文章编号: 1000-5862(2012)02-0135-06

Ad hoc 网络安全协议仿真系统设计与实现

邱修峰[1,2], 刘建伟[1], 陈　杰[1], 刘　哲[1]

(1. 北京航空航天大学电子信息工程学院, 北京 100191; 2. 赣南师范学院数学与计算机系, 江西 赣州 341000)

摘要: 设计并实现了集成可视化网络拓扑的生成、安全协议的配置、数据流的设置、攻击事件的设置、仿真脚本的自动生成到攻击过程的仿真运行和协议性能的分析等功能于一体的仿真系统, 可对不同网络安全协议在不同攻击下的延迟、控制开销、吞吐量、丢包率、包交付率和抖动等参数进行对比分析, 动画演示了网络拓扑和数据流的动态变化.

关键词: 网络安全协议仿真; 网络攻击; 性能分析
中图分类号: TP 319　　文献标志码: A

0　引言

在 Ad hoc 网络协议的研究过程中, 若在真实的网络环境中进行性能研究、网络协议设计和开发不仅耗资巨大, 而且在统计数据的收集和分析上也困难重重, 因此建模与仿真是评估不同设计方案对系统性能影响的一个非常重要的手段. 网络仿真软件可以分为 2 大类: 商用软件和开源软件, 其中商用软件功能强大、界面友好但源代码不公开, 开源软件公开源码但是一般使用方式不友好. 目前流行的网络仿真软件主要有 OPNET[1]、GloMoSim[2]、NS2[3]、NS3[4]、QualNet[5]和 OMNet++[6]等. 对于特定形式的网络仿真, 研究人员开展了形式多样的探索, 文献[7]分析、设计并实现了自组织网络仿真平台, 为自组网的结构设计和网络协议算法性能的进一步测试、 评估、优化提供了仿真环境下的测试平台; 文献[8]基于 Visual C#.Net 开发工具, 设计并实现了无人机通信仿真系统, 以 Ad hoc 网络形式模拟动态空基信息传输网络. 由于 NS2 的开源特性, 不少研发人员基于 NS2 开发了和具体领域相关的仿真系统. 文献[9]通过扩展 NS 2 中的节点类与链路类, 支持对光突发交换网络的运行及关键算法的仿真; 文献[10]运用 NS2 研究并仿真了无线传感器网络多种模式下的吞吐量、丢包率和路由效率等性能的分析. 文献[11]提出了一种基于决策树的协同网络入侵检测模型. 设计了不同的代理对不同的网络数据协议类型分别检测. 但在目前的文献中, 所有的网络仿真软件都没有综合考虑协议的安全性能分析来辅助设计网络安全协议. 因此, 设计一个界面友好(集成网络拓扑场景自动生成, 性能分析比较结果数据图形表示自动生成)和可以分析协议安全性能的 Ad hoc 网络安全协议仿真软件系统具有重大意义.

本 Ad hoc 网络安全协议仿真系统基于 NS2 综合实现了从网络拓扑的生成、安全协议的配置、网络数据流的设置、脚本的自动生成、攻击事件的设置到攻击过程的仿真运行和协议性能的对比分析等一系列网络安全协议仿真具体过程. 基于对网络协议进行各种网络攻击仿真, 系统可以实现不同 Ad hoc 网络安全协议的性能分析.

1　Ad hoc 网络安全协议仿真系统的功能需求

网络安全协议仿真系统除了要考虑一般网络仿真需要的功能(如网络拓扑绘制)外, 还需要考虑协议安全性能问题, 本系统采用仿真各种网络攻击的方式来分析安全协议性能. 整个系统的主要功能有: (1) 不同安全协议在不同攻击下的延迟、控制开销、吞吐量、丢包率、包交付率和抖动等性能对比分析. 对应用不同安全协议的场景跟踪文件(trace 文件)进

收稿日期: 2012-01-12
基金项目: 国家"973"计划(2012CB315905), 国家"863"计划(2009AA01Z418)和高等学校博士学科点专项科研基金(20091102110004)资助项目.
作者简介: 邱修峰(1973-), 男, 江西兴国人, 讲师, 博士研究生, 主要从事网络安全和网络仿真的研究.

