软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2011,22(9):2182−2192 [doi: 10.3724/SP.J.1001.2011.03920]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

抗 DDoS 攻击的主动队列管理算法*

张长旺+, 殷建平, 蔡志平, 刘新旺, 林加润, 朱 明

(国防科学技术大学 计算机学院,湖南 长沙 410073)

Active Queue Management Algorithm to Counter DDoS Attacks

ZHANG Chang-Wang+, YIN Jian-Ping, CAI Zhi-Ping, LIU Xin-Wang, LIN Jia-Run, ZHU Ming

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

+ Corresponding author: E-mail: mleoking@163.com

Zhang CW, Yin JP, Cai ZP, Liu XW, Lin JR, Zhu M. Active queue management algorithm to counter DDoS attacks. *Journal of Software*, 2011,22(9):2182−2192. http://www.jos.org.cn/1000-9825/3920.htm

Abstract: A resilient stochastic fair blue (RSFB) algorithm is proposed to preserve the existing normal network throughput under DDoS attacks. RSFB algorithm identifies benign flows according to their marking probability, which is derived from the stochastic fair blue algorithm. All the identified benign flows are then recorded in a benign flow queue (BFQ). Finally, the RSFB algorithm ensures the transportation of the packets from benign flows to the BFQ. A series of simulations are carried out to evaluate the anti-attack performance of RSFB and a serial of well known AQM algorithms. The results show that the RSFB algorithm i) is highly robust, ii) can well preserve the TCP throughput in the presence of DDoS attacks, iii) and obviously over performs the existing AQM algorithms when facing DDoS attacks.

Key words: active queue management; distributed denial-of-service; stochastic fire blue algorithm

摘 要: 提出一种能够在 DDoS(distributed denial-of-service)攻击下保证现有正常网络流量的弹性随机公平蓝色(resilient stochastic fair blue,简称 RSFB)算法.RSFB 算法根据数据流标记概率来识别良性数据流,并将识别出的良性数据流记录更新到一个良性数据流队列(benign flow queue,简称 BFQ)中.算法再根据 BFQ 中的良性数据流记录来保证良性数据流数据包的顺利传输.通过开展一系列实验,评估对比了 RSFB 算法和几个著名主动队列管理(active queue management,简称 AQM)算法的抗 DDoS 攻击性能.实验结果表明,RSFB 算法具有如下优点:1) 具有高度的健壮性;2) 能够在发生 DDoS 攻击时有效保证现有正常 TCP 数据流的吞吐率;3) 抗 DDoS 攻击性能明显优于现有的主动队列管理算法.

关键词: 主动队列管理;分布式拒绝服务攻击;随机公平蓝色算法

中图法分类号: TP393 **文献标识码**: A

近年来,主动队列管理(active queue management,简称 AQM)算法一直是网络研究领域的研究热点之一.为了缓解网络拥塞和提高网络性能,研究人员在近几十年已经提出了大量的主动队列管理算法.这些主动队列管理算法包括著名的随机早检测(random early detection,简称 RED)算法[1],以及注重公平性的主动队列管理算法,

* 基金项目: 国家自然科学基金(60970034, 61070198, 60803002, 60903040)
收稿时间: 2010-01-24; 修改时间: 2010-04-27; 定稿时间: 2010-07-06

如随机公平蓝色(stochastic fair blue,简称 SFB)算法[2]和优先丢弃随机早检测(RED with preferential dropping,简称 RED-PD)算法[3]等.虽然这些主动队列管理算法能够适用于多样的网络条件,但是绝大部分现有主动队列管理算法在设计时都没有考虑到自己应对网络攻击时的鲁棒性.它们对于现今公认是 Internet 服务最主要威胁之一的分布式拒绝服务(distributed denial-of-service,简称 DDoS)攻击就存在明显的性能漏洞[4].本文的实验结果显示:发生中度和重度 DDoS 攻击时,现有主动队列管理算法(RED[1],RED-PD[3],SFB[2])的性能较未发生 DDoS 攻击时下降 90%以上,它们的性能甚至比传统的尾丢弃(DropTail)队列管理算法还要差.因此,如何提高主动队列管理算法的抗 DDoS 攻击能力是一个亟待解决的问题.

本文提出了一种能够在 DDoS 攻击下保证现有正常网络流量的弹性随机公平蓝色(resilient stochastic fair blue,简称 RSFB)算法来应对 DDoS 攻击.RSFB 算法根据从数据流标记丢弃模块(FMDM)中得到的数据流标记概率来识别良性数据流,并将识别出的良性数据流记录更新到 BFQ 中.算法再根据 BFQ 中的良性数据流记录来保证良性数据流数据包的顺利传输.实验分析表明,RSFB 算法具有高度的健壮性,并且能够在发生 DDoS 攻击时有效保证现有 TCP 数据流的吞吐率,极大地减轻攻击的危害程度.本文的主要工作包括:

(1) 分析并实验验证了现有主动队列管理算法对于 DDoS 攻击存在严重的性能漏洞;
(2) 提出了一种能够在 DDoS 攻击下保证现有正常网络流量的弹性随机公平蓝色算法;
(3) 实验评估了 RSFB 算法在存在单个瓶颈链路的简单网络拓扑下和存在多个瓶颈链路的复杂网络拓扑下的性能.使用 RSFB 算法后,网络因为 DDoS 攻击而损失的正常数据流吞吐率被控制在 1%以内;
(4) 实验对比了 RSFB 算法与多个现有著名主动队列管理算法的抗 DDoS 攻击性能.结果显示, RSFB 算法的抗攻击性能显著优于现有主动队列管理算法.

