

应用层负载均衡技术研究

胡安波, 苏金树, 陈曙晖

HU An-bo, SU Jin-shu, CHEN Shu-hui

国防科学技术大学 计算机学院, 长沙 410073

School of Computer, National University of Defense Technology, Changsha 410073, China

E-mail: genahuanbo@163.com

HU An-bo, SU Jin-shu, CHEN Shu-hui. Research on application-level load balancing technology. *Computer Engineering and Applications*, 2009, 45(14): 84-86.

Abstract: This paper introduces an application-level load balancing model, it distributes the network traffic by the types of applications. The model divides the network traffic into two parts, the ordinary application traffic and the P2P application traffic, and distributes the ordinary application traffic by using static CRC16 algorithm. It divides the P2P application traffic into packet sets, and distributes the original traffic by using Hash mapping functions, and redistributes the flow bundles dynamically by the historical traffic based load balancing algorithm. The experimental results show that the model has an outstanding performance on feasibility and stability, and achieves the goal of application-level load balancing.

Key words: application-level load balancing; load distribution technology; computer network

摘要:提出了一种应用层负载均衡模型,将网络流量按照应用类型进行分流。该模型将网络流量分为普通应用和 P2P 应用两部分,根据应用类型将 P2P 应用流量划分为应用层报文集。初始化时采用 Hash 映射算法分配流量,使用静态 CRC16 算法分流普通应用流量,设计了基于历史流量的负载均衡算法进行动态调整。仿真结果显示,该模型具有较好的可行性与稳定性,实现了应用层次的负载均衡。

关键词:应用层负载均衡;分流技术;计算机网络

DOI: 10.3778/j.issn.1002-8331.2009.14.025 文章编号: 1002-8331(2009)14-0084-03 文献标识码: A 中图分类号: TP393.08

1 引言

随着计算机网络的发展,入侵检测系统(IDS)、内容审计系统等分析系统的性能越来越不能满足链路带宽的要求。单个处理节点已无法独立处理如此大流量的报文负载,必须采用分流技术,将负载分配到多个处理节点并行处理。

传统的分流技术按照分流的层次可分为两类:报文层负载均衡和流层负载均衡。报文层负载均衡的最小分流单位是单个报文。它一般使用随机调度、轮询调度等算法,将输入接口的报文平均分配到各输出接口,实现报文负载的均匀分配。报文层负载均衡具有实现简单,效果好,记录状态少,易于动态调整各输出接口的流量等优点。然而在 IDS、内容审计等系统中,往往需要将同一连接的所有报文发送到同一个分析节点进行处理,以保证分析内容的完整性。因此分流系统需要实现流保持,进行流层负载均衡。流层负载均衡的最小分流单位是一条流或流束(Flow Bundle),首先将输入接口的流量分成若干流束,然后使用移位、XOR、CRC 等算法计算流束五元组信息的 Hash 值,根据 Hash 值将其分配到相应的输出接口,实现了负载均衡,并且保持了同流同宿。

负载均衡技术在网络应用中具有十分重要的作用,一般采用 Hash 函数分配网络流量。Zhiruo Cao 等^[1]总结了 5 种直接 Hash 算法和 2 种基于表的 Hash 算法,用实验对它们的性能进行了比较,从结果可以看出,不同分流算法的分流效果的差异是巨大的。由于网络流量的复杂性与多变性,单纯依靠静态 Hash 技术往往难以取得理想的分流效果。因此,需要向算法中加入动态调整策略。IBM 算法^[2]保存了每一流束的时间戳,在动态调整时,优先调整时间戳最小的流束。Weiguang Shi 等^[3]为了达到均衡时较高的资源利用率,将网络中的流量进行分类,采取动态调整具有侵略性流(Aggressive Flows)的调度算法。为了提高路由表 cache 命中率,A. Shaikh 等^[4]根据 IP 流的持续时间与链路状态更新频率等特点,将网络的流量分为长流和短流,并建议动态分配长流而静态分配短流,以减少表的频繁更新。在 Web 应用中,HRW 算法^[5]将网络请求分配给使用 Hash 算法计算出此请求最大权重的服务器去处理,提高了 Web cache 命中率并降低了服务器重新配置的最小中断率。基于网络流量长度的 Zipf 特性和突发传输特性,HABS 算法^[6]结合了 AHH 算法和 ABS 算法的优点,具有较小的流重映射率和报文乱序率。在

