



基于滑动地址窗口的 IPv6 动态地址隧道模型

马子川, 黄小红, 闫 岫, 张 沛

(北京邮电大学网络技术研究院 北京 100876)

摘 要: IPv6 网络地址跳变传输技术能够有效阻止网络攻击者通过流量分析方式获知通信双方的行为特征。对相关技术进行了研究, 将动态地址链的概念应用于 IPv6 网络地址跳变传输技术, 提出了基于滑动地址窗口的 IPv6 动态地址隧道模型并开发了相应的原型系统。通过对原型系统的各项指标进行测试和分析, 证明该模型能够达到设计目标。

关键词: 网络地址跳变; 动态地址链; 滑动地址窗口; 动态地址隧道

doi: 10.11959/j.issn.1000-0801.2015231

IPv6 Dynamic Address Tunnel Model Based on the Sliding Address Window

Ma Zichuan, Huang Xiaohong, Yan Shen, Zhang Pei

(Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: IPv6 network address hopping transmission technology can effectively prevent network attacks through traffic analysis. On the basis of the study of some related technologies, the concept of dynamic address chain was applied to the IPv6 network address hopping transmission technology and an IPv6 dynamic address tunnel model based on the sliding address window was proposed. The testing result and analysis show that the model can achieve the design goals.

Key words: network address hopping, dynamic address chain, sliding address window, dynamic address tunnel

1 引言

随着 IPv6 应用的不断推广和普及, 其安全性问题也逐渐得到人们的关注。目前 IPv6 网络普遍使用无状态地址配置 (stateless address autoconfiguration, SLAAC) 机制实

现 IPv6 主机的地址自动配置, 简化了 IPv6 地址的配置方式。但是这种机制也给网络攻击者可乘之机, 网络攻击者只要对特定的 IPv6 地址的流量进行监听, 就能够收集到特定主机的全部 IPv6 流量^[1]。

然而现有的对网络层以上的数据安全保护机制所采

收稿日期: 2015-06-26; 修回日期: 2015-09-08

基金项目: 国家国际科技合作与交流专项项目 (No.2013DFE13130), 国家科技支撑计划基金资助项目 (No.2015BAH04F02), “北京市支持中央高校共建项目——青年英才计划”项目专项资助课题

Foundation Items: The National International Technology Cooperation and Exchanges in Special Projects (No.2013DFE13130), The National Science & Technology Pillar Program (No.2015BAH04F02), “Beijing Pillar Program for the Central Universities-Youth Talent Plan” Fund

论文引用格式: 马子川, 黄小红, 闫岫等. 基于滑动地址窗口的 IPv6 动态地址隧道模型. 电信科学, 2015231

Ma Z C, Huang X H, Yan S, et al. IPv6 dynamic address tunnel model based on the sliding address window. Telecommunications Science, 2015231



用的手段主要是对传输的数据内容进行加密处理,无法对传输双方的“信道”进行保护。攻击者只要对特定 IPv6 地址所对应的“信道”中的完整 IPv6 流量进行分析,就能够从流量特征中获知所使用的网络协议、操作系统甚至物理位置等信息,并对此分析得到用户身份,达到窃取用户隐私数据的目的。

为了应对这个问题,网络地址跳变传输技术应运而生,网络地址跳变传输技术通过将数据流量分散到多对 IP 地址所构成的多个“信道”中进行传输,达到隐藏双方真实 IP 地址,对双方的传输“信道”进行隐藏,进而保护传输内容不被监听的目的。由于 IPv6 有着巨大地址空间的天然优势,在 IPv6 网络中使用网络地址跳变传输技术,传输双方可以使用足够多的动态 IPv6 地址建立动态地址连接,可以将监听者通过流量分析窃取信息的可能降到最低。

国内外对 IPv6 网络地址跳变传输机制及其相关技术已经有过一些研究,并且提出了多种 IPv6 网络地址跳变传输方案以及类似的其他方案。参考文献[2]提出一种基于地址和端口进行跳变传输的方案,该方案支持多个客户端间的地址跳变传输,并通过使用时间戳的方式计算所需的动态地址及端口,但这种方式依赖同一个跳变同步服务器对各个客户端设备进行跳变同步。参考文献[3]通过将主机的流量分散到不同的链路之中来完成地址的跳变传输,但同样依赖中心节点对主机的动态地址进行配置和管理。参考文献[4,5]提出的方案通过对 IPv6 地址进行动态混淆,从而得到动态 IPv6 地址进而达到跳变传输的目的,但依赖精确的时钟进行同步和更新。参考文献[6,7]针对时钟同步的需求提出了相应分布式时间戳同步方案。参考文献[8]中的方案使用共享伪随机种子的方式生成多组动态 IPv6 地址,并另辟蹊径使用分组计数器的方式触发动态 IPv6 地