本文第 1 节对相关工作进行阐述.第 2 节提出能够在 DDoS 攻击下保证现有正常网络流量的弹性随机公平蓝色算法.第 3 节利用对比仿真实验详细分析、比较本文算法与具有代表性的 RED 算法[1]、SFB 算法[2]、RED-PD 算法[3]在单个瓶颈链路的简单网络拓扑和多个瓶颈链路的复杂网络拓扑下的性能.第 4 节总结全文并展望未来工作.

1 相关工作

近年来,主动队列管理算法是网络拥塞控制领域的研究热点之一.通过在路由器端主动丢弃或标记部分数据包,主动队列管理算法能够有效避免网络拥塞和提高 TCP 协议的性能[5].研究人员已经提出了大量的主动队列管理算法及其相关的改进技术[1−3,6−9],虽然这些算法能够适用于多样的网络条件,但是绝大部分现有主动队列管理算法在设计时都没有考虑到自己的抗攻击性能.

我们在以前的工作[10]中提出了一种抗低速率拒绝服务(low-rate doS,简称 LDoS)攻击[11]的主动队列管理算法:健壮随机早检测(robust reD,简称 RRED)算法.RRED 算法提出了可疑攻击数据包的概念,并成功利用其对 LDoS 攻击进行检测和过滤.RRED 算法主要针对 LDoS 攻击中攻击包的行为特征采取相应的检测策略,它无法有效应对采用 IP 地址欺骗的 DDoS 攻击.本文算法采取与 RRED 算法不同的思路,不着眼于检测和记录攻击数据流,而是识别并记录良性数据流.在攻击发生时,算法再根据良性数据流记录来保证良性数据流数据包的顺利传输.理论分析和实验结果表明,本文算法能够有效应对包括地址欺骗 DDoS 攻击在内的大规模 DDoS 攻击.

现有关于主动队列管理算法的研究工作中,与本文工作比较相近的是“限制非响应数据流速率”的研究.这里,非响应数据流是指对于网络拥塞不进行响应的数据流,响应数据流是指对于网络拥塞进行响应的数据流[3].响应数据流在网络拥塞时会通过主动降低数据发送速率来避免网络的进一步拥塞.TCP 数据流就是响应数据流的典型代表,UDP 数据流是非响应数据流的典型代表.这些非响应数据流往往大量占用网络带宽,影响其他正常响应数据流的性能.近年来,有很多基于公平性的主动队列管理算法被提出来解决以上“非响应数据流”的问题. RED-PD 算法[3]和 SFB 算法[2]是这些公平主动队列管理算法的典型代表.

RED-PD 算法利用路由器上的数据包丢弃历史来检测高带宽占用数据流,并在网络拥塞时优先丢弃来自这些高带宽占用数据流的数据包[3].这种算法需要为网络中每个数据流建立数据记录,它无法有效处理 DDoS 攻

击产生的海量数据流信息.

SFB 算法通过为每个数据流维持一个标记概率 p_m 来检测和限制非响应数据流[2].SFB 算法的主要问题在于,它无法有效处理 DDoS 攻击产生的海量数据流信息,尤其是发生地址欺骗 DDoS 攻击时.

总体来说,抗 DDoS 攻击的主动队列管理算法现在还处于起步阶段.虽然已经有一些探索性的工作,但是还没有比较完善可用的解决方案.

2 弹性随机公平蓝色算法

2.1 算法基本思想

DDoS 攻击,特别是地址欺骗 DDoS 攻击往往会产生海量的攻击数据流.依靠记录攻击数据流来实现攻击检测和过滤的算法[2,3]一般只能够应对小规模的 DDoS 攻击,对于大规模的 DDoS 攻击,它们会产生巨大的空间开销,性能也会出现急剧地下降.因此,为了有效应对大规模 DDoS 攻击,本文算法不着眼于检测和记录攻击数据流,而是识别并记录良性数据流.RSFB 算法根据良性数据流记录来保证良性数据流数据包的顺利传输.

弹性随机公平蓝色算法的组成示意图如图 1 所示.该算法由一个数据流标记丢弃模块(flow mark and drop module,简称 FMDM)、一个良性数据流队列(benign flow queue,简称 BFQ)和一个数据包队列(packet queue,简称 PQ)组成.其中,数据流标记丢弃模块为每个数据流 f 计算并维持一个标记概率 $f.p_m$,同时,根据这个标记概率 $f.p_m$ 丢弃来自每个数据流的数据包.BFQ 用于维持良性数据流记录,RSFB 算法将识别出的良性数据流实时记录更新到 BFQ 中.对于被数据流标记丢弃模块丢弃的数据报文,RSFB 算法根据 BFQ 记录来判断其是否来自于良性数据流.如果是良性数据流中的数据包,就将其重新加入数据包队列中发送,否则就丢弃.

