

一种面向会话的自适应负载均衡算法*

陈一骄⁺, 卢锡城, 时向泉, 孙志刚

(国防科学技术大学 计算机学院, 湖南 长沙 410073)

A Session-Oriented Adaptive Load Balancing Algorithm

CHEN Yi-Jiao⁺, LU Xi-Cheng, SHI Xiang-Quan, SUN Zhi-Gang

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

+ Corresponding author: E-mail: chenyj@nudt.edu.cn

Chen YJ, Lu XC, Shi XQ, Sun ZG. A session-oriented adaptive load balancing algorithm. Journal of Software, 2008,19(7):1828-1836. <http://www.jos.org.cn/1000-9825/19/1828.htm>

Abstract: Based on the related work on load balancing, a load balancing algorithm named adaptive load balancing algorithm based on Minimum Sessions First is proposed. Simulation results show that the scheme achieves significant improvement in session's integrality disruption and has a fairly good load balance both in packets level and bits level. It is a simple algorithm and also can be easily implemented in hardware. The algorithm had been implemented in the high speed network security device designed by National University of Defense Technology.

Key words: sessions; load balancing; adaptive scheduling algorithm; sessions integrality

摘要: 在论述负载均衡技术相关工作的基础上,基于 IP 报文头多域分类方法,提出自适应负载均衡算法 MSF(minimum sessions first),通过动态调整 TCP 流数目最少的流束,能够在各处理节点间保持动态负载均衡的同时维持会话的完整性。模拟结果表明,MSF 算法具有设计简洁、负载均匀度好、重映射破坏度小、会话完整性破坏度小等优点,对不同负载具有良好的综合性能。该算法已经成功地应用在国防科学技术大学计算机学院研制的高速网络安全设备中,在保持较好的负载均衡效果的前提下保证了会话的完整性,提高了网络安全设备的性能。

关键词: 会话;负载均衡;自适应调度算法;会话完整性

中图法分类号: TP301 **文献标识码:** A

随着网络规模的急剧扩大和数据传输技术的飞速发展,网络速率剧增,导致高速链路的待处理数据与系统处理能力之间的矛盾越来越突出,利用并行处理方法缓解系统处理能力的不足已成为必然。并行处理方法需要在各处理节点之间进行负载分配以实现最佳性能。在负载分配过程中,采用动态调整方式自适应地保持负载均衡具有更高的性能,但这样可能会破坏原始流特性。如果属于同一会话的报文被分配到不同的处理节点,这对路由查表的性能可能没有影响,但在网络安全应用(如 NIDS(network-based intrusion detection system),IPS (intrusion prevention system)等)中,这可能是致命的:目前许多网络攻击行为隐藏了攻击特征,基于独立报文检测技术不能检测出其攻击行为,只有捕获会话的所有报文,通过报文拼接、重组以后,才可以检测出其攻击行为。如

* Supported by the National Natural Science Foundation of China under Grant Nos.90412011, 90604006 (国家自然科学基金); the National Basic Research Program of China under Grant No.2005CB321801 (国家重点基础研究发展计划(973))

Received 2007-02-26; Accepted 2007-08-03

果负载均衡系统不能将整个会话分配到一个处理机节点,则会导致入侵检测系统不能检测出这次攻击,形成漏报.因此研究面向会话(sessions)的负载均衡对网络安全应用具有重要的意义.

根据网络数据传输的特点,定义具有相同源/目的 IP 地址、源/目的端口和协议的报文属于同一个半流,由任意一个半流及其反射半流(交换 IP 地址以及端口后形成的半流)确定一个会话.为简单起见,以后我们使用会话五元组(c_ip,s_ip,c_port,s_port,protocol)来描述一个会话.其中“c”代表“客户”方;“s”代表“服务器”方.

本文在介绍负载均衡相关工作的基础上,基于 IP 报文头多域分类方法,提出动态负载均衡算法 MSF (minimum sessions first),通过动态调整 TCP 流数目最少的流束(flow bundle),达到在维持网络流量负载的动态均衡的同时保持会话完整性的目的,从而使后端系统能够正确理解报文的语义.最后对提出的 MSF 算法进行模拟,以验证算法的性能.