址的更新与切换。

由此可以看出,由于 IPv6 动态地址跳变传输要求通信双方均使用动态地址,因而对动态地址的更新进行同步是动态地址跳变传输机制中非常重要的组成部分,也是目前相关研究的重点所在。

本文针对 IPv6 网络地址跳变传输机制的跳变同步技术进行研究和探索,提出了动态地址链(dynamic address chain)的概念,并将滑动地址窗口和动态地址链应用于 IPv6 网络地址跳变传输,能够在保证 IPv6 网络地址高效跳变的同时降低对同步时钟精度的要求,并在此基础上提出了基于滑动地址窗口的 IPv6 动态地址隧道(IPv6 dynamic address tunnel, 6DAT)模型。

2 6DAT 模型

2.1 6DAT 模型结构

6DAT 模型的基本思路是通过在本地子网的边界处部署 6DAT 网关,子网中的所有主机通过 6DAT 网关连接到互联网。同时,由 6DAT 网关将子网中主机的 IP 数据分组封装于使用动态 IPv6 地址作为源地址和目的地址的隧道数据分组中,从而达到通过动态 IPv6 地址建立动态地址隧道进行跳变传输的目的。

6DAT 模型的基本结构如图 1 所示,子网中的主机通过网关连接到互联网,当需要使用 6DAT 进行传输时,主机的 IP 数据分组在本地网关处被封装为隧道数据分组然后发送到目标网关,目标网关在接收到隧道数据分组之后对其进行解析,然后将原始数据分组转发到目标子网中的目标主机上。

6DAT 网关是由动态地址链所驱动的网络设备,如图 2 所示,可分为动态地址层和数据转发层两个部分。其

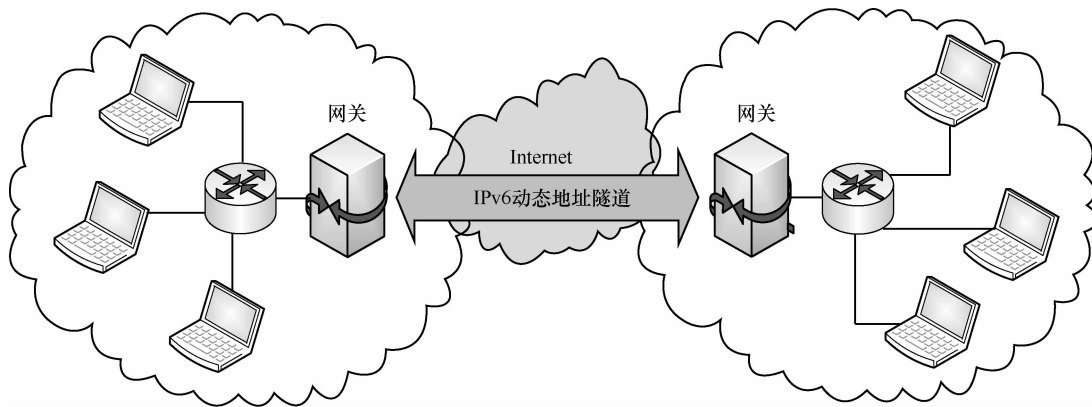


图 1 6DAT 组成结构

中动态地址层又包括本地动态地址链和对端动态地址链，分别为数据转发层提供动态源地址和动态目的地址。数据转发层提供了面向主机的接口和面向互联网的接口，对来自主机的数据分组进行封装，并对来自互联网的数据分组进行解析。

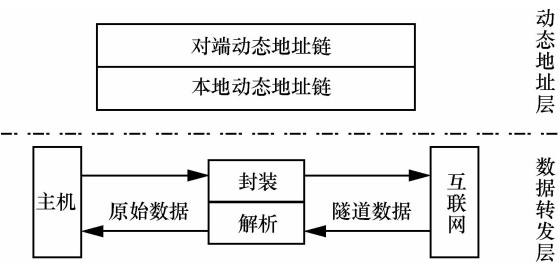


图 2 6DAT 网关架构

2.2 动态地址链

动态地址链是由多个动态 IPv6 地址组成的有序的线性动态 IPv6 地址池。每个 6DAT 网关都包含一个本地动态地址链和至少一个对端动态地址链，6DAT 网关必须在初始化本地动态地址链和对端动态地址链之后才能和对端网关建立动态地址隧道。

