МОДЕЛИ И СИСТЕМЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ РАСПРОСТРАНЕНИЯ СЕТЕВЫХ ЧЕРВЕЙ

И. В. Котенко[1], В. В. Воронцов[2], А. В. Уланов[3]

[1–3]Санкт-Петербургский институт информатики и автоматизации РАН

[1–3]СПИИРАН, 14-я линия ВО, д. 39, Санкт-Петербург, 199178

[1]<ivkote@comsec.spb.ru>, [2]<vorontsov@comsec.spb.ru>, [3]<ulanov@iias.spb.ru>
[1–3]<www.comsec.spb.ru>

УДК 004.94

Котенко И. В., Воронцов В. В., Уланов А. В. **Модели и системы имитационного моделирования распространения сетевых червей** // Труды СПИИРАН. Вып. 4. — СПб.: Наука, 2007.

Аннотация. *Рассматриваются существующие модели и системы имитационного моделирования распространения сетевых червей. Приводится описание программных систем имитационного моделирования DDosVax, NWS, SSF.App.Worm, GTNetS, PDNS и DIB:S/TRAFEN. Представлены достоинства и недостатки применения указанных систем для исследования распространения сетевых червей.* — Библ. 8 назв.

UDC 004.94

Kotenko I. V., Vorontsov V. V., Ulanov A. V. **Models and systems of network worm propagation simulation** // SPIIRAS Proceedings. Issue 4. — SPb.: Nauka, 2007.

Abstract. *The paper considers the existing models and systems of network worm propagation simulation. The simulation systems NWS, SSF.App.Worm, GTNetS, PDNS and DIB:S/TRAFEN are described. The paper analyzes the advantages and disadvantages of applying these systems for network worm propagation research.* — Bibl. 8 items.

1. Введение

Одним из актуальных направлений исследований в области компьютерной безопасности является моделирование распространения сетевых червей. Результат этих исследований применяется для решения следующих задач [7, 8]:

1) исследования и лучшее понимание явлений вирусных эпидемий, происходивших в прошлом;

2) оценка потенциальных угроз, связанных с сетевыми червями;

3) прогнозирование влияния новых червей, которые появятся в будущем, на глобальные сети;

4) определение параметров описывающих поведение червей;

5) *разработка механизмов обнаружения сетевых червей и защита от них*.

Спектр возможных подходов к моделированию распространения компьютерных червей и область их применения показаны на рис. 1 [3]. Разнообразие предлагаемых моделей можно объяснить разнородностью целей исследования, что обусловливает различие в постановках задачи на моделирование. На рис. 1 также отображены рассмотренные в данной статье модели и программные системы моделирования, которые могут быть использованы для моделирования сетевых червей.

Все представленные на рис. 1 модели отображаются в двух основных разрезах, а именно *масштабируемости* и *точности* моделирования.