行分析，以数据和图表的形式展现给用户，力求准确、真实地反应出不同安全协议的性能差异; (2) 用动画演示协议受攻击时的性能动态变化过程. 可以演示数据流的动态变化和网络在各种攻击下的流量变化，表现出 Ad hoc 网络所具有的动态拓扑的特性，将安全协议抵抗攻击的整个仿真过程以动画的形式呈现给用户，力求能直观、形象地向用户呈现丰富的信息，尤其是安全协议的运行过程；并提供便捷的人机交互接口，方便用户观察仿真过程，捕获仿真过程中表现出来的各种数据流信息和攻击信息; (3) 可视化网络拓扑配置和管理. 网络拓扑主要有节点对象组成，可以控制拓扑图中节点对象以及节点对象的位置、信号覆盖范围、信号干扰范围和节点对象之间连接线等的绘制和显示；一个传输层代理对象依附于一个节点对象，可以控制拓扑图中传输层代理对象以及传输层代理对象的位置、传输层代理对象和节点对象的依附关系以及传输层代理对象之间的连接的绘制和显示；一个应用层对象依附于一个传输层代理对象，可以控制拓扑图中应用层对象以及应用层对象的位置、应用层对象和传输层代理对象的依附关系的绘制和显示；可以手工控制绘制或随机生成各种对象; (4) 安全协议配置和管理. 可以查看现有系统支持的各种路由协议(包括安全路由协议)，查看安全协议支持的加密和 Hash 算法；添加、修改和删除新的安全协议; (5) 攻击模型配置和管理. 可以查看系统支持的各种攻击方法；添加、修改和删除新的攻击方法; (6) 仿真过程脚本的自动生成. 整个网络拓扑生成和参数配置完毕后，可以利用 tcl 脚本管理功能自动生成和编辑相应的 tcl 脚本和相关配置文件，调用 NS2 运行 tcl 脚本，产生 trace 和 nam 文件, trace 文件可以用来进行性能数据分析, nam 文件用于动画演示.

2 系统设计与实现

2.1 系统设计方案

系统由前台子系统和后台子系统组成，如图 1 所示. 其中前台子系统包含工程管理、场景生成和参数设置、tcl 脚本文件自动生成和执行、仿真结果分析和系统管理 5 大模块(用 java 语言开发)；后台子系统是在 NS2 的基础之上扩充了安全协议库和攻击模型库 2 个模块(用 C++语言开发).

在系统中，工程被定义为所有一次或若干次 Ad hoc 网络安全协议仿真过程活动的总和. 使用本系统进行协议仿真首先必须创建工程，工程管理模块用来建立、删除、打开和关闭一个工程，和工程有关的数据会自动保存和更新至数据库中. 场景被定义为在一个具体的网络拓扑环境中进行一个协议性能分析仿真的情形. 场景生成和参数设置模块用来在一个工程中创建、删除、打开和关闭一个场景，在一个场景中按照 NS2 对象层次结构绘制生成节点对象、传输层代理对象、应用层对象和连接对象并配置相关的参数如节点初始位置、节点运动轨迹、路由协议、传输层协议、应用层协议、数据流的大小、

