软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2011,22(9):2166−2181 [doi: 10.3724/SP.J.1001.2011.03854]

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

基于 Fluid 的大规模带宽限制蠕虫仿真模型*

聂晓峰+, 荆继武, 王跃武, 向 继

(中国科学院 研究生院 信息安全国家重点实验室,北京 100049)

Fluid-Based Large-Scale Bandwidth-Limited Worm Simulation Model

NIE Xiao-Feng+, JING Ji-Wu, WANG Yue-Wu, XIANG Ji

(State Key Laboratory of Information Security, Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)

+ Corresponding author: E-mail: xfnie@is.ac.cn

Nie XF, Jing JW, Wang YW, Xiang J. Fluid-Based large-scale bandwidth-limited worm simulation model. *Journal of Software*, 2011,22(9):2166−2181. http://www.jos.org.cn/1000-9825/3854.htm

Abstract: In this paper, based on the subnet abstraction mechanism, a fluid-based large-scale bandwidth-limited worm simulation model is proposed. In this model, the fluid simulation paradigm is leveraged to take away the huge volume of scanning traffic to reduce the requirements of computational capability and memory usage. Through extensive comparisons with the packet-level worm simulation and the measurement datasets, experiments demonstrate that the proposed simulation model is capable of high fidelity and low resource consumption, and thus, can fulfill the needs of the analysis of bandwidth-limited worm propagation characteristics and the verification of worm defense strategies.

Key words: bandwidth-limited worm; model; Fluid simulation; ripple effect; multi-level fidelity-preserving

摘 要: 在利用子网抽象技术的基础上,进一步针对带宽限制蠕虫高速扫描的特点,提出了一种基于 Fluid 的大规模带宽限制蠕虫仿真模型.通过 Fluid 仿真技术对蠕虫高速扫描产生的数据包进行抽象,降低其对仿真系统计算能力和存储能力的要求,进而提高仿真执行效率.仿真结果和数据包级仿真以及和实测数据的对比表明,该仿真模型可以在消耗较少资源的情况下具有较高的仿真保真度,能够满足带宽限制蠕虫传播特性分析和防御策略验证的需求.

关键词: 带宽限制蠕虫;模型;Fluid 仿真;涟漪效应;多级仿真保真度

中图法分类号: TP309 **文献标识码**: A

带宽限制蠕虫(bandwidth-limited worm)是目前已经实现的传播速度最快的一类蠕虫.UTC 时间 2003 年 1 月 26 日 05:30 爆发了 Slammer 蠕虫,在 10 分钟内迅速感染了大约 75 000 台脆弱主机[1].PST 时间 2004 年 3 月 19 日 08:45 爆发了 Witty 蠕虫,在大约 45 分钟内感染了 Internet 中的大部分脆弱主机[2].Witty 蠕虫的爆发时间距离所利用漏洞的发布时间相差不到 24 小时,标志着“0day”攻击的真实来临.带宽限制蠕虫的快速传播特性,使其成为 Internet 安全的一个重要威胁.因此,研究其传播特性,设计、实现相应的检测防御策略对于保障 Internet 安全具有重要的实际意义[3].

* 基金项目: 国家自然科学基金(60573015, 70890084/G021102, 61003273); 国家高技术研究发展计划(863) (2006AA01Z454); 中国博士后科学基金(20080440072)

收稿时间: 2009-05-07; 修改时间: 2009-11-26; 定稿时间: 2010-03-15

基于安全性和实验成本的考虑,蠕虫仿真实验是理解蠕虫传播特性和对蠕虫防御策略进行验证分析的主要手段.首先,由于蠕虫传播的巨大危害性和不可操控性,在真实的网络环境中进行实验可能会引起巨大灾难;其次,网络蠕虫是发生在 Internet 上的大规模网络安全事件,场景规模超出了大多数研究机构可提供的真实实验环境的能力.带宽限制蠕虫为了实现快速传播,通常具有极高的扫描速率.在带宽限制蠕虫传播过程中,极高的扫描速率所产生的巨大扫描流量引发了严重的网络拥塞.带宽限制蠕虫所表现的这些特性对现有的蠕虫仿真模型提出了严峻挑战.

作为一种常用的蠕虫仿真模型,解析模型使用一组微分方程(如 Two-Factor 模型[4])或者一组离散迭代表达式(如 AAWP 模型[5])描述蠕虫传播的动态过程.解析模型最大的优点在于计算的有效性,通常用于蠕虫传播特性的宏观分析.但是,解析模型通常把所有实体当作同构实体统一处理,无法针对某个实体做特殊处理.此外,由于解析模型中过多的假设,解析模型难以对蠕虫传播的复杂过程进行有效刻画.带宽限制蠕虫传播的一个显著特点是:大量的扫描数据包将引起严重的网络拥塞,并且该拥塞是一个复杂的动态过程.解析模型由于对蠕虫传播过程中的网络环境进行了过分的简化,无法对此动态过程进行高准确度的模拟,影响了分析效果.

相对于数学解析模型,数据包级仿真能够更为准确地反映蠕虫传播过程中的网络特性变化,可以提供高精度的蠕虫传播模拟[6].但数据包级蠕虫仿真要消耗大量的系统资源,难以进行仿真规模扩展.而网络蠕虫是在 Internet 范围内传播的,仿真系统必须具备一定的仿真规模,才能正确、完整地反映蠕虫传播的特性.此外,带宽限制蠕虫还具有高速扫描速率的特点.测量数据显示,在 Slammer 蠕虫爆发过程中,被感染主机的最高扫描速率高达 26000 scans/sec/worm;在 Witty 蠕虫爆发过程中,尽管被感染主机的最高扫描速率低于 Slammer 蠕虫下的最大速率,也达到了 9700 scans/sec/worm.如此高速率扫描产生的数以亿计的数据包给仿真系统的计算能力和存储能力带来了严峻挑战.随着网络带宽的加大,蠕虫的扫描率会进一步提高,进一步加剧了对仿真系统计算能力和存储能力的要求.尽管并行离散事件仿真(PDES)技术可以在一定程度上通过划分网络空间提高网络仿真的尺度规模,但该技术不适用于大流量仿真.这就要求要有一种更好的抽象机制对蠕虫高速扫描产生的大量数据包进行处理,节约计算资源.

在利用子网抽象技术解决蠕虫仿真尺度规模的基础上[7],本文进一步针对带宽限制蠕虫高速扫描的特点,利用 Fluid 仿真技术对蠕虫数据包流量进行抽象,降低其对仿真系统计算和存储能力的要求,进而提高仿真执行效率.Fluid 仿真技术把离散发送的数据包看成连续流动的 Fluid,不再发送任何数据包,而只是在 Fluid 流速发生变化时,发送一个表示流速大小的流速声明(fluid advertisement).通过 Fluid 仿真技术对网络流量进行抽象,可以有效降低仿真离散事件的数量,实现仿真规模的有效扩展.

本文首先提出了基于 Fluid 仿真技术的大规模带宽限制蠕虫仿真模型,并对 Fluid 仿真技术在带宽限制蠕虫仿真应用中可能存在的问题进行了讨论,并提出了相应的优化措施和解决方案.随后,就数据包级仿真和 Fluid 仿真结合实现多级仿真保真度的仿真模型进行了分析.最后,在 Georgia Tech 的开源网络仿真器 GTNetS[8] 上实现了一个完整的仿真系统,对仿真模型的性能和精度进行了分析,并利用该系统实现了 Witty 蠕虫的传播模拟.本文的主要贡献包括:① 提出了一个基于 Fluid 的大规模带宽限制蠕虫仿真模型,解决了大规模带宽限制蠕虫仿真中仿真规模和仿真保真度之间的矛盾,并根据带宽限制蠕虫特点对 Fluid 仿真技术在带宽限制蠕虫仿真中的应用提出了若干优化措施,减少 Fluid 模型下由于涟漪效应和 Fluid 状态信息维护而导致仿真性能降低和额外内存开销;② 在基于 Fluid 的大规模带宽限制蠕虫仿真模型中引入了 Packet 模型,实现 Fluid 仿真和数据包级仿真的结合,为多级仿真保真度模型下的蠕虫防御策略验证奠定了基础;③ 从系统仿真时间、内存开销和已处理离散事件量等方面对仿真模型的性能进行了测试分析,并为利用该模型开展相关研究提供了方法指导.

本文第 1 节描述蠕虫仿真相关研究工作.第 2 节描述带宽限制蠕虫传播特性.第 3 节描述基于 Fluid 的带宽限制蠕虫仿真模型构建,以及 Fluid 模型与 Packet 模型的结合.第 4 节通过仿真实验对仿真模型的性能和精度进行分析.第 5 节用该模型实现 Witty 蠕虫的仿真分析.最后是全文总结和工作展望.