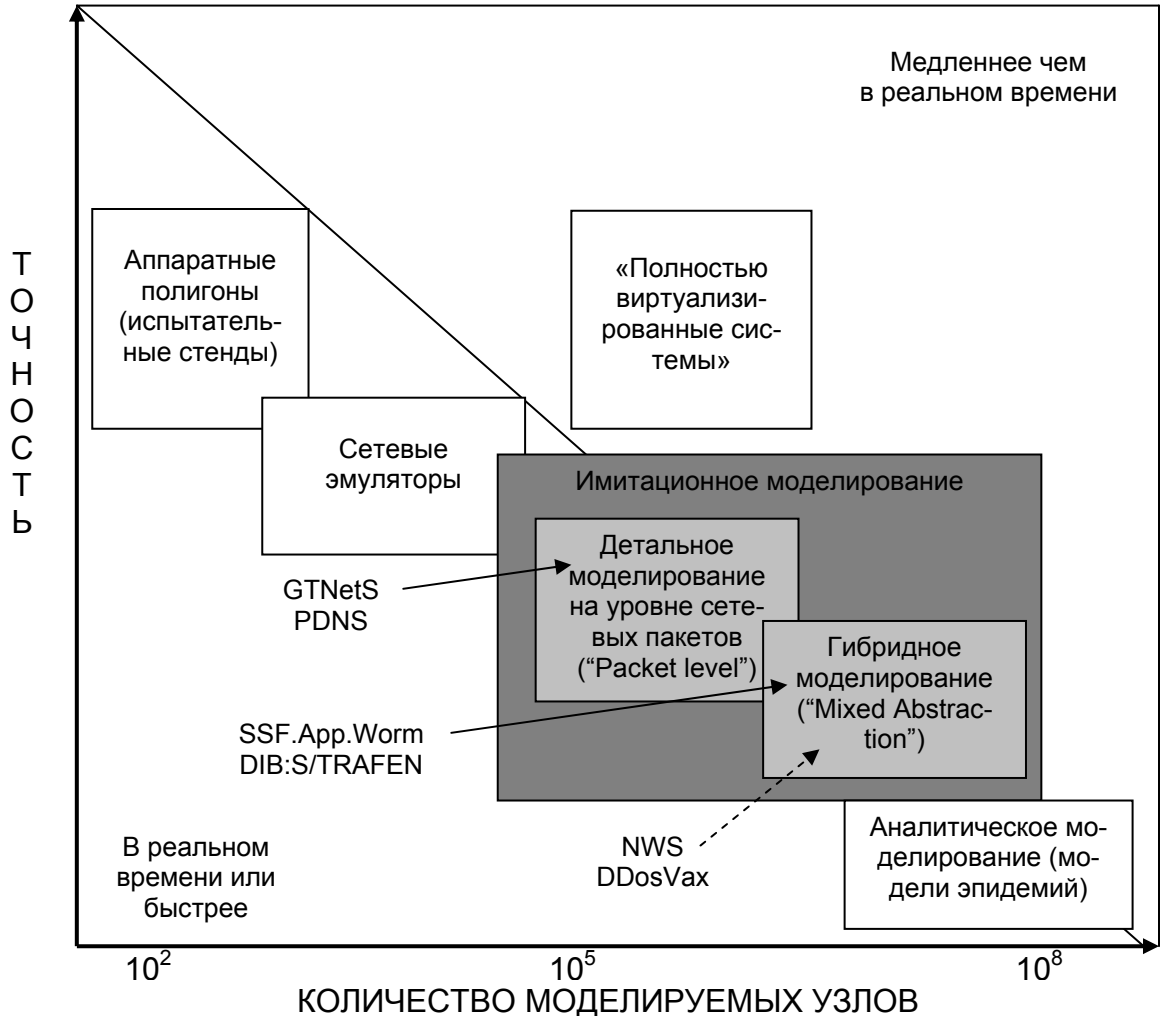

Рис. 1. Подходы к моделированию компьютерных червей [3].

При рассмотрении также учитывается *скорость* моделирования (соотнесенная со временем, затрачиваемым моделируемым процессом во время его реального протекания).

Количество моделируемых узлов определяется как число узлов сети (конечных хостов и маршрутизаторов), которое можно моделировать, используя применяемый метод.

Точность метода соответствует используемой при моделировании степени детализации сети и конечных хостов.

Нетрудно видеть, что методы, используемые для моделирования сетевых червей, фактически включают в себя весь спектр методов, применяемых для моделирования сетей вообще.

Аппаратные полигоны (испытательные стенды) функционируют в реальном времени и позволяют получить наиболее точные данные о распространении червей. Однако очевидным является тот факт, что невозможно создать испытательный стенд, приближающийся по своим масштабам к сети Интернет.

Эмуляция представляет собой комбинацию имитационного моделирования и реализации процессов в реальных сетях в режиме реального времени. Другими словами, эмуляция — это использование реальных испытательных стендов с искусственным внесением в процесс их функционирования тех или иных параметров, например задержек, ошибок, атак и т.д. Таким образом, эмуляция

позволяет осуществлять моделирование с использованием реальных сетевых устройств и приложений.

Так называемые *полностью виртуализированные системы* базируются на применении для моделирования виртуальных машин (в частности, VMWare).

Аналитическое моделирование основывается на использовании математических формул для задания зависимостей между параметрами моделируемых процессов. Данный подход отличается наибольшей масштабируемостью, однако точность моделирования обычно не высока.

Самым распространенным способом моделирования сетевых червей является *имитационное моделирование*, потому что другие подходы (использование реальных червей, натурные эксперименты) имеют те или иные ограничения, например ограниченные возможности по созданию испытательного полигона большого размера. Данный подход предусматривает построение модели исследуемого процесса и имитацию его выполнения. Основной сложностью при таком подходе, пожалуй, следует считать моделирование среды распространения, а именно Интернета, в силу масштабности и неоднородности этой сети. С другой стороны, имитационное моделирование позволяет подробно описать какой-либо участок сети и детально его проанализировать, включая передачу пакетов по каналам связи, возникновение очередей, протоколы маршрутизации и т.д.

Данная статья является обзором ряда наиболее интересных моделей и систем имитационного моделирования, служащих для исследования процессов распространения сетевых червей.

Статья имеет следующую структуру. Во введении описываются существующие подходы к моделированию сетевых червей. Далее последовательно описываются модели и программные системы *имитационного* моделирования эпидемий сетевых червей. Рассматриваются следующие системы имитационного моделирования: *DDosVax, NWS, SSF.App.Worm, GTNetS, PDNS* и *DIB:S/TRAFEN*. В заключении делаются выводы и будущие направления исследований.

2. Система имитационного моделирования DDosVax

В работе [8] описывается система имитационного моделирования сетевых червей, разработанная на языке Perl. Будем называть ее **DDosVax** по названию проекта, в рамках которого она создана.

Не рассматривая каких-либо конкретных уязвимостей, авторы исходят из того, что моделирование червя должно сводиться к моделированию следующих трех шагов (рис. 2):

- шаг 1 — определение уязвимого хоста;
- шаг 2 — компрометация цели;
- шаг 3 — передача тела червя с последующей активацией.

Для некоторых уязвимостей все эти шаги могут быть реализованы в одном сетевом пакете, как в случае червя Sapphire. Для других эти шаги реализуются отдельно.

Шаги 2 и 3 могут быть смоделированы как обмен некоторым объемом данных по определенному протоколу с определенной задержкой по времени.

Шаг 1 — более сложен, но может быть смоделирован, не учитывая детали конкретной уязвимости.

Модель распространения сетевого червя, используемая в системе моделирования DDosVax, описывается 13 параметрами.

Рис. 2. Упрощенная модель червя [8].

