Netfilter;Iptables;P2P 特征码;流量控制
中图分类号:TP393 **文献标识码**:A **文章编号**:1673-629X(2010)06-0101-04

P2P Flow Control Based on Netfilter/Iptables Kernel Extension

XU Su-lei, LIANG Wei
(School of Computer, Nanjing University of Posts & Telecommunications, Nanjing 210003, China)

Abstract: In order to alleviate the network bandwidth traffic impact caused by P2P flow for the rational use of network resources, better protect the network QoS by accurate identification and measurement of P2P flow traffic. With the development of P2P application, the traditional method that detects the P2P flow by the according port can't meet the requirement of P2P management. Describe the advantages and disadvantages of P2P applications, analysis of the Netfilter framework and Iptable expansion technology, as well as the characteristics of P2P protocols. Describe how to use Netfilter / Iptables framework for kernel extension to implement P2P traffic identification and management, and conduct a brief analysis and summary. From the results of analysis, P2P traffic identification and management is improved obviously.

Key words: P2P; Netfilter; Iptable; P2P signature; flow control

0 引 言

P2P 使得网络上的沟通变得容易、更直接共享和交互,真正地消除中间商。P2P 就是人们可以直接连接到其他用户的计算机、交换文件,而不是像过去那样连接到服务器去浏览与下载。P2P 另一个重要特点是改变互联网现在的以大网站为中心的状态,重返"非中心化",并把权力交还给用户[1]。

当前 P2P 技术主要应用于文件共享和交换,且大多是用于大型的多媒体文件的共享和交换。据统计,P2P 应用已占 ISP 业务总量的 60%~80%,成为网络带宽最大的消费者[2,3]。然而 P2P 对带宽的影响至今还不能给出一个定量的评估,这一切源于 P2P 特有的协议特征,给识别、测量带来了巨大的挑战,传统基于端口的流量分类和测量的方法已经失效。为此,文中提出在 Netfilter/Iptalbes 框架下实现的一个 P2P 流量测量系统,该系统能较准确地识别并测量此 P2P 流量,进而可定量研究 P2P 流量对网络的性能影响[4]。

1 Netfilter/Iptables 的扩展技术

Linux 中防火墙 Netfilter/Iptables 系统主要包括两个基本组件:定义在内核空间中的通用框架 Netfilter,即:包分割框架,数据包选择系统。其中后者又由两部分构成:在 Netfilter 框架上定义的数据结构"IP 表"、"链"和"规则",在用户空间实现的应用程序 Iptables 用于插入、修改和删除信息包过滤表中的规则。由于 Netfilter 架构的加入,可以通过简单的内核模块化来实现新功能的扩展,在现有的(Netfilter/Iptables)中可以通过两种方式对现有的防火墙进行扩充:一种是扩展 Netfilter 通过编写相关内核模块调用,直接在相关的钩子上注册从而获得新特性;一种是扩展 IP 表通过

收稿日期:2009-10-26;**修回日期**:2010-01-28
基金项目:江苏省科技计划项目(BG2007045)
作者简介:徐苏磊(1983-),男,江苏连云港人,硕士生,研究方向为计算机通信网及其嵌入式网络;梁 伟,博士,高级工程师,研究方向为计算机通信网及 IP 技术。

编写相关的匹配标准和目标来实现新特性。扩展 IP 表是对现有表的匹配规则的扩充与具体表无关。扩展 IP 表需要编写内核和用户两方的代码,内核模块提供了实际的数据包匹配规则代码,用户方代码提供了 Iptables 新的命令行选项的共享库。

Netfilter 组件是 Linux 内核的一部分,是 Linux2.4 下防火墙的系统基础,它支持基于状态监测(通过 connection track 模块实现)的防火墙结构。Netfilter 是一个设计很合理的框架,可以在适当的位置上登记一些需要的处理函数,如在 NF_IP_FORWARD 点上登记具有转发包过滤功能的函数,包过滤等功能便是由这些登记的函数实现的。也可以登记自己的处理函数,或对原有的函数进行修改,以实现对防火墙过滤功能的修改和扩展。Netfilter 的结构如图 1 所示,它由一些表(table)组成,而这些表包含了内核用来控制信息包过滤处理的规则集。