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.90604006)。

作者简介:胡安波(1982-),男,硕士,主要研究方向:计算机网络安全;苏金树(1962-),男,教授,博士生导师,CCF 高级会员,主要研究方向:计算机网络与网络安全;陈曙晖(1974-),男,博士,副研究员,主要研究方向:计算机网络安全。

收稿日期:2008-03-17 修回日期:2008-05-23

高速网络入侵检测系统的负载分配时,最小会话数优先(MSF)的自适应负载均衡算法^[7]通过动态调整会话数最小的流束,满足入侵检测系统需要维持会话完整性的要求。以上的研究成果,是在分析网络流量特征后,从报文层和流层对负载进行分流,在其具体的应用背景中,具有很好的性能。

近年来,分布式的文件共享,基于P2P的网络应用在网络流量中所占的比例不断增加,分布式的网络攻击也不断出现。为了保证网络安全,在园区网的入口处需部署并行入侵检测系统和内容审计系统。在分布式应用或攻击中,单纯分析一条或几条流无法获得应用或攻击的全部信息,只有尽可能多地将同一应用或攻击的全部流量发送给同一个处理节点进行分析,才能最大限度的获得准确的分析结果。因此,应用层分流是一个极具研究价值的课题。

将网络中的流量划分为普通应用和P2P应用两部分,采取静态分流普通应用流量,动态分流P2P应用流量的策略。在静态分流普通应用流量时采取源、目的IP地址的CRC16算法,并实验论证该算法具有很好的分流效果。为保证处理节点获取内容的完整性,将P2P流量分成若干应用层报文集,使用Hash技术将其映射为流束,设计了基于历史流量的动态调整算法,实现应用层的负载均衡。

2 应用层次的负载均衡

在基于P2P技术的分布式网络应用中,通信的基本形式是群组(group)。对分布式网络应用进行分析时,需要将参与此应用的全部主机的相关流量分配给同一节点分析处理。报文层和流层的分流技术,受其分流层次的局限性,无法满足此需求。应用层负载均衡是将相同网络应用的流量分配给同一处理节点的分流技术。它是作用于应用层的分流技术,与报文层和流层负载均衡相比具有更高层次的抽象。其分流的最小单位是包含来自相同应用的多条数据流的集合。应用层负载均衡的思想是将网络中的流量分为普通应用和P2P应用两部分,分别对这两类应用使用适当的分流算法,将同一应用的流量分配给一个节点处理。

为达到应用层分流的目的,提出了应用层负载均衡模型。该模型具有3个处理阶段:确定最小分流单位、流量的初始分配和流量的动态调整。

2.1 确定最小分流单位

对网络流量进行应用层负载均衡首先要确定最小分流单位。普通应用的参与者是两个主机,将这两个主机的通信流量分配给相同的节点处理,即可得到完整的交互信息。因此,普通应用的最小分流单位仍然是流。对于P2P应用,假设在某一时间段内,网络的通信图如图1所示,按照节点上运行的P2P应用的共享内容进行分类,得到图2,图2中网络中的节点分成了3个子图,每个子图的节点维护着相同的P2P共享内容。因此可以根据某一时间段内网络节点的P2P应用情况来将网络节点划分为若干个节点集 n_i 。

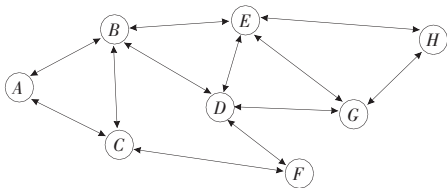


图1 P2P应用通信示例图

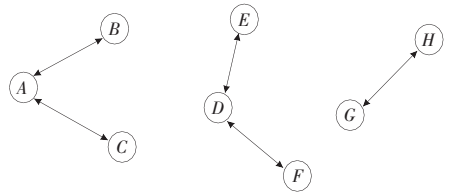


图2 P2P应用节点集

在节点集的划分过程中需要记录每个IP地址所对应的集合标识,随着节点集内节点的不断增加,当收到一个源IP地址和目的IP地址分别处在不同节点集中的数据报时,需要将两个节点集合并成一个节点集,从而需改变大量的IP地址对应的集合标识,导致算法的效率低下。在P2P应用中,同一目的地址的报文必然局限在几种共享内容中,将目的地址相同的报文分配给同一个处理节点进行分析,可以获得此目的地址的节点下载内容的全部信息。因此可以把具有相同目的地址的P2P应用报文看作同一类应用,将其划为同一个报文集 p_i 。