Наиболее важными из них (описывающие самого червя, а не параметры протокола или глобальные характеристики системы) являются следующие:

1. *Используемая стратегия сканирования*. Рассматривается случайное сканирование (random scanning), сканирование по заранее подготовленному списку (hitlist scanning) и сканирование со смещением к локальным адресам (local forced scanning). Авторы используют упрощенный способ расчета вероятности успешного заражения при одном сканировании, рассчитывая в каждый момент времени эту величину как отношение разности числа уязвимых и зараженных хостов к общему числу хостов в сети. Ошибка, вносимая таким расчетом, связана с вероятностью события, что на некотором шаге несколько хостов выберут для сканирования один и тот же адрес. Поскольку доля уязвимых хостов в сети мала, вероятностью этого события пренебрегают.
2. *Используемый протокол транспортного уровня*. Рассматривается два возможных варианта — протокол с установлением соединения (TCP) и протокол без установления соединения (UDP). При использовании протокола UDP накладные расходы, вносимые протоколом транспортного уровня, минимальны и скорость распространения червя в наибольшей степени определяется *пропускной способностью* используемых каналов. Зараженный хост может послать столько зараженных пакетов, сколько ему позволит это сделать пропускная способность канала и стек протоколов. Как известно, в случае использования TCP для установления соединения используется процедура трехстороннего рукопожатия. Таким образом, в отличие от UDP главным фактором, сдерживающим распространение червя, использующего TCP, становится *задержка* Round Trip Time. Также, например, при попытке установить соединение с несуществующим хостом решение о невозможности это сделать будет принято по истечении длительного интервала ожидания. Заметим сразу, что работа TCP в указанной системе реализована достаточно упрощенно, в частности, не рассматривается процедура медленного старта.

3. *Скорость сканирования* (для червей, основанных на UDP) или *максимальное число одновременно открытых соединений* (для червей, основанных на TCP).
4. *Размер червя.*
5. *Возможная дополнительная задержка, необходимая для активации*.

Таким образом, моделирование функционирования червя в работе [8] в самом общем виде сводится к определению новых зараженных хостов, обмену данными с помощью заданного протокола и возможным ожиданиям дополнительного времени перед активацией.

Наибольший интерес для дальнейших исследований в работе [8] *представляет использованная для моделирования сетевых червей модель сети Интернет*, суть которой состоит в следующем. Отдельные хосты при моделировании не рассматриваются, а Интернет представляется в виде *n* групп хостов, которые принадлежат к подсетям со сходными характеристиками. Каждой группе присвоены два определяющих параметра: пропускная способность каналов между хостами группы и вносимая хостами задержка (рис. 3). Причем если взаимодействуют хосты двух разных групп, то параметры соединения рассчитываются по худшим характеристикам (устанавливается минимальная полоса пропускания и максимальная задержка).

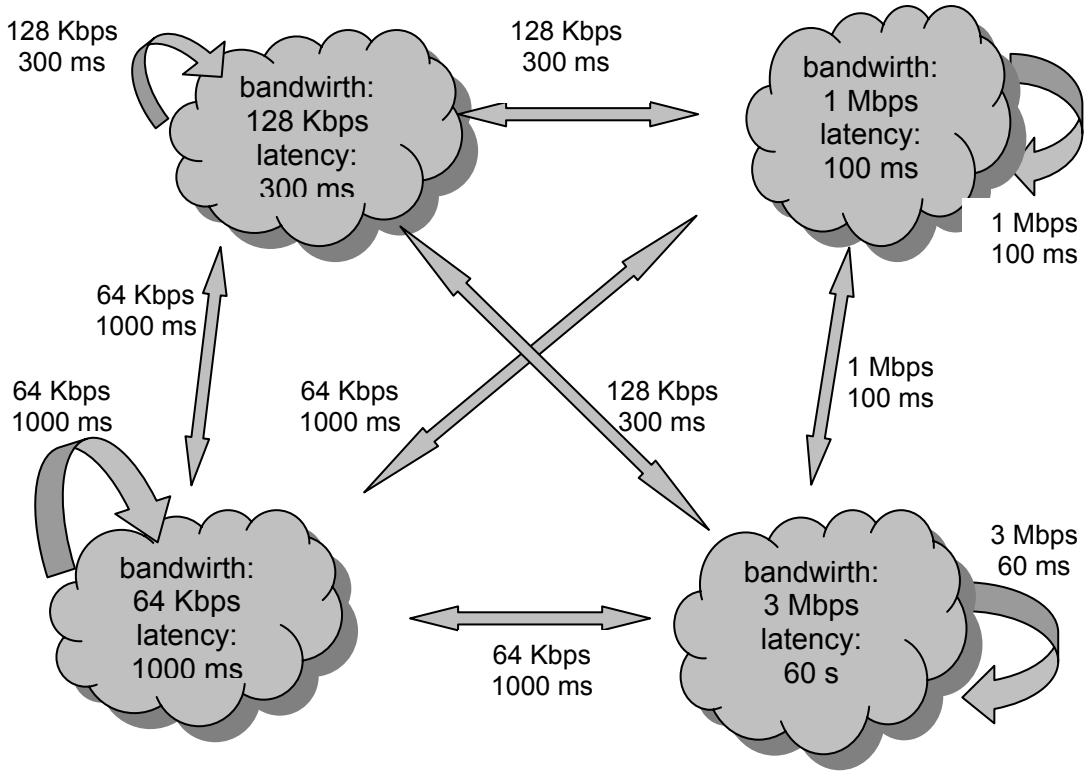

Рис. 3. Модель сети Интернет [8].

