Journal of Computer Applications
计算机应用,2014,34(4):950 -954
ISSN 1001-9081
CODEN JYIIDU
2014-04-10
http://www.joca.cn

文章编号:1001-9081(2014)04-0950-05
doi:10.11772/j.issn.1001-9081.2014.04.0950

基于改进反向探测的 IPv6 邻居缓存保护方法

孔亚洲*,王振兴,王　禹,张连成
(数学工程与先进计算国家重点实验室,郑州 450002)
(*通信作者电子邮箱 zhouking2013@163.com)

摘　要:针对 IPv6 邻居缓存(NC)易被攻击的问题,提出一种改进的反向探测方法(RD+)。该方法首先引入时间戳和报文序列两个选项,分别用于限制报文响应时长以及响应报文匹配;之后,定义 RD+队列存储时间戳和报文序号等信息,并设计基于时间戳的随机早期检测(RED-T)算法对 RD+队列实施管理以防范拒绝服务(DoS)攻击。实验结果表明,RD+能够有效抵抗邻居缓存欺骗和 DoS 攻击,与启发式和显式相结合的方法(HE)以及安全邻居发现协议(SEND)相比,其资源消耗较少。

关键词:IPv6;邻居缓存;反向探测;队列管理;拒绝服务攻击
中图分类号:TP393.08　　**文献标志码**:A

Method of IPv6 neighbor cache protection based on improved reversed detection

KONG Yazhou*, WANG Zhenxing, WANG Yu, ZHANG Liancheng
(*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou Henan 450002, China*)

Abstract: IPv6 Neighbor Cache (NC) was very vulnerable to be attacked; therefore, an improved method named Reversed Detection Plus (RD+) was proposed. Timestamp and sequence were firstly introduced to limit strict time of response and response matching respectively; RD+ queue was defined to store timestamp and sequence, and Random Early Detection Based on Timestamp (RED-T) algorithm was designed to prevent Denial of Service (DoS) attacks. The experimental results show that RD+ can effectively protect IPv6 NC to resist spoofing and DoS attacks, and compared with Heuristic and Explicit (HE) and Secure Neighbor Discovery (SEND), RD+ has a low consumption of resources.

Key words: IPv6; Neighbor Cache (NC); reversed detection; queue management; Denial of Service (DoS) attack

0　引言

邻居发现协议(Neighbor Discovery Protocol,NDP)[1]是 IPv6 的一个关键协议,它组合了 IPv4 中的地址解析协议(Address Resolution Protocol,ARP)及 Internet 控制报文协议(Internet Control Message Protocol,ICMP)中的路由器发现和重定向等协议及功能,并对它们作了改进,解决了同一链路上不同节点之间的信息交互问题。由于其并没有针对链路内的安全威胁给出任何的安全机制,攻击者利用 NDP 存在的安全漏洞,可以对 IPv6 子网实施拒绝服务(Denial of Service,DoS)攻击和重定向攻击[2-3]。

IETF 标准规定由 IPSec AH 来保证 NDP 中数据包的可靠性和完整性,但是并未给出使用方案。安全邻居发现协议[4]通过引入加密生成地址(Cryptographically Generated Address,CGA)[5]和签名机制来保证 NDP 的安全,但其产生的计算开销过大,并没有得到广泛使用。

邻居缓存(Neighbor Cache,NC)是一组有关单个邻居的信息,它包括的信息有邻居 IP 地址与邻居链路层地址映射、邻居可达性状态等。邻居缓存的更新是通过 ND 报文的交互实现的,而 NDP 在设计时并未对报文进行有效保护,因此邻居缓存易遭受欺骗攻击和 DoS 攻击等。文献[6]提出了启发式类型、显式类型和启发式与显式相结合类型三种邻居缓存保护方法,但其并没有给出具体的实施方案,且其工作流程过于复杂,协议修改较大;文献[7]则提出了一种反向检测方法,虽然其一定程度上降低了邻居缓存受攻击的可能,由于反向探测报文没有任何保护机制,攻击者仍然可以发送大量的虚假应答报文,进而轻易绕过反向检测机制;文献[8]描述了 NDP 存在的一些操作问题,并给出了部分管理员的操作规范以缓解存在的问题,但是并未从协议实现上对邻居缓存进行保护。本文提出一种改进的反向检测方法(Reversed Detection Plus,RD+),当节点收到 NDP 报文时,向源节点发送RD+探测报文,同时引入时间戳、报文序号和队列管理等保护机制,确保邻居缓存的正确更新。最后,将其与其他邻居缓存保护方法进行分析比较,结果表明该方法可以有效抵抗邻居缓存欺骗攻击和 DoS 攻击,并且资源消耗少、协议兼容性强。