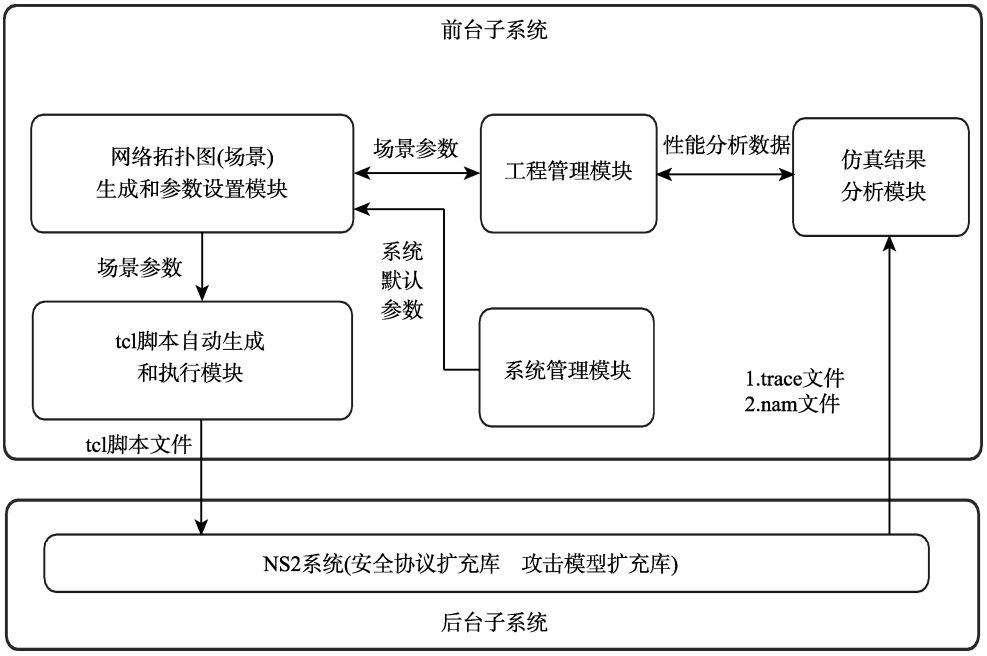

图 1　Ad hoc 网络安全协议仿真系统总体设计示意图

开始时间和结束时间等, 特别指出参数还包含了和安全协议仿真有关的数据, 如协议使用的 Hash 函数、加密算法和攻击行为描述数据等. 一个工程可以包含若干个场景 tcl 脚本自动生成和执行模块根据一个场景的所有具体参数自动生成一个 tcl 脚本文件, 并调用基于 NS2 的后台子系统生成场景跟踪文件(trace 文件)和动画描述文件(nam 文件). 仿真结果分析模块有 2 个功能. 第 1 个功能是根据 trace 文件分析相应场景的丢包率、包交付率、网络控制开销、吞吐量和时延等性能, 可以对一个网络协议在不同网络攻击下的性能或不同协议在相同攻击下的性能比较分析, 从而对协议的安全性能状况做出定量的评价, 为定性评价提供可靠依据; 如果一个协议在某种攻击下相关性能参数没有异常, 则说明此协议可以抵抗这种攻击, 否则说明不能抵抗. 第 2 个功能主要是根据 nam 文件来动画演示一个场景中一个协议在特定攻击下的表现, 可以演示各个节点的运动轨迹、路由动态寻找、数据包发送、数据包接收、数据包丢弃、路由包丢弃等细节, 动画演示画面可以放大缩小和控制动画播放速度. 后台子系统是以 NS2 为基础, 附加了 2 个和网络安全相关的模块: 安全协议扩充库和攻击模型扩充库.

2.2　系统关键技术说明

2.2.1　*场景生成和保存*　系统采用了层次式架构和面向对象技术. 层次式架构在此有双重含义, 一种是用户绘制的网络拓扑图对应 NS2 有 3 层结构: 节点对象、传输层代理对象和应用层对象; 另一种含义是采用了视图层-控制层-模型层-数据库多层结构. 视图层是和用户交互的接口, 用户绘制网络拓扑图, 配置网络每个层次的参数, 配置攻击行为事件. 模型层为系统建立数据模型, 系统建立了工程类、场景类、节点类、节点运动轨迹类、传输层代理类、应用对象类、传输层连接类、攻击行为类、路由协议类型类、攻击类型类等. 数据库为所有数据建立了相应的数据表, 模型层是将所有数据存储在内存中, 而数据库则将数据永久保存. 控制层将视图层中的用户操作和模型层保持一致, 即视图层中的网络拓扑图及其相应参数配置和模型层数据即时同步更新. 网络拓扑图存储在场景数据表、节点对象数据表、节点运动轨迹表、传输层代理对象数据表、应用层对象数据表和攻击行为数据表等表中.