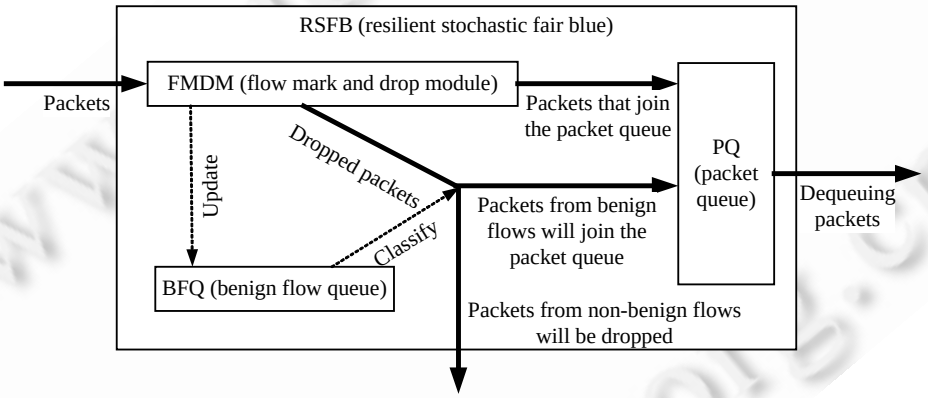


Fig.1 Schematic diagram of RSFB algorithm

图 1 弹性随机公平蓝色算法组成示意图

本文算法使用 BFQ 记录良性数据流信息,并通过将被丢弃的良性数据流数据包重新加入数据包队列保证良性数据流数据包的顺利传输.算法所涉及到的两个核心问题为:如何识别良性数据流和如何保证良性数据流数据包的顺利传输.

如何识别良性数据流是本文算法要解决的核心问题之一,RSFB 算法采用阈值法识别良性数据流(公式(1)):

$$f=\begin{cases}\text{benign flow}, & f.p_m \leqslant p_{mt}\\ \text{non-benign flow}, & f.p_m > p_{mt}\end{cases} \tag{1}$$

在公式(1)中,$f.p_m$ 是数据流标记丢弃模块为数据流 f 更新并维持的标记概率,p_{mt} 是识别良性数据流的阈值.$f.p_m$ 的设计思想和计算过程与随机公平蓝色算法[2]一致,其更新过程如下面的伪码所示[2]:

Variables:
pkt: denotes an arrival packet
f: denotes *pkt*'s corresponding flow
$f.p_m$: denotes the marking/dropping probability p_m for flow *f*
PQ: denotes the packet queue
$B[l][n]$: used for the Bloom Filters constructed with *L* levels with each level containing *N* bins
$B[j[h_j].qsize$: The number of packets in PQ, which were hashed into $B[j][h_j]$.
mean_size_each_bin: (max size of PQ)/*N*.
drop_size_each_bin: *mean_size_each_bin*+*mean_size_each_bin*/2.
For each arrival packet *pkt*:
Compute hashes $h_0,h_1,...,h_{L-1}$ for *pkt*;
for *j*=0 to *L*−1
if ($B[j][h_j].qsize$>*mean_size_each_bin*)
$B[j][h_j].p_m$+=delta;
if ($B[j][h_j].qsize$>*drop_size_each_bin*)
Drop *pkt*;
else if ($B[j][h_j].qsize$==0)
$B[j][h_j].p_m$−=delta;
$f.p_m$=min($B[0][h_0].p_m...B[L][h_{L-1}].p_m$);
if ($f.p_m$==1)
Drop *pkt*;
else
Drop *pkt* with probability $f.p_m$;

这里使用*L*层(每层包括*N*个记录仓)布鲁姆过滤器(Bloom filter,简称 BF)[12]记录和更新数据流的标记概率.每个数据流*f*都对应一个标记概率$f.p_m$,随着数据包的到来,这个标记概率会根据伪码中给出的更新机制实时更新.这个更新机制使:1) 非响应数据流很快地将它们的p_m增大到 1;2) 响应数据流的p_m值保持在 0 左右[2].响应数据流就是本文要识别并记录的良性数据流.本文根据数据流标记丢弃模块将良性数据流的p_m值保持在 0 左右的特性,将阈值p_{mt}设定为 0.一系列仿真实验的结果表明,这种设定能够使 RSFB 算法在多样的网络条件下都具有优秀的抗 DDoS 攻击性能,能够在发生 DDoS 攻击时有效保证现有正常 TCP 数据流的吞吐率.当然,研究能够动态自动确定p_{mt}取值的算法也是本文的未来工作之一.

如何保证良性数据流数据包的顺利传输,是本文算法所涉及的另外一个核心问题.现有主动队列管理算法对于 DDoS 攻击存在明显的性能漏洞,在 DDoS 攻击发生时,正常数据流的吞吐率出现了明显下降[4].本文算法中的数据流标记丢弃模块由于使用了随机公平蓝色算法为每个数据流更新并维持标记概率,在DDoS攻击发生时,此模块也会大量丢弃来自正常数据流的数据包,严重地影响了网络性能.RSFB 算法通过把被数据流标记丢弃模块丢弃的良性数据流数据包重新加入到数据包队列中来保证良性数据流数据包的顺利传输.在检测到有良性数据流的数据包 *pkt* 将要被数据流标记丢弃模块丢弃时,RSFB 算法首先判断数据包队列是否已满:如果数据包队列未满,就直接把 *pkt* 插入到 PQ 中;如果数据包队列已满,就尝试丢弃一个非良性数据流的数据包,然后再把 *pkt* 插入到 PQ 中.

2.2 算法流程

RSFB 算法的具体实现伪码如下:

RSFB algorithm.