表 1 中展示了一个动态地址链的模型，其中每个动态 IPv6 地址有各自对应的序号，各个动态 IPv6 地址在动态地址链中始终保持有序排列，并按照一定的规则在动态地址链上划分出发送地址窗口和接收地址窗口，分别驱动 6DAT 的封装模块和解析模块。

表 1 IPv6 动态地址链

序号	动态 IPv6 地址
0	2001:470:1f04:3cf::2
1	2001:470:1f04:3cf::4
2	2001:470:1f04:3cf::6
3	2001:470:1f04:3cf::8
4	2001:470:1f04:3cf::a
5	2001:470:1f04:3cf::c

2.2.1 动态地址链同步机制

由于建立动态地址隧道的两个对等网关需要独立计算并更新各自使用的动态地址链，若要保证动态地址以及动态地址隧道的持续有效，要求对等网关的动态地址链必须有保持同步的能力。

6DAT 模型所采用的动态地址链同步机制要求分布于不同网关的各个动态地址链拥有相同的更新周期。每隔一个更新周期，各个网关分别计算出需要添加的动态地址序号，并根据动态地址序号计算得到动态地址并添加到各自

的动态地址链中。因而只要约定一个共同的起始时间戳 $Time_{initial}$ ，就能根据当前的时间戳 $Time_{current}$ 以及周期 T 计算得到当前需要添加的地址序号。

$$Index_{add}=Round\left(\frac{Time_{current}-Time_{initial}}{T}\right) \tag{1}$$

采用了上述的地址序号算法之后，并不要求两个对等网关的时钟完全精确一致，只要之间的时间戳误差小于周期 T ，就能够同步计算得到相同的地址序号，完成同步操作，极大降低了对时钟精度的要求。

2.2.2 动态地址算法

根据动态地址链同步机制，动态地址算法必须满足：根据确定的网关地址前缀和地址序号，必须能够唯一计算得到一个确定的动态 IPv6 地址。

根据上述要求，可以提出一个适用于 6DAT 模型的动态地址算法。给定网关前缀 $Prefix$ 以及地址序号 N ，就可以计算得到动态 IPv6 地址。

$$Addr=H(Prefix,N)=Prefix\|Hash(Prefix+N) \tag{2}$$

由于网关前缀和地址序号确定且唯一，该算法能够计算得到确定且唯一的动态 IPv6 地址。

2.3 动态地址窗口

动态地址窗口指的是在动态地址链上特定的动态 IPv6 地址区间，包括发送地址窗口和接收地址窗口。

每个动态地址链都有对应的发送地址窗口和接收地址窗口，本地动态地址链对应着本地发送窗口和本地接收窗口，对端动态地址链对应着对端发送窗口和对端接收窗口。

2.3.1 发送地址窗口

根据地址窗口算法，动态地址链上按照发送窗口容量划分出特定的 IPv6 地址区间作为发送地址窗口。在建立动态地址隧道时，需要从本地发送地址窗口和对端发送窗口中随机选择一对动态地址，分别作为动态地址隧道数据分组的源地址和目的地址。

2.3.2 接收地址窗口

相应的，动态地址链上按照接收窗口容量划分出特定的 IPv6 地址区间作为接收地址窗口。在接收到动态地址隧道数据分组时，需要检查数据分组的源地址和目的地址是否分别位于对端接收地址窗口和本地接收地址窗口，然后再进行后续处理和转发。

2.3.3 地址窗口算法

发送地址窗口和接收地址窗口的特性决定了发送地



址窗口的容量必须小于接收地址窗口的容量,即发送地址窗口是接收地址窗口的子集。

并且,为了保证发送网关所选择的发送地址窗口中的动态 IPv6 地址尽可能落在接收网关的接收地址窗口中,发送地址窗口必须尽可能接近接收地址窗口的中间位置,从而可以得到用于计算发送地址窗口和接收地址窗口的地址窗口算法。

当接收地址窗口起始序号为 N 以及接收窗口和发送窗口容量分别为 Win_R 和 Win_S 时,接收地址窗口为:

$$[N, N+Win_R-1] \quad (3)$$

相应的发送地址窗口为:

$$\left[N + \frac{Win_R - Win_S}{2}, N + Win_R - 1 - \frac{Win_R - Win_S}{2} \right] \quad (4)$$