1　IPv6 邻居缓存安全威胁分析

邻居发现协议的模型最初是在基于完全可信网络的前提下提出的,节点用一系列 Cache 缓存了通过 ND 协议获取的邻节点的相关信息和网络的相关参数,其中邻居缓存存储了邻居 IP-MAC 映射及可达性状态等关键信息。

收稿日期:2013-10-14;**修回日期**:2013-12-19。

作者简介:孔亚洲(1989-),男,河南濮阳人,硕士研究生,CCF 会员,主要研究方向:IPv6 网络安全;　王振兴(1959-),男,河北晋州人,教授,博士,主要研究方向:IPv6 网络安全;　王禹(1984-),男,河南郑州人,博士研究生,主要研究方向:网络安全;　张连成(1982-),男,河南商丘人,讲师,博士,主要研究方向:流量分析、网络安全。

邻居缓存状态机如图 1 所示。

可达性状态的取值是下列 6 个值之一:

1)incomplete(未完成状态):表示正在解析地址,但邻居链路层地址尚未确定。

2)reachable(可达状态):表示地址解析成功,该邻居可达。

3)stale(失效状态):表示可达时间耗尽,未确定邻居是否可达。

4)delay(延迟状态):表示未确定邻居是否可达。delay 状态不是一个稳定的状态,而是一个延时等待状态。

5)probe(探测状态):节点会向处于 probe 状态的邻居持续发送邻居请求报文。

6)empty(空闲状态):表示节点上没有相关邻节点的邻居缓存表项。

由于在接收到邻居请求(Neighbor Solicitation, NS)/邻居应答(Neighbor Advertisement, NA)/路由器请求(Router Solicitation, RS)/路由器应答(Router Advertisement, RA)/重定向(Redirect)等报文时,节点并不对报文信息进行任何检查和认证,因此,邻居缓存易遭受欺骗攻击和 DoS 攻击等。

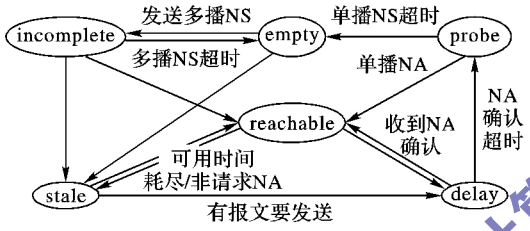

图 1　邻居缓存状态机

1.1　邻居缓存欺骗攻击

邻居缓存欺骗攻击指的是攻击者向目标节点发送包含虚假信息或错误信息的报文,使目标节点邻居缓存存储虚假或错误信息,进而破坏目标节点的通信。

由于邻居缓存默认总是用新的信息来覆盖旧的,与 IPv4 网络中的 ARP 缓存攻击类似,攻击者通过发送包含虚假信息的报文,可以对目标节点实施缓存欺骗攻击等。RFC 2461 规定,节点在收到发向自己的 NS/NA 包时,若缓存中没有对应源 IP 的记录或该记录中的 MAC 地址与 NS/NA 报文中的源 MAC 地址字段相异时,节点将根据 NS/NA 报文更新缓存中该 IP 和 MAC 的映射记录,并将其状态设置为 stale。这样,攻击节点向目标节点发送使用伪造的链路层地址的 NS/NA 报文,以致改变目标节点数据报的发送路径,但是,这种攻击时间受限,攻击者必须一直对虚假的链路层地址作出响应以维持攻击效果。