Variables:

```
2       pkt: denotes an arrival packet
3       f: denotes pkt's corresponding flow
4       f.p_m: denotes the marking/dropping probability p_m for flow f in SFB algorithm
5       PQ: denotes the packet queue
6       pkt_n: denotes a packet in PQ which is not from benign flows
7   For each arrival packet pkt:
8       Feed pkt to FMDM that updates and maintains a marking/dropping probability p_m for each flow f.
9   Step 1: Update the Benign Flow Queue
10      if f.p_m≤p_mt then
11          if f is in BFQ then
12              delete f from BFQ
13              insert f into BFQ
14          else
15              if BFQ is full then
16                  delete the head flow from BFQ
17                  insert f into BFQ
18              else
19                  insert f into BFQ
20  Step 2: Ensure the transportation of the packets from benign flows
21      if SFB algorithm dropped pkt then
22          if f is in BFQ then
23              if PQ is full then
24                  if exists a packet pkt_n in PQ which is not from benign flows then
25                      drop pkt_n from PQ
26                      insert pkt into PQ
27                  else
28                      drop pkt
29              else
30                      insert pkt into PQ
```

伪码中,*pkt* 代表一个到来的数据包,*f* 是 *pkt* 对应的数据流,PQ 代表数据包队列.RSFB 算法是数据包驱动的.当一个数据包 *pkt* 到达路由器时,它首先通过数据流标记丢弃模块,然后,再分两步处理这个数据包:

步骤 1(第 9 行~第 19 行):更新良性数据流队列.标记概率(p_m)值小于或等于标记概率阈值 p_{mt} 的数据流 *f* 被认为是良性数据流,并且被插入到 BFQ 中.其中,BFQ 被设计为一个修改后的先进先出(FIFO)队列.在 BFQ 中,删除操作能够删除任意元素而不仅仅是队首元素,而插入操作只能在队尾进行.如果 *f* 已经在 BFQ 中,那么先在 BFQ 中删除 *f*,然后再在 BFQ 中插入 *f*.这样实现的目的在于使最新发现的良性数据流始终处于 BFQ 的尾部.随着良性数据流的增加,BFQ 中头部的数据流会被逐渐删除.如果一个记录于 BFQ 的良性数据流后来转变为攻击数据流,那么它的 p_m 值就会大于 p_{mt},以后不会再被识别为良性数据流;而它在 BFQ 中的记录也会被新识别的良性数据流逐渐推向 BFQ 的头部,直至被删除.这样,就在 BFQ 的更新中不仅能够有效记录良性数据流,还可以有效剔除非良性数据流.

步骤 2(第 20 行~第 30 行):保证良性数据流数据包的顺利传输.在检测到有良性数据流的数据包 *pkt* 将要被 SFB 算法[2]丢弃时,RSFB 算法通过下面两步尝试把它重新插入到数据包队列中:如果数据包队列未满,就直接把 *pkt* 插入到 PQ 中;如果数据包队列已满,就尝试丢弃一个非良性数据流的数据包,然后再把 *pkt* 插入到 PQ 中.

如果数据包队列已满而且其中没有非良性数据流的数据包时,就直接把 *pkt* 丢弃.在最后这种情况下,数据包队列中都是良性数据流的数据包,说明没有攻击发生或是攻击没有任何效果,所以本文算法在这种情况下不再对 *pkt* 作入队操作.

3 实验结果与性能分析

本节将通过在 NS2[13]网络仿真实验平台上开展一系列的实验,评估 RSFB 算法在 DDoS 攻击下的性能.其中:第 3.1 节通过在一个典型的哑铃(dumbbell)状实验网络拓扑上开展实验,评估了 RSFB 算法在单个瓶颈链路的简单网络下的性能;第 3.2 节通过在存在多个瓶颈链路的网络拓扑中开展实验,评估了 RSFB 算法在复杂网络下的性能.实验中也引入了现有的一些著名的 AQM 算法来与本文算法作对比,这些参与对比的 AQM 算法包括 RED 算法[1]、RED-PD 算法[3]、SFB 算法[2]和尾丢弃(DropTail)算法.

3.1 算法在单个瓶颈链路下的性能实验

本节采用的实验网络拓扑如图 2 所示.其中,瓶颈链路位于节点 R_0 和 R_1 之间,其上运行参与实验的 AQM 算法并设定数据包队列长度为 50 个数据包,其他链路运行尾丢弃算法.30 个正常用户($User_1$~$User_{30}$)各产生一条基于 TCP(newreno)协议的 FTP 数据流,其数据包大小为 1000 字节.20 个攻击者($Attacker_1$~$Attacker_{20}$)产生基于 UDP 协议的 DDoS 攻击数据流,其数据包大小为 50 字节.RSFB 算法中的良性数据流队列大小凭经验设定为 50 个数据流.所有 AQM 算法的其他参数都采用 NS2 的默认取值.