В работе [8] рассматриваются два случая — для числа групп $n = 4$ и $n = 10$, причем характеристики каждой группы и распределение числа хостов по этим группам взяты из [5], где приведены результаты измерений параметров клиентских соединений в файлообменных сетях Napster и Gnutella. Таким образом, фактически вводится основное допущение работы [8], а именно — распределение хостов по группам, полученное для одноранговых файлообменных сетей, отражает структуру всего Интернета в целом.

В [8] приводятся графические зависимости объема передаваемого трафика и количества зараженных хостов от времени. Моделируется распространение червей Sapphire (использует UDP) и Code Red I v2 (применяет TCP). Полученные результаты сравниваются с реальными измерениями (заимствованными из соответствующих источников). Для получения с помощью имитационной модели более реалистичных результатов выполняется «подстройка» модели, например, хосты, первоначально инфицированные Sapphire, помещаются в «быстрые» сети.

Данная система имитационного моделирования по сравнению с другими ранее освещенными имеет следующие *достоинства*:

- используется полный набор параметров, характеризующих сетевого червя;
- моделирование основано на реальных количественных параметрах сетевых червей.

Вообще же, оценить адекватность предложенной модели сети Интернет достаточно сложно в силу относительно небольшого числа доступных измерений при реальных эпидемиях, которые можно было использовать для проверки имитационной модели.

3. Система имитационного моделирования NWS

Network Worm Simulator (NWS) [1] представляет собой систему имитационного моделирования сетевых червей, разработанную, как и DDosVax, на языке Perl. Сеть в этой системе представляет собой полносвязный граф, т.е. любая пара объектов в сети может посылать друг другу сообщения. В NWS реализован объектно-ориентированный подход и пользователь должен создавать объекты некоторых классов, из которых состоит моделируемая сеть.

Перечислим кратко для примера основные классы, по названию которых можно понять их назначение и взаимосвязь. Объект класса Network (сеть) содержит несколько объектов класса Host (хост), который в свою очередь включает в себя несколько объектов класса Software (приложение). При передаче объектов класса Message (сообщение) между хостами указывается не только адрес хоста-назначения, но и имя приложения, запущенного на этом хосте. Тем самым фактически моделируется аналог сокета в TCP/IP сетях. Сам функционал приложений, равно как и червей, может реализовываться пользователем.

Автор NWS использует элементарную математическую модель эпидемии, называемую Susceptible–Infected–Susceptible (SIS). Для такой модели поведение каждого объекта в популяции описывается простейшим марковским процессом с двумя состояниями («болен»/«здоров») и двумя переходными вероятностями (часто равными). Для моделирования эпидемий компьютерных червей, при которых не рассматриваются контрмеры, вероятность перехода «болен»→ «здоров» принимается равной нулю, что еще больше упрощает модель, которая описывается в этом случае единственным параметром — вероятностью заражения.

В [1] приводятся примеры моделирования в NWS по нескольким интересным сценариям. В качестве примера можно привести моделирование одновременного распространения по сети двух различных версий червя Code Red, которые соревновались за заражение общего множества уязвимых хостов.

Как отмечает автор NWS, для проведения более реалистичного моделирования в системе должны быть в дальнейшем реализованы механизмы моделирования межсетевых фильтров и маршрутизаторов.

4. Система имитационного моделирования SSF.App.Worm

Система имитационного моделирования сетевых червей SSF.App.Worm разработана на основе программного пакета Scalable Simulation Framework Network (SSFNet) [6].

SSF представляет собой стандарт для моделирования сложных систем на основе дискретных событий на Java и C++. SSFNet — основанный на SSF набор Java-компонент, предназначенных для моделирования сетей и протоколов начиная с IP-уровня и выше.

Экземпляр класса SSFNet может автоматически конфигурировать себя, посылая запросы к конфигурационной базе данных. Эта база данных содержит файлы в формате Domain Modeling Language (DML) [6], чем обеспечивается стандартизованный и простой для чтения и написания синтаксис, применяемый всеми пользователями системы моделирования.

Система SSF.App.Worm основана на использовании гибридного двухуровневого подхода (рис. 4):

- для более грубого моделирования всего Интернета в целом выделен «*макроскопический» (макро-) уровень*;
- для детального моделирования (до уровня передачи пакетов) отдельных частей сети используется «*микроскопический» (микро-) уровень*.