类似地,攻击者也可以向目标节点发送包含虚假信息的 RS/RA/Redirect 报文,实现对目标节点邻居缓存的欺骗攻击。

1.2　邻居缓存 DoS 攻击

IPv6 中节点在进行地址解析时,需要在 NC 中创建一个待解析邻居节点的缓存表项,并将其状态设置为 incomplete。由于 IPv6 默认子网前缀长度为 64 位,即同一链路上可以有多达 2^{64} 个节点,而节点邻居缓存容量远小于 2^{64},那么,攻击者可以在短时间内发送大量包含虚假信息的 NS 或 NA 报文,使得目标节点的资源耗尽,邻居缓存充满虚假信息,从而对目标节点或目标网络实施 DoS 攻击。

2　改进反向探测的 IPv6 邻居缓存保护方法

邻居缓存作为存储 IP-MAC 映射关系的关键载体,只有保证节点接收到 ND 报文时能够正确更新邻居缓存,才能确保数据传输的安全可靠。对此,本文提出了一种改进的反向探测邻居缓存保护方法。

2.1　基本思想

RD + 方法基本思想如图 2 所示,当目标节点收到 NS/NA 等报文时(图 2 中步骤(1)),首先创建一个新的表项,存储 IP-MAC 映射关系等信息(图 2 中步骤(2));之后,根据基于时间戳的队列管理算法(见第 3 节),向源节点发送 RD + NS 报文(图 2 中步骤(3));最后,目标节点若收到 RD + NA 报文(图 2 中步骤(4)),根据其是否满足更新条件(图 2 中步骤(5))进行相应操作。

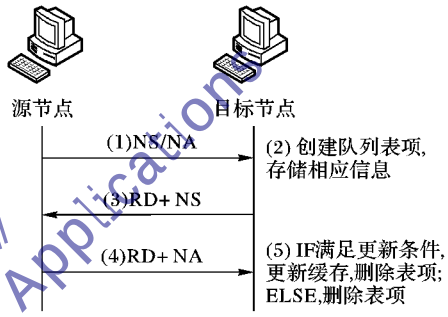

图 2　Reversed Detection Plus 基本思想

2.2　基本结构

节点队列和 RD + 报文是该方法的核心内容,其中:节点队列结构存储了 IP-MAC 映射及其他缓存更新条件;RD + 报文在现有协议的基础上,添加了一个 RD + Option 报文选项,可以很好地兼容现有协议。

2.2.1　节点队列结构

为了确保节点能够正确更新邻居缓存,每个节点建立一个 ND 报文信息存储队列(如图 3 所示)。节点队列结构各字段含义如下:IP 表示源节点的 IP 地址;MAC 表示源节点的 MAC 地址;Timestamp 表示向源节点发送 RD + NS 报文时的时间,初始化为 0;Sequence 表示向源节点发送 RD + NS 报文时的序号;Status 表示是否已向源节点发送 RD + NS 报文。其中 Status 字段的功能为:当攻击节点企图发送虚假 RD + 应答报文时,即使报文中的 Timestamp 字段和 Sequence 字段均合法,但是接收者的队列中并没有将相应记录的 Status 字段置位,那么接收者仍然不更新邻居缓存且删除此队列表项。

IP	MAC	Timestamp	Sequence	Status

图 3　节点队列结构

2.2.2　RD + 报文选项结构

为了保证提出的方法能够很好地兼容现有协议,设计了如图 4 所示的 RD + Option 报文选项。RD + Option 各字段含义如下:Type 表示选项类型值,用于标识 RD + 报文;Length 表示整个选项的长度(包括 Type 和 Length 字段);Sequence 表示 RD + 报文序号,用于响应报文的匹配;Reserved 表示保留字段;Timestamp 表示 RD + 报文时间戳,用于限制报文响应时长。其中,Sequence 字段和 Timestamp 字段结合使用以更好地抵抗欺骗攻击。