1 相关工作介绍

如何在仿真规模和仿真保真度之间寻找一个合适的平衡点,一直是网络蠕虫仿真研究面临的一个重要问题.解决网络仿真规模和仿真保真度之间的矛盾存在 3 条途径[9]:(1) 为增加仿真系统运行平台的计算能力.该方向的主要工作为并行分布式仿真;(2) 改进仿真算法.对于基于离散事件的网络仿真,仿真技术改进的主要工作为采用新的算法加速离散事件队列处理,如采用日历队列算法等;(3) 仿真模型改进.其主要思路是,通过改变仿真模型抽象层次,进行仿真规模和仿真保真度的转换.提高模型抽象层次,可以降低仿真保真度,提高仿真规模;降低模型抽象层次,可以提高仿真保真度,但仿真规模随之降低,选择合适的抽象模型可以实现仿真规模和仿真保真度的统一.相对于前两条途径,改变仿真模型抽象层次,可以更为有效地实现仿真规模和仿真保真度的调节.

针对蠕虫仿真建模,有不同的抽象方法.解析模型建模方法具有较高的抽象层次,所以其仿真规模的可扩展性好,但是仿真保真度难以保障,而且不适合进行防御策略的验证分析.数据包级蠕虫仿真模型具有较低的抽象层次,所以其仿真保真度较好,但是仿真规模却难以扩展.带宽限制蠕虫作为网络蠕虫的一个特例,还需要额外关注极高的扫描速率以及子网带宽限制问题.Kesidis 等人通过对 Kermack-McKendrick 蠕虫传播模型进行扩展,提出了在同构子网下的带宽限制蠕虫传播解析模型[10].Riley 等人在 GTNetS 上构建了一个数据包级网络蠕虫仿真系统.通过采用一系列优化措施,实现了对数据包级网络蠕虫仿真尺度规模的提升;但实验部分对带宽限制蠕虫的扫描率仅设置为 5,10,20 scans/sec/worm,严重偏离了实际的攻击数值[6].Vlachos 等人在基于离散时间的蒙特卡洛网络蠕虫仿真中,利用 Bandwidth-Aware 图模拟带宽限制蠕虫传播中的网络拥塞现象,并在每个时间步长变换具有相同特性的 Bandwidth-Aware 图,以提高仿真精度[11].这些方案都没有很好地解决带宽限制蠕虫仿真中仿真规模和仿真保真度之间的矛盾.

Fluid 仿真是一种针对高带宽网络仿真的网络流量抽象技术.在 Fluid 仿真模式下,网络流量被抽象成为持续流动的 Fluid,而不是单独的数据包.Fluid 模型根据实现形式不同,可以分为解析 Fluid 模型和离散事件 Fluid 模型两种.解析 Fluid 模型使用一组微分方程描述 Fluid 行为,在每个固定时间间隔,依据 Fluid 流速的变化和队列状态信息产生一组微分方程,多次迭代至收敛后,获得更新后的所有 Fluid 和队列状态信息[12].Nicol 利用解析 Fluid 模型进行了 Internet 骨干网络上的蠕虫流量模拟[13].离散事件 Fluid 模型则将流速更新消息抽象成一个 Fluid 流速声明,并将这个流速声明沿路由信息发送到下一跳队列,依次更新整个系统的 Fluid 队列状态[14].相对于数据包级仿真,Fluid 仿真可以有效地减少离散事件的产生数量,从而节约计算资源,提高仿真规模,特别是在实现大流量网络仿真.同时,由于 Fluid 仿真保留了节点队列信息,所以可以很好地模拟网络流量的动态拥塞过程.此外,由于离散事件 Fluid 模型和数据包均基于离散事件仿真推动机制,相对于解析 Fluid 模型,离散事件 Fluid 模型更容易与数据包仿真模型相结合,实现具有多级仿真保真度的仿真模型.基于离散事件 Fluid 模型的优点,本文将基于离散事件 Fluid 模型构建带宽限制蠕虫仿真模型.本文以后的 Fluid 模型均指离散事件 Fluid 模型.

2 带宽限制蠕虫传播特性

与其他主动扫描蠕虫一样,带宽限制蠕虫传播过程中,被感染主机通过发送扫描数据包发现脆弱主机,进而感染该脆弱主机.该脆弱主机被成功感染后,将重复被感染主机动作进行下一次感染.被感染主机发送扫描数据包的速率越高,单位时间内新增的被感染主机数将会越多,蠕虫传播速度也将更快.

为了提高扫描速率,带宽限制蠕虫在代码设计方面一般具有如下 3 个特点:① 带宽限制蠕虫采用无连接的传输层协议,如 UDP 协议,这样被感染主机不需要等待目标主机的连接请求回应,可以尽可能快地发送蠕虫数据包;② 带宽限制蠕虫具有较小的载荷,在带宽一定的情况下,可以提高被感染主机单位时间内蠕虫数据包的发送数量,如 Slammer 蠕虫的数据包大小仅为 376Byte,Witty 蠕虫的数据包大小在 796Byte~1 307Byte 之间.而非带宽限制蠕虫通常具有较大的载荷,如 Code Red 蠕虫的大小为 8KB,Nimda 蠕虫的大小更是达到了 60KB;

③ 为了减少蠕虫代码量,带宽限制蠕虫一般采用简单的扫描策略,如 32 位地址空间完全随机扫描策略.

如前所述,在带宽限制蠕虫爆发过程中,被感染主机被观测到具有极高的扫描速率.但是由于每个子网与 Internet 之间的连接带宽是有限的,被感染主机的扫描速率又极高,少量被感染主机发送的蠕虫数据包即可很快消耗完其所在子网与 Internet 之间全部的连接带宽.当连接带宽消耗殆尽时,将会发生严重的拥塞.此时,即使子网内的被感染主机数量继续增加,该子网发送到 Internet 的扫描数据包数量将不再会有显著地增加,而只有成功发送到 Internet 的蠕虫数据包才能够有效地感染其他脆弱主机.所以在带宽限制蠕虫传播过程中,随着拥塞的不断加剧,被感染主机发送到 Internet 的有效扫描速率将呈现出不规则的快速下降趋势.这类蠕虫因此也称为带宽限制蠕虫.

Weaver 等人对 Slammer 蠕虫和 Witty 蠕虫传播过程中被感染主机的平均有效扫描速率进行了分析,如图 1 所示[15].可以看出,随着蠕虫传播,Slammer 蠕虫和 Witty 蠕虫的平均有效扫描速率急剧降低,并且伴随着大幅度的振荡.其中,Witty 蠕虫的平均扫描速率变化幅度较 Slammer 小,其主要原因是 Witty 蠕虫曲线没有包括变化幅度最大的前 1.5 分钟的测量数据.此外,由于 Witty 蠕虫数据包比 Slammer 蠕虫要大,其扫描数据包速率基数也低于 Slammer 蠕虫.但是两者平均有效扫描速率呈现明显相似的变化趋势.

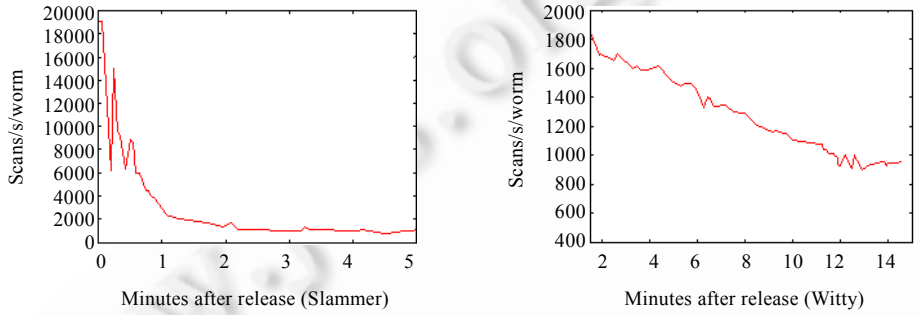


Fig.1 Effective scan rate per infective in the propagation of Slammer and Witty

图 1 Slammer 蠕虫和 Witty 蠕虫传播过程中平均有效扫描速率的变化

带宽限制蠕虫传播过程中平均有效扫描率的动态变化过程可以用数学模型简单表示:假设 Internet 模型由 N 个子网组成,每个子网包含 s 台脆弱主机,子网带宽为 c.被感染主机的扫描率为 δ scans/s/worm,每个蠕虫数据包的大小为 m.由于带宽限制蠕虫采用 32 位地址空间完全随机扫描策略,则扫描空间的大小 $\Omega=2^{32}$.

由于子网外出链路带宽限制,t 时刻子网 i 成功发送到 Internet 中的蠕虫数据包数量可以表示为

$$Scan_i(t)=\begin{cases} Inf_i(t)\cdot\delta, & \text{if } Inf_i(t)\cdot\delta<\xi \\ \xi, & \text{otherwise} \end{cases},\ \xi=\frac{c}{m}, i=1,2,...,N \tag{1}$$

其中,$Inf_i(t)$表示 t 时刻子网 i 中被感染主机的数量,$0\leqslant Inf_i(t)\leqslant s$.

t 时刻发往 Internet 的总有效扫描率 $Scan(t)=\sum_{i=1}^{N}Scan_i(t)$,$t$ 时刻被感染主机的总量 $Inf(t)=\sum_{i=1}^{N}Inf_i(t)$,子网 i 中被感染主机数量的变化可以表示为

$$\frac{dInf_i(t)}{dt}=Scan(t)\cdot\frac{1}{\Omega}\cdot(s-Inf_i(t)),\ i=1,2,...,N \tag{2}$$

则 t 时刻被感染主机的平均有效扫描率为

$$\bar{\delta}(t)\equiv\frac{Scan(t)}{Inf(t)}=\frac{\sum_{i=1}^{N}Scan_i(t)}{\sum_{i=1}^{N}Inf_i(t)} \tag{3}$$

从以上分析可以看出,极高的扫描速率以及巨额扫描流量由于子网带宽限制而引发的严重网络拥塞,是被感染主机平均有效扫描率动态变化、急剧降低的主要原因,也是带宽限制蠕虫在传播过程中区别于其他蠕虫的显著特性.因此,在带宽限制蠕虫仿真中,必须针对这两个特性进行特别处理.数据包级仿真可以精确地模拟网

络拥塞过程,但无法解决巨额扫描流量的资源消耗问题.而 Fluid 仿真可以有效地兼顾对这两个特性的模拟.

