doi:10.3969/j.issn.1000-436x.2012.z2.003

NS2 中 TCP 连接建立模拟的改进

姜誉[1,2]，任健[3]，周黎明[1]
（1. 黑龙江大学 计算机科学技术学院，黑龙江 哈尔滨 150080；2. 黑龙江省数据库与并行计算重点实验室，黑龙江 哈尔滨 150080；
3. 黑龙江大学 信息科学与技术学院，黑龙江 哈尔滨 150080）

摘 要：传输控制协议(TCP, transmission control protocol)连接建立的“三次握手”过程中涉及对半连接表和连接表的管理。但是，已得到广泛应用的网络模拟器 NS2 对 TCP 连接建立过程只有一个形式的表示，没有完整的具体实现。对此进行了改进，为 NS2 增加了半连接表结构，并将 Linux 内核管理半连接表的方式移植到了 NS2 中。仿真结果可清晰地看到 TCP 连接建立过程中半连接表变化，从而满足 TCP SYN 洪泛攻击防控等研究中对 TCP 连接建立过程模拟的需要。

关键词：网络模拟；传输控制协议；连接建立；半连接
中图分类号：TP393 **文献标识码**：A **文章编号**：1000-436X(2012)Z2-0015-05

Improvement on simulating TCP connection establishment in NS2

JIANG Yu[1,2], REN Jian[3], ZHOU Li-ming[1]
(1. School of Computer Science and Technology, Heilongjiang University, Harbin 150080, China;
2. Key Laboratory of Database and Parallel Computing of Heilongjiang Province, Harbin 150080, China;
3. School of Information Science and Technology, Heilongjiang University, Harbin 150080, China)

Abstract: In the procedure of three-way handshake of transmission control protocol connection establishment, it involves the management of half-connection and connection tables. However, in the famous networks simulator NS2, there is only a formal description instead of a concrete and complete implementation for the procedure of TCP connection establishment. An improvement was beed made on the point. Table structures for half-connections and connections are added, and the way of managing half-connection table in Linux kernel is also imported into NS2. Simulation results show that it is clear to monitor the change in half-connection table in the TCP connection establishment procedure, and thus the requirement of simulating the connection establishment procedure in studies such as protection from TCP SYN flooding attack can be met.

Key words: network simulation; transmission control protocol; connection establishment; half-connection

1 引言

传输控制协议的连接建立采用“三次握手”过程，服务进程接收到客户端的连接请求 TCP SYN 报文之后，如果能够建立连接，则将有关的连接信息放入半连接表中，并发出 TCP SYN+ACK 报文(即控制比特 SYN 和 ACK 同时置 1 的 TCP 报文)；当收到客户端响应的 TCP ACK 报文后，再将对应的连接信息从半连接表移入连接表中，从而完成连接建立过程。此外，还会将超时的半连接从半连接表

收稿日期：2012-07-10
基金项目：黑龙江省自然科学基金资助项目（F200823）；黑龙江省教育厅科学技术研究项目基金资助项目（11551342, 12521432）
Foundation Items: The National Natural Science Foundation of Heilongjiang Province (F200823); The Science and Technology Research Project Foundation of Heilongjiang Education Office (11551342, 12521432)

中删除。因此，实际上，TCP 连接建立的“三次握手”过程中涉及对半连接表的管理过程。

NS2 (network simulator version2)[1]是一个免费开源的网络模拟平台软件，具有强大的网络模拟功能，它的开放性使人们可以根据研究需要方便地增加新的模块。特别地，NS2 中关于 TCP 协议许多功能的模拟实现十分完善，因此，有关的已有研究大致可以分为两类，一类是以 NS2 作为平台，对所设计的新协议或算法进行模拟比较研究，例如，文献[2~5]介绍了在 NS2 中添加新协议模块等的有关方法；另一类是根据新的应用环境，对包括 TCP 在内的有关协议的已有功能方法或有关参数进行改进，例如，针对传统的 TCP 拥塞控制方法不适用于无线网络环境的问题，设计和实现新的拥塞控制方法，从而对 NS2 中 TCP 协议的功能进行扩充[5,6]。但是，没有发现关注 NS2 中 TCP 协议的功能是形式上的实现还是具体的实现的研究。