2.2.2　*tcl 脚本自动生成*　tcl 脚本自动生成模块能根据场景生成模块模型层中的存储的一个场景所有对象的即时参数来生成 tcl 脚本. 算法依次由以下步骤组成: (a)根据场景对象的属性生成包含物理层参数配置、链路层参数配置、网络层参数配置、场景大小(节点活动范围)设置、trace 文件名设置、nam 文件设置名等功能的 tcl 脚本作为整个脚本的开始部分; (b) 根据节点对象与节点运动轨迹对象的属性生成节点定义和节点运动脚本; (c)根据传输层代理对象与传输层连接对象的属性生成传输层定义脚本; (d)根据应用层对象属性生成应用层定义脚本; (e)根据所选择的安全协议的属性生成安全协议属性(如协议使用的密钥生成算法)定义相关脚本; (f)根据攻击行为对象生成攻击行为脚本; (g)根据场景属性生成整个脚本的结束部分, 包含脚本运行的开始时间和结束时间定义.

2.2.3　*安全协议库*　安全协议的形式多样, 网络不同层的协议不同, 相同层的协议之间也差异较大. 因此, 要设计一个安全协议的统一模板是几乎不可能的. 本系统中实现的安全协议是各自设计的和保持本身特色的, 为了减轻设计新的协议安全有关的代码编写的工作量, 系统设计了一个公用的可扩充的密钥管理中心, 可以根据需要简洁地给不同对象分配密钥(如 RSA, ECC 等). 目前本系统中扩充的安全协议包括单签名 SAODV 协议、双签名 SAODV 协议、基于信任机制的 AODV 协议、基于签名与信任机制的 AODV 路由协议和基于信任机制的 DSR 路由协议等.

系统实现安全协议一般规则是对协议分组内容的不变部分的 Hash 值进行数字签名, 防止身份假冒和篡改; 对协议分组内容的可变部分(主要是跳数)采用 Hash 链防止跳数的减少.

系统中节点对于其它节点计算信任值时, 是根据人类交际心理学信任值慢速增加快速减少的原则, 对疑似恶意节点加速其信任值的递减. 算法不采用其他节点间接的推荐值以防止谎言欺骗; 算法局部化即算法只和局部邻居相关, 具有分布式的特点; 算法不需要附加设备、不需要节点位置与精确时间信息和不需要严格假设条件. 节点 N_i 对其邻居节点 N_j 进行监听, N_i 对 N_j 的信任值 C_{ij} 的计算公式为 $C_{ij}=S_{ij}/(S_{ij}+F_{ij})$, S_{ij} 为包转发成功次数, F_{ij} 为包转发失败次数, 算法过程如下: (1)对一条广播消息, 若 N_j 成功转发, 则 $S_{ij}=\alpha_1 S_{ij}+1$, $F_{ij}=\alpha_2 F_{ij}$, 若转发失败, 则 $S_{ij}=\alpha_1 S_{ij}$, $F_{ij}=\alpha_2 F_{ij}+1$; (2)对一条单播消息, 设到达目的地址的下一条不同路由有 m 条, L_j 为下一条是 N_j 的路由的长度, m 个 L_j 的最小值和次小值差异为 d, 如果 $d<T_d$ (路由长度差异异常临界