以图 3 所示的动态地址链为例,接收地址窗口是一个容量为 16 的动态 IPv6 地址区间,动态地址序号范围为 0~15。发送地址窗口是一个容量为 8 的动态 IPv6 地址区间,动态地址序号范围为 4~11。

2.3.4 地址窗口滑动

由于发送地址窗口和接收地址窗口均为有限长度的地址区间,为了保证源地址和目的地址的动态性,发送窗口和接收窗口必须定时进行滑动更新。地址窗口滑动情况如图 4 所示。

按照动态地址链的同步机制,每个动态地址链都有各

自的更新周期,每隔一个周期,都将计算得到一个新的动态地址并添加到动态地址链的末端。与此同时,如果将发送地址窗口和接收地址窗口向地址序号增加的方向移动一个地址空间,就可以完成对发送地址窗口和接收地址窗口的滑动更新操作。

由于发送地址窗口和接收地址窗口均按照一定的周期在动态链上进行滑动更新,其动态更新的平滑性就能够得到保证,能够最大限度地避免因动态地址切换而带来的分组丢失等问题。

然而,虽然要求各个网关的动态地址窗口尽力保持同步,但由于不可避免的时钟精度误差的存在,总是会有动态地址窗口不一致的情况发生。试想一种情况,由于发送地址窗口没有精确同步,发送端网关从其发送地址窗口中选取了一对动态地址作为隧道数据分组的源地址和动态地址,但该动态地址对没有位于接收端网关的发送地址窗口中,如图 5 所示。此时由于接收端网关使用接收地址窗口进行动态地址对验证,只要发送端网关的发送地址窗口分组含于接收端网关的接收地址窗口,就能通过动态地址验证。6DAT 模型中将发送地址窗口和接收地址窗口进行分离式设计思想正是基于这样的考虑。

3 6DAT 基本工作过程

假设网关 A 和网关 B 需要建立动态地址隧道。在网关 A

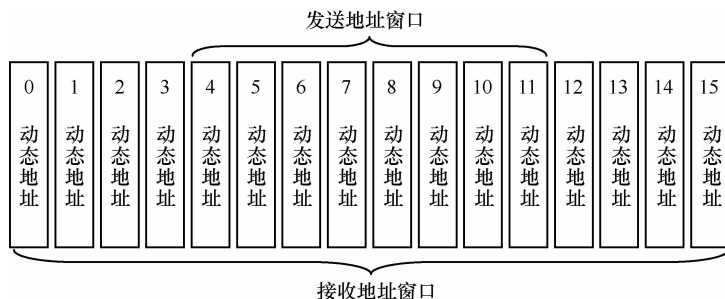


图 3 发送地址窗口和接收地址窗口

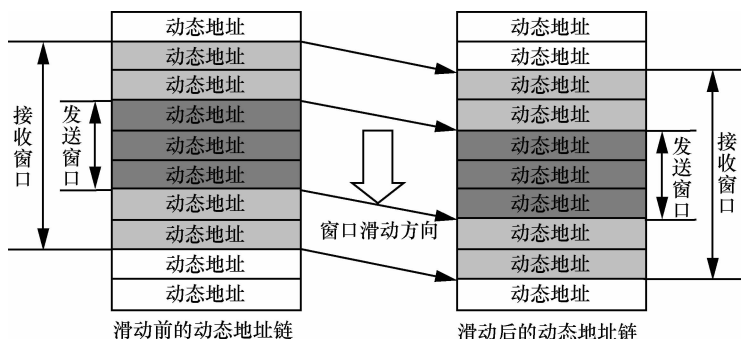


图 4 地址窗口滑动

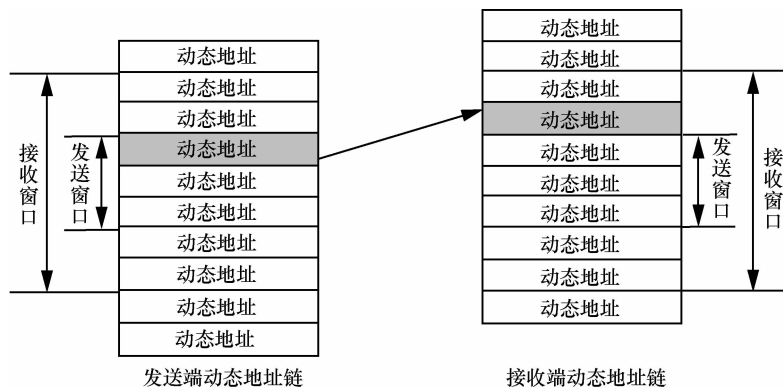


图 5 动态地址窗口非同步状态

和网关 B 之间建立动态地址隧道时,各自所需的动态 IPv6 地址来源于动态地址链 A 和动态地址链 B。为了使网关间的动态地址链保持同步,需要共享表 2 的网关配置信息。