1 相关工作

负载均衡主要讨论在各处理节点之间进行任务分配以及调度问题.从处理粒度来看,负载均衡主要分为报文级调度和流级调度两种.报文级调度一般使用随机调度算法、轮询调度算法等,可以实现报文级负载的均匀分配,但不能保证报文的顺序,可能破坏流的相关性,导致属于相同流的报文分配到不同处理机节点;基于流级的报文分配算法一般采用 Hash 技术,使用移位、XOR、CRC(cyclic redundancy check)算法等计算 Hash 值,它们可以保证报文顺序以及流的相关性,但不能均匀分配流量.从调度方式来看,负载均衡主要分为静态调度和自适应调度两种方式.静态报文调度没有考虑网络流量的局部性和突发性等特点,采用固定的报文分配方法,在微观上会导致负载分配不均衡.自适应负载分配算法可以根据各处理节点的负载自适应地决定负载的分配,因此有更好的性能.

Kenc1等人提出了一个路由器内部各转发引擎间的自适应负载均衡模型^[1].该模型通过防止转发引擎过载的反馈控制机制来实现多个转发引擎间的负载均衡,其报文分配算法通过扩展HRW(highest random weight)算法^[2],采用动态修改转发引擎的权重向量的方式来达到自适应负载均衡的目的.

Shi等人基于对Internet流量的分析,阐明了仅依靠Hash技术不能获得高效的负载平衡结果^[3],提出将网络数据流量划分为正常流(normal flows)和具有侵略性的流(aggressive flows),采取动态调整Aggressive流的调度技术,达到在所有转发性能下的自适应负载均衡效果,以获得高效的资源利用率.

在对负载敏感的动态路由转发系统中,为了提高路由表Cache的稳定性并在各负载均衡链路之间平衡流量,Shaikh等人^[4]根据IP流的持续时间与链路状态更新频率等特点,建议动态路由长流而静态路由短流.通过动态调整对负载敏感的长流,可以减少链路状态信息的频繁更新,从而提高路由表的Cache命中率.

Dittmann等人^[5]描述了一个并行转发系统的负载平衡器(根据作者研究单位后面简称IBM),它采用基于表的Hash技术来划分流量,通过查表得到目标转发引擎(forward engine,简称FE).算法为每一个具有相同索引的流束保留其最近到达报文的时间戳,当触发动态调整时,将时间戳最小的流束分配给当前负载最轻的处理器以实现动态负载均衡.

Shi等人^[6]描述了一个多处理器路由器中的自适应负载平衡方法(根据作者姓名后面简称SHI),它采用Hash技术来划分流量,同时考虑输入负载流量特性,选取其中 m 个流量最大的流束,当触发动态调整时,将这 m 个流束分配给当前负载最轻的处理器实现自适应负载均衡.

上述研究工作主要采用并行处理方法缓解高速报文流与系统处理能力之间的矛盾.采用的负载均衡报文调度器一般由 Hash 分类器、流分类器、自适应控制器和输出选择器 4 部分组成.Hash 分类器在负载均衡时选择报文的输出端口,流分类器选择重映射流.自适应控制器在检测到负载不均衡时选择重映射流的输出端口,并通过输出选择器暂时覆盖 Hash 分类器的输出选择结果,从而保持动态负载均衡.这种处理机制的优点是可以动态检测负载的变化,实现自适应负载均衡,但也存在如下缺点.其一,自适应调整时没有考虑对话务完整性的影响;其二,在负载状态转换过程中,由于输出控制器的影响,极有可能会引起调整的振荡,从而严重影响会话完整性的保持.因此,这类算法不能直接应用于像 NIDS 之类的系统中.

文献[7-10]研究了高速网络环境下 NIDS 的负载均衡问题,所提出的方法都是采用多个 NIDS 并行处理来解决高速链路环境下单个 NIDS 的处理性能瓶颈问题,这些研究工作也没有考虑负载动态调整时对会话完整性的影响.

本文主要研究高速网络环境下可以保持会话完整性的负载均衡算法,在高速网络 NIDS、网络流量管理、网络安全监控等应用中具有重要意义.

2 面向会话的负载均衡问题与评价准则

基于网络内容的处理系统(如并行入侵检测系统和安全监控系统等)需要保持网络会话的完整性,否则,系统不能完整解析会话的语义,从而影响系统处理的正确性以及性能.因此,在对此类系统进行负载均衡时必须完整保持报文的相关性以及会话的完整性,这是基于网络内容的处理系统进行负载均衡的基础.

2.1 面向会话的负载均衡模型

面向会话的负载均衡系统基本体系结构模型如图 1 所示,其中 *Processor* 是基本的处理单元.在由 m 个 *Processor* 组成的具有负载均衡能力的并行处理系统 $\{P_1, P_2, \dots, P_m\}$ 中,当 $P_i (1 \leq i \leq m)$ 空闲时,被分配的报文可以立即处理,否则报文缓冲在大小为 B 的共享缓冲区中.