值)，则长度越小信任值越大，若 N_j 转发成功，则 $S_{ij}=\alpha_1 S_{ij}\sum_{k=1}^{m}L_k\Big/(mL_j)+1$, $F_{ij}=\alpha_2 F_{ij}\sum_{k=1}^{m}L_k\Big/(mL_j)$，若转发失败，则 $S_{ij}=\alpha_1 S_{ij}\sum_{k=1}^{m}L_k\Big/(mL_j)$，$F_{ij}=\alpha_2 F_{ij}\cdot\sum_{k=1}^{m}L_k\Big/(mL_j)+1$；如果 $d\geqslant T_d$，则长度越小信任值越小，若 N_j 转发成功，$S_{ij}=\gamma_1 mL_j\alpha_1 S_{ij}\Big/\sum_{k=1}^{m}L_k+1$, $F_{ij}=\gamma_2 mL_j\alpha_2 F_{ij}\Big/\sum_{k=1}^{m}L_k$，若转发失败，则 $S_{ij}=\gamma_1 mL_j\alpha_1 S_{ij}\Big/\sum_{k=1}^{m}L_k$，$F_{ij}=\gamma_2 mL_j\alpha_2 F_{ij}\Big/\sum_{k=1}^{m}L_k+1$. 其中 α_1、α_2 为调节因子，且 $0<\alpha_1$，$\alpha_2<1$，$\alpha_1<\alpha_2$，调节信任值增加速度比减小速度慢，$\gamma_1=m$，$\gamma_2=\begin{cases}2m, 若L_j的最小值等于2或3,\\ m, 其它,\end{cases}$ 当 L_j 的最小值等于 2 或 3 且 $d\geqslant T_d$ 时则认为附近有恶意节点的可能性大，加速信任值的减小速度.

2.2.4 *攻击模型库* 整个系统有一个统一的可扩展的攻击中心，可以对不同的协议发起攻击. 攻击中心在网络底层(数据链路层)实现，因此可以对高层的数据进行各种攻击操作如复制、修改、丢弃、延迟和非法转交等，从而形成各种表现多样的攻击如泛洪、sybil、虫洞等. 一次攻击行为描述如下:

```
class attacking{
int uid; //表示攻击行为的唯一编号
int attackingnodeid; //表示攻击节点
int attackednodeid;
//表示被攻击节点(虫洞攻击时表示攻击合作节点)
int thirdid; // 表示第 3 方节点(例如修改 IP 地址时会存在第 3 方节点)
string attackingtype; // 表示攻击类型
double starttime; //表示攻击开始时间
double endtime; //表示攻击结束时间
double attackingpower; //表示攻击强度(例如泛洪攻击时表示恶意数据发送的频率)
LL* ll; // 虫洞攻击时对等虫洞节点对象 LL 层的链接地址
}; //所有攻击行为形成攻击事件列表 attacklist.
```

目前本系统攻击模型扩充库已实现 flooding、packet_droping1、packet_droping2、modify_sourceip、modify_destinationip、modify_routing_hopcount、fake_new_routing_reply 和 wormhole 等共 8 种攻击类型，其中 flooding 为泛洪攻击，攻击节点在攻击期间给被攻击节点根据攻击强度恶意发送大量的包; packet_droping1 为丢弃数据包攻击，攻击期间攻击节点将经过本节点的来自被攻击节点的数据包丢弃; packet_droping2 为丢弃路由包和数据包攻击，攻击期间攻击节点将经过本节点的来自被攻击节点的路由包和数据包丢弃; modify_sourceip 为修改数据包的源节点 IP 地址，攻击期间攻击节点将经过本节点的来自被攻击节点的数据包的源节点 IP 地址改为第 3 方节点的 IP 地址; modify_ destinationip 为修改数据包的目的节点 IP 地址，攻击期间攻击节点将经过本节点的来自被攻击节点的数据包的目的节点 IP 地址改为第 3 方节点 IP 地址; modify_routing_hopcount 是修改路由应答包跳数为 1，攻击期间攻击节点将经过本节点的来自被攻击节点的路由应答包跳数修改为 1; fake_new_routing_reply 为伪造路由应答，攻击期间攻击节点将对经过本节点的来自被攻击节点的所有路由请求包伪造一个路由应答包发送给路由请求源节点，无论本节点是否存在到达目的节点的路由; wormhole 攻击是虫洞攻击，攻击期间将在攻击节点到攻击合作节点之间架设一条双向的直通“隧道”，所有攻击节点接收的路由包和数据包在 LL 层直接交给攻击合作节点，所有攻击合作节点接收的路由包和数据包在 LL 层直接交给攻击节点.