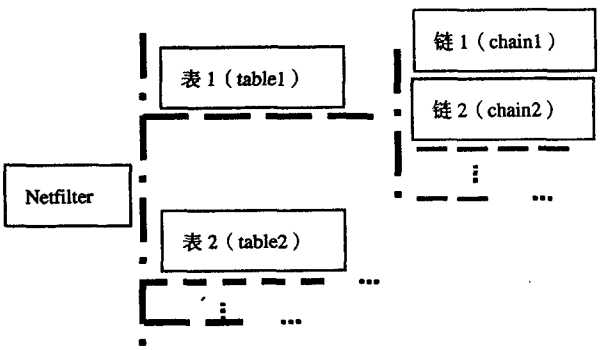

图 1 Netfilter 框架结构

Iptables 在内核中定义了 3 个表:Filter, Mangle 和 Nat。一个表就是一组相关的防火墙规则(rules)的集合[5]。Filter 表实现包过滤(如果只实现包过滤,则只用到 Iptables 中的 Filter 表),Mangle 表提供修改数据报的方法,Nat 表实现源和目的地址变换。每个表又由若干链(chain)组成,而每个链又由一条或多条规则组成。这些规则是决定 Iptables 如何处理网络数据包的最终依据。Iptables 在 Linux 中的操作完全用 Iptalbes 命令来实现。

此外,由于 Netfilter 框架并不是防火墙的具体实现,而是用于扩展网络服务的结构化底层框架。为了利用该结构,Linux 在 IPv4 协议栈中定义了 5 个挂载点,以便对网络数据包进行自定义操作。挂载点的位置如图 2 数字所示。

通过 Netfilter 框架提供的 API,即可在挂载点定义自己的函数来控制网络数据包[6]。对应的挂载点说明如表 1 所示。

2 P2P 协议特征码

要有效管理 P2P,前提是必须能识别出 P2P 网络流量。随着 BitTorrent 程序源代码的公开,P2P 应用软件越来越多[7]。如何准确快速地识别 P2P 网络数据包成为比较困难的问题。一种比较简单的方法是直接通过已知 P2P 软件常用端口判定 P2P 网络流量,但这可能导致使用其中某端口的其他应用无法正常使用。另外如果 P2P 应用软件能够动态调整端口,则该方法就无能为力了。例如 BitTorrent 通常在 6881 端口监听,若该端口被占用,则一直尝试到 6889 端口[8]。

表 1 Netfilter 框架挂载点说明

序号	挂载点	作用对象	实现功能
1	NF_IP_PRE_ROUTING	刚进入网络层的 IP 数据包	源地址转换
2	NF_IP_LOCAL_IN	发往本机的 IP 数据包	过滤输入包
3	NF_IP_LOCAL_OUT	本地发出的 IP 数据包	过滤输出包
4	NF_IP_FORWARD	需要转发的 IP 数据包	过滤转发包
5	NF_IP_POST_ROUTING	设备发出的所有数据包	目的地址转换

文中采用借鉴和归纳部分各种 P2P 应用软件的网络数据包特征码的方法加以判定。可以通过参考相关 P2P 应用软件规范说明书,如 BitTorrent、eDonkey 和 eMule 等软件都有详细的规范说明书,分析其说明书就可以归纳出特征码,如表 2 所示[9,10];可以通过捕获 P2P 网络数据包分析对应特征码,有些 P2P 应用软件虽然较为流行,对此可以通过 sniffer 程序捕获 P2P 应用软件的网络数据包,再进行手工分析,从中找出特征码,如表 3 所示[11]。