表 2 网关配置信息

网关配置	网关 A	网关 B
地址前缀	$Prefix_A$	$Prefix_B$
发送窗口容量	Win_{SA}	Win_{SB}
接收窗口容量	Win_{RA}	Win_{RB}
窗口滑动周期	T_A	T_B

上述网关间共享配置信息可以通过一定的机制进行带内配置或者带外配置。考虑到上述配置信息并不需要频繁更新,若采用带内配置会导致额外的开销以及潜在的安全性问题,因而采用静态的带外配置方式。

3.1 初始化

由于动态地址链 A 和 B 初始状态均为空,需要先对动态地址链进行动态地址填充,由于发送地址窗口分组含于接收地址窗口,因而需要填充的动态地址数量至少为:

$$Size(Chain_A) = \max(Win_{RA}, Win_{SA}) = Win_{RA} \quad (5)$$

$$Size(Chain_B) = \max(Win_{RB}, Win_{SB}) = Win_{RB} \quad (6)$$

将上述步骤得到的两组动态 IPv6 地址分别添加到动态地址链 A 和 B 中:

$$Addr_{i=1 \rightarrow Win_{RA}} = H(Prefix_A, i) \quad (7)$$

$$Addr_{j=1 \rightarrow Win_{RB}} = H(Prefix_B, j) \quad (8)$$

3.2 发送数据

初始化完成后,根据地址窗口算法,动态地址链 A 和动态地址链 B 的初始发送窗口分别如下。

发送窗口 A [$Start_{SA}$, End_{SA}]:

$$Start_{SA} = \frac{Win_{RA} - Win_{SA}}{2} \quad (9)$$

$$End_{SA} = Win_{RA} - 1 - \frac{Win_{RA} - Win_{SA}}{2} \quad (10)$$

发送窗口 B [$Start_{SB}$, End_{SB}]:

$$Start_{SB} = \frac{Win_{RB} - Win_{SB}}{2} \quad (11)$$

$$End_{SB} = Win_{RB} - 1 - \frac{Win_{RB} - Win_{SB}}{2} \quad (12)$$

当网关 A 所在子网的主机向网关 B 所在子网的主机发送数据时,网关 A 需要分别从发送窗口 A 和发送窗口 B 中随机选取一对动态 IPv6 地址作为源地址和目的地址。

$$Addr_{src} = Chain_A[Start_{SA}, End_{SA}].random() \quad (13)$$

$$Addr_{dst} = Chain_B[Start_{SB}, End_{SB}].random() \quad (14)$$

如图 6 所示,来自主机的数据分组先在网关 A 处添加 6DAT 分组头,封装成以上述动态地址为源地址和目的地址的 6DAT 数据分组,然后再发送出去。

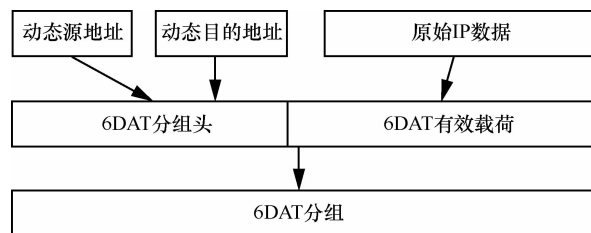


图 6 封装流程

同理,当网关 B 所在子网的主机需要向网关 A 所在子网的主机发送数据时,网关 B 需要分别从发送窗口 B 和发送窗口 A 中随机选取一对动态 IPv6 地址作为源地址和目的地址。

$$Addr_{src} = Chain_B[Start_{SB}, End_{SB}].random() \quad (15)$$

$$Addr_{dst} = Chain_A[Start_{SA}, End_{SA}].random() \quad (16)$$

来自主机的数据分组先在网关 B 处添加 6DAT 分组头,封装成以上述动态地址作为源地址和目的地址的 6DAT 数据分组,然后再发送出去。



3.3 接收数据

初始化完成后,根据地址窗口算法,动态地址链 A 和动态地址链 B 的初始接收窗口分别如下。

接收窗口 A[$Start_{RA}, End_{RA}$]:

$$Start_{RA}=0 \quad (17)$$

$$End_{RA}=Win_{RA}-1 \quad (18)$$

接收窗口 B[$Start_{RB}, End_{RB}$]:

$$Start_{RB}=0 \quad (19)$$

$$End_{RB}=Win_{RB}-1 \quad (20)$$

当网关 B 接收到来自网关 A 的隧道数据分组时,需要分别检查隧道数据分组的源地址和目的地址是否分别位于接收窗口 A 和接收窗口 B 之中,如果是,则对其进行解析并去除隧道数据分组头以获得原始数据分组,然后将其转发至网关 B 所在子网中对应的主机。

$$Chain_A[Start_{RA}, End_{RA}].validate(Addr_{src}) \quad (21)$$

$$Chain_B[Start_{RB}, End_{RB}].validate(Addr_{dst}) \quad (22)$$

相应的,当网关 A 在接收到来自网关 B 的隧道数据分组时,需要分别检查隧道数据分组的源地址和目的地址是否位于接收窗口 B 和接收窗口 A 之中,如果是,则对其进行解析并去除隧道数据分组头以获得原始数据分组,然后其转发至网关 A 所在子网中对应的主机。

$$Chain_B[Start_{RB}, End_{RB}].validate(Addr_{src}) \quad (23)$$

$$Chain_A[Start_{RA}, End_{RA}].validate(Addr_{dst}) \quad (24)$$

4 模型性能分析

为了对 6DAT 模型进行验证,开发了 6DAT 的原型系统,并对核心功能进行了实现。该 6DAT 原型系统采用 Go 语言开发,使用 Netfilter Queue 并结合 Linux 系统上的 iptables 完成对 IPv6 数据分组的捕获和处理。部署该 6DAT 原型系统时使用了如表 3 所示的实验环境配置。

表 3 实验环境配置

6DAT 网关配置	网关 A	网关 B
地址前缀	2001:470:1f04:3cf::/64	2001:470:c:242::/64
接收窗口容量	8	8
发送窗口容量	4	4
窗口滑动周期/s	10	10

为了对 6DAT 模型的部署进行模拟,搭建了如图 7 所示的测试环境。在两台具有双网卡的服务器上部署了上述 6DAT 原型系统作为 6DAT 网关,同时连接另外两台计算机作为 6DAT 主机,并对其进行相关的 IPv6 地址和路由的配置。

4.1 抗时钟误差能力

由于 6DAT 模型仍然需要时钟信号驱动动态地址链和滑动地址窗口进行更新,而作为一个分布式系统,实际应用中难免有时钟的精度误差。通过测量不同时钟误差情况下系统的分组丢失率,对 6DAT 系统的抗时钟误差能力进行评估。

仍然选择发送 100 个 ping 报文的方式对分组丢失率进行测试,调整两个 6DAT 网关之间的系统时钟误差,记录发送 100 个 ping 报文时的分组丢失率。测试结果如图 8 所示。