2.2.5 *性能数据分析* Ad hoc 安全协议仿真系统协议性能数据分析模块有节点信息分析、单个文件分析、一组文件分析和多组文件分析等 4 个子模块组成，对应用不同安全协议的场景跟踪文件(trace 文件)进行分析，以数据和图表的形式展现给用户，力求准确而真实的反应出不同安全协议的抵抗攻击的性能. 该模块将每一个节点的信息呈现给用户，比如: 该节点在传输层接收、发送数据包的数量，在网络层接收、发送、转发数据包和路由包的数量，在数据链路层接收、发送数据包和路由包的数量，丢弃数据包和路由包的数量，据此为用户提供分析安全协议的最基本的信息. 每一次安全协议仿真对应着一个场景跟踪文件(trace 文件)，对每一个文件中的数据进行提取、分析、比较、绘制成图，进而对这一组场景文件或多组场景文件进行性能参数(吞吐量、抖动率、包交付率、丢包率、路由开销和时延等)的分析比较，以数据和图表的形式比较不同协议应对不同攻击时的性能优劣.

2.2.6 *动画演示* 动画演示模块由 nam 文件分析、动作分析、前台监听和动作呈现 4 个子模块组成. nam 文件分析子模块完成 nam 文件的读取、nam 文

件信息的整理工作, 能为调用它的模块提供结构化的信息, 包括节点信息、节点运动信息、数据包信息、数据流信息和攻击信息等, 因为 nam 文件可能有数百兆字节大小甚至更大, 为了解决大文件的读取问题, 采取多线程分块读入的方法和缓存技术. 动作分析子模块将 nam 文件分析模块产生的数据进行进一步分析, 并结合定时器不断的更新各个页面元素(节点、数据包等)的位置信息, 将更新后的信息传输给调用它的模块. 动作呈现子模块根据动作分析模块传递来的页面元素的信息, 将各个页面元素绘制出来, 由于页面元素位置信息的不断刷新, 最终呈现出动作效果. 前台监听子模块捕捉前台的用户操作, 如拖拽时间轴或调整时间比例尺, 将这些操作的信息传输给动作呈现模块, 动作呈现模块会根据这些信息调整动画画面.

3　系统仿真效果

本文以比较分析 NS2 内含的 AODV 路由协议和系统安全协议扩充库基于信任机制的路由协议 AODVT1 抵抗丢弃数据包攻击 packet_droping1 的性能为例说明实际的仿真效果. 在建立新的工程和场景之后, 用户根据需要绘制网络拓扑图(如图 2 数据流源节点和目的节点间有 3 条路径)和配置参数, 设置源节点和目的节点 0~100 s 时间长度的每秒 4 个 512bit 大小数据包的恒定速率数据流, 选择路由协议 AODV, 修改默认的 trace 文件名和 nam 文件名分别为 aodv_d1.tr 和 aodv_d1.nam , 在图 2 的中间路径的一个节点设置丢弃数据包攻击 packet_droping1 (从 5 s 开始到 65 s 结束), 生成和运行 tcl 脚本, 结果产生场景跟踪 trace 文件 aodv_d1.tr 和动画描述 nam 文件 aodv_d1.nam; 然后网络拓扑和其它参数保持不变, 选择路由协议 AODVT1, 修改默认的 trace 文件名和 nam 文件名分别为 aodvt1_d1.tr 和 aodvt1_d1.nam, 再次生成和运行 tcl 脚本, 结果产生场景跟踪 trace 文件 aodvt1_d1.tr 和动画描述 nam 文件 aodvt1_d1.nam. 用性能分析子系统打开 trace 文件 aodv_d1.tr 和 aodvt1_d1.tr 进行分析. 这里以吞吐量来说明协议抵抗丢弃数据包攻击的性能优劣, 图 3 和图 4 就是 2 种不同的协议进行丢弃数据包攻击后吞吐量性能示意图, 攻击期间吞吐量明显差异较大, 从而得出协议 aodvt1 可以较好地抵抗丢弃数据包攻击而 AODV 不能抵抗的结论.