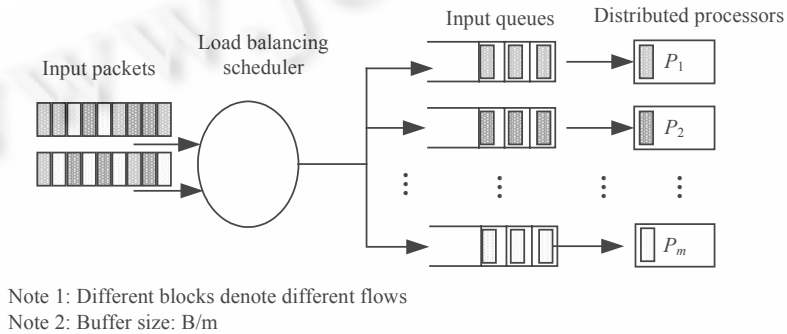


Fig.1 Session-Oriented load balancing module

图 1 面向会话的负载均衡系统基本模型

面向会话的负载均衡系统的重要部件是负责报文分配的负载分配调度器,其性能的好坏直接影响到系统性能.负载分配调度器必须能够均衡地将流量分配给各处理节点,同时确保会话的完整性,以实现并行处理系统的最佳性能.通常情况下,负载分配调度器采用基于 Hash 查表的自适应报文负载分配方法.为了维持报文流的原始流量特性,负载分配调度器需要将每一个会话尽最大可能地分配到同一个处理节点,保持会话的完整性.Hash 映射技术实际上是建立在报文流标识集 S 和具有 n 个表项的流表 FT 之间的 Hash 函数,即:

$$H(\cdot): S \rightarrow FT, FT = \{1, 2, 3, \dots, n\} \quad (1)$$

流标识定义为报文头的一个或多个域的向量,可以是源 IP 地址(SIP)、目的 IP 地址(DIP)、源端口号(SP)、目的端口号(DP)、协议类型(PT)和 TCP 头标识(THF)等报文固定位置参数中的一个或者多个的组合.

2.2 面向会话完整性的负载均衡评价指标体系

由于网络安全应用基于会话处理报文的特点,在并行 NIDS 内部需要将属于同一个会话的报文分配到同一个处理节点,即保持会话的完整性.在面向会话的负载均衡自适应调度中,如文献[3,11]等提出的性能评价标准由于不能准确衡量会话的完整性程度,已经不能正确评价其自适应调度的真实性能.为了评价得客观、真实,我们对文献[3,11]等的负载均衡评价指标进行了修改和扩充,完善了面向会话的负载均衡的评价标准.它主要由报文负载均衡度、位流负载均衡度、流重映射破坏度和会话完整性破坏度等评价标准构成.在评价体系中,我们将负载均衡度分解为位流级和报文级负载均衡两个标准,以分别衡量调度时的位流级和报文级负载均衡效

果;同时新提出了会话完整性破坏度,以衡量负载分配时对会话完整性的保持特性.

事件 1. 流束重映射:当一个流束在流表中从一个处理节点动态地调整到另一个处理节点时,我们称这个流束发生了重映射.

事件 2. 会话破坏:当一个流束在流表中从一个处理节点动态地调整到另一个处理节点时,我们称这个流束中的会话被破坏.

定义 1. 负载均衡度(load balancing metric,简称 LBM).

设 $p_i(t)$ 表示网络链路中传输的报文序列 $p(t)$ 中被处理节点 $P_i(i=1,2,\dots,m)$ 处理的报文序列, $|p_i(t)|$ 表示 $p_i(t)$ 的报文总数, $|Len_i(t)|$ 表示 $p_i(t)$ 中所有报文的字节总数,且 $p(t)=\bigcup_{i=1}^m p_i(t)$, $Len(t)=\bigcup_{i=1}^m Len_i(t)$, 则有 $|Len_i(t)|$ 的自协方差:

$$CV[|Len_i(t)|] = \frac{D(|Len_i(t)|)}{(E(|Len_i(t)|))^2} = \frac{E(|Len_i(t)|^2) - (E(|Len_i(t)|))^2}{(E(|Len_i(t)|))^2} \quad (2)$$