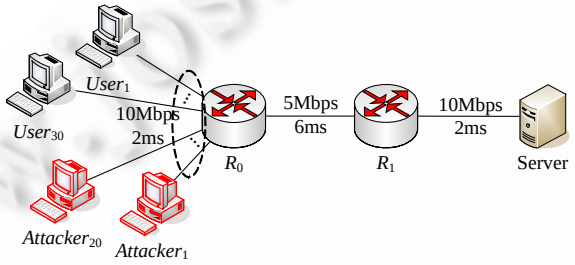


Fig.2 Experimental topology of the single bottleneck network

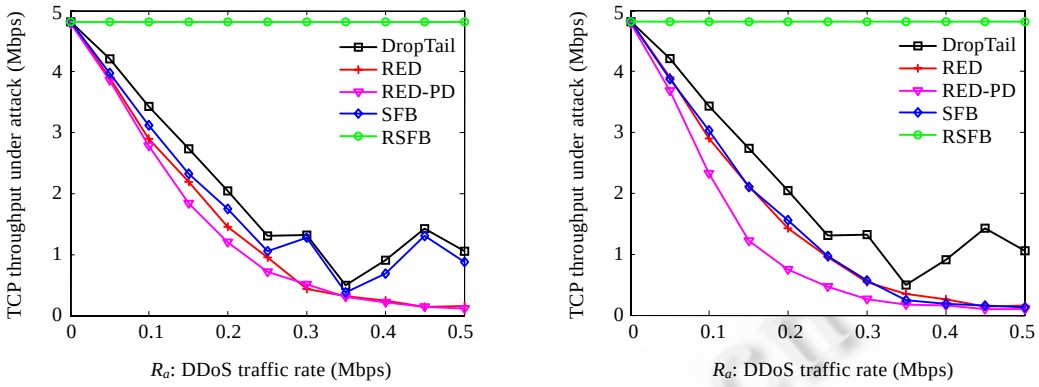
图 2 单个瓶颈链路实验网络拓扑图

每次仿真实验持续 50s,正常用户数据流从 10s 开始到 40s 结束,攻击数据流从 20s 开始到 30s 结束.

本节将开展两组实验,一组为使用真实 IP 地址的 DDoS 攻击实验,另一组为采用虚假 IP 地址的 DDoS 攻击(即地址欺骗 DDoS 攻击)实验.令 R_a 代表每个攻击者的攻击速率,为了研究 AQM 算法的性能受不同速率 DDoS 攻击的影响,这里开展的两组实验将分别把 R_a 从 0Mpbs 逐渐增大到 0.5Mbps.

本文主要通过 DDoS 攻击下正常 TCP 数据流的吞吐率来评估 AQM 算法的抗攻击性能,实验结果如图 3 所示.其中,图 3(a)为真实 IP 地址 DDoS 攻击的实验结果,图 3(b)为地址欺骗 DDoS 攻击的实验结果.在图 3 中,纵轴为 TCP 数据流在 DDoS 攻击下的吞吐率,□线代表尾丢弃算法,+线代表 RED 算法[1],▽线代表 RED-PD 算法[3],◇线代表 SFB 算法[2],○线代表本文提出的 RSFB 算法.

实验结果显示,RSFB 算法具有高度的健壮性,它在发生真实地址 DDoS 攻击和地址欺骗 DDoS 攻击的情况下都能有效保证现有 TCP 数据流的吞吐率.实验结果同时也验证了已有的 AQM 算法对于 DDoS 攻击都存在严重的性能漏洞,它们的性能随着攻击速率的上升而逐渐下降.在发生 DDoS 攻击时,现有 AQM 算法(RED 算法[1]、RED-PD 算法[3]、SFB 算法[2])的性能甚至比传统的尾丢弃算法还要差.特别是那些致力于提高公平性的 AQM 算法(SFB[2]和 RED-PD[3]),它们受地址欺骗 DDoS 攻击的影响更明显(对比图 3(a)和图 3(b)可以看出,对于同样的 R_a,RED-PD 算法和 SFB 算法在地址欺骗 DDoS 攻击下的性能要更差).



(a) TCP throughput under DDoS attacks
(a) DDoS 攻击下 TCP 数据流的吞吐率

(b) TCP throughput under spoofing DDoS attack
(b) 地址欺骗 DDoS 攻击下 TCP 数据流的吞吐率

Fig.3 Experimental results on single bottleneck network

图 3 单个瓶颈链路实验结果图

3.2 算法在多个瓶颈链路下的性能实验

本节通过对如图 4 所示的层次状拓扑网络进行实验,检验本文算法在有多个瓶颈链路的复杂网络拓扑下的性能.图中细线代表的瓶颈链路共有 6 条,根据在网络拓扑中的位置,它们又分为两类:一级瓶颈链路(R_1-R_0, R_2-R_0)和二级瓶颈链路(R_3-R_1,R_4-R_1,R_5-R_2,R_6-R_2),其中,二级瓶颈链路的数据流速率受限于一级瓶颈链路.所有瓶颈链路的带宽均为 5Mps,传输延迟均为 6ms.粗线代表的非瓶颈链路共有 161 条,其带宽为 10Mpbs,传输延迟为 2ms.瓶颈链路运行 AQM 算法并设定数据包队列长度为 50 个数据包,非瓶颈链路运行尾丢弃算法.R_3~R_6 路由节点上各连接有 30 个正常用户节点($User_1$~$User_{30}$).这些正常用户各产生一条基于 TCP(newreno)协议的 FTP 数据流,其数据包大小为 1 000 字节.R_3~R_6 路由节点上又各连接有 20 个攻击者节点($Attacker_1$~$Attacker_{20}$).这些攻击者产生基于 UDP 协议的 DDoS 攻击数据流,其数据包大小为 50 字节.RSFB 算法中的良性数据流队列大小同样凭经验设定为 50 个数据流.所有 AQM 算法的其他参数都采用 NS2 的默认取值.

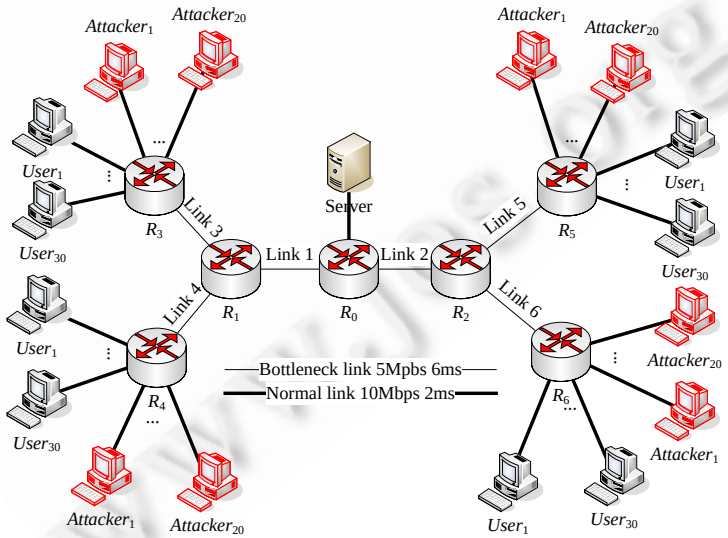


Fig.4 Experimental topology of multiple bottleneck network

图 4 多个瓶颈链路实验网络拓扑图

每次仿真实验持续 50s,正常用户数据流从 10s 开始到 40s 结束,攻击数据流从 20s 开始到 30s 结束.