通过分析 NS2 源码发现，NS2 对 TCP 连接建立功能只有一个形式的表示，它采用“□ ，□ □ TCP SYN □ ”的机制，并记 ACK 序列号为 0□对方收到 SYN 报文后，□ 40byte（无数据)□ACK □ □ SYN+ACK □ 发送 SYN 报文一方采用 last_ack==0 判断是否为第一次收到对方的 ACK 报文，并采用捎带确认发送 ACK 报文，当 last_ack==0 时发出的这个 ACK 报文相当于连接建立“三次握手”中的第 3 个报文。在 NS2 中没有半连接表以及连接表等重要的数据结构，从而既没有实现连接请求进入半连接表这一具体的过程，也不存在对半连接表的管理。因此，在采用 NS2 进行模拟并需要观察半连接表变化情况的研究，例如与半连接表有关的 TCP SYN 洪泛攻击防控研究中，必须对 NS2 进行功能扩展。

本文在分析 Linux 内核的 TCP 协议源代码基础上，对 NS2 的 TCP 协议功能进行扩展，为 NS2 添加了一个带有半连接表和连接表的 TCP 协议代理、半连接表初始化代理，以及对半连接表的管理代理等模块，为相应的模拟研究需要作好准备。

2 Linux 中 TCP 连接建立与管理

选择 Linux 2.6.31 内核版本，从服务器端角度，概要介绍 TCP 连接建立及管理方式。

2.1 Linux 中 TCP 连接的建立过程

内核初始化时为 TCP 协议生成一个结构体实例 tcp_hashinfo，该实例有 3 个散列表，分别为：绑定表 bhash、监听表 listening_hash 和连接表 ehash。绑定表 bhash 与本节讨论内容无关，不再赘述。监听表 listening_hash 的作用是挂接处于监听状态的套接字(socket)实例，当内核收到发给本机的 TCP SYN 连接请求报文时，将连接请求报文交给监听表中相应的监听套接字，以便完成“三次握手”过程。连接表 ehash 的作用是挂接已经建立好连接、可以进行数据通信的套接字实例。所有 TCP 服务进程共享一个监听表 listening_hash 和一个连接表 ehash。

某服务进程 S 完成“三次握手”的具体过程主要涉及 3 种表结构：监听表 listening_hash、半连接表 syn_table 和连接表 ehash。从实现来看，半连接表 syn_table 和连接表 ehash 都是结构体型数组组成的散列表，但连接表 ehash 为共享全局变量，半连接表 syn_table 为局部变量，每个服务进程处于监听状态的套接字各有一个相应的半连接表 syn_table。

服务进程 S 接收到客户端 C 发给 S 的连接请求报文后，解析出源 IP 地址、目的 IP 地址、源端口和目的端口。在监听散列表 linsening_hash 中查找是否有处于监听的套接字正在等待对该目的端口的连接，如果有，则生成一个类型为 request_sock 的结构体实例挂接到半连接表 syn_table 的某一个散列桶中，并发出 TCP SYN+ACK 报文(即控制比特 SYN 和 ACK 同时置 1 的 TCP 报文)。半连接表 syn_table 的结构示意如图 1 所示。

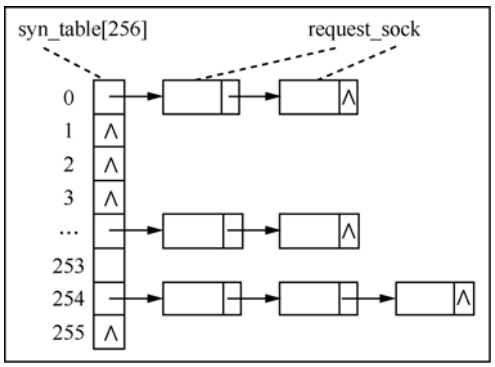

图 1 已挂入某些连接请求的半连接表结构示意

当服务进程收到客户端对 TCP SYN+ACK 响应的 TCP ACK 报文后，将对应的连接请求从半连接表转移到连接表中，从而完成连接建立过程。

2.2 Linux 内核对半连接表的管理

每个服务进程处于监听状态的套接字会启动一个定时器，每隔一段时间对本套接字的半连接表

进行扫描，目的是对半连接表中超时的连接请求进行 TCP SYN+ACK 报文的重传和将超过重传次数的连接请求从半连接表中删除出去。内核如下计算每一次扫描的散列桶数：

$$budget=2(lopt\rightarrow nr\_table\_entries/(timeout/interval))$$

其中，$lopt\rightarrow nr\_table\_entries$ 是与该监听套接字相关联的半连接表的散列桶数，*timeout* 为超时时间，内核初始化时缺省为 3s，*interval* 是对半连接表进行下一次检查的间隔时间，缺省为 0.2s。假设半连接表的散列桶数为缺省的上限值 256，则可以计算出内核一次检查半连接表的 34 个散列桶。