2.2 流量初始分配策略

对普通应用的流和P2P应用的报文集,分别采用不同的负载均衡算法。在经典分流算法中,基于网络通信五元组的16位CRC算法具有很好的负载均衡性能^[8]。而对于某些应用(例如ftp服务),在网络通信过程中,建立连接和传输数据使用不同的端口,基于网络通信五元组的16位CRC算法则把此过程看作两条流来处理,破坏了流的相关性。故模型采用基于源、目的IP地址的16位CRC算法对普通应用的流量进行分配。假设系统有 N 个处理节点,流量分配函数如式(1)所示。

$$\text{Hash}_{\text{nml}} = \text{CRC16}(\text{SrcIP}, \text{DstIP}) \bmod N \quad (1)$$

由于普通应用的数据流一般存在的时间较短,流量相对较小,故采用静态分流方式,避免了记录大量的状态信息,节省了内存空间,提高了算法的性能。

对于P2P应用的流量,采用基于表的动态Hash算法进行分配。假设将P2P应用的流量分为 M 个流束,首先,按照目的地址将P2P应用的流量划分为若干个报文集 p_i ,然后将这些报文集通过Hash函数(2)映射到 M 个流束,最后将 M 个流束使用Hash函数(3)映射到 N 个处理节点。

$$\text{Hash} = \text{DstIP}(p_i) \bmod M \quad (2)$$

$$\text{Hash} = M \bmod N \quad (3)$$

对普通应用和P2P应用分流使用的都是随机分布性较好的Hash函数,理论上保证了系统在初始化时,各处理节点流量的相对平衡性。由于网络流量长度的Zipf特征和突发传输等特性的存在,各处理节点接收的流量会出现不平衡状态,导致系统总体处理性能下降。因此需要对静态分配的流量进行动态调整。

2.3 流量动态调整策略

当各处理节点出现负载不平衡时,需要对分配到各节点的流量进行动态调整,从而保证各节点资源利用率保持平衡。在动态调整时,采用静态路由普通应用的流量,动态调整P2P应用流量的策略。通过改变P2P应用的流束与处理节点的映射关系来调整分配到各处理节点的流量。这样既保证各处理节点的负载均衡,又实现了较低的调整开销。

为了实现上述目的,设计了一个基于历史流量的动态负载均衡算法(Historical Traffic based load Balancing algorithm, HTB)。该算法每隔预设时间间隔检测各输出节点的流量信息,

若发现负载分配不均,则根据各流束的历史流量信息,对相关输出节点所包含的流束进行重映射,使负载均衡分配。算法需要记录在时间段 τ 内各输出节点的流量 f_i ,各流束的流量 b_j 和流束与处理节点的映射关系表。设 φ_{ub} 为系统的不平衡因子:

$$\varphi_{ub} = \frac{\max(f_i) - \min(f_i)}{\text{mean}(f_i)} \quad (4)$$

其中, $\max(f_i)$, $\min(f_i)$ 和 $\text{mean}(f_i)$ 分别为各输出节点流量的最大值、最小值和均值。设 φ_{thrd} 为系统不平衡因子的最大门限值,计时器的时间为 t , λ 为调整因子。算法的流程如下所示:

步骤 1 若 $t < \tau$, 按照流束与输出节点的映射关系表分配 P2P 应用流量,并记录各流束与输出节点的流量。若 $t \geq \tau$, 则跳转步骤 2。

步骤 2 计算系统的不平衡因子 φ_{ub} , 若 $\varphi_{ub} < \varphi_{thrd}$, 跳转步骤 5。若 $\varphi_{ub} \geq \varphi_{thrd}$, 跳转步骤 3。

步骤 3 查找流量最大和最小的输出节点, 计算需要调整的流量 $\Delta f = \lambda (\max(f_i) - \min(f_i))$ ($0 < \lambda < 1$)。 (5)

步骤 4 从流量最大的输出节点包含的流束中选取 n 个流束,使得所选流束的总流量等于或略微小于 Δf ,即 $\sum_{j=0}^n (b_{maxj}) \leq \Delta f$ 。将选取的流束重映射给流量最小的输出节点。跳转步骤 2。

步骤 5 将各输出节点和流束的历史流量信息清空, 设置 $t=0$, 跳转步骤 1。

算法同时调整了流量最大和最小的两个输出节点的流束映射关系,具有较好的快速收敛特性。使用了基于流束历史流量信息的选取策略,算法对调整效果具有预见性,减少了动态调整的触发次数,使系统更加稳定。