对每个参与实验的 AQM 算法,本文对正常情况下(未发生 DDoS 攻击时)其所控制的 6 条瓶颈链路上正常 TCP 数据流的吞吐率进行了统计,结果见表 1.

从表 1 中可以看出,在未发生 DDoS 攻击时,5 种队列管理算法(DropTail,RED[1],RED-PD[3],SFB[2]和 RSFB)的性能是相近的,它们在一级瓶颈链路 R_1-R_0 和 R_2-R_0 上都实现了相同的吞吐率(4.8Mbps).在这种情况下,影响正常数据流吞吐率的因素主要是瓶颈链路的带宽和网络的拓扑结构.

Table 1 TCP throughput on each bottleneck link when there is no DDoS attack

表 1 未发生 DDoS 攻击时各瓶颈链路的正常数据流吞吐率统计表

Bottleneck links	DropTail (Mbps)	RED (Mbps)	RED-PD (Mbps)	SFB (Mbps)	RSFB (Mbps)
R_1-R_0	4.83	4.83	4.82	4.83	4.83
R_2-R_0	4.83	4.83	4.82	4.83	4.83
R_3-R_1	1.20	2.55	2.84	1.78	0.94
R_4-R_1	4.15	3.15	3.02	3.61	4.32
R_5-R_2	1.16	2.69	2.77	1.24	0.96
R_6-R_2	4.20	3.03	3.06	4.19	4.35

这里定义一个瓶颈链路在发生DDoS攻击时的性能保全率(R_{pr})为:发生DDoS攻击时,其上正常数据流的吞吐率与未发生DDoS攻击时其上正常数据流的吞吐率之比.又根据对表1的分析可知,参与实验的5种队列管理算法在正常情况下(未发生 DDoS 攻击时)的性能是相近的.因此,本节以后的实验中就可以根据 R_{pr} 来评估参与实验的队列管理算法的抗 DDoS 攻击性能.

本节将进行 3 组不同攻击速率的 DDoS 攻击实验,第 1 组实验中,每个攻击者的攻击速率(R_a)固定为0.05Mpbs;第2组实验中,每个攻击者的攻击速率固定为0.25Mpbs;第3组实验中,每个攻击者的攻击速率固定为0.5Mpbs.本文将对每组实验中参与实验的队列管理算法在每条瓶颈链路上的性能保全率(R_{pr})进行统计和分析.统计结果见表2~表4.其中,白色背景区域的统计值为发生真实地址 DDoS 攻击时每条瓶颈链路的 R_{pr},灰色背景区域的统计值为发生地址欺骗 DDoS 攻击时各条瓶颈链路的 R_{pr}.

表 2 统计的是一种轻度 DDoS 攻击,其攻击速率较低(R_a=0.05Mbps),在二级瓶颈链路上的汇聚攻击速率为1Mbps(每个攻击者的攻击速率为 0.05Mbps,每个二级瓶颈链路连接有 20 个攻击者).可以看出:对于真实地址DDoS攻击,RSFB算法在所有一级瓶颈链路(R_1-R_0,R_2-R_0)和一个二级瓶颈链路(R_6-R_2)上具有最好的性能,在两个二级瓶颈链路(R_3-R_1,R_4-R_1)具有次好的性能;对于地址欺骗 DDoS 攻击,RSFB 算法在所有一级瓶颈链路(R_1-R_0,R_2-R_0)和两个二级瓶颈链路(R_4-R_1,R_6-R_2)上都具有最好的性能.而且,参与实验的 RED 算法[1]、RED-PD 算法[3]和 SFB 算法[2]除了在两个二级瓶颈链路(R_4-R_1,R_6-R_2)上的性能表现较传统尾丢弃算法好之外,它们在其他瓶颈链路上的性能都比 DropTail 算法要差.

表 3 统计的是一种中度 DDoS 攻击,在二级瓶颈链路上的汇聚攻击速率为 5Mbps(每个攻击者的攻击速率为 0.25Mbps,每个二级瓶颈链路连接有 20 个攻击者).可以看出,对于真实地址 DDoS 攻击和地址欺骗 DDoS 攻击,本文提出的 RSFB 算法在所有瓶颈链路上都具有最好的性能.而且,参与实验的 RED 算法[1]、RED-PD 算法[3]和 SFB 算法[2]在大部分瓶颈链路上的性能都比传统的尾丢弃算法还要差.特别是 RED-PD 算法[3]在中度的地址欺骗 DDoS 攻击下,甚至因为实验超时而无法完成整个实验过程.