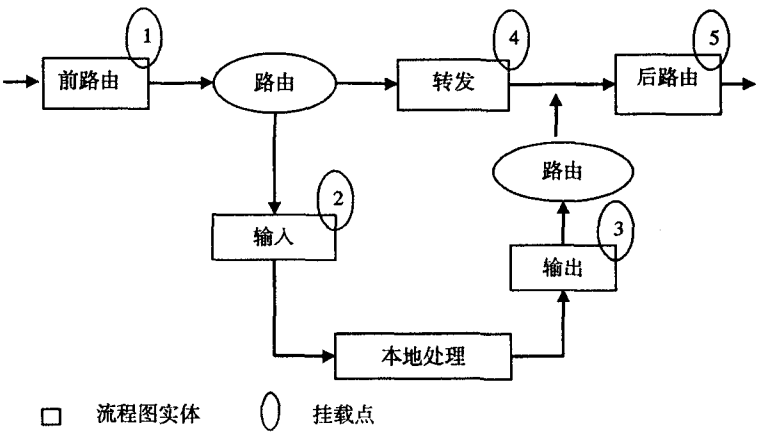

图 2 Iptables 工作流程图及挂载点

3 P2P 流量管理的 Netfilter/Iptables 内核扩展实现

3.1 在 Netfilter 中构造新的匹配函数

在对 Linux 内核扩展 Netfilter 框架时,为了实现

表 2　部分 P2P 数据包特征

应用程序	协议	类型	IP 负载大小
eDonkey	0xe3	0x9a	26
		0x96	14
eMule	0xc5	0x92	10
		0x93	10
Kda	0xc4	0x50	12
		0x59	10

表 3　部分 P2P 软件网络数据包特征

P2P 软件名称	特征码
BitTorrent	BitTorrent protocol
Gnutella	GND 或者 GNUTELLA
KaZaA	KaZaA
Ares Galaxy	PUSH SHA1:
卡盟(KAMUN)	KamunPeers protocol
百度下吧	BaiduP2P
迅雷	query

对 P2P 网络流量的简单管理,便于添加新特征码及调整控制策略,在设计时将特征码及其控制策略都写入配置文件中,从而与具体的实现代码分离,可在不修改源代码的情况下进行升级。配置文件简单示例如下:

BitTorrent|BitTorrent protocol|| -1|

其中,在文件中一种 P2P 软件为一行,彼此参数用"|"分割,方便读取文件的解析。文件的第 1 列为 P2P 软件名;第 2 列为特征码;第 3 列为空以便于添加新特征码;最后 1 列为控制策略,取值为 -1 表示继续传输,为 0 则丢弃,其他正数则表示该类型数据包的最高流速(限速,单位为 KBS),UDP 部分与此类似,可参考表 2,此处不再赘述。

在设计时,选用内核 2.4 版本中,此内核利用 Netfilter 框架时必须用如下函数向内核注册自定义处理函数:

```
void nf_register_hook (struct nf_hook_ops * reg);
```

注销函数如下:

```
void nf_unregister_hook (struct nf_hook_ops * reg);
```

这两个函数的入口参数的数据类型定义如下:

```
struct nf_hook_ops {
    struct list head list;//一般设置{NULL,NULL};
    nf_hookfn * hook; //自定义的挂载函数;
    int pf;//一般为 PF_INET;
    int hooknum;//选择的挂载点;
    int priority;//存在多个挂载函数通过此值判优;
    };
```

扩展内核 Netfilter 主要工作是设计好挂载函数,根据流程图(见图 3)自定义函数:filter_p2p_flow,即:reg->hook=filter_p2p_flow。此函数具体实现的关键代码如下:

```
static unsigned int filter_p2p_flow (unsigned int Hook_Num,
struct sk_buff * * Sk_Buf, const struct net_device * In_Dev,
const struct net_device * Out_Dev, int ( * okfn) (struct sk_buff
* ))
{
struct sk_buff Sk_Buf_In=Sk_Buf;
switch (Sk_Buf_In->nh.iph->protocol)
{
case IPPROTO_TCP:
return (p2p_In_Use = p2p_Parse_Tcp(Sk_Buf_In));
case IPPROTO_UDP:
return (p2p_In_Use = p2p_Parse_Udp(Sk_Buf_In));
}
}
```