3 实验模拟及结果分析

3.1 算法验证与复杂性分析

为验证应用层负载均衡模型的分流效果,设计了模拟实验对算法进行仿真。实验所用的数据为园区网入口处使用 Tcpdump 捕获的网络流量,数据的详细信息如表 1 所示。

表 1 实验数据信息表

捕获时间	报文数	P2P 应用报文比例(%)	P2P 应用报文集数
2007.12.10 10:28:34	4 122 823	57.67	38 425

实验测试了 4 种静态分类函数(基于五元组的 CRC16 函数,基于源目的 IP 的 CRC16 函数,基于协议号、源目的 IP 的 CRC16 函数,基于源目的 IP 分段异或函数)的负载均衡效果,其结果如图 3 所示。

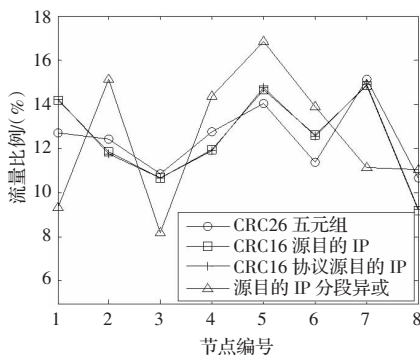


图 3 静态分流函数的负载均衡效果

基于五元组的 CRC16 函数的稳定性最好,但由于其无法保持流的相关性,故使用分流效果仅次于源目的 IP 的 CRC16 函数来进行静态流量分配。

普通应用和 P2P 应用具有不同的流量特征,为提高算法的效率,降低流量的重映射率,采取静态分流其中一种流量,动态调整另一种流量的方式。图 4 显示了使用基于源目的 IP 的 CRC16 函数分别对两种流量进行静态分流的效果,普通应用流量的稳定性更好,由此验证了模型采取的静态分流普通应用流量,动态调整 P2P 流量策略的正确性。

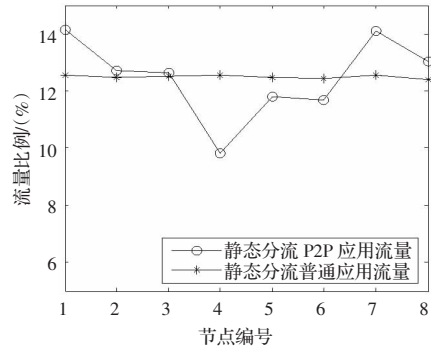


图 4 静态分流普通应用流量与 P2P 应用流量效果比较图

HTB 负载均衡算法与经典的 IBM 算法相比,时间复杂度都是 $O(n)$ 。IBM 算法对每一个报文都需要判断其所在的流是否超时,故其需要 2 次访问存储器,HTB 算法只需 1 次访问存储器,HTB 算法具有更高的运行效率。IBM 算法需保存每一流束的 4 项信息,HTB 算法只需要保存 2 项信息。IBM 算法是实时动态调整算法,HTB 算法是定时的动态调整算法,在队列管理方面,IBM 算法的平均等待队列长度比 HTB 算法的短,但由于处理节点上的内存一般为 GB 级,HTB 算法平均等待队列偏大不会成为模型的瓶颈。

3.2 模型参数的设置

模型的参数主要有以下几个:流束数量 M ,处理节点数量 N ,不平衡因子的门限值 φ_{thrd} ,调整间隔 τ 以及调整因子 λ 。 M 的大小决定动态调整的精度, M 越大,动态调整的精度越高。 N 和 φ_{thrd} 由具体应用方案确定, τ 和 λ 为经验值,由多次实验模拟得出最佳值。模型的各项参数设置如表 2 所示。

表 2 模型参数设置表

参数	取值	参数	取值
M	256	$\varphi_{thrd} / (\%)$	1
N	8	λ	2/3
τ / s	6		

3.3 结果分析

由仿真结果可以看出,用于普通应用分流的基于源、目的 IP 地址的 CRC16 算法本身具有较好的静态分流性能(图 5);用于 P2P 应用分流的基于历史流量的动态负载均衡算法也具有较好的分流效果(图 5)。由处理节点实时分配的流量图(图 6)可以看出,算法具有较强的稳定性与鲁棒性。

4 结束语

本文对应用层负载均衡技术进行了初步的探索,将网络流量分为普通应用和 P2P 应用两部分。确定了 P2P 应用分流的