3 基于 Fluid 的带宽限制蠕虫仿真模型构建

本节首先介绍基于 Fluid 的带宽限制蠕虫仿真模型构建;随后对 Fluid 仿真技术在带宽限制蠕虫仿真应用中可能存在的问题进行了讨论,并提出了相应的优化措施和解决方案;最后,在基于 Fluid 的大规模带宽限制蠕虫仿真模型中引入了 Packet 模型,实现基于 Fluid 仿真和数据包级仿真结合的多级仿真保真度模型.

3.1 基于Fluid的带宽限制蠕虫仿真模型构建

根据利用子网抽象技术解决网络蠕虫仿真尺度规模的思想,基于 Fluid 的带宽限制蠕虫仿真模型将网络环境抽象成两大部分:包含多个路由器的 Internet 骨干网络模型和包含脆弱主机的多个子网模型.两者之间用具有特定带宽的链路模型连接,如图 2 所示.所有的脆弱主机分布在各个子网中.子网可以接受来自 Internet 骨干网络的扫描流量,并模拟子网中脆弱主机的感染过程.一旦子网中的脆弱主机被感染,子网将向 Internet 骨干网络发送扫描流量.子网和 Internet 骨干网络模型之间的蠕虫流量以及蠕虫流量在 Internet 骨干网络模型内不同路由器之间的转发,用 Fluid 模型进行模拟.通过将具体的脆弱主机抽象到一个子网中,我们可以降低流速声明的发送数量,并且能够通过子网和 Internet 骨干网络模型之间的 Fluid 队列,模拟蠕虫传播造成的动态拥塞过程.

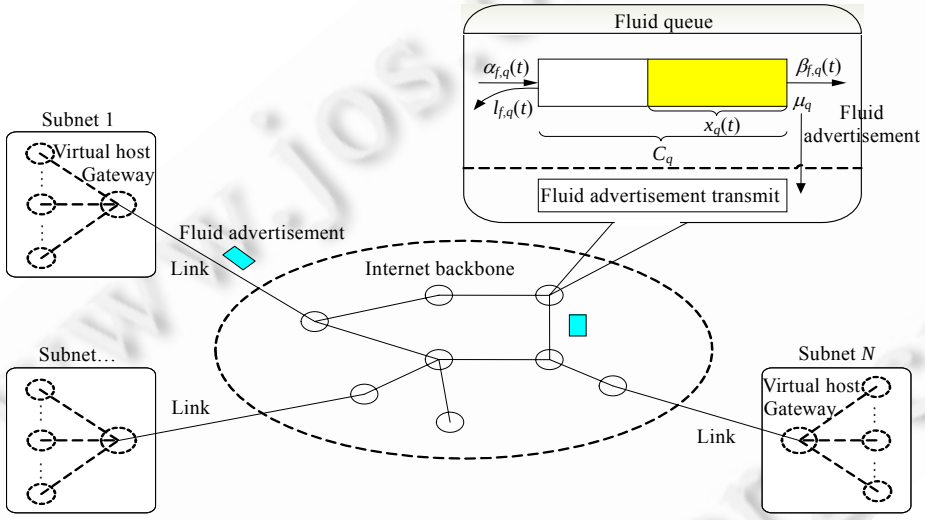


Fig.2 Framework of the fluid-based bandwidth-limited worm simulation model

图 2 基于 Fluid 的带宽限制蠕虫仿真模型框架

在 Fluid 模型中,每个网络接口对应一个 Fluid 队列.Fluid 队列(fluid queue)维持经过该网络接口的每个 Fluid 流(fluid flow)的状态,包括输入速率、输出速率和丢弃速率.我们把 t 时刻经过队列 q 的 Fluid 流 f 的进入速率、离开速率和丢弃速率分别表示为 $\alpha_{f,q}(t)$,$\beta_{f,q}(t)$和 $l_{f,q}(t)$.队列缓冲区维持的其他状态还包括所有 Fluid 流的集中输入速率 $\alpha_q(t)=\sum_{f\in F_q}\alpha_{f,q}(t)$、集中输出速率 $\beta_q(t)=\sum_{f\in F_q}\beta_{f,q}(t)$、集中丢弃速率 $l_q(t)=\sum_{f\in F_q}l_{f,q}(t)$、队列缓冲区使用量 $x_q(t)$、队列缓冲区容量 C_q 以及缓冲区服务率 μ_q.其中,F_q 是经过 Fluid 队列 q 的所有 Fluid 流的集合,缓冲区服务率 μ_q 等于队列所属接口对应的外出链路的带宽.

Fluid 模型用流速声明更新各个 Fluid 队列的状态信息.当 t 时刻,一个新的流速声明到达 Fluid 队列的时候,根据新到达的流速声明计算队列缓冲区使用量、经过该队列的每个 Fluid 流的输入速率、输出速率和丢弃速率.缓冲区使用量根据下式计算:

$$x_q(t)=x_q(t_l)+(\sum_{f\in F_q}\alpha_{f,q}(t_l)-\mu_q)\times(t-t_l)=x_q(t_l)+(\alpha_q(t_l)-\mu_q)\times(t-t_l) \quad (4)$$

其中,t_l 是上一个流速声明到达的时刻.如果根据式(4)计算出来的队列 q 的缓冲区使用量 $x_q(t)<0$,把 $x_q(t)$设置为 0;如果计算出来的缓冲区使用量 $x_q(t)>0$,则把 $x_q(t)$设置为 C_q.然后,t 时刻经过队列 q 的所有 Fluid 流 f 的离开速率和丢弃速率按照如下规则计算:

(1) 假如$\alpha_q(t)<\mu_q$并且$x_q(t)=0$,即所有 Fluid 流的集中输入速率小于队列服务率并且队列为空,则此时不存在数据包丢弃,集中丢弃速率 $l_q(t)=0$,每个 Fluid 流的离开速率等于其进入速率,$\beta_{f,q}(t)=\alpha_{f,q}(t)$,$l_{f,q}(t)=0$;

(2) 假如$\alpha_q(t)<\mu_q$并且 $0<x_q(t)\leqslant C_q$,即所有 Fluid 流的集中输入速率小于队列服务率,队列非空,但是队列缓冲区使用量逐渐减小,则此时集中输出速率等于队列服务率$\beta_q(t)=\mu_q$,集中丢弃速率 $l_q(t)=0$,每个 Fluid 流的离开速率等于按其所在集中输入速率的比值均分集中输出速率,$\beta_{f,q}(t)=\alpha_{f,q}(t)/\alpha_q(t)\times\mu_q$,$l_{f,q}(t)=0$;

(3) 假如$\alpha_q(t)\geqslant\mu_q$并且 $0\leqslant x_q(t)<C_q$,即所有 Fluid 流的集中输入速率大于等于队列服务率,队列非空,但还未超过最大容量.则此时集中输出速率等于队列服务率$\beta_q(t)=\mu_q$,集中丢弃速率 $l_q(t)=0$,每个 Fluid 流的离开速率等于按其所在集中输入速率的比值均分集中输出速率,$\beta_{f,q}(t)=\alpha_{f,q}(t)/\alpha_q(t)\times\mu_q$,$l_{f,q}(t)=0$;

(4) 假如$\alpha_q(t)\geqslant\mu_q$并且 $x_q(t)=C_q$,即所有 Fluid 流的集中输入速率大于等于队列服务率,队列已充满.则此时集中输出速率等于队列服务率$\beta_q(t)=\mu_q$,集中丢弃速率 $l_q(t)=\alpha_q(t)-\mu_q$,每个 Fluid 流的离开速率等于按其所在集中输入速率的比值均分集中输出速率,$\beta_{f,q}(t)=\alpha_{f,q}(t)/\alpha_q(t)\times\mu_q$,$l_{f,q}(t)=\alpha_{f,q}(t)/\alpha_q(t)\times l(t)$.

如果某个 Fluid 流的输出速率发生变化,则新创建一个流速声明,并根据网络连接信息,将流速声明发送到下游各个相关 Fluid 队列,更新队列信息.流速声明到达下一跳的时间可以根据排队时间、传输时间和在链路上的传播时间计算获得.采用 Fluid 模型,可以在不产生很多网络事件的情况下很好地模拟网络流量的延迟、拥塞和丢包率,从而可以实现较为精确的带宽限制蠕虫传播过程模拟.