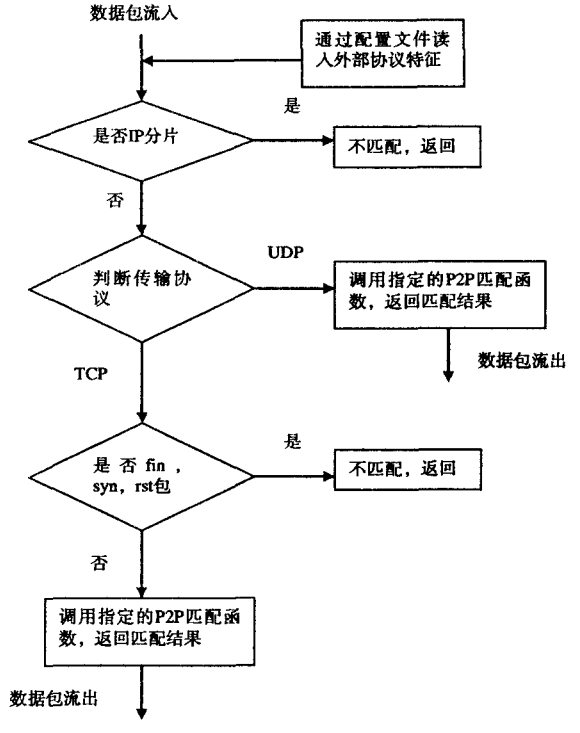

图 3　数据包处理流程图

3.2　Iptables 添加命令行匹配

为了更好地配合 Netfilter 扩展实现对 P2P 流量管理的模块,用户可以使用 Iptables 命令行实现对内核 Netfilter 的具体操作,为了添加新的命令行匹配选项,需要调用函数 register_match 来进行注册,它的入口参数为结构 iptables_match,其中最重要的成员为 parse 函数,在 parse 函数中对接受到的命令行参数进行检查和解释,将命令行输入转换为程序可读的格式,然后在调用 libiptc 库提供的 iptc_commit()函数向内核提交该操作请求。在 libiptc/libiptc.c 中定义了 iptc_commit()(即 TC_COMMIT()),它根据请求设置了一个

struct ipt_replace 结构，用来描述规则所涉及的表(filter)和 HOOK 点(FORWARD)等信息，并在其后附加当前这条规则的一个 struct ipt_entry 结构。组织好这些数据后，iptc_commit()调用 setsockopt()系统调用来启动核心处理该请求。在用户空间调用内核模块和用户空间互动对于 Netfilter 的方法是使用 setsockoPt 机制，每个协议必须被修改以用来为 setsockoPt 不能理解的号码调用 setsockoPt()(对于 setsockoPt 号码是 setsockopto)，并且迄今为止只有 IPv4，IPv6 和 DECnet 被修改[12]。

当把编译好针对 P2P 流量的新的匹配器模块 filter_p2p_flow 后，调用函数 register_match 来进行注册添加好匹配模块后，就可以通过 Iptalbes 定义相应的匹配规则，以及扩展模块的定义，可以实现 P2P 流量检测功能，并根据需要实施有效的控制和管理，甚至是实施特定环境下的丢弃数据包，如在网关式防火墙中实施的管理部署，对应的 Iptables 配置如下(# 代表终端命令行提示符)：

iptables -A FORWARD -p all -m filter_p2p_flow --bit -j DROP

iptables -A FORWARD -p tcp -m filter_p2p_flow --ares -j DROP

iptables -A FORWARD -p udp -m filter_p2p_flow --kaza -j DROP

这样的命令就能把 BitTorrent、ares、kazaa 这 3 种 P2P 网络的流量阻断。接下来的实验分析部分着重是针对 BitTorrent 进行分析，并且给出了对比图。

4 实验分析

通过实验测试，内核扩展实现 P2P 流量管理可以对常见的 P2P 应用，如 BT 和电驴等实施有效控制，可以很好地阻断和控制 P2P 流量，具有很好的扩展能力，更好地利用好网络资源，保障好网络的 QoS。在图 4 中给出了和固定端口在流量方面的比较，虚线为固定端口流量图，实线为文中模块 filter_p2p_flow 的流量图。具体的部署环境为第三部分叙述的那样，是部署在一个网关上，作为网关式防火墙，对齐进行一个可靠时间的流量采样，并且与固定端口的再流量方面的比较。