将式(2)归一化到 $[0,1]$ 区间,得到位流负载均衡度 $bLBM(t)$:

$$bLBM(t) = \frac{CV[|Len_i(t)|]}{CV[|Len_i(t)|]+1} = 1 - \frac{(E(|Len_i(t)|))^2}{E(|Len_i(t)|^2)} \quad (3)$$

由于 $|Len_i(t)|$ 和 $|Len_i(t)|^2$ 的数学期望可分别用它们在 $(t-\Delta t, t)$ 时间内的观察值表示: $E(|Len_i(t)|) = \frac{\sum_{i=1}^m |Len_i(t)|}{m}$, $E(|Len_i(t)|^2) = \frac{\sum_{i=1}^m |Len_i(t)|^2}{m}$, 因此,式(3)可用如下方法计算:

$$bLBM(t) = 1 - \frac{\left(\sum_{i=1}^m |Len_i(t)|\right)^2}{m \sum_{i=1}^m |Len_i(t)|^2} \quad (4)$$

由柯西不等式定理可知, $0 \leq bLBM(t) \leq 1$, 负载越均衡时 $bLBM(t) \rightarrow 0$.

$bLBM(t)$ 以字节流为单位统计了负载分配的均衡性,体现了负载分配的微观均衡度,在基于报文内容的处理系统(如 NIDS,IPS 系统等)中,好的微观均衡度可以获得很好的处理性能.对于式(4),如果使用 $p_i(t)$ 代替 $Len_i(t)$ 作为负载均衡的衡量标准,那么就是文献[11]中定义的报文负载均衡度:

$$pLBM(t) = 1 - \frac{\left(\sum_{i=1}^m |p_i(t)|\right)^2}{m \sum_{i=1}^m |p_i(t)|^2} \quad (5)$$

其实,式(5)是式(4)的特例,如果认为序列中每一个报文的长度都是 1,那么式(4)和式(5)的计算结果是一样的.报文负载均衡度 $pLBM(t)$ 体现了负载分配的宏观均衡度,在基于报文的处理系统(如并行路由转发引擎系统)中,好的宏观均衡度可以获得很好的处理性能.

定义 2. 重映射破坏度(remapping disruption,简称 RD).

设 $F_s(t)$ 表示报文序列 $p(t)$ 在调度过程中发生流束重映射的次数.流束重映射破坏度表示为

$$RD(t) = F_s(t) / p(t) \quad (6)$$

$RD(t)$ 用于评价负载分配中流束的重映射行为,直接反映了流束迁移的频度,一个好的负载均衡算法其重映射破坏度应该尽可能地小,这样才能减小流束重映射开销,降低会话完整性被破坏的风险.

定义 3. 会话完整性破坏度(session integrity disruption,简称 SID).

设 $S_s(t)$ 表示报文序列 $p(t)$ 在重映射过程中发生的会话破坏次数.会话完整性破坏度表示为

$$SID(t) = S_s(t) / p(t) \quad (7)$$

$SID(t)$ 用于评价负载分配中会话的完整性保持行为,它直接反映了负载分配的会话完整性程度.如果属于同一会话的多个报文被重映射到不同的处理节点上,则该会话的完整性被破坏,从而可能使得系统的后端处理节点不能正确分析会话的语义,影响系统性能甚至系统处理的正确性.

面向会话的动态负载均衡除了满足一般低开销、报文负载均衡度和流保序等基本要求以外,还应该满足

位流负载均衡度、重映射破坏度和会话完整性破坏度都尽可能小的要求。

3 基于 MSF 的自适应负载分配算法

在面向会话的负载均衡系统中,我们提出采用最小会话数优先的自适应负载均衡算法来实现各处理节点间的负载均衡.如图 2 所示,算法由流分类、Hash 分类、自适应控制和流表这 4 个模块组成,构成由基于流表的静态流量分配部件和基于最小会话数优先的自适应负载调整部件,完成负载的动态均衡.MSF 算法采用自适应负载调整部件动态修改流表的相关表项,由流表确定输出端口的自适应调度策略,确保一次触发不会产生多次流束迁移,避免由于动态调整引起的振荡。