Type(8位)	Length(8位)	Sequence(16位)
Reserved(32位)		
Timestamp(64位)		

图 4 RD + Option 选项

2.3 基本流程

当目标节点收到 NS/NA 等报文时，首先，在队列中创建一个表项，记录收到 ND 报文的信息，如 IP-MAC 映射关系等；然后，等待任意时间 $t(0 < t < \tau)$，从队列头部选取一个记录，向源节点发送 RD + NS 报文，该报文包含 RD + Option 选项，并将相应记录的 Timestamp 字段和 Sequence 字段进行数据填充，且 Status 字段置位。当源节点发送 RD + NA 报文时，必须在选项中填充相同的 Sequence 字段及发送应答报文时的时间戳。

最后，目标节点对收到的 RD + NA 报文进行检查。首先，查询队列中是否有 Sequence 字段相同的记录；其次，检查队列中 Status 字段是否置位；再次，检验该 RD + NA 报文是否在定义的时间阈值 T 内到达。如果队列的该记录中各字段信息均合法且正确，那么将该记录的信息更新邻居缓存表并删除队列中的相应记录；否则，不更新邻居缓存表同时删除队列中相应记录。

另外，为了避免报文环路问题，当目标节点收到含有 RD + Option 选项的 NS 报文时，不允许该目标节点向源节点发送包含 RD + Option 选项的 NS 报文。

3 抗 DoS 攻击的队列管理算法

从 2.1 节可知，目标节点的队列是否安全直接影响 RD + 方法能否保护邻居缓存的安全更新，例如，攻击者不断发送大量虚假 NS/NA 报文，使目标节点一直处于满队列状态且队列中充满虚假信息，那么目标节点同样无法更新其邻居缓存，攻击者仍然可以达到对目标节点进行 DoS 攻击的目的，因此，队列管理是保证 RD + 方法稳定正确工作的关键之一。

3.1 队列管理技术分析

目前队列管理主要包括被动队列管理(Passive Queue Management, PQM)和主动队列管理(Active Queue Management, AQM)[9-11]。PQM 在缓冲区溢出时才丢包，而 AQM 根据缓冲区队列长度等信息以一定的概率丢包。

目前应用最广泛的 PQM 是弃尾(Drop Tail)队列管理，即随着队列缓冲区的溢出而丢包，其丢包是被动的，如果队列缓冲区不满，不会主动丢弃数据包。弃尾队列管理往往会保持较高的队列占用，队列满时，后来的数据包被全部丢弃。除了弃尾机制，另外两种 PQM 机制是随机丢弃(Random Drop)和弃头(Drop Front)机制。当队列满时，前者从队列中随机找出一个包丢弃以让新来的包进入队列；后者从队列头部丢包，以便让新包进入队列。但这两种机制都存在满队列和链路利用率低等问题[12-13]。

典型的随机早期检测(Random Early Detection, RED)[14]算法属于 AQM。RED 根据节点的平均队列长度 Q_{avg} 计算随队列增加而变大的概率 p，节点根据概率 p 来丢弃到达的数据包。如果平均队列长度小于最小阈值 Q_{min}，不丢包；如果平均队列长度介于最小阈值 Q_{min} 和最大阈值 Q_{max} 之间，则根据随队列增加而变大的概率 p 丢弃到达数据包；如果平均队列长度 Q_{avg} 大于最大阈值 Q_{max}，RED 丢弃所有到达的数据包。

$$p = \begin{cases} 0, & Q_{avg} < Q_{min} \\ \dfrac{Q_{avg} - Q_{min}}{Q_{max} - Q_{min}}, & Q_{min} \leqslant Q_{avg} \leqslant Q_{max} \\ 1, & Q_{avg} > Q_{max} \end{cases} \tag{1}$$

其中 Q_{max} 的值通常设置得比节点队列长度小得多，用来吸收突发数据流。虽然平均队列长度经过平滑计算得到，但如果持续的突发数据流到来，计算的平均队列长度 Q_{avg} 超过 Q_{max} 时，后来的数据包大量被丢弃，这时的 RED 实质上已经变成了弃尾队列管理。