内核默认对每一个超时的SYN+ACK报文最多重传 5 次，超时时间初值缺省为 3s，每经过一次重传，超时重传的超时时间加倍，即 5 次重传时的超时时间依次分别为 6s、12s、24s、48s 和 96s，这表明一个未完成的连接请求在半连接表中最长能够存活 3+6+12+24+48+96=189s。在 5 次重传后还没有收到对该连接请求的 TCP ACK 确认报文，就将其从半连接表的散列桶中删除。

3　扩展、仿真与验证

本节给出在 NS-2.33 中增加的数据结构、扩展的代理模块以及实验设置、仿真结果和分析。

3.1　功能扩展

首先添加了简化的(满足模拟需要即可)、存储连接请求信息的结构体 request_sock 和连接请求监听套接字结构体 listen_sock 的主要字段如下。

```
struct request_sock {
struct request_sock *dl_next;
unsigned int saddr;//源 IP 地址
unsigned int faddr;//目的 IP 地址
unsigned short sport;//源端口
unsigned short fport;//目的端口
double recv_time;//进入半连接表的时间
double keep_alive_time;//存活时间
};
struct listen_sock {
unsigned int max_qlen;//半连接表容量大小
unsigned int qlen;
//半连接表中的当前半连接数量
struct request_sock *syn_table[256];//半连接表
}
```

其中，变量 *max_qlen* 记录半连接表容量，给出了每个服务进程处于监听状态的套接字所允许的半连接数量的最大阈值；变量 *qlen* 记录半连接表中半连接的实际数量，给出了对应的半连接表中当前实际存在的半连接数量；半连接表 *syn_table* 的桶数(即散列入口数)设置为 256 个。

其次，本文添加了 4 个代理，分别是：1) 新的 TCP 代理 P_NewTCP，它可以向指定的服务器代理发送任意指定源 IP 地址的连接请求 SYN 报文，并可指定发出的连接请求的真假属性，如果指定 SYN 报文真假属性为真，则代表正常连接请求，否则代表 SYN 洪泛攻击流量。2) 初始化半连接表代理 P_LinuxCore，并可按要求时刻输出半连接表的占用情况以便观察半连接表的变动。3) 接收连接请求代理 P_NewTCPSink，将收到的连接请求存入半连接表中。4) 半连接表管理代理 P_SynTableMge，管理和定时扫描半连接表，将正常的连接请求建立成连接，并清除超时的攻击流量连接请求。

再给出服务进程的总服务率的定义。

定义 1　服务进程总服务率 $r(t)=\sum_{i=0}^{t} ce_i / \sum_{i=0}^{t} cr_i$，其中，$ce_i$ 为截止到第 *i*s 时已完成连接建立的正常连接请求总数，cr_i 表示截止到第 *i*s 时已收到的正常连接请求总数。

3.2　实验设置

本文需要验证的是绑定在节点上的扩展功能的效果，与网络结构无关，为清晰起见，仿真时采用了星型的拓扑结构。其中星型的中心节点对应路由节点，只是简单转发报文，星型一个的边缘节点与服务节点对应，其他边缘节点与客户节点对应。链路带宽都设置为 1GB/s，保证模拟不受带宽的限制。上述添加的代理与节点的绑定关系如图 2 所示。

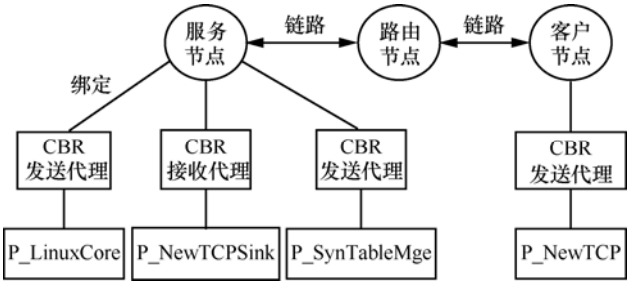

图 2　代理与节点的绑定关系

初始化半连接表代理 P_LinuxCore 和半连接表管理代理 P_SynTableMge 是通过调用自身的处理请求函数 sendmsg()激活的，因此需要绑定为 CBR

(constant bit rate)发送代理。半连接表容量设置为 1 048 576，即最多可同时容纳 2^{20} 个连接请求。

每次实验进行 400s，以保证观察到 2 段(每段 189s)完整的最大超时时间内半连接表的变化情况。正常的连接请求从第 1s 开始由客户节点发给服务节点，持续 395s。任意时刻正常连接请求的总数不超过半连接表表元总数的 90%，以保证没有攻击流量时所有的正常连接请求都完成建立连接。