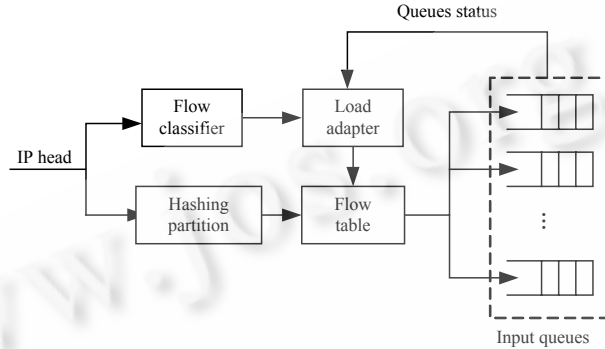


Fig.2 Architecture of adaptive load balancing scheduler

图 2 自适应负载分配调度器框图

3.1 基于流表的静态负载分配方法

基于报文的,如 Round-Robin 等负载调度算法具有开销低、效率高等特点,但是无法保持会话的完整性要求.基于 Hash 技术的负载均衡是近年来研究的重点,也在应用中获得很好的应用.Cao,Wang 等人在文献[12]中对负载均衡的目标、衡量标准、性能评价等方面进行了深入分析,得出了 CRC 算法具有良好的负载均衡性能,其次异或算法以及 IP 头校验和算法也有很好的性能.程光等人在文献[13]中从位流熵的角度证明了异或-移位算法具有较高的均匀度,并使用真实网络数据进行了验证。

本文采用基于流表的负载分配方法来实现负载的静态分配.如图 2 所示,系统采用 Hash 函数 $f(x)$ 将输入流量划分成 n 等份,然后均匀地将这 n 等份分派到 m 个处理节点,同时将分配结果记录到负载分配流表中.对于每一个输入报文,首先通过 Hash 函数 $f(x)$ 计算得到流表索引,然后通过流表得到报文的最终输出处理节点。

对于 Hash 函数 $f(x)$ 的选取,文献[12,13]中有较多的论述,但是由于 CRC 算法的计算结果与关键字的顺序相关,从而可能导致一个会话的两个半流的报文被分配到不同的处理节点,因此需要进行预处理才能适合面向会话的负载均衡应用.由于网络链路中的流量绝大部分都是 TCP 报文,因此本文以 TCP 报文四元组{源 IP 地址、目的 IP 地址、源端口、目的端口}作为关键字,通过 Hash 函数 $f(x)$ 计算以后得到一个 16 位的计算结果,以此结果为基础创建流表,实现基于流表的静态负载分配。

3.2 MSF 自适应负载分配算法

在理想情况下,基于流表的静态分配方法可以获得较好的报文级负载均衡.由于网络实际流量的突发性等特点,可能会引起处理节点间的负载不均衡,从而导致某个流束负载过大,超过后端处理节点的处理能力而致使报文丢失.此时需要采用自适应方法动态调整流束的映射关系,保持负载的均衡.自适应负载分配算法主要考虑触发策略、调度对象的选定和目的端口的选择等几个方面。

3.2.1 触发策略

在自适应动态负载均衡方法中,有多种触发方式.例如周期性触发策略方法,特点是容易实现,也不需要收

集任何系统信息;缺点是效率低,且可能引起误操作,即当前负载处于均衡状态却启动了负载均衡调度.另一种方法是监测输入队列的长度,利用它作为后端处理节点的负载指示.如果某一参数(例如空闲的输入共享缓冲区容量、最大队列长度,或者采用低通滤波以后的队列长度等)超过预设的阈值,就可以触发动态调整.

MSF 算法的触发策略是动态监测各处理节点的输入队列长度,当某一个队列的长度超过预设的阈值时,启动负载动态调整器.

3.2.2 调度对象的选定

当负载不均衡的时候,需要选择合适的流束进行重映射.选择调度对象时应该把握一个原则:应该尽可能地保持数据流的原始流量特性,对流量特性的破坏尽可能地小.对于面向会话的负载均衡策略,最重要的就是在保持会话的完整性的前提下保持负载的动态均衡.

由于无论从报文的角度还是从位流的角度来说,TCP 报文都占据了网络流量的绝大部分,因此,MSF 算法只动态调整 TCP 报文流,其他报文流不作为 MSF 算法的调度候选对象.

对于到达的每一个报文,动态调整器检查报文类型,如果是 TCP 报文,则统计该流束的会话个数.如果该流束的会话个数低于某一个阈值,则可以将该流束分配给当前负载最轻的处理节点.算法的伪代码如下所示:

Algorithm 1. Select_Scheduled_Unit.

When adaptive schedule required {

 Input: *flow_bundle*;

 Output: *SelectScheduledUnit*

 If (*flow_bundle* is new) or (*Sessions_Table*[*flow_bundle*] < *scheduled_value*)

 //Select the scheduled flow bundle which session's number is the Minimum;

SelectAdjustUnit ← True; //current flow bundle can be selected as a candidate for adaptive scheduling.

 else

SelectAdjustUnit ← False //current flow bundle can not be selected as a candidate for adaptive scheduling.