3.2 基于时间戳的 RED 算法

现有的 RED 算法不能准确描述 RD + 方法的队列管理需求，本文提出了一种基于时间戳的 RED 算法 RED-T。在 RED 算法的基础上，通过引入报文时间戳对队列管理的影响，使得原始 RED 算法能够满足 RD + 邻居缓存保护方法的要求。

$$p = \begin{cases} 0, & Q_{avg} < Q_{min} \\ \dfrac{Q_{avg} - Q_{min}}{Q_{max} - Q_{min}} \times \dfrac{T_{current} - T_{timestamp}}{T}, & Q_{min} \leqslant Q_{avg} \leqslant Q_{max} \\ 1, & Q_{avg} > Q_{max} \end{cases} \tag{2}$$

其中：$T_{current}$ 是当前时间，$T_{timestamp}$ 是当前队列头部记录的 Timestamp 字段值，T 为定义的时间阈值。

RED-T 算法伪代码如下：

算法 Random Early Detection Based on Timestamp(RED-T)。

输入 队列最小长度 Q_{min}，队列最大长度 Q_{max}，当前时间 $T_{current}$，队列记录集合 $R = \{R_1, R_2, R_3, \cdots, R_{Q_{max}}\}$，队列头部记录时间戳 $T_{timestamp}$，超时时间阈值 T。

输出 数据包丢弃概率 p。

```
Start
 /* 每个队列记录启动计时器 */
 for each R_i ∈ R, do
   if R_i. timestamp > 0, SetTimer (T_i)
 end for
 /* 计时器超时且没有收到 RD + 应答报文 */
 if T_i TimeOut And No RD + Respond then
   Remove R_i from R
 endif
 /* 计算数据包丢包概率 */
 If Q_avg < Q_min then
   p = 0;
 eles if Q_avg < Q_max And Q_avg > Q_min, then
   p = (Q_avg - Q_min)/(Q_max - Q_min) × (T_current - T_timestamp)/T;
 else p = 1;
 endif
End
```

RED-T 算法主要分为两部分：

1)为队列中每个 Timestamp 字段非 0 的表项启动一个定时器，若定时器超时还没有收到相应的 RD + 应答报文，则删除队列表项；

2)根据队列平均长度 Q_{avg} 计算节点丢包概率，使得节点队列能够及时丢弃部分数据包，保证目标节点队列的健壮性。

RED-T 算法结果受 Q_{avg}、$T_{current}$、$T_{timestamp}$ 和 T 等因素的影响，同时，网络延时也会对目标节点接收报文产生影响，如果

网络延时较大,超过时间阈值 T(忽略节点处理 RD + 请求报文和发送 RD + 应答报文时间的情况下),那么目标节点将删除队列表项,就不能够正确更新邻居缓存,需要对时间阈值 T 作相应调整。

4　实验分析

本文提到的等待任意时间 t、时间阈值 T、最小队列长度 Q_{min} 和最大队列长度 Q_{max} 等应该根据不同网络的实际需求进行设定,本文选定 t 的上限 τ = 20 ms, T = 10 ms, Q_{min} = 100, Q_{max} = 500。

4.1　实验环境

为了验证 RD + 方法,采用如图 5 所示的交换式网络拓扑结构。

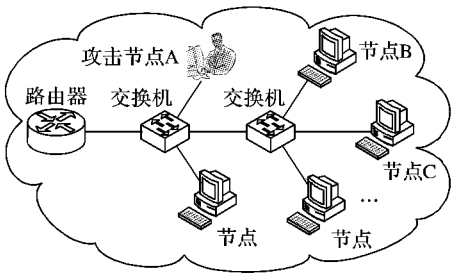

图 5　实验网络拓扑图

网络中各主机节点均安装 Ubuntu 11.04 操作系统,采用 Pentium Dual-Core 3.0 GHz 的 CPU,2 GB 的内存。同时,为了与 RD + 方法进行对比分析,本文还实现了文献[6]中的 Heuristic、Explicit 和 HE 等方法以及文献[7]中的 RD 方法。下面以攻击节点 A 和主机节点 B、C 为例进行实验,参数设置见表 1。