还可以用动画演示子系统分别打开 aodv_d1.nam 和 aodvt1_d1.nam 动画描述文件, 观察 2 个协议在丢弃数据包攻击时抵抗性能的动态表现.

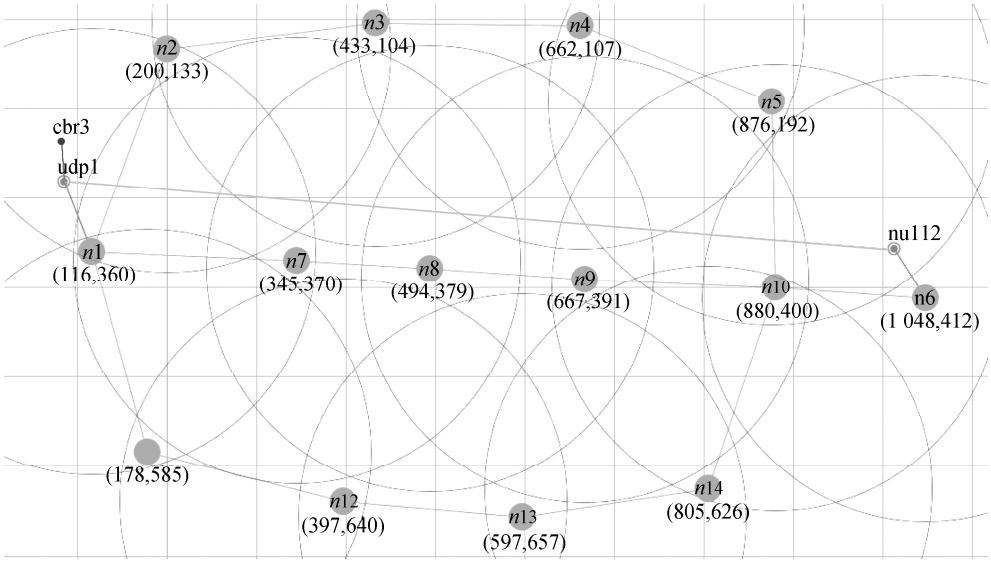

图 2　网络拓扑示意图

4　下一步工作

总之, 整个系统设计合理, 功能比较完整, 但存在许多不足之处, 下一步的工作主要有: (1)扩充安全协议库, 尽可能包含已知的典型协议; (2)扩充攻击模型库, 尽可能包含已知的各种攻击; (3)扩充系统的应用范围使其可以应用在其它类型的网络, 成为通用的网络安全协议仿真分析软件.