}

3.2.3 队列长度计算

负载分配系统主要通过输入队列的状态来衡量后端处理节点的负载情况.MSF 算法使用具有简单低通滤波功能的输入队列长度计算方法来表示输入队列长度.负载分配系统触发队列长度计算以后,计算平均队列长度,算法如下所示:

$$Queue_Len = Current_Queue_Len \times (1 - w) + Real_Queue_Len \times w \quad (8)$$

其中:

Queue_Len: 报文到达后的平均队列长度;

Current_Queue_Len: 报文到达前的平均队列长度;

Real_Queue_Len: 当前队列的实际长度;

w: 队列权值,它指定当前队列实际长度对平均队列长度的贡献.

为了避免流量突发造成的抖动影响性能,平均队列长度的计算简单过滤了突发流量.选取合适的 *w* 对平均队列长度的影响很大,如果 *w* 过大,那么流量的突体会使转发平面的负载急剧变化,造成抖动.反之,如果 *w* 过小,那么流量变化对平均队列长度的影响很小,无法反映当前的负载状况.

3.2.4 MSF 算法

MSF 自适应负载分配方法在负载不平衡时采用自适应负载分配方法动态分配占流量绝大部分的 TCP 流量.算法的主要思想是当负载不均衡时,启动负载调整器,根据调度策略选出需要调整的流束,修改流表,将该流束重新映射到负载最轻的处理节点.这样,该流束的每一个后续报文,都将转发到新的目的处理节点.MSF 算法的伪代码如下:

Algorithm 2. MSF based load balancing scheduled algorithm.

Input:

packet 5 tuple (*sip,dip,sport,dport,protocol*);packet length *p_length*;

output:

target processor ID;

Init:

```
for  $i=0,i<65535,i++$  {
  FTable [ $i$ ]= $i \% m$ ; /*init the static flow table*/
}
```

Processing:

```
While (When packet arrived) {
  index←Hash(Sip,Dip,Sport,Dport); /*computing the hash value*/
  Processor_ID←FTable[index]; /*select the output processor ID*/
  If (protocol==6){
    adjust_req=Buffer_Status(Processor_ID);
    /*check the status of current processor's input packet buffer, if it's necessary schedule the packet to a
    new processor.*/
    if (adjust_req==True) {
      If (Select_Scheduled_Unit(index)) /*current flow bundle can be scheduled*/{
        Processor_ID←MIN_Load_Prossor();
        /*assigned the current packet to the processor which it's payload is the minimum, active schedule
        the load dispatch*/
        FTable[index]=Processor_ID; /*update the static flow table*/
      }
    }
  }
}
```

4 模拟

为了验证MSF负载均衡调度算法的性能,我们使用真实的网络链路流量Traces,通过软件对算法各方面的性能进行了模拟,并进行算法间的参数比较.在模拟过程中,我们选择HRW^[1],SHI^[6]和IBM^[5]三种算法,与MSF调度算法进行比较.模拟中,数据源选择IPLS 10G traces以及中南某省级电信中心的10G链路的实际流量,后端处理机数目 $m=8$,输入缓冲区队列长度 $Q=64\text{KB}$.如果这些算法中有涉及到用于存储流分配信息的映射表,则其表项规模都设置为4K项.表1给出了不同算法中的参数配置情况.模拟结果如图3所示.

图3(a)和图3(b)显示,SHI算法的负载均衡度最差,HRW和IBM算法在不同Traces下的负载均衡度差异较大,而MSF针对各种Traces的负载均衡度比较平稳.说明MSF对不同负载具有良好的负载调节能力.

图3(c)显示SHI算法在4种Traces下都具有较高的流重映射比例,HRW的流重映射比例最低,MSF与IBM算法的流重映射破坏度相当,但MSF的重映射破坏度在所有Traces上的平均值较低.

图3(d)显示的结果与图3(c)的结果相对应,SHI算法和IBM算法在4种Traces下都具有较高的会话完整性破坏度,MSF算法的会话完整性破坏度最好,且在所有Traces上的值比较平均.

总的说来,MSF和IBM算法在各个性能指标上都具有比较均衡的性能优势,但是,IBM的会话完整性破坏度比较差,这主要与它的只基于目的IP地址的Hash分类方法、调度方法以及自适应调整方法等相关.MSF算法由于采用了基于IP四元组的Hash方法、使用基于流表的动态自适应负载调整策略,可以确保一次触发动态调整不会产生如文献[3,6]那样的两次迁移的后果,避免了调整的振荡,因此具有更好的会话完整性.MSF算法具有负载均衡度好、算法简单、易于硬件实现、重映射会话破坏度小等优点,对不同负载具有更好的综合性能,

适合于面向会话的高速链路流量负载均衡应用.