图 2 中的子网模型,根据其收到的流速声明推断出其他子网扫描到该子网的蠕虫数据包的到达时刻.假设单位时间内数据包的到达个数符合 Poisson 分布,即数据包到达时间间隔符合指数分布,则子网 i 发出的、到达子网 j 的蠕虫数据包的时间间隔可以根据如下公式计算:

$$t_{i,j}\sim Exponential(\omega/\alpha_{i,j}) \tag{5}$$

其中,ω为蠕虫数据包的平均大小,$\alpha_{i,j}$为子网 i 发给子网 j 的流速声明的大小.蠕虫数据包的源地址可以根据源子网内被感染主机的地址随机生成,目的地址可以根据目的子网地址空间随机生成.

在子网内部,仅保留脆弱主机的配置信息和状态信息.当根据公式(5)从 Fluid 流产生蠕虫数据包后,再根据所产生蠕虫数据包的目的地址,对子网内脆弱主机的状态进行更新.如果具有该目的地址的脆弱主机存在,则感染该脆弱主机,将其状态变为被感染状态,然后根据该主机的扫描率更新该主机所在子网扫描到其他子网的蠕虫流速.产生的流速声明将通过 Internet 骨干网络模型,逐一传播到各个相关 Fluid 队列,更新 Fluid 队列状态信息,最后到达各个目的子网.子网 i 发给子网 j 的流速声明的大小与子网 i 中被感染主机的数量的关系为

$$\alpha_{i,j}(t)=\sum_{k\in I_i(t)}\delta_{i,k}\cdot N_j\cdot\omega/\Omega \tag{6}$$

其中,Ω表示蠕虫扫描空间大小,N_j表示子网 j 的地址空间大小,$I_i(t)$为子网 i 中被感染主机的集合,$\delta_{i,k}$表示子网 i 中被感染主机 k 的扫描率.子网 i 中被感染的主机发给仿真场景中不存在的子网的流速声明大小如下(这些流量将根据路由信息在边界路由器处丢弃):

$$\alpha_{i,\Lambda}(t)=\sum_{k\in I_i(t)}\delta_{i,k}\cdot(\Omega-\sum_j N_j)\cdot\omega/\Omega \tag{7}$$

Internet 骨干网络模型由包含一系列路由器的传输(transit)子网组成.每个路由器包含 Fluid 队列和路由信息,Fluid 队列维护经过该路由器的流状态信息,模拟蠕虫流量经过该路由器的延时和丢包率,路由信息用于转发流速声明.当一个子网中的脆弱主机状态发生改变时,子网发出流速声明更新该子网到其他子网的蠕虫流速,流速声明经过 Internet 骨干网络模型转发到相应目的子网.当一个路由器收到一个流速声明后,将根据流速声明更新自己的 Fluid 队列信息,同时根据路由信息将流速声明信息转发到相应的下游路由器.下游路由器重复这样的动作,直到所有的相关路由器的流状态信息都得到更新.这样,整个 Internet 的网络流量就得到了更新.根据此前对 Fluid 模型的分析,Fluid 队列的缓冲区使用量 $x_q(t)$可以模拟蠕虫流量经过传输子网路由器的延时.当 $x_q(t)$

等于 Fluid 队列的容量 C_q 时,Fluid 队列可以模拟 Internet 骨干网络的拥塞.

Internet 骨干网络模型和子网模型之间的 Fluid 队列可以模拟两者之间的动态拥塞过程.Fluid 队列的输入 $\alpha_q(t)$会随着被感染主机数量的增加而增大.当超过 Fluid 队列的服务率μ_q 时,Fluid 队列的缓冲区使用量 $x_q(t)$就会逐渐增加.当 $x_q(t)=C_q$ 时,持续增加的蠕虫流量导致 Fluid 队列的丢弃率 $l_q(t)$就开始增加,从而模拟了蠕虫流量由于子网带宽限制而导致的动态拥塞过程.

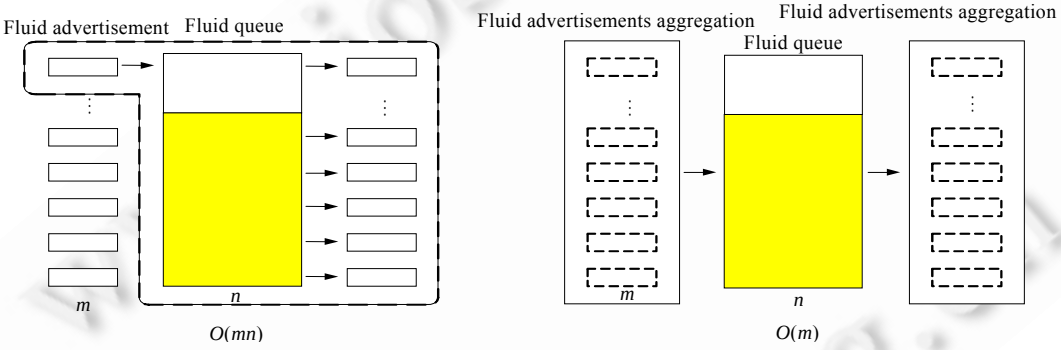
3.2 针对带宽限制蠕虫的Fluid模型优化机制

3.2.1 下一跳地址相同流速声明聚合和 On-Demand 拆分

从第 3.1 节可以看到,一个 Fluid 流输入速率的改变,可能引起经过同一队列的其他流输出速率改变,速率发生改变的 Fluid 流将更新后的流速向下游传播,在下游队列中引起更多的 Fluid 流输出速率改变.这个流速改变传播和放大的过程称为涟漪效应(ripple effect).涟漪效应对 Fluid 仿真的效率具有重大影响.

当一个子网内的主机状态发生变化后,如脆弱主机由易感染状态变成被感染状态,会同时更新该子网到其他子网的蠕虫流速.由于离散事件仿真的特性,与这些流速声明相关的离散事件会被依次处理;而且当这些依次处理的流速声明先后经过同一个非空 Fluid 队列的时候,每个流速声明都会导致经过这个队列的其他 Fluid 流的流速发生改变,流速发生改变的 Fluid 流亦将发送流速更新声明到下游 Fluid 队列,如此短时间内不断的流速改变将引起严重的涟漪效应.

通过把下一跳地址相同的 Fluid 流速声明集中聚合发送,把经过最大相同路由的流速声明集中处理,不仅减少了流速声明的数量,而且降低了由于在短时间内不断依次更新非空 Fluid 队列造成的涟漪效应,如图 3 所示.



(a) Without fluid advertisements aggregation (a) 不使用流速声明聚合

(b) With fluid advertisements aggregation (b) 使用流速声明聚合

Fig.3 Effect of merge optimization of fluid advertisements

图 3 流速声明聚合优化效果

算法 1 和算法 2 分别描述了如何根据流速声明的下一跳(next-hop)地址对子网发出的流速声明进行聚合以及如何在网络接口处对接收到的流速声明聚合执行 On-Demand 拆分.

算法 1. 根据流速声明 $f_{i,j}$ 的下一跳地址对子网 i 发出的流速声明 $f_{i,j}$ 进行聚合.

1. 初始化下一跳流速声明聚合集 S,$S\leftarrow\varnothing$
2. **for** each subnet $j\in G$ ▷计算发往各个子网的流速声明
3. $f_{i,j}\leftarrow\{i,j,\alpha_{i,j}\}$
4. $nhop[f_{i,j}]\leftarrow LookupRoure(i,f_{i,j})$
5. **if** $nhop[f_{i,j}]\neq NIL$ ▷next-hop 信息存在
6. **then** $S[nhop[f_{i,j}]]\leftarrow S[nhop[f_{i,j}]]\cup f_{i,j}$
7. $f_{i,\Lambda}\leftarrow\{i,IPADDR\_NONE,\alpha_{i,\Lambda}\}$ ▷计算发往 Internet blackhole 的流速声明
8. $nhop[f_{i,\Lambda}]\leftarrow LookupRoure(i,f_{i,\Lambda})$

9. **if** $nhop[f_{i,A}]\neq NIL$ ▷next-hop 信息存在
10. **then** $S[nhop[f_{i,A}]]\leftarrow S[nhop[f_{i,A}]]\cup f_{i,A}$
11. **for** each pair $\{nhop,F\}\in G$
12. **if** $F\neq\varnothing$
13. **then** 发送流速声明聚合 F 到下一跳 $nhop$

当中间路由器的网络接口接收到流速声明聚合 F 以后,根据流速声明 $f\in F$ 的下一跳地址执行 On-Demand 拆分.如果流速声明聚合 F 中仅包含一个流速声明或者所有流速声明 f 的下一跳地址相同,则不对流速声明聚合 F 进行拆分;如果流速声明聚合 F 中包含在当前网络接口具有不同下一跳地址的流速声明,则此时才根据对流速声明 f 的下一跳地址对流速声明聚合 F 进行拆分,最大限度地保持了路由相同的流速声明聚合发送.

算法 2. 网络接口处对收到的流速声明聚合 F 执行 On-Demand 拆分.