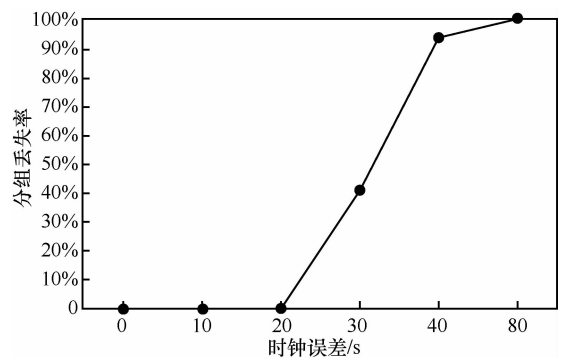


图 8 分组丢失率与时钟误差

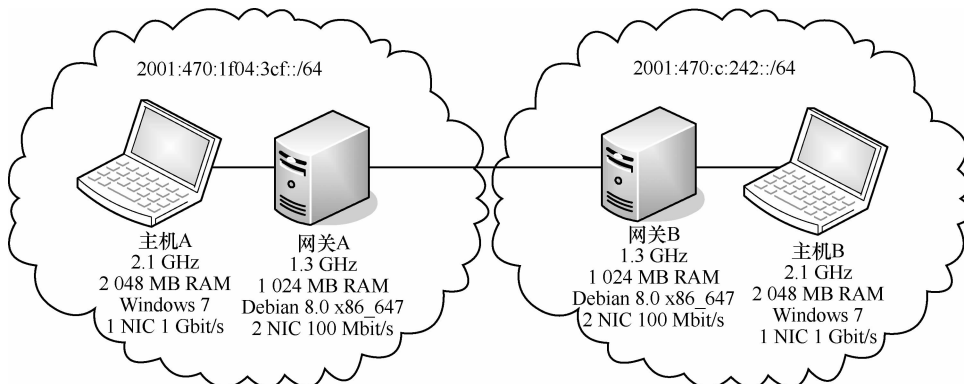


图 7 6DAT 原型系统测试环境

根据动态地址链的同步机制以及地址窗口算法不难得到,若要避免分组丢失,则必须保证发送地址窗口的动态地址全部落入接收地址窗口之中,即发送端的发送地址窗口的边界在接收端的接收地址边界以内。据此,可以推算得到最大能够允许的时钟误差为:

$$Time_{deviation} = \frac{Win_R - Win_S}{2} \cdot T \tag{25}$$

在本次实验中,接收地址窗口为 8,发送地址窗口为 4,窗口滑动周期为 10 s,根据式(25)计算可得最大理论时钟允许误差为 20 s。而实验结果显示,当时钟误差大于 20 s 时系统分组丢失率急剧上升,也验证了推算正确的。

4.2 流量分散能力

将发送单位个数的 IP 数据分组时所使用的“信道”数量,即唯一 IP 地址对数量,定义为系统的流量分散能力,并且使用该指标评估 6DAT 系统对恶意流量分析的抵抗能力。

$$D = \frac{count_{channel}}{count_{packet}} = \frac{count_{ip-pair}}{count_{packet}} \tag{26}$$

当 6DAT 并未启用时,发送端和接收端均使用静态的 IP 地址,即自始至终只采用一对 IP 地址进行传输。在这种情况下,流量分散能力与 IP 数据分组的数量成反比,IP 数据分组越多则流量分散能力越弱,也就越容易遭受恶意流量分析攻击。

在开启 6DAT 之后,使用 TCPDUMP 工具对上述实验的 IP 数据分组进行抓取和记录,并对其进行分析,统计其中唯一的 IP 地址对数量,得到表 4 的结果。

表 4 IP 地址对数量统计

报文数量/个	IP 地址对数量	流量分散能力(IP 地址对/报文数量)
100	49	0.490
500	257	0.514
1 000	504	0.504

此外,由于动态 IP 地址对的数量受发送地址窗口的影响,因而选择在发送 100 个报文的前提下,测试 6DAT 原型系统使用不同的发送地址窗口容量时的流量分散能力,测试数据如图 9 所示。

由上述实验结果可以看出,在启用 6DAT 之后,主机之间的流量被分散到大量 IP 地址对所组成的“信道”中。且当发送窗口容量越大,则可用的动态 IPv6 地址越多,动态 IPv6 地址对也就越多,流量分散能力越强。若攻击者对流量进行截获和分析,获得全部的完整数据的难度也就会越大,也就

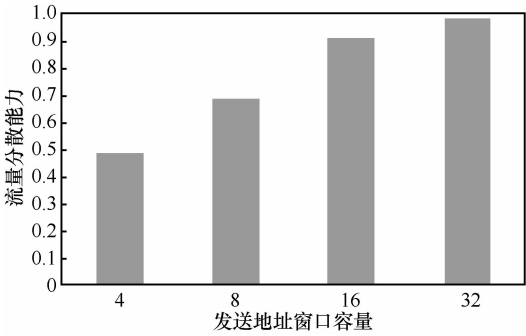


图 9 发送窗口容量与流量分散能力

越难分析得到通信双方的真实身份以及通信方式。

4.3 时延和分组丢失

为了了解系统的时延和分组丢失情况,采用在两个主机间发送 ping 报文的方式进行测试。在开启网关的 6DAT 功能的情况下,从一台主机向另外一台主机分别发送 100、500 以及 1 000 个 ping 报文,分别记录响应时间和分组丢失情况。之后,在关闭网关的 6DAT 功能的情况下,重复上述实验,分别再次记录响应时间和分组丢失状况,实验结果见表 5。