Table 1 Implementing strategy and parameters setting in analyzed adaptive load balancing

表 1 4 种算法的实现策略和内部参数设置

Algorithm	Parameters setting
HRW	Period: $\Delta t=1ms$, the triggering valve: 0.8, $r=3$, $\epsilon_h=0.01$
IBM	Timeout: 200us; the valve of counter: 50000 bytes; the full valve:0.8; the attenuation rate of counter: 0.5
SHI	High-Speed flow checking window: 1000 packets; the checked flows: 5; the triggering IQ valve of remap: 0.8
MSF	The minimum sessions checking window: 1000 packets, the triggering IQ valve of remap: 0.8, the rate of attenuation: 0.5

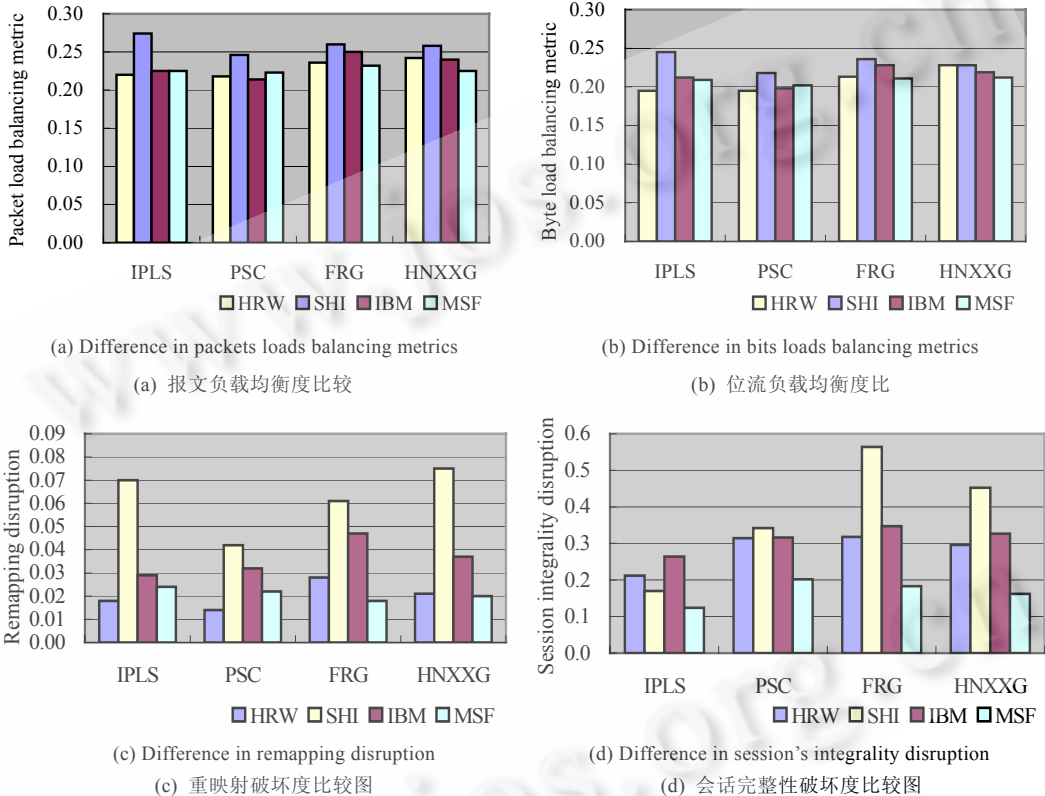


Fig.3 Performance analysis in different scheduling algorithms

图 3 不同调度算法性能比较

5 结 论

基于 IP 报文头多域分类方法,本文采用静态预分配和动态调整相结合的自适应调度思想,通过动态调整会话数最小的流,获得会话级的负载均衡,能够满足需要维持会话完整性应用的要求.通过软件模拟,验证了基于 MSF 的自适应负载均衡算法具有设计简洁、负载均衡度好、会话完整性破坏度小等特点,适合于面向会话的高速链路环境下分布处理时的负载均衡要求.但是,对于自适应调度算法的稳定性问题,我们还没有从理论上很好地证明,同时算法也可能将 DDOS 攻击等流量分配到多台后端处理节点,从而影响并行 NIDS 对这类攻击的检测效率,这都将是今后研究的重点.