SYN 洪泛攻击流量设置如下：在第 5s 时，产生攻击连接请求，持续时间为 380s，每秒产生的攻击连接请求总数为半连接表表元总数 1 048 576 的 4%，即攻击速率恒定为 41 943 个/秒连接请求。攻击流量的 IP 地址随机生成，每个攻击连接请求都认为是不同的，保证攻击流量总量。

经测试，在实际网络中，正常情况下仅需 100ms 左右即可完成连接建立过程，因此，管理半连接表时，每 100ms(即 0.1s)对半连接表进行一次清理，将正常的连接请求从半连接表移入到连接表中，完成连接的建立，并清除总计已超时 189s 的攻击连接请求。仿真实验在安装有 2.6.31 内核版本的 Linux RedHat5.0 操作系统、双二核 CPU、8GB 内存的曙光高性能服务器上进行。

3.3　仿真结果与分析

模拟的正常连接请求流量如图 3 所示。对正常的连接请求流量在 45~55s、145~155s、245~255s 以及 295~305s 这 4 个时段设定了突发流量，目的是为了观察在具有突发流量的情况下半连接表的占用变化情况。

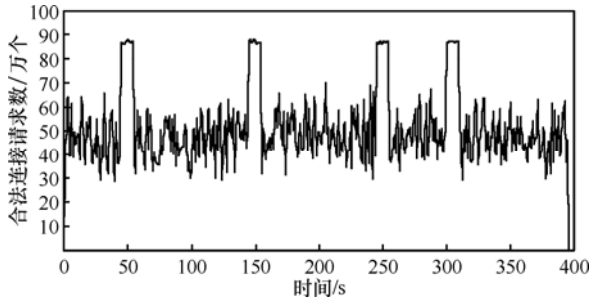

图 3　正常连接请求流量

无攻击流量时半连接表占用情况如图 4 所示。因为每 100ms 进行一次半连接表的管理，所以图中横轴的时间单位为 0.1s，这样观察的粒度更细。

从图 4 中可以看到，在几个突发流量时段上，半连接表的占用率也随着正常流量的突发而基本保持在 5%以上，其他时点的半连接表占用率则处于波动状态，时高时低。

根据管理方式，每个正常连接请求在半连接表中最多存在 100ms，同时，没有攻击流量，因此，半连接表的占用率始终没有超过 11%，模拟的全部正常连接请求都得以建立连接，服务率始终为 100%。

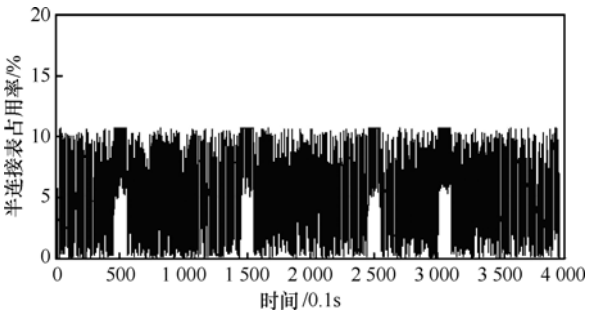

图 4　无攻击流量时半连接表的占用情况

然后，按照上述实验设置加入攻击流量。这时，半连接表的占用情况如图 5 所示。

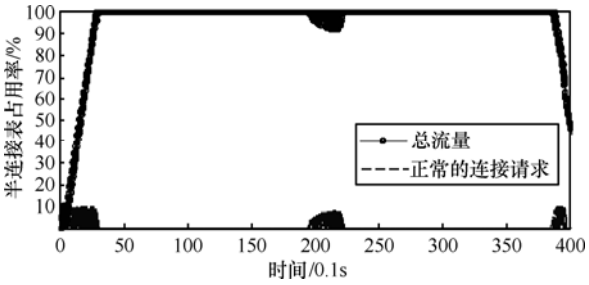

图 5　有 SYN 洪泛攻击流量时半连接表的占用率

从图 5 中可以看到，在 30s 左右时半连接表的占用率达到了 100%，此后大部分时间内半连接表都保持 100%的占用率，但除了在 194s 至 224s 之间和在第 383s 后，其他时间内正常连接请求在半连接表中的数量为 0，半连接表被攻击连接请求所占用，正常连接请求得不到服务。

随模拟时间变化的服务进程的总服务率如图 6 所示。从图 6 中看到，在开始一段时间内，由于半连接表还没有被攻击流量占满，因此，服务进程的总服务率达到并保持在 100%。