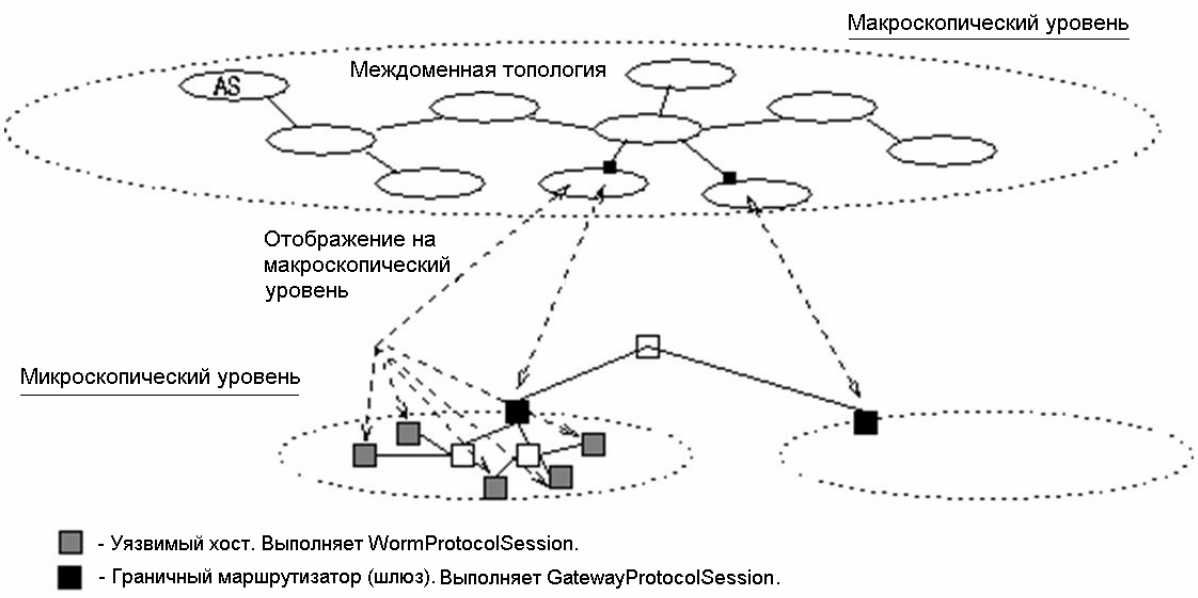

Рис. 4. Микро- и макроуровни в SSF.App.Worm [6].

На *макроскопическом уровне* реализуется некоторая модель эпидемии (детерминированная или стохастическая), а также задаются такие параметры, как например вид распределения уязвимых хостов в сети **(**равномерное, называемое однородным, либо привязанное к Автономным Системам (AS), называемое многослойным**)**, параметры трафика, возникающего при сканировании.

На *микроскопическом уровне* рассматриваются некоторые конкретные сетевые модели с определенными в них хостами, маршрутизаторами и другими компонентами.

Связь между моделями эпидемий и моделями сетей обеспечивается посредством двух протоколов WormProtocolSession и GatewayProtocolSession. Каждый уязвимый хост, который моделируется на микроскопическом уровне, должен поддерживать WormProtocolSession. Если в ходе развития эпидемии на макроскопическом уровне для заражения будет выбран такой хост, то это событие может быть специальным образом обработано в зависимости от функциональности червя. В свою очередь GatewayProtocolSession, выполняемый на пограничных маршрутизаторах, позволяет «перебрасывать» трафик, возникающий при сканировании в ходе эпидемии, в соответствующую модель сети.

По нашему мнению, гибридный подход, заложенный в SSF.App.Worm, безусловно является интересным и перспективным. К очевидному недостатку, имеющемуся на данный момент, можно отнести отсутствие проверок соответствия и непротиворечивости топологий сетей на микро- и макроуровнях.

5. Система имитационного моделирования на основе GTNetS

В работе [4] предлагается подход к детальному моделированию распространения сетевых червей в больших сетях, в том числе на уровне передачи отдельных пакетов (packet level). Данный подход базируется на использовании системы имитационного моделирования *Georgia Tech Network Simulator (GTNetS)*, основанной на C++.

Применение такого высокого уровня детализации позволяет учесть детали установления TCP-соединения, корректную маршрутизацию пакетов, направленных на несуществующие конечные хосты, влияние вредоносного трафика на нормальный и т.п.

При разработке имитационной модели авторы [4] преследовали несколько целей, а именно, что

1) накладные расходы, связанные с моделированием сетевых червей в симуляторе GTNetS, должны быть минимальны;
2) модель должна позволять рассчитывать при передаче пакетов важные параметры, которые влияют на распространение червей;
3) модель должна быть достаточно гибкой, чтобы поддерживать широкий диапазон классов сетевых червей.

В работе [4], так же как и в [8], моделирование поведения сетевого червя сводится к моделированию трех шагов, а сам червь описывается первыми пятью параметрами, которые приведены выше при обзоре системы DDosVax. Кроме этого, дополнительно вводится еще один параметр — порт протокола транспортного уровня, используемый сетевым червем при заражении. Учет номера этого порта позволяет проводить более реалистичное моделирование в том случае, когда одновременно с червем хосту передается нормальный трафик (на тот же самый либо другой порт).

Система имитационного моделирования GTNetS предоставляет значительную гибкость в построении произвольной топологии сети путем соединения большого числа предопределенных элементов. В работе [4] предлагаются интересные расширения, позволяющие проводить моделирование распространения червей более эффективно.

Известно, что в любой системе моделирования сетей значительных вычислительных ресурсов и объемов памяти требует расчет и хранение информации, необходимой для маршрутизации. В тех случаях, когда априори рассчитанные таблицы маршрутизации не являются подходящим решением в силу большой топологии сети, используется метод NIx-векторов. Его суть состоит в том, что маршруты рассчитываются только в случае необходимости методом поиска в ширину [4] (а фактически алгоритмом Дейкстры поиска кратчайшего пути на графе) и хранятся в *заголовке пакета* в специальном сжатом виде (как правило, для маршрута достаточно не более 32 бит). Структура NIx-вектора кэшируется в узле-отправителе и используется при повторном обращении к тому же получателю.