References:

[1] Kencl L, Boudec JL. Adaptive load sharing for network processors. In: Proc. of the IEEE INFOCOM 2002. New York: IEEE Computer Society Press, 2002. 545-554. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4460887

- [2] David G, Ravishankar CV. Using name-based mappings to increase hit rates. *IEEE/ACM Trans. on Networking*, 1998,6(1):1-14.
- [3] Shi WG, MacGregor MH, Gburzynski P. Load balancing for parallel forwarding. *IEEE/ACM Trans. on Networking*, 2005,13(4): 790-801.
- [4] Rexford SJ, Shin KG. Load-Sensitive routing of long-lived IP flows. *ACM SIGCOMM Computer Communication Review*, 1999,29(4):215-226.
- [5] Dittmann G, Herkersdorf A. Network processor load balancing for high-speed links. In: *Proc. of the 2002 Int'l Symp. on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2002)*. San Diego, 2002. 727-735. http://www.research.ibm.com/people/g/gdittma/publications/Loadbalancer_Dittmann.pdf
- [6] Shi W, MacGregor MH, *et al.* Adaptive load balancer for multiprocessor routers. University of Alberta, T6G 2E8, 2004. <http://www.cs.ualberta.ca/~pawel/PAPERS/>
- [7] Yu R, Sun Z, Chen J, Mei SL, Dai YQ. Traffic distributor for high-speed network intrusion detection system. *Journal of Tsinghua University (Sci. & Tech.)*, 2005,145(10):1377-1380 (in Chinese with English abstract).
- [8] Lu ZJ, Zheng J, Huang H. A distributed real time intrusion detection system for high speed network. *Journal of Computer Research and Development*, 2004,41(4):667-673 (in Chinese with English abstract).
- [9] Kruegel C, Valeur F, Vignaetaal G, Kemmerer R. Stateful intrusion detection for high-speed networks. In: *Proc. of the IEEE Symp. Security and Privacy*. Los Alamitos: IEEE Computer Society Press, 2002. <http://portal.acm.org/citation.cfm?id=830527>
- [10] Chen XX, Fang BX, Li L. Architecture of intrusion detection for high-speed networks. *Journal of Computer Research and Development*, 2004,41(9):1481-1487 (in Chinese with English abstract).
- [11] Zhang XM. Research on key technologies of network processor design and implementation [Ph.D. Thesis]. Changsha: National University of Defense Technology, 2006 (in Chinese with English abstract).
- [12] Cao Z, Wang Z, Zegura E. Performance of hashing-based schemes for internet load balancing. In: *Proc. of the IEEE Infocom 2000*, Vol.1. Atlanta: IEEE Press, 2000. 332-341. <http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/6725/17985/00832203.pdf?arnumber=832203>
- [13] Cheng G, Gong J, Ding W, Xu JL. A hash algorithm for IP flow measurement. *Journal of Software*, 2005,16(5):652-658 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/16/652.htm>

附中文参考文献:

- [7] 余荣,孙智,陈嘉,梅顺良,戴一奇.高速网络入侵检测系统流量分配器.清华大学学报(自然科学版),2005,45(10):1377-1380.
- [8] 吕志军,郑璟,黄皓.高速网络下的分布式实时入侵检测系统.计算机研究与发展,2004,41(4):667-673.
- [10] 陈训逊,方滨兴,李蕾.高速网络环境下入侵检测系统结构研究.计算机研究与发展,2004,41(9):1481-1487.
- [11] 张晓明.网络处理器设计的关键技术研究[博士学位论文].长沙:国防科学技术大学,2006.
- [13] 程光,龚俊,丁伟,徐加羚.面向 IP 流测量的哈希算法研究.软件学报,2005,16(5):652-658. <http://www.jos.org.cn/1000-9825/16/652.htm>



陈一骄(1972-),男,湖南益阳人,博士生,副研究员,主要研究领域为高性能路由器,网络安全.



时向泉(1972-),男,博士生,副研究员,主要研究领域为高性能路由器,网络安全.



卢锡城(1946-),男,教授,博士生导师,中国工程院院士,CCF 高级会员,主要研究领域为高性能计算机,虚拟计算环境,高性能路由器,网络通信,信息安全.



孙志刚(1974-),男,博士,副研究员,CCF 高级会员,主要研究领域为高性能路由器,高性能计算机,网络安全,信息安全.