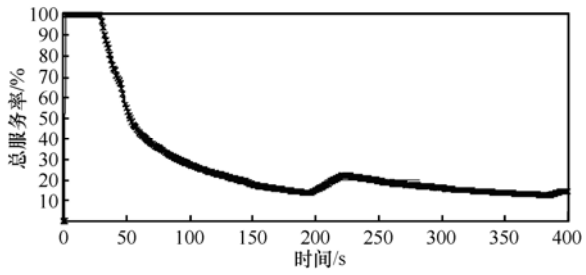

图 6　有 SYN 洪泛攻击流量时的总服务率变化情况

当半连接表短时间内被攻击连接请求占满后，由于未到 189s 的超时清理时间，半连接表内的攻击连接请求得不到清除，但正常连接请求仍持续发出，因此，随时间推移的服务进程服务率呈现大幅度下降的趋势。在 194s 至 224s 之间，一部分超时的攻击连接请求被清除，一些正常连接请求得以进入半连接表，使总服务率有所上升。但由于攻击流量持续存在，在第 224s 后半连接表很快又被新的攻击连接请求占满，半连接表中的正常连接请求数量又降为 0，使总服务率又呈下降趋势。在第 383s 时，新的超时攻击连接请求才被清除，第 386s 开始攻击结束，被清除的攻击连接请求在半连接表中得不到补充，正常连接请求才能再次进入到半连接表，总服务率又有所回升。实验中一共发出 20 266.7 万个连接请求，完成连接建立的请求总计 2 955.9 万个，总服务率仅为 14.6%，SYN 洪泛攻击产生了效果，正常连接请求受到很大影响。

4　结束语

本文针对 NS2 中 TCP 连接建立模拟的具体实现中存在的不足，在深入分析 Linux 内核服务器端 TCP 连接建立过程以及对半连接表管理方式的基础上，对 NS2 中 TCP 连接建立进行了功能扩展，添加了半连接表和连接表结构、相应的代理模块，以及管理半连接表的机制，并对实验结果进行分析。从仿真结果来看，本文的模块扩展是有效的，并能够清晰地观察到半连接表的变化情况，达到了目的。今后将在此基础上，进行 SYN 洪泛攻击的缓解方法研究。

参考文献：

[1] The network simulator - ns-2. [EB/OL]. http://www.isi.edu/nsnam/ns/ns-documentation.html, 2011.

[2] 李彪, 周文安, 解冰等. NS2 中异构网络的 QoS 监测模块的扩展[J]. 系统仿真学报, 2012, 24(3): 618-623.
LI B, ZHOU W A, XIE B, *et al*. NS2 extension for QoS monitoring in heterogeneous networks[J]. Journal of System Simulation, 2012, 24(3): 618-623.

[3] 王晓曦, 王秀利, 周津慧等. NS2 网络仿真器功能扩展方法及实现[J]. 小型微型计算机系统, 2004, 25(6): 1009-1014.
WANG X X, WANG X L, ZHOU J H, *et al*. Function extension of NS2 network simulator and its implementation[J]. Mini-micro Systems, 2004, 25(6): 1009-1014.

[4] 杨锦亚, 郭虹, 于宏毅等. NS-2 新功能模块的开发[J]. 计算机仿真, 2006, 23(11): 120-123.
YANG J Y, GUO H, YU H Y, *et al*. Development of new modules in NS-2[J]. Computer Simulation, 2006, 23(11): 120-123.

[5] 叶晓国. 基于 NS-2 的无线传感器网络仿真模块扩展方法的研究[J]. 计算机研究与发展, 2011, 48(Suppl): 302-306.
YE X G. NS-2-based simulation module extension method for wireless sensor networks[J]. Journal of Computer Research and Development, 2011, 48(Suppl): 302-306.

[6] ZHOU B, FU C P, ZHANG K, *et al*. An enhancement of TCP-veno over light-load wireless networks[J].IEEE Communications Letters, 2006, 10(6): 441-443.

作者简介：

姜誉（1968-），男，黑龙江伊春人，博士，黑龙江大学教授，主要研究方向为网络与信息安全、Internet 测量。

通信作者 **任健**（1971-），男，黑龙江哈尔滨人，硕士，黑龙江大学副教授，主要研究方向为网络与信息安全。E-mail: ren-jian@tom.com。

周黎明（1983-），男，黑龙江齐齐哈尔人，黑龙江大学硕士生，主要研究方向为网络与信息安全。