表 5 ping 响应时间

	开启 6DAT			关闭 6DAT		
报文数量	100	500	1 000	100	500	1 000
最小 RTT/ms	0.604	0.659	0.655	0.369	0.279	0.270
最大 RTT/ms	27.338	29.113	28.549	1.480	7.308	3.140
平均 RTT/ms	1.479	1.412	1.343	0.618	0.563	0.573
标准差/ms	2.611	2.003	1.778	0.151	0.337	0.167
分组丢失率	0	0	0	0	0	0

从表 5 中不难发现,开启 6DAT 之后,由于存在对 IP 数据分组的封装和解析操作,6DAT 对 IP 数据分组的转发存在一定的时延。对于大部分的 IP 数据分组,在进行转发的时候会增加约 1 ms 的时延,但也有极少部分的 IP 数据分组在转发的时候产生了较大的时延。总体来看,6DAT 网关对 IP 数据分组的操作所带来的时延并没有给使用过程带来较大的困扰,但仍然存在优化的空间。

此外,根据测试结果,该 6DAT 原型系统在分组丢失率方面表现良好,并没有表现出严重的分组丢失情况。

5 结束语

网络地址跳变技术能够将通信双方的流量分散到多组 IP 地址所构成的信道中进行传输,能够有效防止攻击者通过流量分析的方式获知用户行为,窃取用户信息。由



于 IPv6 拥有巨大的地址空间,能够提供足够多的动态地址用于网络地址跳变,因而 IPv6 技术的发展与普及为网络地址跳变传输技术的发展提供了机遇。

本文从 IPv6 地址跳变传输的跳变同步技术这一技术入手,对现有的相关技术进行总结和改进,提出了动态地址链的概念,并通过使用动态地址链将滑动窗口应用于 IPv6 跳变地址传输技术,在此基础上提出了基于滑动地址窗口的 IPv6 动态地址隧道模型。IPv6 动态地址隧道模型能够将流量分散到多对动态 IPv6 地址所组成的信道中进行传输,能够抵抗恶意的流量分析行为。此外,由于采用了动态地址链和滑动地址窗口机制,能够大大降低对时钟同步的精确度依赖。通过对 IPv6 动态地址隧道模型的原型系统开发和测试,证明该原型系统在时延、分组丢失率、时钟精度要求以及流量分散能力方面能够达到模型的设计目标。在未来的工作中,将会对该模型的加密模式进行研究,并对该模型的关键算法进行优化,在保证模型的安全性的同时优化性能以应对应用中的各种实际场景。

参考文献

- 1 Stephen G, Matthew D, Randy M, *et al.* IPv6: nowhere to run, nowhere to hide. Proceedings of the 44th Hawaii International Conference on System Sciences(HICSS), Kauai, USA, 2011
- 2 Shi L, Jia C, LV S, *et al.* Port and address hopping for active cyber-defense. Proceedings of the Conference on Intelligence and Security Informatics, Chengdu, China, 2007
- 3 刘慧生,王振兴,郭毅.一种基于多穴跳变的 IPv6 主动防御模型. 电子与信息学报, 2012, 34(7): 1715~1720
Li H S, Wang Z X, Guo Y. An IPv6 proactive network defense model based on multi-homing hopping. Journal of Electronics & Information Technology, 2012, 34(7): 1715~1720
- 4 Stephen G, Matthew D, Randy M, *et al.* MT6D: a moving target IPv6 defense. Proceedings of the Military Communications Conference, Baltimore, USA, 2011
- 5 Stephen G, Matthew D, Randy M, *et al.* Implementing an IPv6 moving target defense on a live network. Proceedings of the National Symposium on Moving Target Research, Annapolis, USA, 2012
- 6 Lin K, Jia C F, Weng C. Distributed timestamp synchronization for end hopping. China Communications, 2011, 8(4): 164~169
- 7 Lin K, Jia C F, Shi L Y. Improvement of distributed timestamp synchronization. Journal of China Institute of Communications, 2012, 33(10): 110~116
- 8 Sifalakis M, Schmid S, Hutchison D. Network address hopping: a mechanism to enhance data protection for packet communications. Proceedings of the 40th Annual IEEE International Conference on Communications (IEEE ICC), Seoul, Korea, 2005

[作者简介]



马子川,男,北京邮电大学网络技术研究
院硕士生,主要研究方向为下一代互联网。



黄小红,女,博士,北京邮电大学网络技
术研究院副教授,主要研究方向为下一代互
联网关键技术。



闫岫,男,博士,北京邮电大学网络技
术研究院在站博士后,主要研究方向为移
动性管理。



张沛,男,博士,北京邮电大学网络技
术研究院讲师,主要研究方向为网络空间安全。