表 4 统计的是一种重度 DDoS 攻击,在二级瓶颈链路上的汇聚攻击速率为 10Mbps(每个攻击者的攻击速率为 0.5Mbps,每个二级瓶颈链路连接有 20 个攻击者).可以看出,本文提出的 RSFB 算法在所有瓶颈链路上都具有最好的性能.RED-PD 算法[3]在重度地址欺骗 DDoS 攻击下也因为实验超时而无法完成整个实验过程.对于同样的攻击速率,SFB 算法[2]在地址欺骗 DDoS 攻击下的性能比在真实地址 DDoS 攻击下也要差得多.

Table 2 R_{pr} on each bottleneck link when there is an DDoS attack with R_a=0.05Mbps

表 2 发生 DDoS 攻击时各瓶颈链路的性能保全率统计表(R_a=0.05Mbps)

Bottleneck links	DropTail (%)	RED (%)	RED-PD (%)	SFB (%)	RSFB (%)
R_1-R_0	78.8 78.8	66.5 67.1	55.7 48.3	69.6 66.6	99.5 99.5
R_2-R_0	79.9 79.9	67.2 66.5	54.8 46.8	67.8 65.5	99.5 99.5
R_3-R_1	266.4 266.4	64.9 75.8	61.3 50.9	150.8 119.5	263.4 84.4
R_4-R_1	32.4 32.4	72.0 65.6	45.3 42.3	39.0 49.0	68.1 101.4
R_5-R_2	186.7 186.7	90.8 84.2	60.6 37.2	110.5 122.1	53.6 70.4
R_6-R_2	58.3 58.3	50.5 54.9	44.9 52.2	60.8 53.6	106.5 104.2

Table 3 R_{pr} on each bottleneck link when there is an DDoS attack with R_a=0.25Mbps

表 3 发生 DDoS 攻击时各瓶颈链路的性能保全率(R_{pr})统计表(R_a=0.25Mbps)

Bottleneck links	DropTail (%)	RED (%)	RED-PD (%)	SFB (%)	RSFB (%)
R_1-R_0	6.5 6.5	6.2 5.7	4.4 Time out	3.7 6.5	99.5 99.5
R_2-R_0	6.0 6.0	5.4 5.5	4.8 Time out	2.1 5.5	99.5 99.5
R_3-R_1	25.2 25.2	10.2 7.6	6.4 Time out	9.5 10.9	62.0 84.6
R_4-R_1	5.3 5.3	7.3 8.4	5.1 Time out	2.4 7.1	105.4 102.1
R_5-R_2	25.9 25.9	8.7 8.4	7.6 Time out	5.5 15.9	79.6 89.7
R_6-R_2	4.7 4.7	6.2 6.8	4.8 Time out	2.2 4.2	102.7 102.4

Table 4 R_{pr} on each bottleneck link when there is an DDoS attack with R_a=0.5Mbps

表 4 发生 DDoS 攻击时各瓶颈链路的性能保全率(R_{pr})统计表(R_a=0.5Mbps)

Bottleneck links	DropTail (%)	RED (%)	RED-PD (%)	SFB (%)	RSFB (%)
R_1-R_0	5.8 5.8	2.2 1.8	1.5 Time out	4.6 1.9	99.5 99.5
R_2-R_0	5.8 5.8	2.1 1.6	1.3 Time out	9.5 2.0	99.5 99.4
R_3-R_1	22.0 22.0	3.1 3.4	2.2 Time out	11.2 3.3	129.6 81.2
R_4-R_1	4.4 4.4	2.8 2.0	1.7 Time out	1.3 1.3	95.8 102.6
R_5-R_2	15.3 15.3	3.4 2.4	1.6 Time out	17.6 3.3	50.7 82.8
R_6-R_2	6.5 6.5	2.2 2.0	1.9 Time out	6.5 1.4	106.9 101.0

综合表 2~表 4 可以看出,发生中度和重度 DDoS 攻击时,现有的 DropTail 算法、RED 算法[1]、RED-PD 算法[3]和 SFB 算法[2]的性能都出现了急剧的下降,它们控制下的大部分瓶颈链路的正常数据流吞吐率较未发生 DDoS 攻击时均降了 90%以上.而 RSFB 算法在 DDoS 攻击发生时,成功保全了网络中大部分瓶颈链路的正常数据流吞吐率.使用 RSFB 算法后,所有一级瓶颈链路(R_1-R_0,R_2-R_0)因为 DDoS 攻击而损失的正常数据流吞吐率都被控制在 1%以内.同时,从以上实验结果也可以看出,本文算法在应用于复杂网络拓扑时也存在一定的不足,即 RSFB 算法控制下的二级瓶颈链路(R_3-R_1,R_4-R_1,R_5-R_2,R_6-R_2)出现了流量不均衡的现象.如何改进二级瓶颈链路的流量均衡性和公平性,需要进一步研究.

本文的实验结果同时也验证了已有的 AQM 算法(RED 算法[1]、RED-PD 算法[3]和 SFB 算法[2])对于 DDoS 攻击存在严重的性能漏洞.它们在 DDoS 攻击下的性能甚至比传统的尾丢弃算法还要差.特别是最近出现的公

平性 AQM 算法(RED-PD 算法[3]和 SFB 算法[2]),它们的性能受地址欺骗 DDoS 攻击的影响更为明显.因此,如何提高 AQM 算法的抗 DDoS 攻击能力是亟待解决的问题.本文的研究为这个问题提供了一个可行的解决方案.

4 总结与展望

本文提出了一个弹性随机公平蓝色算法来对抗DDoS攻击,并且对该算法在存在单个瓶颈链路的简单网络下和存在多个瓶颈链路的复杂网络下的性能都进行了仿真实验.