在具体对 BitTorrent 软件以半小时为累加，获得一段时间的流量数据比较图，明显在识别和管理上有所提高。文中所实现的扩展模块 filter_p2p_flow 对几个具体的 P2P 软件在具体的应用中的效果参照图表 4 所示。

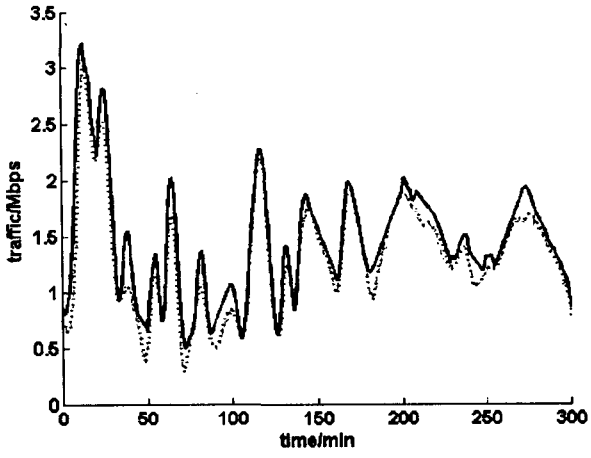

图 4 与固定端口对比图

表 4 扩展模块对几种具体应用的测试比较

模块选项	具体应用	协议类型	测试结果
-bit	BitTorrent	TCP/UDP	良好
-xunlei	迅雷	TCP/UDP	良好
-kaza	KaZa	CP/UDP	欠佳
-emu	eMule	TCP/UDP	良好

5 结束语

文中介绍了 P2P，以及由此带来的流量问题，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，文中提出了利用特征码识别 P2P 网络流量的方法，并利用 Linux 内核中的 Netfilter/Iptables 框架初步实现对 P2P 的管理。而且 Netfilter/Iptables 具有很好的灵活性，在该框架上可以方便地对防火墙进行功能扩展，在内核中保证了控制的响应速度，可以迅速更新规则库，同时通过该方式也可以按照自身需要扩展防火墙功能。下一阶段将进一步研究识别 P2P 网络数据包的新方法，以期对 P2P 网络数据包有更高的识别率。

参考文献：

[1] 张联峰，刘乃安，钱秀槟，等. 综述：对等网(P2P)技术[J]. 计算机工程与应用，2003，39(12)：142-145.

[2] Stein J H. Architecture and algorithms for internet-scale (P2P) data management[R]. Berkeley: Intel Research, 2004.

[3] Yagi S, Waizumi Y, Tsunoda H, et al. Network application identification based on transition pattern of payload length[C]//WCNC. LasVegas, USA:[s.n.], 2008.

[4] Sen S, Spatscheck O, Wang D, et al. Accurate, scalable in-network identification of p2p traffic using application signatures[C]// Proceedings of the 13th International Conference on World Wide Web. New York, NY, USA: ACM Press, 2004:512-521.

(下转第 108 页)

割效果。二维 Tsallis 交叉熵直线型分割方法，不仅考虑边界区域的信息，还考虑目标和背景间的信息量差异，有效地实现了图像的分割。另外其具有算法简单，运行速度快的特点，可见二维 Tsallis 交叉熵直线型分割方法是一种有效的分割方法。