Несмотря на высокую эффективность этого метода при традиционном моделировании сетей, в случае с моделированием сетевых червей, осуществляющих случайные сканирования, возникают следующие проблемы, приводящие к увеличению требований к ресурсам, необходимым для моделирования:

1) возрастает частота выполнения алгоритм поиска кратчайшего пути, поскольку вероятность обращения к кэшу мала;
2) при поиске пути до несуществующего хоста алгоритм проверяет все пути, пока не определит, что искомого маршрута не существует;
3) в случае если получатель пакета не найден (Nix-вектор не найден), пакеты все равно должны быть направлены по разумному маршруту (в свою подсеть или на соответствующий шлюз, где и будут отброшены).

Для решения указанных проблем в [4] предлагается дополнить GTNetS так называемыми *агентами маршрутизации* (routing proxy), манипулирующими сетевыми адресами (предполагается маршрутизация с использованием масок), присвоенными интерфейсу некоторого хоста в том случае, если они доступны через этот интерфейс.

Авторы [4] предлагают размещать такие агенты на узлах, являющихся шлюзами некоторых подсетей. Тогда работу алгоритма поиска кратчайшего пути можно приостанавливать, как только встретился узел, содержащий агента маршрутизации с подсетью, включающей адрес хоста назначения.

Рассчитанный NIx-вектор будет использоваться для маршрутизации до агента, который и направит пакет конечному хосту (тем самым решается 3-я проблема). Более того, использование агентов маршрутизации позволяет также повысить эффективность реализации алгоритма поиска кратчайшего пути. Действительно, учитывая свойства агента маршрутизации, можно отказаться от рассмотрения некоторой группы последующих узлов (в [4] эта идея называется *BFS pruning*). Это справедливо в том случае, если на некотором шаге агент показывает, что через заданный интерфейс адрес хоста назначения не доступен.

Последнее улучшение в методике расчета маршрутов (правда, на момент написания [4] еще не реализованная в GTNetS) — это, так называемая, *агрегация NIx-векторов*, суть которой состоит в следующем. При нахождении маршрута к заданному хосту с использованием агентов фактически рассчитывается маршрут к некоторому *диапазону адресов* хостов. Тогда скорость расчета последующих кратчайших путей может быть ускорена, если в узле назначения (вместе с самим NIx-вектором) кэшировать также информацию о том диапазоне адресов, для которого он может быть использован.

Подводя итог по работе [4], отметим, что в ней не содержится каких-либо

попыток оценки качества получаемых в ходе имитационного моделирования результатов, однако методы повышения эффективности детального моделирования сетевых процессов представляются весьма интересными.

6. Система имитационного моделирования на основе PDNS

В работе [3] рассматривается детальная модель компьютерных червей, относящаяся к классу “packet level”. Эта модель разрабатывается с использованием систем имитационного моделирования *Georgia Tech Network Simulator (GTNetS)* и *Parallel and Distributed Network Simulator (PDNS)*.

PDNS является расширением известной системы моделирования ns2, позволяющим распараллеливать вычисления. Сам ns2 также использовался для моделирования распространения сетевых червей с использованием гибридного подхода, схожего с SSFNet [4]. Однако отсутствие в ns2 встроенного механизма для присвоения IP–адресов узлам сети ведет к проблемам при моделировании механизма сканирования.

В отличие от подходов, предлагаемых авторами [4, 8], в [3] вводится более детальная модель распространения сетевого червя, учитывающая открытие скрытого (backdoor) порта (рис. 4).

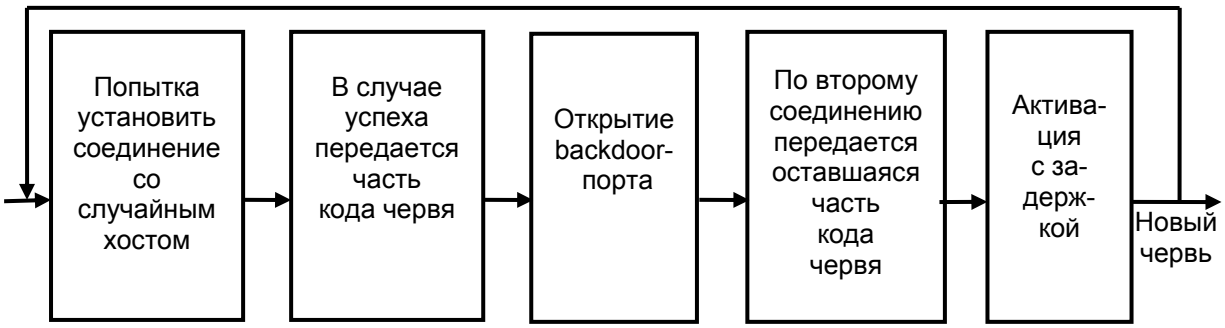

Рис. 4. Модель распространения червя [3].

Параметры, описывающие поведение сетевого червя, в работе [3] подробно не описаны, но основные из них совпадают с параметрами, введенными в [4] и рассмотренными выше.