1. 得到该接口所属的节点 n
2. **if** $|F|=1$ ▷流速声明聚合仅包含一个流速声明,不需要拆分
3. **then** 根据下一跳地址转发流速声明聚合 F 到相应网络接口
4. **else**
5. 初始化下一跳流速声明聚合集 S_s,$S_s\leftarrow\varnothing$
6. **for** each fluid advertisement $f\in F$
7. $nhop[f]\leftarrow LookupRoure(n,f)$
8. **if** $nhop[f]\neq NIL$ ▷next-hop 信息存在
9. **then** $S_s[nhop[f]]\leftarrow S_s[nhop[f]]\cup f$
10. **if** $|S_s|=1$ ▷所有流速声明的下一跳地址相同,亦不需要拆分
11. **then** 根据下一跳地址转发流速声明聚合 F 到相应网络接口
12. **else** ▷存在下一跳地址不同的流速声明,需要对接收到的流速声明聚合 F 进行拆分
13. **for** each pair $\{nhop,F_s\}\in S_s$
14. **if** $F_s\neq\varnothing$
15. **then** 发送拆分后的流速声明聚合 F_s 到下一跳 $nhop$

3.2.2 无状态 Fluid 队列(stateless fluid queue)

由于某个 Fluid 流输入流速的改变将引起经过该队列的其他 Fluid 流的输出流速改变,因此 Fluid 队列需要维护每个经过它的 Fluid 流的状态信息.除了与子网链路相关的 Fluid 队列外,Internet 骨干网络模型中路由器的 Fluid 队列需要维护 $O(n^2)$个 Fluid 流状态信息.随着网络规模的增大,队列维护的 Fluid 流状态信息将消耗大量的内存,可能成为限制网络仿真规模的瓶颈.从第 3.1 节可以看到,如果一个队列的总输入流速小于服务速率,那个队列的缓冲区使用量将永远为 0,不存在流速损失,所有经过队列的 Fluid 流的输出流速等于输入流速.此时,维护 Fluid 流状态信息没有多大意义.在这种情况下,可以考虑使用无状态 Fluid 队列.在无状态 Fluid 队列中,不对流速声明进行处理,直接将流速声明转发到下一跳节点.在 Internet 骨干网络拥塞不严重的情况下,可以使用该方法进行仿真模型的优化,降低内存消耗.

3.3 基于Fluid仿真和数据包级仿真结合的多级仿真保真度模型

基于 Fluid 的大规模带宽限制蠕虫仿真模型可以很好地解决带宽限制蠕虫仿真的大规模和高带宽问题,满足带宽限制蠕虫传播特性的研究需要.但由于抽象层次的提高,可能会导致某些细节信息的丢失;而蠕虫防御策略验证的需求又要求我们在保证大规模网络特性的前提下,实现局部网络或者特定数据包的细节仿真.为了满足以上两个需求,需要对不同的仿真抽象技术进行有机的结合,实现具有多级仿真保真度的仿真模型,如 Liljenstam 等人提出了基于 Packet 模型和解析模型的混合仿真模型[16].本文通过在 Fluid 模型中引入 Packet 模型,可以在必要的场景实现更高仿真保真度的仿真,满足蠕虫防御策略验证分析的需要.

根据蠕虫防御策略研究,我们重点关注 Fluid 模型和 Packet 模型在两个方面的结合:① 在 Internet 骨干网络

模型实现 Packet 模型和 Fluid 模型结合,即 Internet 骨干网络模型中的路由器能同时处理数据包和流速声明.由于带宽限制蠕虫具有极高的传播速度,如何及早地发现蠕虫并采取相关措施,对于遏制蠕虫在 Internet 上的传播具有重要意义,如 Dübendorfer 等人提出了利用蠕虫传播产生的 ICMP 数据包在 Internet 骨干网络路由器上检测蠕虫爆发[17].在第 3.1 节提出的仿真模型中,该场景可以用 Fluid 模型仿真网络蠕虫流量,Packet 模型仿真蠕虫传播产生的 ICMP 包,ICMP 包在跨越 Internet 骨干网络的过程中保留了完整的数据包信息;② 在局部的子网模型实现完全的 Packet 模型,Internet 骨干网络模型使用 Fluid 模型.由于子网一般在一个统一的安全策略管理之下,因此在子网层配置和部署蠕虫防御设施具有较高的效率和可实现性,如 PWC 主动蠕虫遏制策略[18].通过在某个子网实现完全的 Packet 模型,Internet 骨干网络和其他子网使用 Fluid 模型,可以在保持子网内数据包精度的条件下,充分考虑外部蠕虫扫描对本地子网的影响,满足基于子网的蠕虫防御策略研究的要求.已有的混合仿真模型由于仿真机制不同,大多只实现了第 2 种结合模式,而在实现第 1 种结合模式时存在较多困难.

为了实现 Fluid 模型和 Packet 模型的结合,我们在 Fluid 队列中加入处理数据包的能力[14].具体方法为:对通过 Fluid 队列 q 的数据包,进行采样获得数据包对应的数据包 Fluid 流速 $\alpha_{p,q}$.但与原有 Fluid 流所不同的是,不会对这个由数据包采样获得的数据包 Fluid 流计算输出流速,更不会发出这个数据包 Fluid 流的流速声明到下一跳.同时,Fluid 队列根据队列缓冲区使用量,决定丢弃数据包还是将数据包转发到下一跳.数据包的转发根据路由信息完成,转发延时根据队列缓冲区的使用量计算得到.

$\alpha_{p,q}(t)$的具体计算过程为:在每个数据包到达的时刻 t,根据采样窗口 W 中最新的$|W|$个数据包,重新计算 $\alpha_{p,q}(t)$,$|W|$为数据包采样窗口 W 的大小.$\alpha_{p,q}(t)$的计算方法是

$$\alpha_{p,q}(t)=\sum_{p\in W} s_p \Big/ (t_{|W|}-t_1) \tag{8}$$

其中,s_p 为采样窗口 W 中数据包 p 的大小,t_1 为采样窗口中第 1 个数据包的到达时刻,$t_{|w|}$为采样窗口中第$|W|$个数据包的到达时刻.根据公式(8)计算新的 $\alpha_{p,q}$ 后,与当前的 $\alpha_{p,q}$ 进行比较,如果差值超过预先设定的门限,就用新的 $\alpha_{p,q}$ 取代旧的 $\alpha_{p,q}$.当 $\alpha_{p,q}$ 被更改时,设置一个超时事件.如果在预期的时间间隔内没有数据包到达,则将 $\alpha_{p,q}$ 重置为 0;如果在预期的时间间隔内有数据包到达,则取消当前超时事件,并按照上面的机制重新计算 $\alpha_{p,q}$.

数据包的转发过程为:当 t 时刻一个数据包到达时,通过 $D_p=x_q(t)/\mu_q$ 计算数据包在队列中的延时.当数据包 $s_p+x_q(t)\leqslant C_q$ 时,接受数据包,并经过延时 D_p 后根据路由信息转发数据包.当 $s_p+x_q(t)>C_q$ 时,队列以$(\alpha_q-\mu_q)/\alpha_q$ 的概率丢弃该数据包,其中,$\alpha_q=\sum_{f\in F_q}\alpha_{f,q}(t)+\alpha_{p,q}(t)$.如果当 $s_p+x_q(t)>C_q$ 时数据包被接受,按照延时 D_p 转发数据包,并置队列为 Pending 状态.当队列为 Pending 状态时,所有经过队列的数据包将被丢弃.经过$(s_p+x_q(t)-C_q)/\mu_q$ 时间,队列状态由 Pending 状态恢复至正常.

在引入了 Packet 模型之后,Fluid 队列对接收到的 Fluid 流速声明的处理方式同上节一样.不同的是,这时输入流$\alpha_q(t)$应包含数据包 Fluid 流$\alpha_{p,q}$.当 t 时刻,一个数据包到达时,如果根据式(8)计算出的$\alpha_{p,q}$ 发生改变,则相当于此时新到达一个流速声明,应按照第 3.1 节描述的步骤更新 Fluid 队列状态信息.通过把经过 Fluid 队列的数据包抽象成额外的数据包 Fluid 流,可以实现 Packet 模型对 Fluid 模型的影响,保证 Fluid 模型的仿真精确性.

当 Internet 骨干网络模型中所有节点的网络接口所对应的队列都具备了上述混合队列的工作模式后,就可以在 Internet 骨干网络模型同时模拟 Fluid 流和数据包,实现前面所描述的第 1 种方式的结合.

对于第 2 种方式的结合,可以通过在子网模型的边界网关处设置数据包/流速声明转换组件实现.首先,用基于 Packet 模型的主机和路由器模型构建子网内部仿真场景.子网内部的数据包发送到子网的边界网关,边界网关用前面提到基于采样窗口的转换机制,将数据包流量转换为 Fluid 流速声明,并将相应的流速声明逐一转发到相关节点.子网边界网关接受来自 Internet 骨干网络模型的 Fluid 流速声明,根据 Fluid 流速信息生成数据包.

4 仿真模型性能分析

通过对 Georgia Tech 的开源网络仿真器 GTNetS 进行扩展,我们对其增加了 Fluid 仿真机制,并在此基础上实现了上节描述的基于 Fluid 的大规模带宽限制蠕虫仿真模型.后续两节,我们将通过两个层面对基于 Fluid 的

带宽限制蠕虫仿真模型的有效性和精度进行验证.首先进行和数据包级仿真的对比实验,对仿真结果进行精度比较,并从仿真消耗时间、峰值内存使用量和已处理的离散事件量等方面对仿真效率进行比较.然后进行和实际攻击数据的对比实验,并与实际测量数据进行对比分析.