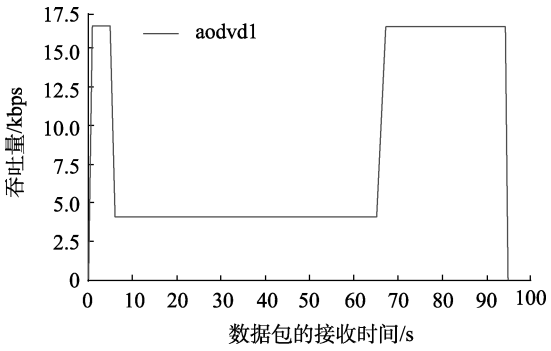

图 3 **aodv** 在丢弃数据包攻击时的吞吐量

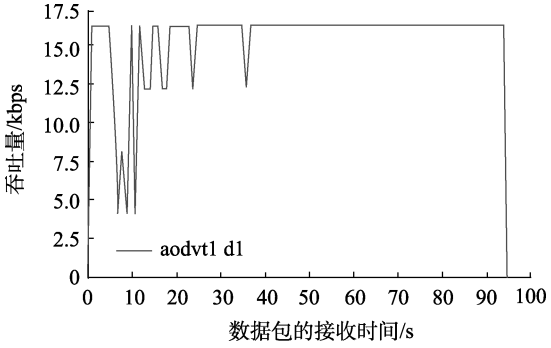

图 4 **aodvt1** 在丢弃数据包攻击时的吞吐量

5 参考文献

[1] OPNET.White papers and power briefs [EB/OL].[2012-02-05]. http://cn.opnet.com/whitepapers/.

[2] GloMoSim. GloMoSim manual [EB/OL].[2012-02-05].http://pcl.cs.ucla.edu/projects/glomosim/GloMoSimManual.html.

[3] Bajaj S, Breslau L, Estrin D, et al. Improving simulation for network [EB/OL]. [2012-01-12].http://ilab.cs.byu.edu/zappala/pubs/usc-cs-tr-99-702.pdf.

[4] NS3.NS3 tutorial [EB/OL].[2012-01-12].http://www.nsnam. org/docs/release/3.13/tutorial/ns-3-tutorial.pdf.

[5] Qualnet.Qualnet tutorial [EB/OL].[2012-01-13].http://www.qualnet.com/pdf/QualNetTutorial.pdf.

[6] András Varga, Rudolf Hornig.An overview of the OMNeT++ simulation environment [EB/OL].[2012-01-16]. http://dl.acm.org/citation.cfm?id=1416290.

[7] 刘军, 李喆, 岳磊. 自组织网络仿真平台的设计与实现 [J]. 计算机科学, 2008, 35(1): 24-26, 30.

[8] 吴迪, 赵小刚, 朱凤仙. 想定环境中无人机 Ad Hoc 网络仿真研究和实现 [J]. 系统仿真学报, 2008, 20(23): 6409-6413, 6428.

[9] 朱智俊, 乐孜纯, 朱冉. 基于 NS2 的 OBS 网络仿真平台研究与实现 [J]. 通信学报, 2009, 30(9): 128-134.

[10] Tao Yang, Mino, Spaho G, et al. A simulation system for multi mobile events in wireless sensor networks[EB/OL].[2012-01-15]. http://dl.acm.org/citation.cfm?id=1989772.

[11] 蒲元芳, 张巍, 滕少华, 杜红乐. 基于决策树的协同网络入侵检测 [J]. 江西师范大学学报: 自然科学版, 2010, 34(3): 302-307.

The Design and Realization for Ad hoc Network Secure Protocols Simulation Platform

QIU Xiu-feng[1,2], LIU Jian-wei[1], CHEN Jie[1], LIU Zhe[1]

(1. School of Electronics and Information Engineering, Beihang University, Beijing 100191, China;
2. Department of Mathematics and Computer, Gannan Normal College, Ganzhou Jiangxi 341000, China)

Abstract: A simulator that integrates functions of drawing network topology, configuring secure protocol, setting up data flows and attack events, generating simulation scripts automatically, running attacks and comparing protocol performances is designed and implemented. Through comparative analysis to delay, control overhead, throughput, packet loss rate, package delivery rate, and Jitter of simulation results from running different secure protocols under different attacks, the system can realize the performance analysis to various Ad hoc network secure protocols, and demonstrate the dynamic changing of a network under attacks in the form of animation.

Key words: network secure protocol simulation; network attack; performance analysis

(责任编辑: 冉小晓)