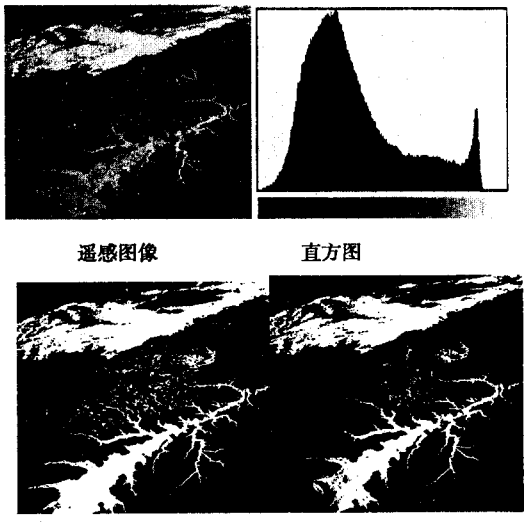

图 3 遥感图像分割结果

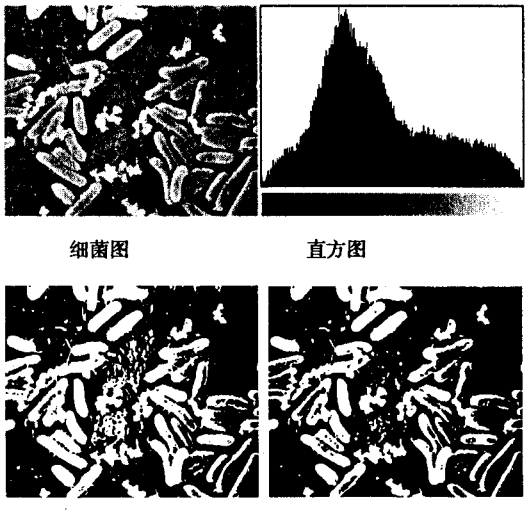

图 4 细菌图像分割结果

参考文献：

[1] Pun T. A new method for grey－level picture thresholding using the entropy of the histogram[J]. Signal Processing, 1980 (2):223－237.

[2] Kapur J N, Sahoo P K, Wong A K C. A new method for grey－level picture thresholding using the entropy of the histogram [J]. Computer Vision, Graphics and Image Processing, 1985 (3):273－285.

[3] Abutaleb A S. Automatic thresholding of gray－level pictures using two－dimensional entropy[J]. Computer Vision, Graphics and Image Processing, 1989(1):22－32.

[4] Albuquerque M P, Esquef I A. Image thresholding using Tsaillis entropy[J]. Pattern Recognition Letters, 2004, 25(9):1059－1065.

[5] Tang Yinggan, Di Qiuyan, Cuan Xinping. Method for thresholding image segmentation based on minimum Tsallis－cross entropy[J]. Chinese Journal of Scientific Instrument, 2008, 29:1868－1872.

[6] Sahoo P K, Arora G. Image thresholding using two－dimensional Tsallis－Havraa－Charvat entropy[J]. Pattern Recognition Letters, 2006, 27:520－528.

[7] Fan Jiulun, Zhao Feng. Two－Dimensional Ostu's Curve Thresholding segmention method for gray－level image[J]. Acta Electronica Sinica, 2007(4):751－755.

[8] Wang Junnian, Shen Quntai. A Clustering－Based Niching Particle Swarm Optimization[J]. Informational and control, 2005(5):213－217.

[9] 纪 震，廖惠连，吴青华，等. 粒子群算法及应用[M]. 北京：科学出版社，2009.

（上接第 104 页）

[5] 姚晓宇，赵 晨. Linux 内核防火墙 Netfilter 实现与应用研究[J]. 计算机工程，2003，29(8)：112－116.

[6] 余青霓，周 刚. Linux 防火墙[M]. 北京：人民邮电出版社，2000.

[7] 王春枝，李涛向. 基于双层特征的 P2P 流量检测[J]. 计算机技术与发展，2009，19(7)：238－241.

[8] 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.

[9] Karagiannis T, Broido A, Faloutsos M, et al. Transport Layer Identification of P2P Traffic[C]// Proceedings of the 4th ACM SIGCOMM conference on Internet measurement. New York, USA: ACM Press, 2004.

[10] Nicoll J R, Bateman M, Ruddle A, et al. Challenges in measurement and analysis of the BitTorrent content distribution model[C]// Proc of International Postgraduate Symposium on the Conver－gence of Telecommunications, Networking and Broadcasting. Liver－pool: Liverpool John Moores University, 2004.

[11] Constantinou F, Mavrommatis P. Identifying known and unknown peer－to－peer traffic[C]// Fifth IEEE International Symposium on Network Computing and Applications. Cambridge, MA, USA: IEEE Xplore, 2006:93－102.

[12] 周 诚，戴 忠，江 林. 基于 Netfilter 技术的复合防火墙系统研究与实现[J]. 计算机测量与控制，2007(6)：790－791.