首先进行和数据包级仿真的对比实验,实验的网络测试场景设置如下:该网络测试场景包含 1 000 个子网,每个子网的寻址空间为 512,子网外出链路带宽为 10Mb,每个子网包含 10 台脆弱主机,脆弱主机的 IP 地址在子网空间内随机生成.这 1 000 个子网由包含 50 个路由器的 Internet 骨干网络连接起来,Internet 骨干网络路由器间的链路带宽为 2 000Mb.网络拓扑采用 GT-ITM 生成.数据包的大小根据 Witty 蠕虫大小设置,为 1 070 字节(包含了 UDP、IP 和 Ethernet 头)[19].每次实验开始时,随机选取一台脆弱主机进行感染,当所有脆弱主机被感染完毕后该次实验结束.实验硬件平台为 Dell OptiPlex 755,Intel 2.66G 酷睿 2 双核 CPU,内存扩展至 4G.

4.1 Fluid模型和Packet模型蠕虫仿真性能分析

仿真实验的结果见表 1.首先进行最小扫描空间下的性能分析,即蠕虫扫描空间刚好为 1 000 个子网,扫描速率分别设置为 500,1000 和 2000 scans/s/worm.可以看出,随着蠕虫扫描速率的增加,仿真消耗时间、峰值内存使用量和处理的离散事件量均呈线性增加.这是由于扫描速率越快,网络中同时存在的数据包和单位时间内产生的离散事件量就越多.对比采用了流速声明聚合的 Fluid 模型可以发现:Fluid 模型下仿真消耗时间和处理的离散事件量均小于 Packet 模型下的数值;而且随着扫描速率的增加,仿真消耗时间、峰值内存使用量和处理的离散事件量均随之减少.这是由于:在 Fluid 模型下,只有子网内被感染主机数目发生改变时才更新流速声明,而不像 Packet 模型下被感染主机持续不断地发送数据包;而且由于子网带宽的限制,当子网内被感染主机的扫描速率之和超过 1168 scans/s/worm 之后,即使子网内新增被感染主机,当前子网到其他子网的蠕虫扫描流速也不会发生改变.Packet 模型和 Fluid 模型下的蠕虫传播曲线如图 4 所示,可以看到,Fluid 模型在保持了较小的计算资源消耗的前提下,很好地保证了蠕虫传播仿真的精度.

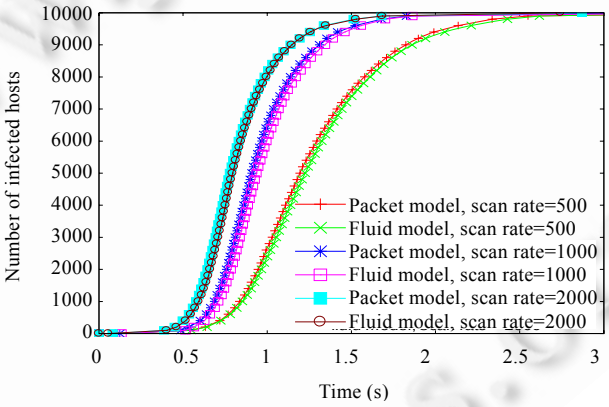


Fig.4 Simulation results of packet model and fluid model

图 4 数据包模式和 Fluid 模式下的仿真结果

然后进行扫描空间为 2^{24} 和 2^{32} 下的两组实验,扫描速率均为 2000 scans/s/worm.可以看出:在 Packet 模型下,随着扫描空间增大,仿真需要的内存也随之增加;当达到 32 位用户态应用程序的内存使用上限时,也仅仅分别感染了 46%和 1%的脆弱主机,不能完成蠕虫传播仿真的任务.在 Fluid 模型下,由于将脆弱主机所在子网外的空间统一对待,当扫描空间发生变化时,所消耗的内存和处理的离散事件量近似相等.

Table 1 Comparison of performance metrics between packet model and fluid model

表 1 数据包模式和 Fluid 模式下的仿真性能指标对比

	Parameter		Packet model				Fluid model (with fluid ad. aggregation)			
	Scan space	Scan rate	Infectious percentage	Elapsed time	Peak memory usage (MB)	Total events processed	Infectious percentage	Elapsed time	Peak memory usage (MB)	Total events processed
1	1000×512	500	100%	2h46m01s	709	34 157 029	100%	40m56s	708	12 053 730
2	1000×512	1 000	100%	5h37m42s	1 809	54 411 166	100%	22m23s	660	8 576 100
3	1000×512	2 000	100%	6h11m27s	2 489	66 780 984	100%	8m28s	601	5 361 598
4	2^{24}	2 000	46.11%	14h03m56s	3 079	71 542 354	100%	8m02s	600	5 680 849
5	2^{32}	2 000	0.92%	85h22m28s	3 057	160 597 856	100%	8m16s	600	5 196 407

4.2 针对带宽限制蠕虫的Fluid仿真优化机制的性能分析

本节我们研究流速声明聚合在基于 Fluid 的带宽限制蠕虫仿真模型中的作用,实验参数设置见表 2.可以看出,流速声明聚合不仅减少了离散事件量,而且减轻了流速声明可能额外引起的涟漪效应,流速声明聚合优化对基于 Fluid 的带宽限制蠕虫仿真模型性能有重大提升.

Table 2 Effect on performance metrics of fluid advertisements aggregation

表 2 流速声明聚合优化对性能指标影响

	Parameter		Fluid model (without fluid ad. aggregation)				Fluid model (with fluid ad. aggregation)			
	Scan space	Scan rate	Infectious percentage	Elapsed time	Peak memory usage (MB)	Total events processed	Infectious percentage	Elapsed time	Peak memory usage (MB)	Total events processed
1	1000×512	500	100%	24h29m49s	1 162	60 930 098	100%	40m56s	708	12 053 730
2	1000×512	1 000	100%	14h32m10s	1 019	46 514 809	100%	22m23s	660	8 576 100
3	1000×512	2 000	100%	6h30m50s	871	31 862 454	100%	8m28s	601	5 361 598

以上实验为了精确考虑,所有队列均为有状态 Fluid 队列,即 Fluid 队列保持所有经过它的 Fluid 流的状态信息.接下来测试无状态 Fluid 队列对仿真所需内存的节约,实验参数设置与表 2 保持一致.除了与子网链路相关的 Fluid 队列由于要保证子网带宽限制特性使用有状态队列外,Internet 骨干网络模型中路由器的其他队列均为无状态队列.仿真实验采用流速声明聚合优化措施,仿真结果如图 5 所示.可以看到,无状态队列可以很好地节约内存开销,因此可在遭遇内存仿真瓶颈时,将那些外出链路带宽大于进入链路带宽的 Fluid 队列设置为无状态队列,以增大网络仿真规模.

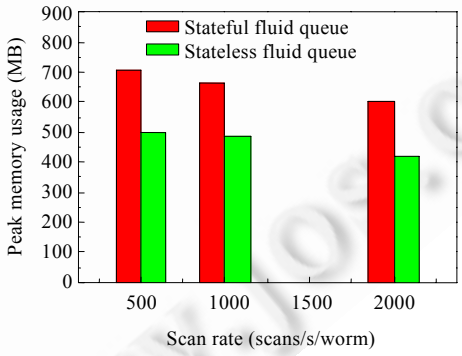


Fig.5 Effect on peak memory usage of stateless fluid queue

图 5 无状态队列对最大内存使用量的影响

以上仿真实验说明,基于 Fluid 的带宽限制蠕虫仿真模型可以在保证仿真精度的条件下,有效地降低仿真硬件需求,降低实验成本,提高仿真规模.为了更进一步验证模型的有效性和精度,在下一节将利用基于 Fluid 的带宽限制蠕虫仿真模型仿真 Witty 蠕虫,并与 CAIDA 提供的 Witty 蠕虫实际测量数据[20]进行对比分析.

5 利用基于 Fluid 的带宽限制蠕虫仿真模型仿真 Witty 蠕虫

5.1 Witty蠕虫仿真实验参数确定

基于 Fluid 的带宽限制蠕虫仿真模型是一个通用模型.为了实现特定带宽限制蠕虫的传播仿真,需要根据蠕虫的传播特性确定必要的仿真参数.由于蠕虫爆发的突然性和不可重现性,而且蠕虫传播过程涉及实体众多,通常缺乏蠕虫爆发时刻各实体和网络参数的精确记录.我们通过可获得的数据,如 CAIDA 提供的 Witty 蠕虫测量数据和蠕虫爆发时刻的 BGP 路由表快照,对 Witty 蠕虫仿真中的各重要参数进行推断,进而确定 Witty 蠕虫仿真参数,包括脆弱主机数量及其分布、蠕虫可用的子网有效带宽、主机带宽以及初始感染主机列表等.