Интересным является то, как упомянутая выше проблема моделирования маршрутизации на несуществующие хосты решается в работе [3]. Авторами [3] предлагается способ решения этой проблемы на основе введения в сеть маршрутизатора, который будет «обособлять» подсеть неиспользуемых адресов. При пересылке пакета на промежуточном маршрутизаторе сверяется адрес получателя данного пакета с таблицей маршрутизации. В случае попадания адреса получателя в подсеть неиспользуемых адресов данный пакет уничтожается. В противном случае пакет направляется к своему получателю.

Авторы [3] отмечают также, что значительные вычислительные затраты при моделировании связаны с установлением TCP-соединений, причем симулятор PDNS ведет себя с этой точки зрения хуже, чем GTNetS, поскольку переключение между компонентами, разработанными на OTcl и C++, занимает значительное время.

Вообще же результаты имитационного моделирования, проведенные в [3], являются достаточно уникальными.

С использованием параллельных вычислений на 128 процессорах моделировалось распространение сетевого червя Code Red II в сети, состоящей из 1.28 миллионов узлов. При этом сеть включала 128 связанных магистральных маршрутизатора, каждый из которых представлял собой корень двухуровневого дерева с коэффициентом ветвления 100.

7. Система имитационного моделирования DIB:S/TRAFEN

В [2] рассматривается задача моделирования функционирования системы обнаружения и отслеживания сетевых червей DIB:S/TRAFEN и способы моделирования трафика сетевого червя. В DIB:S/TRAFEN используется допущение, что черви при размножении опрашивают большое количество IP-адресов. Однако поскольку их выбор случаен, многие из них окажутся недоступными (немаршрутизируемыми). При этом маршрутизатор возвратит хосту, пославшему такой запрос, пакет “ICMP Destination Unreachable” (ICMP Type 3). В этом пакете будет содержаться исходный IP-заголовок и по крайней мере 8 байт заголовка протокола, которые вместе составляют адреса и порты отправителя и получателя для протоколов UDP и TCP.

На маршрутизаторах количество таких пакетов в секунду обычно ограничено. Кроме того, в соответствии с политикой безопасности, они могут отсылаться не отправителю пакета с немаршрутизируемым адресом, а на хост-анализатор. DIB:S/TRAFEN анализирует такие пакеты и, если их было больше определенного количества за заданное время, генерирует различные варианты тревоги. Пример типичной ситуации, при которой генерируется сигнал тревоги, когда один и тот же хост по одному и тому же протоколу контактирует с тем же портом N хостов за время t. Затем компонент TRAFEN определяет, в соответствии с заданными моделями, свидетельствует ли эта тревога о распространяющемся сетевом черве.

Для *моделирования трафика сетевого червя* используется система моделирования сети Интернет SSFNet, в которой реализованы протоколы стека TCP/IP и маршрутизации с определенной долей детальности.

Основное допущение, используемое при моделировании трафика сетевого червя, заключается в следующем. Черви генерируют так много сетевых пакетов, что с их имитацией не справится ни один суперкомпьютер. Например, для червя Slammer, который заразил около 75000 хостов, характерна скорость сканирования 4000 хостов в секунду на экземпляр. Трафик составляет около 300 миллионов пакетов в секунду. Предлагается подробно рассматривать лишь эффекты, к которым приводит размножение червя (отказы, динамика маршрутизации или перегрузки).

Эта так называемая «макроскопическая модель» соответствует эпидемической модели распространения вируса. Используются детерминированная простая эпидемическая модель и стохастическая модель на ранних стадиях размножения.

Для анализа вирусного трафика сканирования на базе этих моделей формируется пространственная эпидемическая модель (Spatial Epidemic Model), в которой хосты разбиваются на группы, и рассматривается трафик между этими группами.

Пусть обнаружены сканирования из сети j с интенсивностью $\sigma_j^{\text{gen}}(t)$ и сканирования, направленные в сеть j с интенсивностью $\sigma_j^{\text{dest}}(t)$. Количество «входящих» $\sigma_j^{\text{ingr}}(t)$ и “исходящих” $\sigma_j^{\text{egr}}(t)$ сканирований для сети j определяется в [2] следующим образом:

$$\sigma_j^{\text{ingr}}(t) = \sigma_j^{\text{dest}}(t) - \sigma_j^{\text{gen}}(t) \cdot \frac{A_j}{2^{32}},$$

$$\sigma_j^{\text{egr}}(t) = \sigma_j^{\text{gen}}(t) \cdot \left(1 - \frac{A_j}{2^{32}}\right),$$

где A_j — размер адресного пространства сети j.

Если предположить, что каждая группа хостов представляет собой автономную систему (AS), а трафик передается по шлюзам, то эта модель дает возможность моделировать топологию Интернет на уровне AS. Соответственно шлюзы будут посылать пакеты ICMP для немаршрутизируемых адресов.

Рассмотрим, как входящие потоки сканирования будут преобразовываться в пакеты. Так как сканирование исходит из различных фрагментов сети, то приход пакетов сканирования было решено моделировать пуассоновским процессом. В начале каждого шага (интервала) приход пакетов устанавливается как $\sigma_j^{\text{ingr}}(t)$. Промежутки между приходами сэмплируются, и определяется время следующего прихода пакета. Если оно не попадает в данный интервал, то происходит переход к следующему интервалу и время снова определяется. Это будет соответствовать пуассоновскому процессу при $\sigma_j^{\text{ingr}}(t) \cdot \Delta t >> 1$.