表 1　节点参数表

节点	IPv6 地址	MAC 地址
A	fe80::212:3fff:fe7c:9a5d	00-12-3f-7e-9a-5d
B	fe80::213:46ff:fe65:bc7d	00-13-46-65-bc-7d
C	fe80::20c:29ff:fee9:abd9	00-0c-29-e9-ab-d9

分析比较各类邻居缓存保护方法抵抗缓存欺骗攻击和 DoS 攻击的能力,并对 RD + 方法参数的选择及影响因子进行实验分析。

4.2　结果及分析

4.2.1　RD + 方法性能分析

从前面章节可知,RD + 方法与 RD 方法相比,增加了 16 字节的报文选项,引入的算法增加了一定的处理复杂度,为了对 RD + 方法的性能进行分析,对两种方法的 CPU 占用率和处理时间进行了测试,结果如图 6、7 所示。

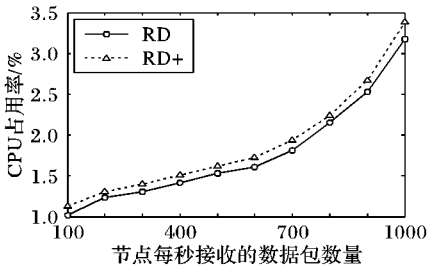

图 6　RD + 与 RD 方法 CPU 占用率比较

从图 6 可以看出,在节点接收数据包数量相同的情况下,RD + 方法比 RD 方法的 CPU 占用率要高,但是 CPU 占用率最多只高出 0.21%,可见 RD + 方法增加报文长度对节点 CPU 资源消耗的增加并不明显。

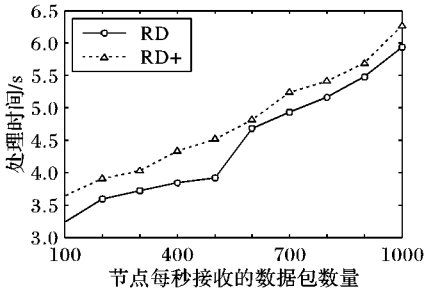

图 7　RD + 与 RD 方法处理时间比较

图 7 是 RD + 方法与 RD 方法在节点接收数据包数量相同的情况下,节点处理时间的对比分析。从图中可以看出,RD + 方法比 RD 方法的数据包处理时间要多,但是最多只高出了 0.6 s,可见 RD + 方法的处理复杂度并未明显降低协议性能。

RD + 方法利用较少的资源消耗及较小协议性能的牺牲换取邻居缓存的安全,是在可接受范围内的。

4.2.2　RD + 方法与其他方法的对比

首先,进行抗欺骗攻击能力测试。攻击节点 A 构造包含虚假信息的 NS 报文(IP_C-MAC_A),将其发送至节点 B,测试节点 B 分别部署 5 种不同的保护方法时抵抗欺骗攻击的能力,最后通过查看节点 B 的邻居缓存,总结 5 种保护方法抗欺骗攻击的能力。

然后,进行抗 DoS 攻击能力测试。攻击节点 A 分别以每秒发送 1 000、2 000、3 000 个数据包的速度构造并发送包含虚假信息的 NS 报文(主要伪造子网中不存在的 IP-MAC 映射),测试 5 种保护方法抵抗 DoS 攻击的能力。