5.1.1 蠕虫脆弱主机数量及其分布

CAIDA 测量结果显示:Witty 蠕虫爆发过程中,大约共有 55 909 个不同 IP 地址向 Internet 发送蠕虫数据包.但是考虑 Witty 蠕虫的破坏性载荷很大程度上会引起被感染主机的重启,由于动态 IP 地址分配,同一台被感染主机重启后可能拥有不同的 IP 地址.Moore 等人对测量数据进行分析,判定 Internet 中全部 Witty 蠕虫脆弱主机的数量约为 12 000 台.所以,本节 Witty 蠕虫仿真实验设定脆弱主机总数为 12 000 台.

CERNET BGP View 提供了 Witty 蠕虫爆发时的 BGP 路由表快照,利用 BGP 路由表快照,我们可以得到 Witty 蠕虫爆发时整个 Internet 中所有可以路由到的 Prefix.假设每个 Prefix 对应于仿真模型中的一个子网,通过将 CAIDA 提供的被感染主机的 IP 地址映射到相应的 Prefix 中,即可得到脆弱主机在所有子网中的分布参数.为了消除动态 IP 地址分配因素的影响,我们选取 Witty 蠕虫爆发前 45 分钟感染的 IP 地址进行 Prefix 映射匹配,然后依据此结果外推至全部脆弱主机.因为在蠕虫爆发初期,动态 IP 地址分配对蠕虫传播的影响较小,而且经过 45 分钟,Internet 中的全部脆弱主机已经大部分被感染.处理结果显示,Witty 蠕虫脆弱主机分布在大约 1 670 个 Prefix,Prefix 地址空间从 2^8~2^{24} 不等,但是 80%的 Prefix 的地址空间集中在 2^{12}~2^{14} 之间.所以,仿真实验设定包含脆弱主机的子网模型个数为 1 670,子网模型的地址空间为 2^{13}.每个子网模型拥有的脆弱主机数存在较大差异,多数子网拥有较少的脆弱主机,少数子网拥有较多的脆弱主机,具体分布见表 3.

Table 3 Distribution of the number of vulnerable hosts per subnet for Witty

表 3 子网脆弱主机数分布

Number of vulnerable hosts per subnet	Number of subnets	Proportion of the subnets	Number of vulnerable hosts per subnet	Number of subnets	Proportion of the subnets
1	530	0.317 37	10	40	0.023 95
2	300	0.179 64	11~20	120	0.071 86
3	170	0.101 80	21~30	10	0.005 99
4	110	0.065 87	31~40	40	0.023 95
5	60	0.035 93	41~50	20	0.011 98
6	110	0.065 87	51~100	20	0.011 98
7	40	0.023 95	101~200	10	0.005 99
8	50	0.029 94	>200		0
9	40	0.023 95			

5.1.2 子网有效带宽

由于其他非脆弱主机以及网络应用程序的存在,蠕虫可用的子网有效带宽小于该子网外出链路的全部带宽.此外,蠕虫流量和其他网络流量之间的相互作用也使得蠕虫可用的子网有效带宽在蠕虫传播过程中不断变化.由于蠕虫爆发的突然性,往往缺乏蠕虫爆发时刻各子网其他网络流量大小的精确记录.这里,我们通过 CAIDA 提供的 Witty 蠕虫测量数据对蠕虫可用的子网有效带宽进行推算.

假设蠕虫可用的各子网有效带宽存在相似性,子网有效带宽在一定的范围内小幅变动,并且包含脆弱主机多的子网模型拥有较高有效带宽的概率稍高.在 Witty 蠕虫爆发期间,某一时刻发送到 Internet 骨干网络的总扫描流量可以通过 CAIDA 提供的 Witty 蠕虫测量数据获得;同时,该时刻包含被感染主机的子网数量可以通过对观测到的被感染主机的 IP 地址进行映射获得.两者相除,即可得到该时刻蠕虫可用的子网平均有效带宽.通过对多个时刻进行上述计算,即可得到蠕虫可用的子网有效带宽变化范围.根据这个方法,我们以 0.5 分钟为时间间

隔,计算了从蠕虫爆发后 2 分钟~14 分钟这一系列时刻的子网平均有效带宽.计算结果显示,蠕虫可用的子网有效带宽变化范围:1700~2400 packets/s.因此,在本节 Witty 蠕虫仿真实验中,每个子网模型从该范围中随机选定一个有效连接带宽,包含脆弱主机台数多的子网拥有较高有效带宽的概率稍微增加.

5.1.3 主机带宽

CAIDA 提供的 Witty 蠕虫测量数据包括对被感染主机连接类型的统计信息,被感染主机的连接类型、带宽大小以及各种连接类型所占全部被感染主机的比例见表 4.本节实验中,根据被感染主机连接类型的比例设定脆弱主机的连接类型和主机带宽,并假设被感染主机的全部带宽都可被 Witty 蠕虫利用.

Table 4 Distribution of connection types of witty-infected computers

表 4 被感染主机连接类型分布

Connection type	Bandwidth	% of total infected population
Broadband	100Mb	43.70
XDSL	1Mb	26.04
Dialup	56Kb	17.34
Cable	10Mb	11.43
T1	1.544Mb	1.45
ISDN	144Kb	0.03

5.1.4 初始感染主机列表 hit-list 和被感染主机持续时间

相对于其他带宽限制蠕虫,如 Slammer 蠕虫,Witty 蠕虫具有其他两个显著特点:初始感染主机列表 hit-list 和破坏性载荷.CAIDA 测量数据显示,Witty 蠕虫前 10s 内迅速感染了大约 110 台主机,随后增长速度明显降低.这些被 Witty 蠕虫首先感染的 110 台主机多数集中于 140.152.0.0/14 网段.因此可以推断,Witty 蠕虫采用了 hit-list 扫描策略以提高蠕虫传播初始阶段速度.本节 Witty 蠕虫仿真实验设定初始感染主机列表大小为 110.

Witty 蠕虫携带的破坏性载荷会导致被感染主机重新启动或者崩溃.但被感染主机重新启动或者崩溃的时刻,即被感染主机的持续时间是一个不规则的分布,很难用一个统一的函数进行有效表示.本节 Witty 蠕虫仿真实验主要关注蠕虫传播的前期阶段,此时被感染主机重新启动或者崩溃的现象比较小.此外,由于被感染主机扫描率很高,少量被感染的主机所发送的蠕虫流量即可充满主机所在子网的有效带宽.即使少量被感染主机重新启动或者崩溃,也不影响所在子网发送到 Internet 的扫描流量,从而不影响被感染主机的增长速度.基于以上考虑,本节 Witty 蠕虫仿真实验设定在仿真实验关注时间范围内,被感染主机将一直处于持续状态.

5.2 仿真实验结果分析

在之前构建的基于 Fluid 的大规模带宽限制蠕虫仿真平台之上,根据上述分析设定 Witty 蠕虫传播仿真参数,对 Witty 蠕虫传播特性和流量特性进行分析.

5.2.1 Witty 蠕虫传播特性仿真结果分析

本节通过设定初始感染主机台数与分布来模拟 hit-list 扫描策略,并将蠕虫传播仿真结果和实际测量结果进行比较.其中,ACM IMC 2008 的数据是在 CAIDA 测量数据的基础上考虑网络拥塞因素所做的修订[21].

实验首先把 110 台初始感染主机集中到一个子网模型,蠕虫传播情况如图 6 中仿真结果 1 所示.可以看出:蠕虫开始的传播速度低于实际蠕虫传播速度;虽然快速传播阶段与实际传播速度接近,但整体传播时间明显增加.这主要是由于在进行子网有效带宽估算时,由于缺乏蠕虫传播开始变化幅度最大的前 1.5 分钟的测量数据,导致初始传播阶段发送到 Internet 的扫描流量偏小所致.可通过提高 110 台初始感染主机所在子网的蠕虫可用子网有效带宽,或者通过将 110 台初始感染主机所在子网拆分成具有类似有效带宽的更多子网,提升初始传播阶段发送到 Internet 的扫描流量.通过将 110 台初始感染主机拆分至 80 个子网模型进行仿真实验,蠕虫传播情况如图 6 中仿真结果 2 所示.此时,蠕虫传播曲线与实际测量结果有较高的拟合度.实验结果说明,hit-list 中的脆弱主机的数量和分布以及蠕虫可用的子网有效带宽对蠕虫传播有较大影响.虽然实测数据显示多数初始被感染主机集中于一个网段,但是在初始目标脆弱主机选择时,攻击者可能很好地考虑了带宽限制因素.实验结果同时表明,在初始感染主机设定合适的情况下,该仿真模型具有较高的仿真精度,可以为带宽限制蠕虫传播与防御

研究提供有效的实验平台.

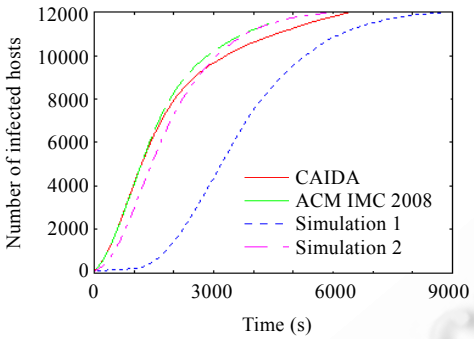


Fig.6 Simulation results of Witty propagation

图 6 Witty 蠕虫传播仿真实验结果

5.2.2 Witty 蠕虫流量特性仿真结果分析

多数主动扫描蠕虫自动检测、防御策略都是基于蠕虫爆发时产生的异常流量设计的.所以,仿真系统进行蠕虫防御策略研究时,必须能够比较准确地反映蠕虫传播过程中的流量特性.为了验证该仿真模型在蠕虫流量特性仿真方面的性能,本文在 Witty 蠕虫传播仿真实验 2 进行过程中,记录了蠕虫爆发后 1.5 分钟~14.5 分钟的被感染主机平均有效扫描速率,并将仿真结果与实测数据进行对比,如图 7 所示.