RSFB 算法能够在发生 DDoS 攻击时有效保证现有正常 TCP 数据流的吞吐率,但是对于在攻击进行中新出现的正常 TCP 数据流,RSFB 算法往往无法对其进行保证.因此,本文的未来工作之一在于:进一步优化 RSFB 算法,使其也能够保证在 DDoS 攻击进行中新出现的正常 TCP 数据流的吞吐率.一个改进思路是,可以结合我们以前工作[10]中采用的可疑攻击数据包技术,综合考虑攻击数据流和正常数据流的特征,使算法既能有效保证正常数据流的吞吐率,又能过滤攻击数据流.但进行以上改进的同时,也需要考虑减小数据流状态保存时的空间开销问题.

将 RSFB 算法在实际网络中应用并进一步优化改进该算法,是本文进一步研究内容之一.另外,如何改进本文算法在二级瓶颈链路的流量均衡性和公平性,也是今后应考虑的问题.这个问题的一个解决思路是,可以通过限制来自不同链路的良性数据流在良性数据流队列中的比例,改进本文算法在二级瓶颈链路的流量均衡性和公平性.当然,真正解决以上问题还需要进一步的研究与分析.

References:

[1] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance. IEEE/ACM Trans. on Networking, 1993,1(4): 397−413. [doi: 10.1109/90.251892]

[2] Feng WC, Kandlur DD, Saha D, Shin KG. Stochastic fair blue: A queue management algorithm for enforcing fairness. In: Proc. of the IEEE INFOCOM. 2001. 1520−1529. [doi: 10.1109/INFCOM.2001.916648]

[3] Mahajan R, Floyd S, Wetherall D. Controlling high-bandwidth flows at the congested router. In: Proc. of the IEEE Int'l Conf. on Network Protocols (ICNP). 2001. 192−201. [doi: 10.1109/ICNP.2001.992899]

[4] Luo XP, Chang RKC, Chan EWW. Performance analysis of TCP/AQM under denial-of-service attacks. In: Proc. of the IEEE Int'l Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS). 2005. 97−104. [doi: 10.1109/MASCOTS.2005.50]

[5] Ryu SW, Rump C, Qiao CM. Advances in Internet congestion control. IEEE Communications Surveys and Tutorials, 2003,5(1): 28−39. [doi: 10.1109/COMST.2003.5342228]

[6] Wang XL, Wang YJ, Zhou H, Cai KY. Optimal design of AQM routers with D-stable regions based on ITAE performance. Journal of Software, 2007,18(12):3092−3103 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/18/3092.htm [doi: 10.1360/jos183092]

[7] Yang JW, Gu DY, Zhang WD. An analytical design method of PID controller based on AQM/ARQ. Journal of Software, 2006, 17(9):1989−1995 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/1989.htm [doi: 10.1360/jos171989]

[8] Lu XC, Zhang MJ, Zhu PD. An adaptive PI active queue management algorithm. Journal of Software, 2005,16(5):903−910 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/16/903.htm [doi: 10.1360/jos160903]

[9] Ji QJ, Dong YQ. A load-adaptive active queue management algorithm. Journal of Software, 2006,17(5):1140−1148 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/1140.htm [doi: 10.1360/jos171140]

[10] Zhang CW, Yin JP, Cai ZP, Chen WF. RRED: Robust RED algorithm to counter low-rate denial-of-service attacks. IEEE Communications Letters, 2010,14(5):489−491. [doi: 10.1109/LCOMM.2010.05.091407]

[11] Kuzmanovic A, Knightly EW. Low-Rate TCP-targeted denial of service attacks and counter strategies. IEEE/ACM Trans. on Networking, 2006,14(4):683−696. [doi: 10.1109/TNET.2006.880180]

[12] Broder A, Mitzenmacher M. Network applications of Bloom filters: A survey. Internet Mathematics, 2004,1(4):485−509. [doi: 10.1080/15427951.2004.10129096]

[13] McCanne S, Floyd S. The network simulator—ns-2. 2008. http://www.isi.edu/nsnam/ns/

附中文参考文献:

[6] 王秀利,王永吉,周辉,蔡开元.基于D稳定域和ITAE准则的主动队列管理算法.软件学报,2007,18(12):3092−3103. http://www.jos.org.cn/1000-9825/18/3092.htm [doi: 10.1360/jos183092]

[7] 杨吉文,顾诞英,张卫东.主动队列管理中 PID 控制器的解析设计方法.软件学报,2006,17(9):1989−1995. http://www.jos.org.cn/1000-9825/17/1989.htm [doi: 10.1360/jos171989]

[8] 卢锡城,张明杰,朱培栋.自适应 PI 主动队列管理算法.软件学报,2005,16(5):903−910. http://www.jos.org.cn/1000-9825/16/903.htm [doi: 10.1360/jos160903]

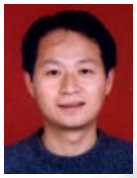
[9] 纪其进,董永强.一种链路负载自适应的主动队列管理算法.软件学报,2006,17(5):1140−1148. http://www.jos.org.cn/1000-9825/17/1140.htm [doi: 10.1360/jos171140]

张长旺(1984－),男,河北唐山人,博士生,主要研究领域为网络安全.

刘新旺(1983－),男,博士生,主要研究领域为机器学习.

殷建平(1963－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为信息安全,模式识别,网络算法.

林加润(1987－),男,博士生,主要研究领域为网络安全.



蔡志平(1975－),男,博士,副教授,CCF 高级会员,主要研究领域为网络安全,网络测量,虚拟化.

朱明(1985－),男,博士生,主要研究领域为网络安全.