测试结果如表 2 所示。从表中可以看出,HE 方法与 RD + 方法都能够抵抗邻居缓存欺骗攻击和 DoS 攻击。

表 2　各种邻居缓存保护方法抵抗攻击能力比较

保护方法	欺骗攻击	DoS 攻击
Heuristic	√	×
Explicit	×	√
HE	√	√
RD	×	×
RD +	√	√

其中:√表示可以抵抗此类攻击,×表示不能抵抗此类攻击。

另外,在目标节点每秒接收数据包数量相同的条件下,对两种方法的 CPU 占用率进行了比较,测试结果如图 8 所示。

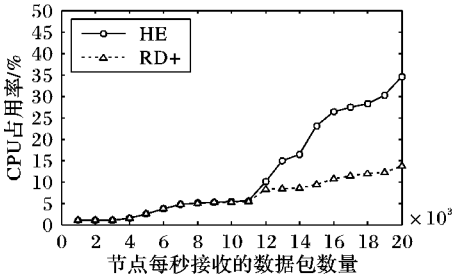

图 8　RD + 方法与 HE 方法 CPU 占用率比较

从图 8 中可以看出,在流量较小的情况下,RD + 方法与

HE 方法 CPU 占用率并没有明显区别,但是,随着流量的增加,RD + 方法的 CPU 占用率增加缓慢,HE 方法的 CPU 占用率增加较快,最大差值可达 20.87%,证明 RD + 方法资源消耗更少,大流量情况下优势更加明显。因为 RD + 方法采用自定义队列结构对收到的 ND 数据包进行存储,并采用基于时间戳的 RED 算法对其进行管理筛选,使得实际需要处理的数据包数量远小于 HE 方法,而 HE 方法需要对所有收到的数据包进行逐一处理,致使数据包数量过大时,HE 方法占用的 CPU 时间比 RD + 方法要多。

本文采用了 RED-T 算法,并对该算法与原始 RED 算法进行了比较,测试结果如图 9 所示。

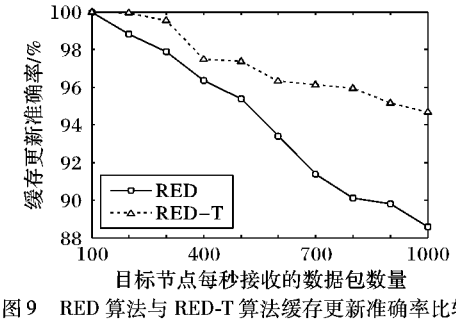

图 9 RED 算法与 RED-T 算法缓存更新准确率比较

图 9 显示了两种算法对 RD + 方法准确率的影响,从图中可以看出,采用 RED-T 算法比 RED 算法准确率要高,且最多高出 6.07%。因为 RED-T 算法引入了报文时间戳的影响,由 3.2 节式(2)可知,RED-T 算法不仅考虑了队列长度等因素的影响,还充分考虑了响应报文时间戳等因素,一方面提高了抗欺骗攻击能力,另一方面提高了缓存更新准确率。

4.2.3 RD + 方法与 SEND 协议比较

RD + 方法与 SEND 协议均能保证邻居发现报文的安全,但是 SEND 协议引入的 CGA 和 RSA(Rivest-Shamir-Adieman)数字签名等机制使得计算开销不容忽视。为了证明 RD + 方法的优势,对两种方法的 CPU 占用率进行了比较分析。本文采用 SEND 协议的一种实现方法 NDProtector 进行测试,测试结果如图 10 所示。

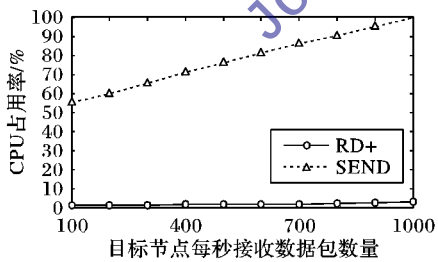

图 10 RD + 与 SEND CPU 资源消耗比较

从图 10 中可以发现,SEND 协议在进行 CGA 地址认证和 RSA 签名认证时,需要消耗的 CPU 资源非常大,甚至能造成 DoS 攻击,而 RD + 方法消耗的 CPU 资源则可以忽略不计。这也是 SEND 协议未能广泛部署使用的原因之一。