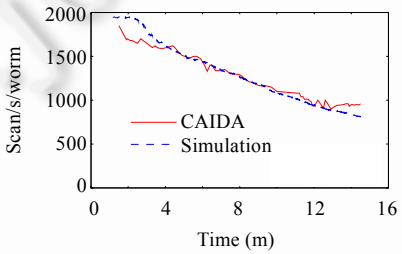


Fig.7 Simulation results of Witty traffic

图 7 Witty 蠕虫流量仿真实验结果

可以看出,仿真结果与实测数据比较吻合,具有明显相同的变化趋势.随着蠕虫传播,被感染主机平均有效扫描速率逐渐降低,说明该仿真模型能够很好地模拟带宽限制蠕虫传播的流量特性,可以为相应的蠕虫防御策略分析验证提供有效的实验平台.仿真结果与实测数据之间的误差,主要是由于子网模型带宽及其分布估计不够精确以及测量数据本身可能存在的误差造成的,通过适当的参数调整可以进一步减小仿真误差.

6 结束语

在利用子网抽象技术解决蠕虫仿真尺度规模的基础上,本文进一步针对带宽限制蠕虫高速扫描的特点,提出了一种基于 Fluid 的大规模带宽限制蠕虫仿真模型.该仿真模型能够在保证较高仿真精度的情况下,有效地节约计算资源和内存消耗,降低仿真成本,提高仿真执行效率.为了验证分析该仿真模型的有效性,本文在 GTNetS 上实现了一个完整的基于 Fluid 的大规模带宽限制蠕虫仿真系统,并利用该系统和数据包级仿真的性能进行了对比分析,以及根据 CAIDA 测量数据对 Witty 蠕虫传播进行模拟.实验结果显示,该仿真模型具有较高的仿真精度和较小的资源消耗.该仿真模型为我们研究分析带宽限制蠕虫传播特性和防御策略提供了一个有效的仿真实验平台.此外,本仿真模型可以方便地和数据包级仿真进行结合,实现局部细节仿真.在后续研究中,我们将主要致力于在该仿真平台之上对带宽限制蠕虫防御策略进行研究和评估.

致谢 感谢 CAIDA 为本文研究提供详尽的 Witty 蠕虫传播实测数据.感谢美国 Pennsylvania State University 网络安全实验室主任 Peng Liu 教授在解决本文研究问题的过程中给予的指导和建议.

References:

[1] Moore D, Paxson V, Savage S, Shannon C, Staniford S, Weaver N. Inside the slammer worm. IEEE Security and Privacy, 2003, 1(4):33−39. [doi: 10.1109/MSECP.2003.1219056]

[2] Shannon C, Moore D. The spread of the witty worm. IEEE Security and Privacy, 2004,2(4):46−50. [doi: 10.1109/MSP.2004.59]

[3] Wen WP, Qing SH, Jiang JC, Wang YJ. Research and development of Internet worms. Journal of Software, 2004,15(8): 1208−1219 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/15/1208.htm

[4] Zou CC, Gong WB, Towsley D. Code red worm propagation modeling and analysis. In: Proc. of the 9th ACM Symp. on Computer and Communication Security (CCS 2002). New York: ACM Press, 2002. 138−147. [doi: 10.1145/586110.586130]

[5] Chen Z, Gao L, Kwiat K. Modeling the spread of active worms. In: Proc. of the IEEE INFOCOM 2003. Piscataway: IEEE Press, 2003. 1890−1900. [doi: 10.1109/INFCOM.2003.1209211]

[6] Riley GF, Sharif MI, Lee W. Simulating Internet worms. In: Proc. of the 12th IEEE/ACM Int'l Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2004). Washington: IEEE Computer Society, 2004. 268−274. [doi: 10.1109/MASCOT.2004.1348281]

[7] Wang YW, Liu P, Jing JW, Jia XQ. A multi-level fidelity-preserving bandwidth-limited worm simulation model and its application. In: Proc. of the 41st IEEE Annual Simulation Symp. (ANSS 2008). Washington: IEEE Computer Society, 2008. 308−318. [doi: 10.1109/ANSS-41.2008.29]

[8] The georgia tech network simulator (GTNetS). http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS/ [doi: 10.1145/944773.944775]

[9] Liu B, Figueiredo DR, Guo Y, Kurose J, Towsley D. A study of networks simulation efficiency: Fluid simulation vs. packet-level simulation. In: Proc. of the IEEE INFOCOM 2001. Piscataway: IEEE Press, 2001. 1244−1253. [doi: 10.1109/INFCOM.2001.916619]

[10] Kesidis G, Hamadeh I, Jin YM, Jiwasurat S, Vojnović M. A model of the spread of randomly scanning Internet worms that saturate access links. ACM Trans. on Modeling and Computer Simulation (TOMACS), 2008,18(2):6:1−6:14. [doi: 10.1145/1346325.1346327]

[11] Vlachos V, Kalliamvakou E, Spinellis D. Simulating bandwidth-limited worms: One graph to rule them all? In: Papatheodorou TS, Christodoulakis DN, Karanikolas NN, eds. Proc. of the Current Trends in Informatics: The 11th Panhellenic Conf. on Informatics (PCI 2007). Athens: New Technologies Publications, 2007. 151−162.

[12] Nicol DM, Yan GH. High-Performance simulation of low-resolution network flows. SIMULATION: Trans. of the Society for Modeling and Simulation International, 2006,82(1):21−42. [doi: 10.1177/0037549706066093]

[13] Nicol DM. Efficient simulation of Internet worms. ACM Trans. on Modeling and Computer Simulation (TOMACS), 2008,18(2): 5:1−5:32.

[14] Kiddle C, Simmonds R, Williamson C, Unger B. Hybrid packet/fluid flow network simulation. In: Proc. of the 17th ACM/IEEE/SCS Workshop on Principles of Advanced and Distributed Simulation (PADS 2003). Washington: IEEE Computer Society, 2003. 143−152. [doi: 10.1109/PADS.2003.1207430]

[15] Weaver N, Hamadeh I, Kesidis G, Paxson V. Preliminary results using scale-down to explore worm dynamics. In: Proc. of the 2nd ACM CCS Workshop on Rapid Malcode (WORM 2004). New York: ACM Press, 2004. 65−72. [doi: 10.1145/1029618.1029628]

[16] Liljenstam M, Yuan Y, Premore BJ, Nicol DM. A mixed abstraction level simulation model of large-scale Internet worm infestations. In: Proc. of the 10th IEEE/ACM Int'l Symp. on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2002). Washington: IEEE Computer Society, 2002. 109−116. [doi: 10.1109/MASCOT.2002.1167067]

[17] Dübendorfer T, Wagner A, Hossmann T, Plattner B. Flow-Level traffic analysis of the blaster and sobig worm outbreaks in an Internet backbone. In: Julisch K, Kruegel C, eds. Proc. of the 2nd Conf. on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 2005). LNCS 3548, Berlin: Springer-Verlag, 2005. 103−122.

[18] Jhi YC, Liu P, Li LQ, Gu QJ, Jing JW, Kesidis G. PWC: A proactive worm containment solution for enterprise networks. In: Proc. of the 3rd Conf. on Security and Privacy in Communication Networks (SecureComm 2007). Piscataway: IEEE Press, 2007. 433−442.

[19] Kumar A, Paxson V, Weaver N. Exploiting underlying structure for detailed reconstruction of an Internet-scale event. In: Proc. of the 5th ACM SIGCOMM Internet Measurement Conf. (IMC 2005). Berkeley: USENIX Association, 2005. 351−364.

[20] The CAIDA dataset on the Witty worm. http://www.caida.org/data/passive/witty_worm_dataset.xml

[21] Wei SJ, Mirkovic J. Correcting congestion-based error in network telescope's observations of worm dynamics. In: Proc. of the 8th ACM SIGCOMM Internet Measurement Conf. (IMC 2008). New York: ACM Press, 2008. 125−130. [doi: 10.1145/1452520.1452536]

附中文参考文献:

[3] 文伟平,卿斯汉,蒋建春,王业君.网络蠕虫研究与进展.软件学报,2004,15(08):1208−1219. http://www.jos.org.cn/1000-9825/15/1208.htm

聂晓峰(1982－),男,安徽阜南人,博士生,主要研究领域为网络与系统安全技术,大规模网络蠕虫仿真.

王跃武(1975－),男,博士,讲师,CCF 会员,主要研究领域为网络与系统安全技术,大规模网络蠕虫仿真.

荆继武(1964－),男,博士,教授,博士生导师,CCF 会员,主要研究领域为网络与系统安全技术, PKI 技术,入侵容忍技术,蠕虫仿真技术.

向继(1976－),男,博士,讲师,CCF 会员,主要研究领域为网络与系统安全技术.