Если червь использует транспортный протокол, в котором предусмотрена повторная посылка пакетов, например TCP–SYN, то такие пакеты моделируются. Например, реализация TCP от Microsoft предусматривает первую переposылку TCP–SYN через 3 с, а вторую — через 6 с.

Для каждой подсети (фрагмент сети) выделяется диапазон адресов u_j. Случайным образом с вероятностью $1 - u_j$ определяется, должен ли быть сгенерирован пакет ICMP на приходящий пакет сканирования. В соответствии с наблюдениями авторов, количество пакетов ICMP обычно 3–4 в секунду.

Сгенерированный пакет составляется следующим образом:

- IP-заголовок: адреса получателя и отправителя генерируются автоматически;
- ICMP-заголовок: пакет type 3 subtype 1 “host unreachable”;
- вложенный заголовок IP/TCP/UDP: адрес отправителя выбирается случайно из всех адресов хостов с вирусами, адрес получателя выбирается случайно из всего пространства адресов. Порт отправителя зависит от типа вируса, порт получателя выбирается случайно из диапазона 1024–65535.

В генерируемый трафик добавляется шум, то есть сканирование, вызванное не вирусной активностью.

Авторами [2] были проведены эксперименты с использованием стохастической эпидемической модели для червей Code Red II и Slammer.

В результате сравнения с реальным трафиком этих червей были получены следующие их параметры: скорость сканирования 4.65 и 4000 хостов в секунду соответственно, количество используемых адресов (пространства IPv4) — 50%,

количество доступных адресов (маршрутизируемых) тоже 50%. Проведены эксперименты по обнаружению вирусной активности по сгенерированному трафику в системе DIB:S/TRAFEN. Получены наилучшие значения параметров для компонент TRAFEN.

Выделим следующие *достоинства* данного подхода:

- генерируется не весь вирусный трафик, а только его предполагаемые последствия. В данном случае генерируются ICMP-пакеты, возвращаемые маршрутизатором при обработке немаршрутизируемого адреса в пакете;
- моделируется вся сеть Интернет на уровне AS.

Недостатки подхода можно сформулировать так:

- генерируются только пакеты ICMP–T3. Этих пакетов достаточно только для обнаружения методом TRAFEN;
- не подходит для имитации вирусного трафика на свитчах, так как количество пакетов ICMP–T3 будет небольшим в связи с малым размером сети.

8. Заключение

На основе анализа результатов исследований по тематике моделирования распространения сетевых червей в статье приводится обзор ряда имитационных моделей и программных систем имитационного моделирования, применяемых для исследования распространения сетевых червей. Рассмотрены недостатки и достоинства существующих моделей.

Представленные подходы к моделированию сетевых червей, основанные на детальном моделировании на уровне сетевых пакетов (“Packet level”) и гибридном моделировании (“Mixed Abstraction”) могут быть с успехом использованы для исследования механизмов сетевой защиты. С учетом выявленных особенностей имитационных моделей авторами планируется разработка собственной системы моделирования распространения сетевых червей для исследований перспективных механизмов сдерживания эпидемий сетевых червей.

Работа выполнена при финансовой поддержке РФФИ (проект №07–01–00547), программы фундаментальных исследований ОИТВС РАН (контракт №3.2/03), Фонда содействия отечественной науке и при частичной финансовой поддержке, осуществляемой в рамках проектов Евросоюза POSITIF (контракт IST–2002–002314) и RE–TRUST (контракт № 021186–2).

Литература

1. *Ediger B*. Simulating Internet Worms [Электронный ресурс] // <http://www.users.qwest.net/ ~eballen1/nws/> (по состоянию на 12.03.2007).
2. *Liljenstam M., Nicol D. M., Berk V. H., Gray R. S.* Simulating Realistic Network Worm Traffic for Worm Warning System Design and Testing // Proceedings of the 2003 Workshop on Rapid Malcode (WORM). October 27, 2003, Washington, DC, USA. P. 60–70.
3. *Perumalla K. S., Sundaragopalan S.* High-Fidelity Modeling of Computer Network Worm // Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC'04), December 06–10, 2004. P. 126–135.
4. *Riley G. F., Sharif M. I., Lee W.* Simulating Internet Worms // 12th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS’04), Oct. 2004. P. 268–274. [Электронный ресурс] // <http://www.cc.gatech.edu/~wenke/papers/worms_simulation.pdf> (по состоянию на 12.03.2007).
5. *Saroiu S., Gummadi P. K., Gribble S. D.* A measurement study of peer-to-peer file sharing sys-

tems // Proc. of Multimedia Computing and Neworking 2002 (MMCN'02), San Jose, CA, USA, January 2002. P. 156–170.
6. Scalable Simulation Framework. [Электронный ресурс] // <http:// www.ssfnet.org/homePage.html> (по состоянию на 12.03.2007).
7. *Talkad H.* Survey of Worm Traffic Simulators: Course project for Security and Privacy in Computing, Csci 8980-002 Fall, 2003. 7 p. [Электронный ресурс] // < http://www.users.cs.umn.edu/~htalkad/files/worm.pdf> (по состоянию на 12.03.2007).
8. *Wagner A., Dubendorfer T., Plattner B., Hiestand R.* Experiences with worm propagation simulations // Proceedings of ACM CCS Workshop on Rapid Malcode (WORM'03), 2003. P. 34–41.