5 结语

本文主要针对 IPv6 邻居发现协议中邻居缓存容易遭受欺骗攻击和 DoS 攻击的问题,提出一种基于改进反向探测的邻居缓存保护方法。同时,本文也完成了该方法的报文结构、数据结构及基于时间戳的队列管理算法设计,并对该方法的性能、功能及其优势进行了测试和对比分析。实验结果表明,该方法不仅能够有效保护邻居缓存,而且其资源消耗比 HE 方法和 SEND 协议都少。本文方法尚存在一些不足之处:本文设计的 RED-T 算法虽然能够有效提高邻居缓存更新准确率,但是尚不能保证准确率达到 100%。在今后的工作中将致力于进一步研究队列管理算法,根据不同网络环境选择算法的合适参数,使得缓存更新准确率进一步提高。

参考文献:

[1] NARTEN T, NORDMARK E, SIMPSON W, *et al*. RFC 4861, Neighbor Discovery for IP version 6 (IPv6) [S]. Geneva: IETF, 2007.

[2] HUANG M. The IPv6 technology research of the next-generation Internet [D]. Nanjing: Nanjing University of Posts and Telecommunications, 2013. (黄明超. 下一代互联网 IPv6 技术的研究[D]. 南京: 南京邮电大学, 2013.)

[3] ZHAO Y. Research of security in IPv6 transition phase [D]. Beijing: Beijing Jiaotong University, 2013. (赵延. IPv6 过渡阶段中的安全研究[D]. 北京: 北京交通大学, 2013.)

[4] ARKKO J, ERICSSON ED, KEMPF J, *et al*. RFC 3971, Secure Neighbor Discovery (SEND) [S]. Geneva: IETF, 2005.

[5] AURA T. RFC 3972, Cryptographically Generated Address (CGA) [S]. Geneva: IETF, 2005.

[6] KITAMURA H, ATA S, MURATA M. IPv6 neighbor cache update [EB/OL]. (2009 - 10 - 19) [2013 - 05 - 12]. http://tools.ietf.org/id/draft-kitamura-ipv6-neighbor-cache-update-00.txt.

[7] JIANG S, CHEN X, SONG X. Neighbor cache protection in neighbor discovery protocol [EB/OL]. (2010 - 03 - 02) [2013 - 04 - 15]. http://tools.ietf.org/html/draft-jiang-v6ops-nc-protection-01.

[8] GASHINSKY I, JAEGGLI J, KUMARI W. RFC 6583, Operational neighbor discovery problems [S]. Geneva: IETF, 2012.

[9] ZHANG L. The research of priority checking RED algorithm based on IPv6 network [D]. Changchun: Jilin University, 2012. (张黎丽. 基于 IPv6 网络的优先级检测 RED 算法的研究[D]. 长春: 吉林大学, 2012.)

[10] LIU X. Active queue management algorithm research based on fuzzy model [D]. Nanjing: Nanjing University of Science and Technology, 2013. (刘雪梅. 基于模糊控制理论的主动队列管理算法研究[D]. 南京: 南京理工大学, 2013.)

[11] LI R. Active queue management algorithm research based on robust control [D]. Nanjing: Nanjing University of Science and Technology, 2013. (李睿. 基于鲁棒控制的主动队列管理算法研究[D]. 南京: 南京理工大学, 2013.)

[12] YIN X, Research on congestion control mechanism over heterogeneous networks [D]. Hangzhou: Zhejiang University, 2013. (尹翔. 异构网络环境下拥塞控制方法研究[D]. 杭州: 浙江大学, 2013.)

[13] WANG J, LIN B. Active queue management algorithm based on particle swarm optimization [J]. Journal of Computer Applications, 2013, 33(2): 390 - 396. (王军祥, 林柏钢. 基于粒子群优化的主动队列管理方法[J]. 计算机应用, 2013, 33(2): 390 - 396.)

[14] YANG X, HE W. Adaptive nonlinear RED algorithm based on routing queue resources [J]. Journal of Computer Applications, 2013, 33(3): 621 - 624. (杨晓亚, 何万生. 基于路由队列资源自适应的非线性随机早起检测算法[J]. 计算机应用, 2013, 33(3): 621 - 624.)